Vol. 3

Optimization Methods

Computer Aided and Integrated Manufacturing Systems

A 5-Volume Set

Cornelius T Leondes

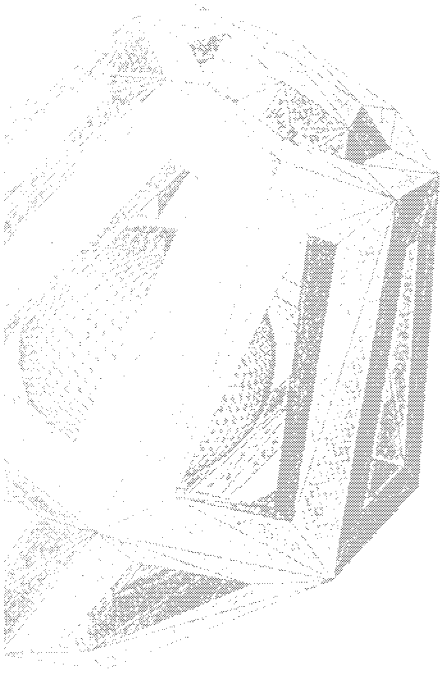
Vol. 3

Optimization Methods

Computer Aided and Integrated Manufacturing Systems

A 5-Volume Set

This page is intentionally left blank



Vol. 3

Optimization Methods

Computer Aided and Integrated Manufacturing Systems

A 5-Volume Set

Cornelius T Leondes

University of California, Los Angeles, USA

World Scientific

New Jersey • London • Singapore • Hong Kong

Published by

World Scientific Publishing Co. Pte. Ltd.

5 Toh Tuck Link, Singapore 596224

USA office: Suite 202, 1060 Main Street, River Edge, NJ 07661

UK office: 57 Shelton Street, Covent Garden, London WC2H 9HE

British Library Cataloguing-in-Publication Data
A catalogue record for this book is available from the British Library.

COMPUTER AIDED AND INTEGRATED MANUFACTURING SYSTEMS
A 5-Volume Set
Volume 3: Optimization Methods

ISBN 981-238-339-5 (Set)
ISBN 981-238-981-4 (Vol. 3)

Desk Editor: Tjan Kwang Wei
Typeset by Stallion Press

Printed by FuIsland Offset Printing (S) Pte Ltd, Singapore

Preface

Computer Technology

This 5 volume MRW (Major Reference Work) is entitled "Computer Aided and Integrated Manufacturing Systems". A brief summary description of each of the 5 volumes will be noted in their respective PREFACES. An MRW is normally on a broad subject of major importance on the international scene. Because of the breadth of a major subject area, an MRW will normally consist of an integrated set of distinctly titled and well-integrated volumes each of which occupies a major role in the broad subject of the MRW. MRWs are normally required when a given major subject cannot be adequately treated in a single volume or, for that matter, by a single author or coauthors.

Normally, the individual chapter authors for the respective volumes of an MRW will be among the leading contributors on the international scene in the subject area of their chapter. The great breadth and significance of the subject of this MRW evidently calls for treatment by means of an MRW.

As will be noted later in this preface, the technology and techniques utilized in the methods of computer aided and integrated manufacturing systems have produced and will, no doubt, continue to produce significant annual improvement in productivity — the goods and services produced from each hour of work. In addition, as will be noted later in this preface, the positive economic implications of constant annual improvements in productivity have very positive implications for national economies as, in fact, might be expected.

Before getting into these matters, it is perhaps interesting to briefly touch on Moore's Law for integrated circuits because, while Moore's Law is in an entirely different area, some significant and somewhat interesting parallels can be seen. In 1965, Gordon Moore, cofounder of INTEL made the observation that the number of transistors per square inch on integrated circuits could be expected to double every year for the foreseeable future. In subsequent years, the pace slowed down a bit, but density has doubled approximately every 18 months, and this is the current definition of Moore's Law. Currently, experts, including Moore himself, expect Moore's Law to hold for at least another decade and a half. This is impressive with many significant implications in technology and economies on the international scene. With these observations in mind, we now turn our attention to the greatly significant and broad subject area of this MRW.

"The Magic Elixir of Productivity" is the title of a significant editorial which appeared in the *Wall Street Journal.* While the focus in this editorial was on productivity trends in the United States and the significant positive implications for the economy in the United States, the issues addressed apply, in general, to developed economies on the international scene.

Economists split productivity growth into two components: Capital Deepening which refers to expenditures in capital equipment, particularly IT (Information Technology) equipment: and what is called Multifactor Productivity Growth, in which existing resources of capital and labor are utilized more effectively. It is observed by economists that Multifactor Productivity Growth is a better gauge of true productivity. In fact, computer aided and integrated manufacturing systems are, in essence, Multifactor Productivity Growth in the hugely important manufacturing sector of global economies. Finally, in the United States, although there are various estimates by economists on what the annual growth in productivity might be, Chairman of the Federal Reserve Board, Alan Greenspan — the one economist whose opinions actually count, remains an optimist that actual annual productivity gains can be expected to be close to 3% for the next 5 to 10 years. Further, the Treasure Secretary in the President's Cabinet is of the view that the potential for productivity gains in the US economy is higher than we realize. He observes that the penetration of good ideas suggests that we are still at the 20 to 30% level of what is possible.

The economic implications of significant annual growth in productivity are huge. A half-percentage point rise in annual productivity adds $1.2 trillion to the federal budget revenues over a period of ten years. This means, of course, that an annual growth rate of 2.5 to 3% in productivity over 10 years would generate anywhere from $6 to $7 trillion in federal budget revenues over that time period and, of course, that is hugely significant. Further, the faster productivity rises, the faster wages climb. That is obviously good for workers, but it also means more taxes flowing into social security. This, of course, strengthens the social security program. Further, the annual productivity growth rate is a significant factor in controlling the growth rate of inflation. This continuing annual growth in productivity can be compared with Moore's Law, both with huge implications for the economy.

The respective volumes of this MRW "Computer Aided and Integrated Manufacturing Systems" are entitled:

Volume 1: Computer Techniques
Volume 2: Intelligent Systems Technology
Volume 3: Optimization Methods
Volume 4: Computer Aided Design/Computer Aided Manufacturing (CAD/CAM)
Volume 5: Manufacturing Process

A description of the contents of each of the volumes is included in the PREFACE for that respective volume.

Optimization methods will become an increasingly important factor in manufacturing systems as they have proven to enhance productivity. The design of cellular manufacturing systems will also be optimized, again with significant implications for productivity. Computer Aided Design (CAD) methods for manufacturing processes, such as the ubliquitous injection molding process, will increasingly utilize optimization methods. Rapid prototyping has become a way of life in manufacturing systems and this will benefit greatly from the optimization methods. The CAD/CAM (Computer Aided Design/Computer Aided Manufacturing) process will necessarily require visual assessment techniques of free-form surfaces in order to assure an optimal product. These and numerous other significant topics are treated rather comprehensively in Volume 3.

As noted earlier, this MRW (Major Reference Work) on "Computer Aided and Integrated Manufacturing Systems" consists of 5 distinctly titled and well integrated volumes. It is appropriate to mention that each of the volumes can be utilized individually. The great significance and the potential pervasiveness of the very broad subject of this MRW certainly suggests the clear requirement of an MRW for an adequately comprehensive treatment. All of the contributors to this MRW are to be highly commended for their splendid contributions that will provide a significant and unique reference source for students, research workers, practitioners, computer scientists and others, as well as institutional libraries on the international scene for years to come.

This page is intentionally left blank

Contents

CHAPTER 1

OPTIMAL DYNAMIC FACILITY DESIGN OF MANUFACTURING SYSTEMS

TIMOTHY L. URBAN
Operations Management,
The University of Tulsa,
600 South College Avenue,
Tulsa, Oklahoma, USA
E-mail: unbantl@utulsa.edu

The physical layout of manufacturing systems is a major determinant of a firm's efficiency. With the rapidly-changing environment facing most firms today as well as the shortened life cycles of many products and process technologies, facility rearrangement and redesign become critical in sustaining productivity and competitiveness. Consequently, operations managers and researchers have recently focused on the dynamic aspects of facility design. This paper investigates various approaches of analyzing and solving the dynamic facility layout problem. Optimization, bounding, and heuristic methodologies are presented, and issues concerning the application and implementation of these dynamic models are presented.

Keywords: Facility design; dynamic models; optimization.

1. Introduction

It is a well-established fact that products and processes exhibit life cycles evolving through initial development, growth, maturity, and decline stages. The understanding of these life cycles can be quite beneficial in determining appropriate marketing and manufacturing strategies for an organization. Schmenner[1] proposed that facilities also progress through life cycles and that the knowledge of this concept can be exploited to plan the use and change of the facility, leading to improved productivity and prolonged facility life. Nandkeolyar *et al.*[2] further developed a conceptual model for facility life cycles, identifying the characteristics that describe each stage. The throughput, the number of products, the capacity utilization, and the process technology — all of which have an effect on the design of the facility at any particular stage of the life cycle — change throughout the life of a facility which, in turn, necessitates the redesign of the facility. In general, manufacturing facilities tend to be quite capital intensive and have long-range implications for the organization,

which underscores the importance of dynamic facility design in response to the changing demands placed on the facility.

Nicol and Hollier[3] reported on the results of a field study of 33 manufacturing companies in the United Kingdom. One of the aspects investigated was the stability of the firms' facility layout. They found that layouts were frequently designed for a predetermined fixed level of production which could only be marginally exceeded, yet many companies experienced or anticipated volume changes by a factor of two or more. Nearly half of the companies had an average layout stability of two years or less; the mean of all firms was just over three years. The authors concluded "that radical layout changes occur frequently and that management should therefore take this into account in their forward planning".

More recently, Hales[4] reported on a survey of facility management organizations (199 respondents, two-thirds of which were manufacturing firms) designed to measure current business practices and concerns. One of the facilities management practices that was identified as being in serious need of improvement was that of "planning horizon" for major buildings and facilities; over 80 percent of the respondents categorized their organization with poor or inadequate performance on this dimension. Furthermore, three of the eight key findings that were identified are:

(i) Rearranging for cells and continuous flow: This was the greatest management concern, cited by 60% of respondents, including a number in government, insurance, health care, and other non-manufacturing facilities.
(ii) Many facilities organizations lack readiness: They are playing 'one-move chess' with no plan beyond their next major project.
(iii) Planning horizons are still too short among manufacturing organizations: The most common practice is still calendar-based, typically three to five years, rather than being tied to industry cycles or the life cycles of key products and process technologies.

Obviously, facilities managers see the rearrangement and redesign of facilities as an important part of their organizational efficiency and competitiveness, yet this is one aspect of their planning efforts that is apparently underemphasized.

The purpose of this paper is to investigate various approaches of analyzing and solving the dynamic facility layout problem. Current formulations of the problem and techniques for determining the optimal solution are presented. We present a linearization of the problem, such that a commercially-available, linear-programming computer package can be utilized, either in conjunction with a CAD system or as a stand-alone package for a facility manager solving relatively small problems. Bounding techniques are demonstrated to make the problem more tractable. Finally, implementation issues concerning the understanding and application of the dynamic facility layout problem are presented.

2. The Dynamic Facility Layout Problem

Despite the obvious practical relevance concerning the dynamics of facility design, the vast majority of relevant research has focused on the static problem, in which the layout design is determined with no consideration of future requirements. Literally hundreds of papers have been written on the static facility (plant) layout problem since the development of operations sequence analysis[5] and systematic layout planning.[6] Several reviews of the facility layout problem have been published, including Levary and Kalchik,[7] Kusiak and Heragu,[8] and Meller and Gau.[9] Recently, a survey paper on the dynamic facility layout problem was published by Balakrishnan and Cheng.[10]

Hitchings[11] was one of the first authors to recognize the importance of planned changes in the layout of facilities. Based on the observation that the material handling cost for all feasible layouts tends to be approximately normally distributed, Hitchings argued that controlling a layout could be conducted similar to the control of a production process. Thus, the use of control charts was suggested to identify at which point in time a layout change is warranted; that is, when the cost of effecting the change is less than the savings that result from the change. The use of statistical quality control techniques is common in manufacturing systems and could then easily be applied to determine the timing of the facility redesign.

Hicks and Cowan[12] developed an extension of the well-known Craft heuristic,[13] a pairwise-exchange procedure in which the cost of moving a department and the resulting process improvements (resulting from the redesign) are incorporated in the layout decision logic. One shortcoming of their approach, however, is that it only provides one opportunity for facility redesign, changing the layout when the resulting cost improvement exceeds the rearrangement cost; no future rearrangements are considered. Slepicka and Rajchel[14] proposed a dynamic programming procedure to select from a set of feasible layout arrangements over a finite time period. They formulated the problem with two conflicting goals — minimizing the cost of the rearrangement and maximizing the operational efficiency of the resulting layouts. It was assumed the time "when a major expansion takes place" is defined, and a feasible set of layouts within each time frame can be determined.

Rosenblatt[15] was the first to present a comprehensive treatment of the dynamic facility layout problem; he provided an explicit formulation of the problem, developed an optimal solution methodology, identified bounding procedures, and established heuristic techniques. He proposed a model analogous to the quadratic assignment problem (QAP) in which the objective is to assign each of the N departments to one of the N specified locations (dummy departments or locations can be used when an unequal number of departments/locations are available). In the dynamic layout problem, this assignment must be made for each of the T periods in the planning horizon; thus, there are $(N!)^T$ possible solutions to the overall problem. Even symmetric layouts must be considered in the dynamic situation, as different rearrangement costs will be incurred for different layouts. Since the publication of

Rosenblatt's paper, a great deal of research activity has focused on dealing with various aspects of the dynamic facility layout problem.

3. Mathematical Formulation

The dynamic facility layout problem (DFLP) is one in which the layout arrangement of a facility — that is, the relative location of departments, machines, cells, workstations, etc. — is determined for each period of a finite planning horizon. The principal costs associated with this problem are the material handling costs for each period as well as any rearrangement costs involved in changing the layout between periods. On the one hand, we would like to change the layout arrangement over time to minimize the material handling effort; on the other hand, we would like to maintain the same layout from one period to the next to avoid costs associated with the rearrangement of the facility.

3.1. *Extension of the quadratic assignment problem*

The most common formulation of the DFLP is an extension of the quadratic assignment problem; reviews of the QAP can be found in Burkard,[16,17] Finke *et al.*,[18] and Pardalos *et al.*[19] Under this formulation, the performance (efficiency) of a particular layout arrangement is measured as the sum of the workflows (material handling cost per unit distance) times the distance traveled. This measure must be expressed as a cost in the DFLP, as opposed to a distance measure, so it can directly correspond to the rearrangement costs. This is the approach that was presented by Rosenblatt[15] and has been the focus of the majority of research conducted on the dynamic problem.

Let $N = \{1, 2, \ldots, i, \ldots, |N|\}$ represent the number of departments and locations and $T = \{1, 2, \ldots t, \ldots, |T|\}$ represent the time periods in the planning horizon. The problem can then be formulated as a quadratic binary programming problem (see, e.g. Kaku and Mazzola[20]) as follows:

DFLP–1
Minimize

$$C = \sum_{t \in T} \left[\sum_{i \in N} \sum_{j \in N} \sum_{k \in N} \sum_{l \in N} f_{ikt} d_{jl} x_{ijt} x_{klt} + \sum_{i \in N} \sum_{j \in N} c_{ijt} x_{ijt} + \sum_{i \in N} s_{it} y_{it} + r_t z_t \right]. \tag{1}$$

Subject to:

$$\sum_{j \in N} x_{ijt} = 1 \qquad i \in N,\ t \in T, \tag{2}$$

$$\sum_{i \in N} x_{ijt} = 1 \qquad j \in N,\ t \in T, \tag{3}$$

$$x_{ijt} \in \{0, 1\} \qquad i \in N,\ j \in N,\ t \in T, \tag{4}$$

$$y_{it} = \sum_{j \in N} x_{ijt-1} \left(\sum_{l \in N \setminus \{j\}} x_{ilt} \right) \qquad i \in N,\ t \in T \setminus \{1\}, \tag{5}$$

$$z_t \geq y_{it} \qquad i \in N,\ t \in T \setminus \{1\}, \tag{6}$$

$$y_{it}, z_t \geq 0 \qquad i \in N,\ t \in T \setminus \{1\}, \tag{7}$$

where f_{ikt} is the workflow cost from department i to department k in time period t (likely measured as the product of the volume of material flow, assumed to be deterministic, and the cost per unit to move the material), d_{jl} is the distance from location j to location l (assumed to be time invariant), c_{ijt} is the cost of assigning department i to location j in period t, s_{it} is the variable rearrangement cost of moving department i at the beginning of the time period t, and r_t is the fixed rearrangement cost associated with making any layout changes at the beginning of period t. The decision variables are:

$$x_{ijt} = \begin{cases} 1 & \text{if department } i \text{ is placed at location } j \text{ in period } t \\ 0 & \text{otherwise} \end{cases}$$

$$y_{it} = \begin{cases} 1 & \text{if department } i \text{ is moved at the beginning of period } t \\ 0 & \text{otherwise} \end{cases}$$

$$z_t = \begin{cases} 1 & \text{if any rearrangement is made at the beginning of period } t \\ 0 & \text{otherwise} \end{cases}$$

Constraint sets (2) and (3) are the typical constraints associated with an assignment problem and they ensure that each department is assigned to one location and each location contains exactly one department for each time period. Constraint sets (5) and (6) are definitional constraints ensuring that the rearrangement variables (y_{it} and z_t) take on a value of one if a rearrangement takes place. Note that these constraints also allow the rearrangement variables to be expressed as continuous variables. As shown, these definitional constraints and variables assume that there is no existing layout at the beginning of the planning horizon, implying that this is a new facility; hence, they are not required for the first period (any layout arrangement can be used with no rearrangement cost at $t = 1$). If there is an existing layout involving a rearrangement at the current time, this can be represented by incorporating the appropriate assignment variables (i.e. we would have x_{ij0} to take on appropriate values) and extending Constraint sets (5) and (6) over the entire planning horizon ($t \in T$).

This formulation of the dynamic facility layout problem can easily be generalized. For example, Balakrishnan *et al.*[21] formulated the variable rearrangement costs to reflect the origin and destination of a department's move, $\sum_i \sum_j \sum_l s_{ijlt} y_{ijlt}$. This could take into account, for instance, situations in which the rearrangement cost is dependent on the distance the department is moved. Discounting considerations

can also be easily incorporated into the formulation, if desired, by simply including a discount factor for each of the cost coefficients.

3.2. *Strategic interpolative design*

An alternative approach to the DFLP was presented by Montreuil and Venkatadri[22] in which the facility designer has developed a target layout arrangement for the final (mature) phase of a facility expansion. The intent is then to identify the intermediary layout over several phases. It is assumed that each department will remain within the boundaries it is assigned in the final arrangement. Thus, the size of the department must remain less than or equal to its final size during each phase, although the input/output stations of the departments are not required to be stationary and can be moved at no cost. The location of each department relative to the other departments, however, must remain the same.

Since the size of the departments can change over time, the quadratic assignment formulation (in which departments are allocated to specific locations) is not appropriate. Instead, a formulation is used in which the layout is defined on a planar coordinate system (see, e.g. Montreuil[23] and Banerjee *et al.*[24]). The departments are rectangular, and their locations are characterized by the coordinates of the input/output stations and the length and width of the department.

Since each department remains in the same location (although it may grow over time), there are no rearrangement costs associated with this formulation. Thus, the objective is simply to minimize the sum of the material handling costs for each phase over a finite planning horizon. Let $N = \{1, 2, \ldots, i, \ldots, |N|\}$ represent the number of departments, $S = \{1, 2, \ldots, s, \ldots, |S|\}$ represent the number of input/output stations for the departments, and $P = \{1, 2, \ldots, p, \ldots, |P|\}$ represent the phases of the facility expansion. The formulation of the problem is then:

DFLP–2
Minimize

$$C = \sum_{p\in P}\sum_{i\in N}\sum_{j\in N}\sum_{s\in S}\sum_{r\in S} w_p f_{ijsrp}\left(|x_{isp} - x_{jrp}| + |y_{isp} - y_{jrp}|\right). \tag{8}$$

Subject to:

$$\underline{X}_{ip} \leq x_{isp} \leq \overline{X}_{ip} \qquad i \in N,\ s \in S,\ p \in P, \tag{9}$$

$$\underline{Y}_{ip} \leq y_{isp} \leq \overline{Y}_{ip} \qquad i \in N,\ s \in S,\ p \in P, \tag{10}$$

$$0 \leq \underline{LX}_{ip} \leq \overline{X}_{ip} - \underline{X}_{ip} \leq \overline{LX}_{ip} \qquad i \in N,\ p \in P, \tag{11}$$

$$0 \leq \underline{LY}_{ip} \leq \overline{Y}_{ip} - \underline{Y}_{ip} \leq \overline{LY}_{ip} \qquad i \in N,\ p \in P, \tag{12}$$

$$0 \leq \underline{B}_{ip} \leq 2\left[\left(\overline{X}_{ip} - \underline{X}_{ip}\right) + \left(\overline{Y}_{ip} - \underline{Y}_{ip}\right)\right] \leq \overline{B}_{ip} \qquad i \in N,\ p \in P, \tag{13}$$

$$\underline{X}_{i(p-1)} \geq \underline{X}_{ip} \qquad i \in N,\ p \in P\backslash\{1\}, \tag{14}$$

$$\overline{X}_{ip} \geq \overline{X}_{i(p-1)} \qquad i \in N,\ p \in P\backslash\{1\}, \qquad (15)$$

$$\underline{Y}_{i(p-1)} \geq \underline{Y}_{ip} \qquad i \in N,\ p \in P\backslash\{1\}, \qquad (16)$$

$$\overline{Y}_{ip} \geq \overline{Y}_{i(p-1)} \qquad i \in N,\ p \in P\backslash\{1\}, \qquad (17)$$

where w_p is the weight associated with phase p. It is also a function of the length of the phase, a discounting factor, etc. f_{ijsrp} is the workflow between input/output station s of department i and station r of department j in phase p; (x_{isp}, y_{isp}) are the coordinates of input/output station s of department i in phase p; $(\underline{X}_{ip}, \underline{Y}_{ip})$ and $(\overline{X}_{ip}, \overline{Y}_{ip})$ are the coordinates of the lower and upper boundaries of department i in phase p, respectively; $(\underline{LX}_{ip}, \underline{LY}_{ip})$ and $(\overline{LX}_{ip}, \overline{LY}_{ip})$ are the lower and upper bounds on the length of the sides of department i in phase p, respectively; and $\underline{B}_{ip}$ and $\overline{B}_{ip}$ are the lower and upper bounds on the perimeter of department i in phase p. Linear programming can be utilized to solve this problem as the absolute values in the objective function can be easily linearized through the use of additional variables.

Constraint sets (9) and (10) ensure that the input/output station coordinates are within the departmental boundaries. Constraint sets (11) and (12) restrict the length of the sides of the department to be within bounds; the same is done for the perimeter of the department with Constraint set (13). Finally, Constraint sets (14)–(17) ensure that the department in any phase is located within the boundaries of the department in the next phase, allowing growth over time but maintaining its relative location. Montreuil and Venkatadri[22] also identified some variations of their model by considering aspects such as temporary reductions in the size of a department and phasing out a facility.

Montreuil and Laforge[25] introduced a dynamic facility layout model to take into consideration the probabilistic nature of future requirements. They proposed a set of possible future states, each with a probability of occurrence as well as workflow and spatial requirements. The designer is also required to propose a design skeleton for each future, so, as with the Montreuil and Venkatadri[22] model, the relative positions of the departments do not change. The authors argued, however, that an experienced layout designer could investigate multiple design skeletons to identify good layouts.

Lacksonen[26] has since proposed an approach to address the limitation that pre-specifying the design skeleton has on the analysis of the trade-off between material handling costs and rearrangement costs. He proposed a two-stage formulation in which the first stage is a QAP-type of analysis that considers both workflow and department rearrangement, then he fixes the rearrangement costs and estimates the department arrangements. The second stage is a mixed integer program (an extension of the Montreuil formulations) which takes the results of the first stage to minimize the flow costs and define the specific department locations; preprocessing operations were identified to improve on the solution time.[27]

4. Optimal Solution Methodologies

In this section, we will investigate various existing optimal solution methodologies to the dynamic facility layout problem. We will also emphasize on the DFLP-1 formulation, as the majority of research conducted on the DFLP has focused on this formulation. A linearization of DFLP-1 is also developed that will allow the optimal solution of the DFLP to be found using commercially-available, linear-programming software.

4.1. *Dynamic programming*

The first technique developed to identify the optimal solution for the dynamic facility layout problem was presented by Rosenblatt[15] using a dynamic programming algorithm. Each of the states of the dynamic program (DP) corresponds to a particular layout arrangement, and each of the stages corresponds to a time period in the planning horizon, resulting in a problem with $N!$ states and T stages. As mentioned above, symmetric layouts must be included, since different rearrangement costs will result from the different layout arrangements.

A recursive relationship was established to identify the combination of layout arrangements with the minimum total cost as follows:

$$C^*_{tm} = \min_k \left\{ C^*_{t-1,k} + R_{km} \right\} + Q^m_t, \tag{18}$$

where R_{km} is the rearrangement cost as a result of changing from layout arrangement A_k to layout arrangement A_m ($R_{kk} = 0$). This could easily be modified to provide a different cost for different time periods. Q^m_t is the material handling cost for layout arrangement A_m in period t; and C^*_{tm} is the minimum total cost for all periods up to t, in which layout arrangement A_m is being used in period t ($C^*_{01} = 0$, assuming an initial layout is given). To restrict the state space of the model, Rosenblatt noted that the Sweeney and Tatham[28] results for the dynamic location problem are applicable to the dynamic layout problem. In particular, a layout arrangement for a given period does not need to be included in the DP if the difference between the material handling cost of that arrangement, Q^m_t, and the material handling cost of the optimal static solution for that period, Q^*_t, is greater than the difference between the values of the upper bound, C^+, and the lower bound, C^-, of the dynamic problem; that is:

$$Q^m_t - Q^*_t \geq C^+ - C^-. \tag{19}$$

Therefore, it is necessary to include only the best static solutions for each period in the planning horizon. To identify the best ranked static solutions that must be included in the DP state space, a constrained quadratic assignment problem (precluding higher ranked solutions) can be solved. Balakrishnan[29] developed an alternative fathoming procedure in which the right-hand side of Constraint set (19) is replaced by twice the maximum rearrangement cost that could occur in a given

period. This, in the general case, is the sum of the fixed and all variable rearrangement costs.

Due to the computational requirements necessary to optimally solve the quadratic assignment problem as well as that for dynamic programming algorithms (the infamous 'curse of dimensionality'), this approach is practical only for small problems. The development of strong bounds — to reduce the DP state space — and an efficient method of finding the best ranked static solutions have become consequential in finding the optimal solution to the DFLP.

4.2. *Mixed integer programming*

Another approach to solving the dynamic facility layout problem is to linearize the quadratic binary program in order to utilize integer linear programming algorithms, which are readily accessible using commercially-available software. As shown in the previous section, DFLP-1 contains a quadratic objective function and a set of quadratic constraints. To linearize the constraint set, we replace Constraint set (5) with the following constraints:

$$y_{it} \geq x_{ijt-1} - x_{ijt} \quad i \in N,\ j \in N,\ t \in T, \tag{5a}$$

$$y_{it} \geq x_{ijt} - x_{ijt-1} \quad i \in N,\ j \in N,\ t \in T. \tag{5b}$$

While this approach obviously requires additional constraints ($2N^2T$ versus NT), it results in an all-linear constraint set.

We can linearize the objective function in a manner similar to the approaches used in the linearization of the quadratic assignment problem. Various QAP linearizations have been proposed by Lawler,[30] Bazaraa and Sherali,[31] Frieze and Yadegar,[32] and Kettani and Oral[33,34]; we follow the approach of Kaufman and Broeckx[35] which, in turn, is based on the method of Glover.[36] To do this, first consider the quadratic component of the objective function (Eq. 1):

$$\sum_{t\in T}\sum_{i\in N}\sum_{j\in N}\sum_{k\in N}\sum_{l\in N} f_{ikt}d_{jl}x_{ijt}x_{klt}. \tag{20}$$

Now define a set of N^2T continuous variables, w_{ijt}, such that:

$$w_{ijt} = x_{ijt}\sum_{k\in N}\sum_{l\in N} f_{ikt}d_{jl}x_{klt}. \tag{21}$$

If x_{ijt} is equal to zero, then w_{ijt} will also equal zero; if x_{ijt} is equal to one, then w_{ijt} will equal the material handling cost for the corresponding layout arrangement, $\sum_k \sum_l f_{ikt}d_{jl}x_{klt}$. Therefore, this component of the objective function can be simply expressed as the sum of the w_{ijt} variables. For each new variable, we also add the

constraints:

$$u_{ijt}x_{ijt} + \sum_{k\in N}\sum_{l\in N} f_{ikt}d_{jl}x_{klt} - w_{ijt} \leq u_{ijt}, \tag{22}$$

$$w_{ijt} \geq 0, \tag{23}$$

where $u_{ijt} = \max\{\sum_k \sum_l f_{ikt}d_{jl}, 0\}$. The DFLP can now be expressed as the following mixed integer program:

DFLP–1 (MIP)
Minimize

$$C = \sum_{t\in T}\left[\sum_{i\in N}\sum_{j\in N}(w_{ijt} + c_{ijt}x_{ijt}) + \sum_{i\in N} s_{it}y_{it} + r_t z_t\right]. \tag{24}$$

Subject to:

$$\sum_{j\in N} x_{ijt} = 1 \qquad i \in N,\ t \in T,$$

$$\sum_{i\in N} x_{ijt} = 1 \qquad j \in N,\ t \in T,$$

$$y_{it} \geq x_{ijt-1} - x_{ijt} \qquad i \in N,\ j \in N,\ t \in T,$$

$$y_{it} \geq x_{ijt} - x_{ijt-1} \qquad i \in N,\ j \in N,\ t \in T,$$

$$z_t \geq y_{it} \qquad i \in N,\ t \in T,$$

$$u_{ijt}x_{ijt} + \sum_{k\in N}\sum_{l\in N} f_{ikt}d_{jl}x_{klt} - w_{ijt} \leq u_{ijt} \qquad i \in N,\ j \in N,\ t \in T,$$

$$x_{ijt} \in \{0,1\}, w_{ijt}, y_{it}, z_t \geq 0 \qquad i \in N,\ j \in N,\ t \in T.$$

This formulation contains N^2T integer variables and $(N^2 + N + 1)T$ continuous variables. While readily available software can be used to solve this problem (the Ampl model, a modeling language for mathematical programming, and the data presented by Lacksonen and Enscore[37] are contained in the appendix for illustrative purposes), it will still only be practical for relatively small problems. The use of MIP solution strategies — strong bounds, depth-first versus breadth-first branching, special ordered sets, etc. — can be utilized to increase the size of the problem; Crowder *et al.*[38] discussed strategies for solving general, large-scale, zero-one linear programming problems.

4.3. *Special case: fixed rearrangement costs*

In many situations, the fixed cost of rearranging a facility far outweighs the variable (i.e. departmental) rearrangement costs. For example, if the entire facility must frequently be shut down when any rearrangement occurs, the cost of moving any particular department may be negligible as compared to the lost production time.

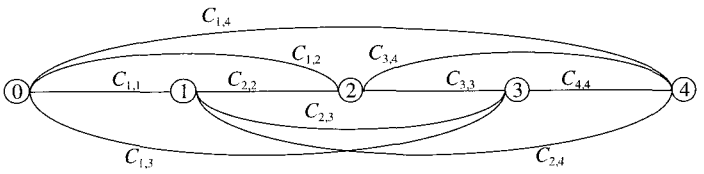


Fig. 1. Network representation of the DFLP with fixed rearrangement costs.

The dynamic facility layout problem in which only fixed rearrangement costs are involved (i.e. $s_{it} = 0, \forall i, t$) was presented by Urban.[39]

To find the optimal solution to this problem, an approach analogous to the "incomplete" dynamic programming algorithm for dynamic facility location[40] can be used. The DFLP can be decomposed into a number of static subproblems, each of which utilizes the same layout arrangement throughout. Since only fixed rearrangement costs are relevant, once a decision to rearrange the layout is made, any layout decision is independent of any previous layout decision and it can be made solely on minimizing material handling costs. Thus, the optimal material handling cost for any subproblem can be found with the static quadratic assignment problem using the sum of the workflows for all periods in the subproblem:

$$f'_{ik} = \sum_{t=p}^{q-1} f_{ikt}, \tag{25}$$

where the subproblem begins with either period 1 or a period p in which a rearrangement takes place, and ends with either period T (the end of the planning horizon) or the period before the next rearrangement. Once each of the $T(T+1)/2$ possible subproblems have been solved, we can model the final phase of the overall problem as a shortest-path network — analogous to the classic equipment replacement problem — as illustrated in Fig. 1.

In the network, the nodes represent each period in the planning horizon (node 0 is the current period), and the cost associated with each of the arcs equals the minimum material handling cost for that subproblem plus the fixed rearrangement cost. Due to the structure of this type of problem, it is possible to solve larger problems to optimality rather than in the general case.

5. Bounding Procedures

The development of strong bounds is essential for the efficient solution of the general dynamic facility layout problem. Tight bounds can reduce the state space of Rosenblatt's dynamic programming algorithm and can hasten the fathoming of the branch-and-bound procedure for the mixed integer program. In this section, we investigate the various bounding techniques that have been developed for the DFLP.

5.1. *Lower bound procedures*

Rosenblatt[15] noted that a lower bound for the DFLP can be obtained by summing the minimum material handling costs of the static facility layout problem in each period and ignoring any associated rearrangement costs, $C^- = \sum_t Q_t^*$. To determine the value of this bound, however, requires finding the optimal solution to the quadratic assignment problem for each of the T periods in the planning horizon.

To devise lower bounds that require less computational effort, Urban[41] developed a class of bounds based on lower bounds, rather than the optimal solution, of the static problem. That is, the lower bound for the DFLP can be obtained by summing the lower bound of the static layout problem in each period, again ignoring any associated rearrangement costs. While these bounds will be dominated by the Rosenblatt bounds, they are computationally more attractive, particularly for larger problems, and were found to perform quite well for compact layout designs as well as when the variability of workflow between departments is low.

An approach used to identify a lower bound that dominates all existing bounds was developed by Urban.[39] To do this, consider the following two situations:

(i) **If a rearrangement takes place**
In this case, at least two departments will be affected. Thus, we will incur the fixed rearrangement cost and the variable rearrangement cost for two departments (at a minimum) in one time period. Thus, a lower bound would be the minimum possible material handling cost (i.e. the Rosenblatt bound) plus this minimum rearrangement cost:

$$C_R^- = \sum_{t \in T} Q_t^* + \min_t \left\{ r_t + \min_i \{s_{it}\} + \min_i \{s'_{it}\} \right\}, \tag{26}$$

where s'_{it} is the set of variable rearrangement cost values with the minimum value from that period removed.

(ii) **If a rearrangement does not take place**
The only circumstance in which rearrangement costs are not incurred is when the same layout arrangement is utilized throughout the planning horizon. In this situation, the minimum material handling cost can be determined by solving a quadratic assignment problem using the sum of the workflows over the entire planning horizon; that is:

$$C_{NR}^- = Q_{1,T}^*,$$

where the objective of the QAP is:

$$\text{Minimize} \quad C_{NR}^- = Q_{1,T}^* = \sum_{i \in N} \sum_{j \in N} \sum_{k \in N} \sum_{l \in N} \left[\sum_{t \in T} f_{ikt} \right] d_{jl} x_{ij} x_{kl}, \tag{27}$$

subject to the typical assignment constraints, Constraint sets (2)–(4).

The lower bound for the DFLP can then be expressed as:

$$C^- = \min\{C_R^-, C_{NR}^-\}. \tag{28}$$

Since both of these values, C_R^- and C_{NR}^-, are assured of being greater than or equal to the bound obtained from Rosenblatt's procedure, this bound obviously dominates it.

5.2. *Upper bound procedures*

In his discussion on reducing the number of layout arrangements to be considered in each period, Rosenblatt[15] noted that "two problems still remain to be answered. The first is how to get a good (small) upper bound $\cdots$". While any feasible solution to the DFLP will provide an upper bound, it is obviously beneficial to identify as small a bound as possible. He presented two alternatives for determining the upper bounds: (1) using a myopic approach which utilizes the optimal layout (minimizing the material handling cost) for each time period, while incurring the necessary rearrangement costs; and (2) selecting a layout among the optimal layouts from each period that will provide the minimum material handling cost for the entire planning horizon, incurring no rearrangement costs.

Batta[42] proposed an upper bound for the DFLP by solving the quadratic assignment problem using the sum of the workflows over the entire planning horizon and utilizing that layout arrangement for each period. This will dominate Rosenblatt's second bound as it provides a lower material handling cost and it will also incur no rearrangement costs. Furthermore, it is more computationally efficient, as it requires the calculation of one QAP, as opposed to one for each period in the planning horizon.

An approach to identify an upper bound that dominates all existing bounds was presented by Urban.[39] To do this, consider the solution methodology for the special case presented in the previous section in which only fixed rearrangement costs are involved. As discussed, the optimal material handling cost for each subproblem is found by solving the quadratic assignment problem using the sum of the workflows for all periods in the subproblem. The relevant rearrangement costs are then added to the material handling costs, and the resulting network is solved (as shown in Fig. 1). Although this bound requires additional computational requirements, it dominates Rosenblatt's first method and Batta's method, as both of those can be shown to be special cases of this approach.

6. Implementation Issues

We now turn our attention to several issues that may arise during the analysis, optimization, and implementation of the dynamic facility layout problem.

6.1. *Heuristics*

The dynamic facility layout problem (DFLP-1) has as its special case the quadratic assignment problem (when $T = 1$) and, since the QAP is known to be NP-complete,[43] we can conclude that the DFLP is also NP-complete. Thus, the development of heuristics is necessary to solve even moderately-sized instances of the dynamic problem.

Rosenblatt[15] proposed a class of heuristics for the DFLP in which the dynamic programming algorithm is used, but with fewer states considered than necessary to determine the optimal solution. The intent is to determine a good, although small, set of layout arrangements (states) to be considered in each period (stage). He proposed using the static optimal QAP solution from each period (thus, requiring N layouts) under a heuristic approach to generate a number of layouts, or randomly generating the layouts. While this approach obviously may not result in the optimal solution, it becomes computationally feasible for larger problems, although it still requires the use of a dynamic programming algorithm.

A heuristic that utilizes the popular steepest-descent, pairwise-interchange procedure (used in Craft[13]), was developed by Urban.[44] To adapt it for use with the DFLP, the concept of "forecast windows" is used. That is, the workflows for a varying number of periods are evaluated at any one time. The heuristic first analyzes the workflows one period at a time, and the associated rearrangement costs are incurred. The heuristic then considers a forecast window of two periods, assuming the same layout will be used over this time frame, and determines appropriate layout arrangements. The length of the forecast window is increased, the layouts determined, and the solution is obtained from the forecast window that provides the minimum total cost.

Lacksonen and Enscore[37] modified and evaluated five existing algorithms to solve the DFLP: (1) a pairwise exchange heuristic that also evaluates exchanging pairs of locations in consecutive time periods; (2) a heuristic utilizing a cutting-plane routine to identify initial solutions for the static problem followed by an exchange routine; (3) a branch-and-bound algorithm limited to the most promising partial solutions and limited in program run length; (4) a dynamic programming algorithm with a limited number of states identified using an exchange routine; and (5) a heuristic based on cut trees including "rearrangement avoidance" costs. Of these five methods, the cutting-plane algorithm identified the best solutions for all test problems considered.

Since then, several authors have used various metaheuristics to solve the dynamic facility layout problem. Genetic algorithms have been used for the DFLP with additional constraints[45] and with a two-period, multiple-floor problem.[46] Kaku and Mazzola[20] developed a tabu search heuristic procedure for the DFLP. Urban[39] utilized a greedy randomized search procedure (Grasp) for the special case of the DFLP with fixed rearrangement costs. Recently, Bozer and Wang[47] presented a simulated annealing algorithm. Due to the computational requirements for identifying the

optimal solution to the DFLP, the development of effective heuristics is a fruitful area of research.

6.2. *Discrete efficient frontier*

The formulation of the total cost of the dynamic facility layout problem (Eq. 1) assumes that we can derive explicit values for the workflow costs (f_{ikt}) and the rearrangement costs (s_{it} and r_t) in comparable terms. However, in practice, static facility layout analyzes typically minimize the distance traveled, not the associated cost. It may be difficult to obtain comparable costs that could be used to solve the DFLP as formulated. The DFLP generally has two conflicting objectives: to minimize material handling and to minimize the rearrangement "effort" of the facility over the planning horizon.

A useful approach to situations with incommensurable objectives is the concept of a discrete efficiency frontier (DEF), which is the set of efficient (nondominated) solutions, such that a given solution is at least equally as good as another solution on all measures (objectives) and strictly better on at least one measure. Rosenblatt and Sinuany-Stern[48] and Malakooti[49] addressed the notion of identifying efficient layout arrangements for the static, multiple-criteria (workflow versus closeness rating) facility layout problem and developed algorithms to identify the DEF. Urban[39] proposed a method of identifying the DEF for the dynamic facility layout problem with no variable rearrangement costs — comparing the volume of material flow and the number of rearrangements made — using the information obtained from the solution to the problem.

When we include variables, as well as fixed, rearrangement costs, the measure of the rearrangement effort may no longer simply be the number of rearrangements. It may be necessary to use another measure, such as the total number of departments moved, the total rearrangement cost (if it can be determined), or the number of rearrangements (if that is still the primary concern for rearrangement). To illustrate the process, let us use the following objectives:

(i) **Material handling**
Minimize the total workflow over the planning horizon, $\sum_t \sum_i \sum_j \sum_k \sum_l f_{ikt}\, d_{jl} x_{ijt} x_{klt}$, where f_{ikt} is not expressed as a cost but measured simply as the volume of material flow, subject to the typical assignment constraints.

(ii) **Rearrangement**
Minimize the total number of departments moved over the planning horizon, $\sum_t \sum_i y_{it}$; again, this could be replaced with other appropriate measures.

A typical efficient frontier is illustrated in Fig. 2. The two extremes of the frontier are easily determined: (1) *the minimal rearrangement point* is determined by utilizing the same layout arrangement during the entire planning horizon, hence no departments are moved, which is determined by minimizing the material handling, as with the Batta upper bound; and (2) *the minimal material handling point* is

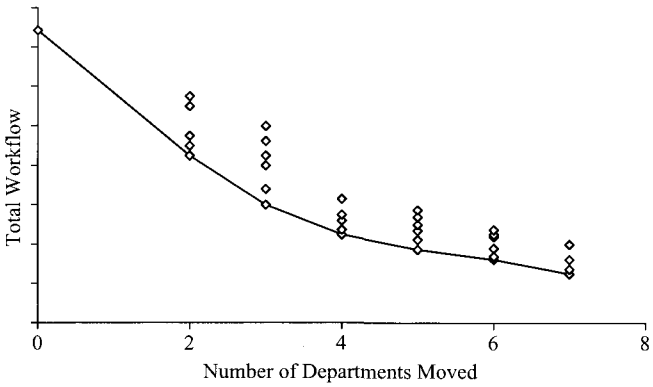


Fig. 2. Discrete efficient frontier for the DFLP.

determined by minimizing the material handling for each time period incurring the necessary rearrangement (number of departments moved) of the facility. We can then find the intermediate points in the following manner:

Step 1: Using the formulation DFLP-1, remove the rearrangement cost portion of the objective function, leaving only $\sum_t \sum_i \sum_j \sum_k \sum_l f_{ikt} d_{jl} x_{ijt} x_{klt}$, and solve. This will provide *the minimal material handling point* as described above.

Step 2: From this solution, define ψ to be the number of departments that are moved over the planning horizon.

Step 3: To the formulation in *Step 1*, include the constraint:

$$\sum_{i \in N} \sum_{t \in T} y_{it} \leq \psi - 1 \tag{29}$$

and solve.

Step 4: Repeat *Steps 2* and *3* until $\psi = 0$. At this time, we will have *the minimal rearrangement point.*

Theoretically, there could be as many as $NT - 1$ iterations required to identify the entire efficient frontier. To reduce the calculations, we may wish to replace the right-hand side of Eq. 29 with $\psi - k$, where k is some fraction of the number of departments moved. We could also use heuristics to solve the mixed integer program, particularly for larger problems; however, we are not assured that all of the points on the efficient frontier will be identified.

6.3. *Flexible facility layout*

An interesting perspective regarding dynamic facility design is the concept of flexible facility layouts. The flexibility of a layout arrangement is the ability to accommodate

changes in the production requirements, either by the insensitivity of the layout to those changes (reactive flexibility) or by the ability of the layout to be easily and rapidly changed (adaptive flexibility). Several papers have been written in this area since the publication of the paper by Shore and Tompkins[50]; however, we will limit our discussion to those that explicitly consider the rearrangement of the facility.

Bullington and Webster[51] evaluated the adaptive flexibility of a layout by considering the rearrangement costs of a change from that layout to one of several potential future layouts. The present layout with the smallest expected rearrangement cost, using the probability of each of the potential future layouts, was considered to be the most flexible. Savsar[52] extended this approach by incorporating reactive flexibility considerations, measured using both material flows and closeness ratings, with the expected rearrangement cost.

Each of these approaches assume that there is only one opportunity for rearrangement. A related approach that considers multiple periods in the planning horizon is presented by Kouvelis *et al.*[53] They proposed a methodology to identify robust layout arrangements — those that are within a certain percent of the optimal material handling cost. First, robust layouts for each period in the planning horizon are identified, then a sequence of layouts that avoids the moving of "monuments," departments that are difficult to relocate, are found. The flexibility of this approach is such that once (in the first period) a decision is made concerning the family of layouts to be used, the only irreversible decision is that the location of the monuments and a near-optimal layout arrangement can then be identified for each period.

6.4. *A quick-and-dirty test of optimality*

Balakrishnan and Cheng[10] noted that the importance of considering the dynamics of facility layout problems is diminished in two situations: (1) when the rearrangement cost is negligible, we need to focus only on the material handling cost and can rearrange the facility as needed; and (2) when the rearrangement cost is prohibitive, we can use the same layout arrangement for the entire planning horizon. The results from Urban's[39] lower bound calculations provide a test of optimality for the second case; in particular, we can make the following claim:

> **Claim:** If $\min_t \{r_t + \min_i\{s_{it}\} + \min_i\{s'_{it}\}\} > Q^*_{1,T} - \sum_{t\in T} Q^*_t$, then it is optimal to avoid facility rearrangement, $\sum_t z_t = 0$, incurring a total cost of $Q^*_{1,T}$.

This observation makes it unnecessary to solve the DFLP in those situations with relatively large rearrangement costs.

7. Concluding Remarks

The results of an evaluation of dynamic layout strategies for a flexible manufacturing system by Afentakis *et al.*[54] indicate that a poor layout can add as much as

36 percent to the material handling requirements. Given the strategic importance of maintaining efficient and productive facilities in a rapidly changing environment, the consequence of incorporating the dynamics of facility design is obvious. Furthermore, the use of CAD/CAE/CAM systems makes it quite straightforward to incorporate this type of analysis into a comprehensive facility design program.

Recent research in the area of dynamic facility layout has extended the basic problem into other areas. For example, Balakrishnan *et al.*[21] and Conway and Venkataramanan[45] incorporated budget constraints that may restrict the amount of rearrangement that can take place in a particular period. Lacksonen and Hung[55] noted that the most common type of facility layout project is re-layout and they proposed a project-scheduling model for a rearrangement project that considers temporary relocations. The dynamics of specific types of manufacturing systems, such as cellular manufacturing systems[56,57] and automated manufacturing systems,[58] have also been studied.

Appendix 1 — Ampl Program for DFLP

```
# Dynamic facility layout problem mixed integer program
# using linearization of Kaufman & Broeckx (EJOR 1978)

set D;                                  # set of departments
set L;                                  # set of locations
set T ordered;                          # set of time periods

param f{D,D,T};                         # workflow between departments
param d{L,L};                           # distance between locations
param c{D,L,T};                         # assignment cost
param s{D,T};                           # rearrangement cost, variable
param r{T};                             # rearrangement cost, fixed

param u{i in D, j in L, t in T} :=
  max(0, sum{k in D, l in L} f[i,k,t]*d[j,l]);

var w{D,L,T} >= 0;                      # linearization variables
var x{D,L,T} binary;                    # assignment variables
var y{D,T} >= 0;                        # rearrangement, variable
var z{T} >= 0;                          # rearrangement, fixed

minimize cost:
  sum{t in T}
    ( (sum{i in D, j in L} w[i, j, t])
      +(sum{i in D, j in L} c[i, j, t]*x[i, j, t])
      +(sum{i in D} s[i, t]*y[i, t]) + r[t]*z[t]) );
```

```
subject to asgndept {i in D, t in T}:
   sum{j in L} x[i, j, t] = 1;

subject to asgnlocn {j in L, t in T}:
   sum{i in D} x[i, j, t] = 1;

subject to def1 {i in D, j in L, t in T: ord(t)> 1}:
   y[i, t] - x[i, j, prev(t)] + x[i, j, t] >= 0;

subject to def2 {i in D, j in L, t in T: ord(t)> 1}:
   y[i, t] - x[i, j, t] + x[i, j, prev(t)] >= 0;

subject to def3 {i in D, t in T}:
   z[t] - y[i, t] >= 0;

subject to def4 {i in D, j in L, t in T}:
   (sum{k in D, l in L} f[i, k, t]*d[j, l]*x[k, l, t])
      + u[i, j, t]*x[i, j, t] - w[i, j, t] <= u[i, j, t];
```

Appendix 2 — Ampl Data for DFLP

```
# Lacksonen and Enscore (IJPR 1993) test problem with no replacement of
# departments, 4 departments/locations (N=4) and 2 time periods (T = 2).

 set D   :=   D1,   D2,   D3,   D4;
 set L   :=   L1,   L2,   L3,   L4;
 set T   :=   T1,   T2;

param f :=
   [*, *, T1]:
        D1   D2   D3   D4:=
  D1     0    0    0     0
  D2    10    0    0     0
  D3     4    8    0     0
  D4     0    2    6     0

[*, *, T2]:
        D1   D2   D3   D4:=
  D1     0    0    0     0
  D2     6    0    0     0
  D3     2    6    0     0
  D4    11    6    5     0;
```

param d :

	L1	L2	L3	L4:=
L1	0	1	2	3
L2	1	0	1	2
L3	2	1	0	1
L4	3	2	1	0;

param c :=

[∗, ∗, T1]:

	L1	L2	L3	L4:=
D1	0	0	0	0
D2	0	0	0	0
D3	0	0	0	0
D4	0	0	0	0

[∗, ∗, T2]:

	L1	L2	L3	L4:=
D1	0	0	0	0
D2	0	0	0	0
D3	0	0	0	0
D4	0	0	0	0;

param s :

	T1	T2:=
D1	0	10
D2	0	10
D3	0	10
D4	0	10;

param r :=

T1	0
T2	0;

References

1. R. W. Schmenner, Every factory has a life cycle, *Harvard Business Review* **61**, 2 (1983) 121–129.
2. U. Nandkeolyar, S. S. Rao and K. Rana, Facility life cycles, *Omega — The International Journal of Management Science* **21**, 2 (1993) 245–254.
3. L. M. Nicol and R. H. Hollier, Plant layout in practice, *Material Flow* **1**, 3 (1983) 177–188.
4. L. Hales, Benchmarking facilities management: recent survey results, *IE News: Facilities Planning & Design* **27**, 2 (1993) 1–3.
5. E. S. Buffa, Sequence analysis for functional layouts, *Journal of Industrial Engineering* **6**, 2 (1955) 12–13, 25.
6. R. Muther, *Systematic Layout Planning*, Boston: Industrial Education Institute, 1961.

7. R. R. Levary and S. Kalchik, Facilities layout — A survey of solution procedures, *Computers & Industrial Engineering* **9**, 2 (1985) 141–148.
8. A. Kusiak and S. S. Heragu, The facility layout problem, *European Journal of Operational Research* **29**, 3 (1987) 229–251.
9. R. D. Meller and K.-Y. Gau, The facility layout problem: recent trends and emerging perspectives, *Journal of Manufacturing Systems* **15**, 5 (1996) 351–366.
10. J. Balakrishnan and C. H. Cheng, Dynamic layout algorithms: a state-of-the-art survey, *Omega—The International Journal of Management Science* **26**, 4 (1998) 507–521.
11. G. G. Hitchings, Control, redundancy, and change in layout systems, *AIIE Transactions* **2**, 3 (1970) 253–262.
12. P. E. Hicks and T. E. Cowan, Craft-M for layout rearrangement, *Industrial Engineering* **8**, 5 (1976) 30–35.
13. G. C. Armour and E. S. Buffa, A heuristic algorithm and simulation approach to relative allocation of facilities, *Management Science* **9**, 2 (1963) 294–300.
14. K. L. Slepicka and D. E. Rajchel, Layout selection over a finite time period, *Annual Industrial Engineering Conference Proceedings*, 1982.
15. M. J. Rosenblatt, The dynamics of plant layout, *Management Science* **32**, 1 (1986) 76–86.
16. R. E. Burkard, Quadratic assignment problems, *European Journal of Operational Research* **15**, 3 (1984) 283–289.
17. R. E. Burkard, Locations with spatial interactions: quadratic assignment problems. *Discrete Location Theory*, eds. P. B. Mirchandani and R. L. Francis (New York: Wiley, 1990) 387–437.
18. G. Finke, R. E. Burkard and F. Rendl, Quadratic assignment problems, *Annals of Discrete Mathematics* **31** (1987) 61–82.
19. P. M. Pardalos, F. Rendl and H. Wolkowicz, The quadratic assignment problem: a survey and recent developments. *Quadratic Assignment and Related Problems, Proceedings of the DIMACS Workshop on Quadratic Assignment Problems*, eds. P. M. Pardalos and H. Wolkowicz, *DIMACS Series in Discrete Mathematics and Theoretical Computer Science* **16** (1994) 1–42.
20. B. K. Kaku and J. B. Mazzola, A tabu-search heuristic for the dynamic plant layout problem, *Informs Journal on Computing* **9**, 4 (1997) 374–384.
21. J. Balakrishnan, F. R. Jacobs and M. A. Venkataramanan, Solutions for the constrained dynamic facility layout problem, *European Journal of Operational Research* **57**, 2 (1992) 280–286.
22. B. Montreuil and U. Venkatadri, Strategic interpolative design of dynamic manufacturing systems layout, *Management Science* **37**, 6 (1991) 682–694.
23. B. Montreuil, A modelling framework for integrating layout design and flow network design. *Progress in Material Handling and Logistics*, Volume 2, eds. J. A. White and I. W. Pence (New York: Springer-Verlag, 1991) 95–116.
24. P. Banerjee, B. Montreuil, C. L. Moodie and R. L. Kashyap, A modelling of interactive facilities layout designer reasoning using qualitative patterns, *International Journal of Production Research* **30**, 3 (1992) 433–453.
25. B. Montreuil and A. Laforge, Dynamic layout design given a scenario tree of probable futures, *European Journal of Operational Research* **63**, 2 (1992) 271–286.
26. T. A. Lacksonen, Static and dynamic layout problems with varying areas, *Journal of the Operational Research Society* **45**, 1 (1994) 59–69.
27. T. A. Lacksonen, Preprocessing for static and dynamic facility layout problems, *International Journal of Production Research* **35**, 4 (1997) 1095–1106.

28. D. S. Sweeney and R. L. Tatham, An improved long-run model for multiple warehouse location, *Management Science* **22**, 7 (1976) 748–758.
29. J. Balakrishnan, Notes: 'The dynamics of plant layout', *Management Science* **39**, 5 (1993) 654–655.
30. E. L. Lawler, The quadratic assignment problem, *Management Science* **9**, 4 (1963) 586–599.
31. M. S. Bazaraa and H. D. Sherali, Benders' partitioning scheme applied to a new formulation of the quadratic assignment problem, *Naval Research Logistics Quarterly* **27**, 1 (1980) 29–41.
32. A. M. Frieze and J. Yadegar, On the quadratic assignment problem, *Discrete Applied Mathematics* **5** (1983) 89–98.
33. O. Kettani and M. Oral, Equivalent formulations of nonlinear integer problems for efficient optimization, *Management Science* **36**, 1 (1990) 115–119.
34. O. Kettani and M. Oral, Reformulating quadratic assignment problems for efficient optimization, *IIE Transactions* **25**, 6 (1993) 97–107.
35. L. Kaufman and F. Broeckx, An algorithm for the quadratic assignment problem using benders' decomposition, *European Journal of Operational Research* **2**, 3 (1978) 207–211.
36. F. Glover, Improved linear integer programming formulations of nonlinear integer problems, *Management Science* **22**, 4 (1975) 455–460.
37. T. A. Lacksonen and E. E. Enscore, Quadratic assignment algorithms for the dynamic layout problem, *International Journal of Production Research* **31**, 3 (1993) 503–517.
38. H. Crowder, E. L. Johnson and M. Padberg, Solving large-scale zero-one linear programming problems, *Operations Research* **31**, 5 (1983) 803–834.
39. T. L. Urban, Solution procedures for the dynamic facility layout problem, *Annals of Operations Research* **76** (1998) 323–342.
40. G. O. Wesolowsky, Dynamic facility location, *Management Science* **19**, 11 (1973) 1241–1248.
41. T. L. Urban, Computational performance and efficiency of lower-bound procedures for the dynamic facility layout problem, *European Journal of Operational Research* **57**, 2 (1992) 271–279.
42. R. Batta, Comment on 'The dynamics of plant layout', *Management Science*, **33**, 8 (1987) 1065.
43. S. Sahni and T. Gonzalez, P-complete approximation problems, *Journal of the Association for Computing Machinery* **23** (1976) 555–565.
44. T. L. Urban, A heuristic for the dynamic facility layout problem, *IIE Transactions* **25**, 4 (1993) 57–63.
45. D. G. Conway and M. A. Venkataramanan, Genetic search and the dynamic facility layout problem, *Computers & Operations Research* **21**, 8 (1994) 955–960.
46. J. S. Kochhar and S. S. Heragu, Facility layout design in a changing environment, *International Journal of Production Research* **37**, 11 (1999) 2429–2446.
47. Y. A. Bozer and C.-T. Wang, A heuristic procedure for multi-year dynamic facility layout problems, presented at the Informs National Conference, Cincinnati, 1999.
48. M. J. Rosenblatt and Z. Sinuany-Stern, A discrete efficient frontier approach to the plant layout problem, *Material Flow* **3**, 4 (1986) 277–281.
49. B. Malakooti, Multiple objective facility layout: a heuristic to generate efficient alternatives, *International Journal of Production Research* **27**, 7 (1989) 1225–1238.
50. R. H. Shore and J. A. Tompkins, Flexible facilities design, *AIIE Transactions* **12**, 2 (1980) 200–205.

51. S. F. Bullington and D. B. Webster, Evaluating the flexibility of facility layouts using estimated layout costs, *9th International Conference on Production Research Proceedings* (1987) 2230–2236.
52. M. Savsar, Flexible facility layout by simulation, *Computers & Industrial Engineering* **20**, 1 (1991) 155–165.
53. P. Kouvelis, A. A. Kurawarwala and G. J. Gutiérrez, Algorithms for robust single and multiple period layout planning for manufacturing systems, *European Journal of Operational Research* **63**, 2 (1992) 287–303.
54. P. Afentakis, R. A. Millen and M. M. Solomon, Dynamic layout strategies for flexible manufacturing systems, *International Journal of Production Research* **28**, 2 (1990) 311–323.
55. T. A. Lacksonen and C.-Y. Hung, Project scheduling algorithms for re-layout projects, *IIE Transactions* **30**, 1 (1998) 91–99.
56. A. J. Vakharia and B. K. Kaku, Redesigning a cellular manufacturing system to handle long-term demand changes: a methodology and investigation, *Decision Sciences* **24**, 5 (1993) 909–930.
57. E. M. Wicks and R. J. Reasor, Designing cellular manufacturing systems with dynamic part populations, *IIE Transactions* **31**, 1 (1999) 11–20.
58. P. Kouvelis and A. S. Kiran, Single and multiple period layout models for automated manufacturing systems, *European Journal of Operational Research* **52**, 3 (1991) 300–314.

This page is intentionally left blank

CHAPTER 2

COMPUTER TECHNIQUES AND APPLICATIONS FOR THE DESIGN OF OPTIMUM CELLULAR MANUFACTURING SYSTEMS

MINGYUAN CHEN

Department of Mechanical and Industrial Engineering,
Concordia University,
1455 de Maisonneuve West,
Montreal, Quebec, Canada H3G 1M8

Cellular manufacturing is a well developed approach for mid-volume and mid-variety production management and control. In this chapter, a number of mathematical programming models for cellular manufacturing problem formulations were introduced. These models address cellular manufacturing problems in both static and dynamic environments. Specific features of these problems and models are discussed. A number of solution methods based on integer programming, decomposition methods and dynamic programming are presented in detail. Several example problems are used to demonstrate the natures of the problems and to illustrate the modeling as well as solution approaches. Computational aspects of the solution methods are also presented with their implications to solving real world practical problems.

Keywords: Cellular manufacturing; cell re-configuration; modeling and applications.

1. Introduction

Manufacturing cells are typically designed for mid-volume and mid-variety production. Flexible manufacturing systems (FMS) and special manufacturing systems may also be used for different levels of mid-volume and mid-variety production.[24] Cellular manufacturing (CM) has been viewed as an application of group technology (GT) philosophy in organizing and managing manufacturing machines, equipment, personnel and production. Surveys conducted by Welmmerlov and others[11,29–31] show that cellular manufacturing approaches provide substantial benefits for the companies in terms of improved productivity and reduced work-in-process inventory. A survey conducted by two Australian universities and an IE consulting company[26] shows that a large percentage of discrete manufacturing companies in Australia have also implemented the CM approaches. Feedbacks from the surveyed companies are similar to those as reported in the surveys conducted by Wemmerlov and others. They are very positive with respect to many critical manufacturing areas.

In designing and implementing cellular manufacturing techniques, some of the primary issues to be considered are:

- the creation of part families;
- the number of manufacturing cells to be configured in the system;
- the types of machines to be placed in the cells;
- the single or multiple units of machines to be placed in cells;
- the machine layout in the cells;
- the ways and methods of material handling;
- the intra-cell and inter-cell material flows;
- the labor issues;
- the quality issues;
- the work-in-process inventory;
- the cell re-configuration as a result of changing production demand;
- production planning in a cellular manufacturing environment; and
- the scheduling of cellular manufacturing production.

In fact, cellular manufacturing is related to many aspects of a manufacturing system if viewed from general system design perspectives. Figure 1 shows some of the manufacturing aspects discussed above and their relations with each other in the general context of cellular manufacturing. This chart is developed for illustration purpose only, while the subtle and complicated relationships among different manufacturing areas may not be accurately reflected. In Fig. 1, we show that CM has two major components, one is group technology (GT) and the other is cell formation. Primarily, GT is concerned with part family generation while cell formation deals with the setting up of manufacturing cells. Generally speaking, there are three types of GT methods in generating part families[3]: visual inspection, coding and classification, and production flow analysis. Visual inspection may require computer vision systems and image processing techniques. Coding and classification are more closely related to computer aided design (CAD) for automated code generation and computer aided process planning (CAPP) using variant as well as generative approaches. Production flow analysis is used for both part family generation and manufacturing cell formation. This is also the primary approach, if not the only one, for cell formation. It is based on part process plans providing part-machine relationship. The machines processing a relatively independent set of parts will be placed in a same machine cell to reduce material handling efforts and cost. Cell formation problems can be characterized by many different features as shown in the large dashed box in Fig. 1. As shown in this figure, in forming machine cells, one may consider a single production plan where each operation can be processed by only one machine in the system. This is based on the assumption that the machines do not have the flexibility to process less desirable operations. In this simplified case, more comprehensive models may be developed to include intra-cell and inter-cell material handling and other aspects. Setting up machine cells to optimize intra-cell material handling is

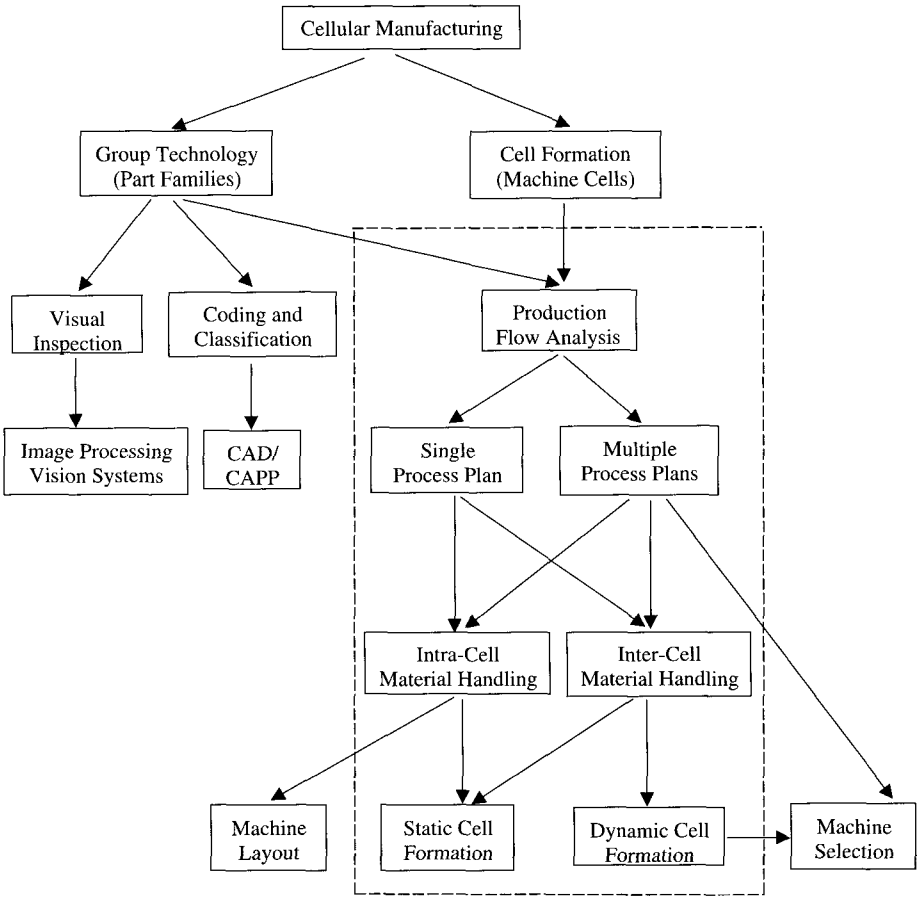


Fig. 1. Cellular manufacturing and related issues.

normally treated as machine layout problems. It could also be part of a comprehensive cell formation model. If machine flexibility is a dominant phenomenon in a system, an operation of a part may be processed by alternative machines. In other words, there may exist alternative process plans for part processing. In this case, the problem is more complicated due to an increased number of possible solutions. The optimal solution requires us to develop new procedures not only to form the cells but also to identify machine/operation combinations, since processing costs vary with different process plans. If the problem is limited by considering a single process plan for each part, then there is no machine selection problem, unless the problem is studied in a dynamic environment with changing production demands in multiple time periods. As discussed by Bedworth *et al.*,[3] one of the difficulties in implementing cellular manufacturing strategies is the possible large cost incurred in rearranging machine cells. Most developed procedures use current production

data, available machine capacities and existing part-machine processing relations to cluster machines into cells. In a dynamic and more realistic manufacturing environment, one may need to reform the cells after the changes in production demand or part mix occur. System changes such as adding or removing machines may be required in order to efficiently operate the system to accommodate new demands. Changing system configurations, however, can be very expensive, difficult or even impossible. This and other issues will be discussed in the section on developing solution procedures to solve dynamic cell formation problems.

2. Solution Methods for Solving Various CM Problems

Research on basic and extended CM problems with many or some of the system characteristics shown in Fig. 1 has been carried out by many researchers. Various solution approaches have been proposed for solving different problems effectively and efficiently. These methods can generally be categorized as:

- mathematical programming;
- optimization-based heuristics;
- rule-of-thumb heuristics;
- artificial neural networks; and
- discrete event simulation.

In this section, we will briefly discuss the basic features of these approaches as they are applied to solving CM problems.

2.1. *Mathematical programming*

Mathematical programming is a very powerful tool in formulating many manufacturing system design and operation problems including cell formation problems. This approach is to construct a number of mathematical functions to reflect the requirements of forming manufacturing cells. In many cases, these are simple linear functions. In other situations, non-linear functions may have to be used to reflect the interests of the particular study. Also, depending on the purpose of the study, one or more of these functions will be used as the objective functions to be maximized or minimized (subjected to the constraints of other functions). The problem solution determines the numbers of cells, the machines in cells, the parts to be processed by the machines in cells, etc. These are the decision variables in the mathematical model. The use of this approach dictates that the solutions, if can be found, will be optimal and that no further improvements can be achieved. However, the real challenge of using mathematical programming models is the efficiency of finding the solutions because very often these models are integer programming models. There is no general solution procedure which can solve integer programming models efficiently. Therefore, models formulated for different problems with various features and assumptions may be investigated individually and effective solution procedures

may be developed to solve these problems with specific features. Later in this section, we will discuss some of these models and solution methods.

2.2. *Optimization-based heuristics*

As discussed above, mathematical programming models can be developed to solve cellular manufacturing problems. Due to the difficulties of optimally solving these models, researchers have developed various algorithms to solve them heuristically. In fact, the use of these algorithms and methods may not be limited to solving cellular manufacturing problems. They can be used for solving many different manufacturing problems as long as these problems can be meaningfully formulated. The solution methods are based on various search techniques such as simulated annealing,[25] Tabu Search[15] and genetic search.[13] Using certain criteria to cut off the number of iterations in a general branch-and-bound process also belongs to this category.[27]

2.3. *Rule-of-thumb heuristics*

Methods developed based on some heuristic approaches may not generate an optimization formulation. One may simply check the available information and follow the rules-of-thumb of the practitioners in solving the problems. These methods are usually simple, fast and easy to use. Optimal solutions are not the target and the quality of the solutions depends on initial solution and other features of the problem. Hence a procedure may be used iteratively to obtain a better solution. This can normally solve very large problems with satisfactory results.

Heuristic construction or improvement procedures have been developed by different researchers. Some construction procedures may be similar to those used in general facility layouts. Many improvement procedures are similar to the well-known CRAFT process. One example of such method can be found in Viswanathan.[28] In that paper, the author presented a quadratic integer programming model. An algorithm based on simple interchange of the facilities was proposed for finding the optimal or near-optimal solutions of the model. Another example is shown in Harhalakis *et al.*[8] where a bottom-up aggregation and a local refinement procedure were proposed to generate manufacturing cells.

2.4. *Artificial neural networks*

Different types of artificial neural networks have been developed to solve cell formation problems. Interested readers may find some of the research reports in Jamal,[12] Rao and Gu,[18] and Zolfaghari and Liang.[32] They will not be further discussed in this chapter.

2.5. *Discrete event simulation*

Simulation models developed to solve CM formation problems are associated with detailed studies of CM production features such as planning, scheduling, material

handling, inventory control, quality control, machine breakdowns, etc. They normally can provide large amounts of information regarding many aspects of the system, including cells and cell formation. Various heuristic algorithms for cellular manufacturing and production scheduling may be included in these simulation models. Similar to the heuristic approaches, the purposes of simulation studies are not optimization. They are the computerized implementation of heuristics, usually with sophisticated software. Some of the cellular manufacturing studies using simulation can be found in Biles *et al.*,[4] Prakash and Chen,[17] Shafer and Charnes,[21] Shang and Tadikamalla,[22] Shinn and Williams,[23] among others. In fact, simulation may be more attractive if the cell formation problems involve scheduling decisions.

2.6. *Summary*

Some of the CM methods proposed by academic researchers may not be directly applicable to solving real world practical problems. If the mathematical models are complicated and the solution methods are difficult to use, the possibility that they will actually be used in practice is very small. Discrete simulation may be the most effective means for solving practical cell formation problems and other manufacturing problems. Most simulation models are relatively simple, straightforward and easily understandable. The fast development of today's computer graphics and animation technology also greatly enhanced the graphic capability of many already powerful simulation packages such as ARENA and Taylor-II, among many others currently available in the market. In the following sections, we will discuss some typical CM problems and models with different features. Most of these problems can be solved by some of the above discussed methods to a certain extent. Obviously, the need to investigate which methods are most effective and most efficient in solving different problems with specific features is still present.

3. Static Cell Formation Problems

In formulating this type of problems, it is assumed that the cells will be formed based on known demand for part processing in the manufacturing system. It is also assumed that such demands may be relatively stable in the foreseeable future. Cells formed based on such assumptions will also be relatively stable. If demands change, the cell formation process will start all over again to form new cells based on newly available information and data. This approach, largely ignoring the future needs to the system, may not be realistic. Models taking into account the future concerns will be discussed later in this chapter. Although we will concentrate on detailed mathematical programming in presenting the problems, the different approaches discussed in the previous sections such as heuristics and simulation can be developed to solve these or other similar problems.

3.1. *Basic mathematical programming models*

The general static cell formation problem can be described as follows. A number of different types of parts are to be processed in a manufacturing system consisting of a number of machines. Each type of part may require some or all of the machines for processing. Machines are to be grouped into relatively independent cells with certain criteria to be optimized (minimized or maximized). Such criteria are typically material flows between machines inside a cell as well as those among the cells. Sometimes, the criteria, or objectives, may include machine cost, equipment cost and other costs.

Let us consider the objective to minimize inter-cell material flows only. This criterion is normally expressed by the summation of the distances multiplied by the materials transported among the cells. To formulate this objective function, we may consider two different situations:

(i) The cells are located at different places and the differences of the distances between the cells are significant; and
(ii) Same distances among the cells are assumed.

For the first scenario as described above, a non-linear quadratic function may be used to express the total material handling. This is similar to that presented in Atmani *et al.*[2] Let x_{ijkl} be a 0-1 variable. $x_{ijkl} = 1$ if operation j of part i is processed by machine k in cell l; otherwise, $x_{ijkl} = 0$. Note that subscripts ijk of the variable x_{ijkl} indicate that machine k is required to process the operation j of part i. This information is known from the part process plan. Let $D_{ll'}$ be the distance between cells l and l', P_i be the units of part i to be processed in the system and CT_i be the unit inter-cell material handling cost. Then the total inter-cell material handling cost can be expressed by:

$$\sum_{i=1}^{I} CT_i P_i \sum_j \sum_k \sum_l D_{ll'} x_{ijkl} x_{i(j+1)k'l'}. \tag{1}$$

In developing cell formation models, one may also want to minimize the cost of the machines in the system used to process the parts. If we let z_{kl} be the number of type k machines to be placed in cell l and H_k be the cost of having one unit of machine k in the system, then the machine cost can be expressed by:

$$\sum_k H_k \sum_l z_{kl}. \tag{2}$$

The cell design model can be developed to minimize the summation of the above two cost items that are subjected to certain constraint conditions. One such condition is that an operation of any part can only be processed by one of the machines in one of the cells. This constraint can be expressed by:

$$\sum_k \sum_l x_{ijkl} = 1. \tag{3}$$

Other constraints may be developed to express various conditions in the cellular manufacturing system. For example, let A_{ijk} be the capacity requirement for machine k to process the operation j of part i, then the relationship between variables x_{ijkl} and z_{kl} would be:

$$\sum_i \sum_j A_{i,j,k} x_{ijkl} \leq z_{kl}. \tag{4}$$

For any manufacturing system, the machines of any type are normally limited. Let M_k be the number of available type k machines in the system, then the following constraint should be imposed:

$$\sum_{l=1}^{L} z_{kl} \leq M_k. \tag{5}$$

In addition, the cell sizes are also limited and only a certain number of machines can be placed in a particular cell. We also need a minimum number of machines for each cell, otherwise the cell may disappear. Let LB_l and UB_l be the minimum and maximum numbers of machines a cell can have, then the cell size constraint can be expressed by:

$$LB_l \leq \sum_k^K z_{kl} \leq UB_l. \tag{6}$$

The above discussed objective function and the four constraint inequalities are the required relations in formulating a basic mathematical programming model for cell formation. The model can be written as follows:
SCF-1

$$\min \sum_{i=1}^{I} CT_i P_i \sum_j \sum_k \sum_l D_{ll'} x_{ijkl} x_{i(j+1)k'l'} + \sum_k H_k \sum_l z_{kl}.$$

Subject to:

$$\sum_k \sum_l x_{ijkl} = 1,$$

$$\sum_i \sum_j A_{ijk} x_{ijkl} \leq z_{kl},$$

$$\sum_{l=1}^{L} z_{kl} \leq M_k,$$

$$LB_l \leq \sum_k^K z_{kl} \leq UB_l,$$

x_{ijkl} are 0-1 variables,

z_{kl} are general integer variables.

The solution of SCF-1 determines the units of different types of machines placed in each cell and the processing of the part taking place in the cells. In fact, the objective function in SCF-1 is similar to that of the well understood quadratic assignment problem (QAP) model. The SCF-1 model can be extended to include other features of specific problems of study. Such possible extensions will be discussed later in this chapter. Unfortunately, this easy and straightforward model is not simple enough to be solved with general purpose integer programming methods. The formulation of a real-world problem usually has a large number of integer variables and the computational amount required to solve such problems increases exponentially to the number of integer variables. Using a general branch-and-bound approach to solve this NP-hard problem is computationally prohibitive in practical applications. In addition, there are two more complicating factors in this model. The first is that the nonlinear objective function needs to be linearized before any general methods can be used. Since the quadratic items in the objective function only has 0-1 integer variables, linearization is relatively simple. For example, for any quadratic item composed of x_1 and x_2, let $w_{1,2}$ be a new 0-1 variable, then the nonlinear items x_1x_2, can be replaced by $w_{1,2}$ with two added constraint inequalities:

$$w_{12} \geq x_1 + x_2 - 1,$$
$$w_{12} \leq \frac{x_1 + x_2}{2}.$$

This linearization process is simple and can be found in many standard operations research text books (e.g. in Ravindran *et al.*[19]). The real difficulty, however, is that the added integer variables and constraints increase the computational time significantly as shown in our experiences. From this point of view, non-linear integer functions should be avoided in formulating cell formation problems if possible. In fact, a reasonable cell formation model can be developed using linear terms only. This is discussed next.

Let s_{il} be a 0-1 integer variable to represent part-cell relationships. $s_{il} = 1$ if any operation of part i is processed by a machine placed in cell l, and $s_{il} = 0$ otherwise. If we further assume that the distance the part travels between every two cells are the same or the differences are insignificant and negligible, then a simple function to reflect the total travel cost can be written as:

$$\sum_{i=1}^{I} CT_i P_i \left[\sum_{l} s_{il} - 1 \right], \tag{7}$$

where P_i and CT_i are the units of type i part to be produced in the system and the unit inter-cell material handling cost, respectively. These are the same as in model SCF-1. This function indicates that if part i is to be processed in 2 cells, then there is 1 inter-cell movement; if it is to be processed in 3 cells, there are 2 inter-cell movements; etc. This function does not have non-linear quadratic items. However, there are some other sets of constraints in the optimization model to represent the relationship among operations, parts, machines and cells. Let x_{ijkl} be the same 0-1

variable as defined in SCF-1, then the minimization of the cost function in Eq. 7 is constrained by the following inequality:

$$x_{ijkl} \leq s_{il}. \tag{8}$$

This constraint function ensures that if any operation j of part i is to be processed by machine k allocated to cell l, part i must stay in cell l for processing. The corresponding s_{il} then will not be 0. All other constraint functions in SFC-1 must be in effect in this new model as given below:
SCF-2

$$\min \sum_{i=1}^{I} CT_i P_i \left[\sum_{l} s_{il} - 1 \right] + \sum_{k} H_k \sum_{l} z_{kl}.$$

Subject to:

$$x_{ijkl} \leq s_{il},$$

$$\sum_{k} \sum_{l} x_{ijkl} = 1,$$

$$\sum_{i} \sum_{j} A_{ijk} x_{ijkl} \leq z_{kl},$$

$$\sum_{l=1}^{L} z_{kl} \leq M_k,$$

$$LB_l \leq \sum_{k}^{K} z_{kl} \leq UB_l,$$

x_{ijkl}, s_{il} are 0-1 variables,

z_{kl} are general integer variables.

In fact, SCF-2 can be further simplified, for example, if we can substitute the rather complicated machine capacity constraint in SCF-1 by

$$x_{ijkl} \leq z_{kl}, \text{ where}$$

z_{kl} are 0-1 variables.

The relaxed machine requirement constraint implies that each unit of any machine has unlimited processing capacity. There would be no multiple units of any type of machine in any cell. If the machine cost item in the objective function is removed, then the model will tend to have one unit of every type of machine in each of the cells if it does not exceed the total machine limit. In the SCF-3 model presented below, however, this will not happen since there is a trade-off between having more machines and less material handling. The model is given below:
SCF-3

$$\min \sum_{i=1}^{I} CT_i P_i \left[\sum_{l} s_{il} - 1 \right] + \sum_{k} H_k \sum_{l} z_{kl}.$$

Subject to:

$$x_{ijkl} \leq s_{il},$$

$$\sum_k \sum_l x_{ijkl} = 1,$$

$$x_{ijkl} \leq z_{kl},$$

$$\sum_{l=1}^{L} z_{kl} \leq M_k,$$

$$LB_l \leq \sum_k^K z_{kl} \leq UB_l,$$

x_{ijkl}, z_{kl}, s_{il} are 0-1 variables.

SFC-3 is a much simplified model as compared to SFC-1. Although it still has a large number of integer variables, experiences showed that it took much less computational time to find the optimal solutions for SFC-3 rather than for SFC-1, if the problems are of similar sizes. The characteristics of the three models can be summarized below.

- SCF-1 considers the different distances between cells. SCF-2 and SCF-3 assume that the distances are same.
- SCF-1 and SCF-2 consider limited machine capacity. This may result in multiple units of the same type machine in one cell. SCF-3 assumes that the machine capacity is unlimited hence the maximum number of any type machine in a cell is one.
- SCF-1 has a large number of integer variables after linearization and it has complicated constraint functions. SCF-3 is, perhaps, the simplest mathematical programming model for cell formation problem formulation.

LINGO programs of the above three models are provided in the Appendix for three similar example problems. Computational features of these models will be discussed later in this chapter.

3.2. *Model variations*

The cell formation models discussed above can be further extended to include many other manufacturing features. Some of these extensions are discussed below.

3.2.1. *Intra-cell material handling*

In formulating cell formation models, intra-cell material handling can also be considered. Let CR_i be unit cost for the material handling of part type i between two machines inside any cell, then the total intra-cell material handling cost can be

expressed by:

$$\sum_{l}^{L}\sum_{i=1}^{I} CR_i P_i \left[\sum_{j} x_{ijkl} - 1\right]. \tag{9}$$

In the above expression, P_i and x_{ijkl} are the same as those defined in SCF-1. This intra-cell material cost is based on the consideration that the distances between machines inside any cell are the same or the differences are negligible. If the distance differences are significant, then the mathematical expressions developed for facilities layout problems (e.g. in Heragu[10]) may be more appropriate. The cell formation model, however, could be much more detailed and complicated. If the above cost function is included in the objective function in SCF-1, SCF-2 or SCF-3, it will not increase the number of integer variables nor the number of constraints.

3.2.2. *Production sequence and scheduling*

If production sequence is considered, then other constraint functions should be added into the model. Let y_{ijk} be the starting time of operation j of part i by machine k, and let T_{ijk} be the required time to process operation j of part type i using machine k. It is obvious that this parameter is directly related to the machine capacity requirement parameter A_{ijk}, and hence $T_{ijk} = T(A_{ijk})$. The following constraint function can be added into SCF-1, SCF-2 or SCF-3, if scheduling is a concern of the problem:

$$y_{ijk} + T_{ijk}\left(\sum_{l}^{L} x_{ijkl}\right) \leq y_{i(j+1)k'}. \tag{10}$$

In fact, this additional constraint will not affect the computational results of the cell formation problems unless a cost function reflecting such a requirement is added into the objective function.

3.2.3. *Multiple process plans*

The main decision variables x_{ijkl} in the above models are defined for part i, operation j, machine k and cell l. In presenting SCF-1 and other models, we assume that subscript k is dependent on subscript j. In other words, the type of machine to process an operation is known once the operation is given. In many situations, there is the possibility that an operation of a part can be processed by different machines. That is, the manufacturing system has certain flexibility with respect to machining processes.[24] In this case, the same symbol for the variable x_{ijkl} still can be used while it may be interpreted in two different ways:

(i) The subscripts j and k are related by a number of process plans. An operation j of a part i may be processed by machine k or machine k' corresponding to different process plans. Once a process plan is decided, however, the machines

to process the operations of a part are determined. In solving this type of problems, decisions are made for selecting the most appropriate process plan from a given number of process plans for the parts.

(ii) The subscripts j and k are independent to each other. An operation of a part may be processed by a number of machines specified for that operation. In solving this type of problems, decisions are made to select the most appropriate machines for the operations. The process plan for a part is to be generated along with the other parts of the problem solution.

Process flexibility can be executed in the formulating of cell formation problems under any of the above discussed ways. Such models will be similar to SCF-1, SCF-2 or SCF-3 with additional constraints related to the given multiple process plans or the set of eligible machines for a specific operation. In the case that no process plans are given while an operation j of part i can be processed by a number of machines, the following equation can be used to substitute the first constraint in SCF-1 (or other corresponding constraints in SCF-2 and SCF-3):

$$\sum_{l=1}^{L} \sum_{k \in K_{ij}} x_{ijkl} = 1, \tag{11}$$

where K_{ij} is the set of machines eligible for processing operation j of part i. This constraint is used to ensure that only one machine in one of the cells will be assigned to an operation j. If a number of alternative process plans are specified, similar but more complicated constraint equations can be developed.

3.3. *Solution methods*

Different methods have been proposed to solve cell formation problems as discussed above. The typical ones are the direct optimization methods, the relaxation-based integer programming methods and the heuristic search methods such as simulated annealing, genetic search and Tabu search.

3.3.1. *Direct method*

By "direct method" we mean that the model is solved directly by a commercial software designed for solving general integer programming problems. Computer software packages such as LINDO, CPLEX or other similar software are widely available. The direct methods are easy to understand and they can provide optimal solutions without the comprehensive analysis of the model nor the extensive development of solution methods based on specific problem features. However, on the other hand, direct methods may not be applied to solve real life large scale problems due to the NP-hardness of most integer programming problems (Nemehauser and Wosley[16]). In the later sections of this chapter, we will present a number of examples to illustrate the computational difficulties of using direct methods in solving different cell formation models as discussed in this chapter.

One of the common approaches to overcome the difficulty of NP-hardness in solving large scale integer programming models is model relaxation. For example, if certain sets of the constraint functions in a model are relaxed or not enforced, the model may be solved through the solving of a series of simpler and smaller models. These simpler models are the results of relaxation and, possibly, decomposition. The solutions of each of the sub-problems must be properly assembled to obtain the optimal or near optimal solutions of the original problem. Such an approach will be briefly discussed next.

3.3.2. *Relaxation-based methods*

As we have discussed above, the use of direct method in solving practical cell formation problems is very limited. The exception may be SCF-3. Sometimes SCF-3 may be too simple to be realistic. On the other hand, if a more complicated problem can be relaxed and solved through solving a series of sub-problems similar to SCF-3, then it could be a practical problem solving approach.

Relaxing the integer requirement for some of the variables can be useful for solving SCF-2. Our computational experience shows that SCF-2 can be solved with CPU times similar to that for solving SCF-3, if z_{kl}, the number of different machines in each cell, are considered as continuous variables. A branch-and-bound procedure must then follow to generate the integer solutions for the optimum integer number of machines in the cells.

Alternative relaxing methods may be used such as the one proposed in Dahel.[6] This method is to relax the machine capacity requirement by letting the aggregate processing requirement be restricted by the total number of machines in a cell, rather than by the individual types of machines. If the result happens to satisfy the requirements, then the problem is solved. If it does not, the constraint will be modified based on the relaxed problem solution and the model will be resolved with the added constraints. It will continue until all the violated constraints are resolved. There is no guarantee that the final solution will be optimal after solving a series of relatively easy problems similar to and simpler than SCF-3.

A more systematic and popular relaxation approach is the Lagrangian relaxation. This method allows one to multiply the more difficult constraints with the Lagrangian multipliers and then move the summation of the products to the objective function from the constraint set. For example, if the machine requirement constraints in SCF-2 are relaxed following this approach, SCF-2 becomes:

SCF-2-L(λ)

$$\min \sum_{i=1}^{I} CT_i P_i \left[\sum_{l} s_{il} - 1 \right] + \sum_{k} H_k \sum_{l} z_{kl} + \sum_{k} \sum_{l} \lambda_{kl} \left[\sum_{i} \sum_{j} A_{ijk} x_{ijkl} - z_{kl} \right].$$

Subject to:

$$x_{ijkl} \leq s_{il},$$

$$\sum_k \sum_l x_{ijkl} = 1,$$

$$\sum_{l=1}^{L} z_{kl} \leq M_k,$$

$$LB_l \leq \sum_k^K z_{kl} \leq UB_l,$$

$$x_{ijkl}, z_{kl} \text{ are 0-1 variables}, \lambda_{kl} \geq 0.$$

Presumably, solving SCF-2-L(λ) is much simpler as compared to SCF-2 since the constraint set is simpler than that in SCF-3. However, the solution of SCF-2-L(λ) corresponds to the set of λ_{kl} used in the model. Sub-gradient search, a standard process to update the set of λ_{kl} based on the current solution, can be used to solve SCF-2-L(λ) iteratively. Detailed description of the Lagrangian relaxation and the sub-gradient search can be found in, for example, the book by Nemhauser and Wolsey.[16] The Lagrangian relaxation is a general problem solving approach and it can be used for solving any mathematical programming problems if they involve complicated constraint functions.

3.3.3. *Optimization-based heuristic search*

This type of method includes simulated annealing, Tabu search and genetic search. The purpose of such search methods is to conduct a limited number of evaluations in search for the global optimum. In using simulated annealing, for example, the search may be set intentionally to a less attractive local direction. That is, the next solution is moved away from the best known solution in the hope that global optimum could be reached along the locally unattractive direction. The idea of a Tabu search is similar at situations where some directions of search are not allowed. Genetic search is used to emulate the natural evolution process with much faster paces. The solutions of a binary integer programming problem can be viewed as a long string of 0-1's. Consider this string as a chromosome or gene. With the evolution and mutation processes taking place, the species evolves to the shape best suited for the environment. The genetic search is to emulate this process with an objective function. The value of the objective function can be used to measure the performance of the solution as it evolves and mutates. All such search methods involve certain randomness and they require pseudo-random number generators. In addition, a set of search parameters need to be pre-determined. These types of search have produced interesting and encouraging results. Sometimes, however, results from such search experiments may be difficult to generalize since the parameters used are specific to the problems in question. Detailed discussion on the applications of

such search methods can be found in the papers by Alfa *et al.*,[1] Joines *et al.*[13] and Kolahan.[14]

3.4. *Summary*

In summary, mathematical programming is a powerful tool for modeling various manufacturing system problems including manufacturing cell formation problems. The difficulty may lie in solving these models since many of them involve a large number of integer variables. Unless they are very simplified such as those in SCF-3, it is almost impossible to find the optimal solutions for problems of reasonable sizes using widely available computing facilities such as the PC computers. Different methods have been developed to solve these problems. They include the relaxation-based approaches and the heuristic search to find the sub-optimal and near-optimal solutions without using explosive amount of computation. If relaxation-based methods are specifically developed for different models, they could be very effective in solving some specific problems. On the other hand, more studies may be needed in the application of various search methods for solving combinatorial optimization problems, including many manufacturing cell formation problems.

4. Dynamic Cell Formation Problems

In the previous section, we discussed some simple mathematical programming models for cell formation. A common assumption in developing these models is that the types and units of parts to be processed as well as the cellular manufacturing system itself are relatively stable. As we mentioned before, this assumption may not be very realistic. With the increasing demand for more production flexibility and efficiency, today's manufacturing systems are dynamic rather than static. In a dynamic manufacturing environment, it is very likely that production demand or part mix may change frequently and the manufacturing cells have to be modified from time to time. The optimal cell configuration generated from earlier data may not be valid after such changes occur in the system. System changes such as adding or removing machines may be required in order to efficiently operate the system in a different manufacturing environment. This problem can be addressed by using more comprehensive mathematical models including time varying aspects of the system.

4.1. *Mathematical programming models*

A dynamic manufacturing system can be considered for a number of time periods t, where $t = 1, 2, \ldots, T$ with $T > 1$. One time period could be a month, a season or a year. The types and number of parts to be processed by the machines may vary with t. An optimized cellular manufacturing system for $t = 1$ may have to be reconfigured for time $t = 2$ and other time periods. If such reconfiguration is necessary, new machines may be acquired, existing machines may be removed or moved from current cells to different cells. Such reconfiguration cost must be

included into the cost function in addition to the inter-cell part travel and machine operation costs as we had worked out when we present the models in Sec. 3.

The objective of the dynamic models considered in this section is to minimize the overall inter-cell part travel cost, the machine holding cost and the machine moving cost for the entire planning time horizon T. A cost function similar to the one shown below may be used in a dynamic cell formation model:

$$\begin{aligned} M(\mathbf{x}(t), \mathbf{y}(t), \mathbf{z}(t)) = & \sum_{t=1}^{T} \sum_{i=1}^{I(t)} f_i(t) P_i(t) \sum_j \sum_k \sum_l D_{ll'} x_{ijkl}(t) x_{i(j+1)k'l'}(t) \\ & + \sum_{t=1}^{T} \sum_{k=1}^{K(t)} H_k(t) \sum_{l=1}^{L} z_{kl}(t) \\ & + \sum_{t=1}^{T-1} \sum_{k=1}^{K(t)} \left[I_k^+ \sum_{l=1}^{L} y_{kl}^+(t) + I_k^- \sum_{l=1}^{L} y_{kl}^-(t) \right]. \end{aligned} \tag{12}$$

Similar to those in the static model SCF-1 in the previous section, the first item of the objective function is the inter-cell part travel cost and the second item is the holding and operating cost needed to maintain the required machines in the system. The only difference is that they are time varying parameters with an additional dimension of t. The third item is the system reconfiguration cost which did not exist in the static models. $y_{kl}^+(t)$ and $y_{kl}^-(t)$ are the number of type k machines to be added/removed from machine cell l at the end of time t. At the end of time t, if a new machine is to be installed then all the items in the objective function may be affected with an additional machine cost $H_k(t)$ and installation cost I_k^+. If there is no need to acquire a new machine but an existing machine needs to be moved from its current cell to another one, then the moving cost is $I_k^+ + I_k^-$, with the total number of machines remaining the same.

Some of the constraints for such models are the same as those in the static models except that the corresponding variables and parameters have an extra dimension t. These constraint functions are given below:

$$\sum_{l=1}^{L} x_{ijkl}(t) = 1, \tag{13}$$

$$\sum_i \sum_j A_{ijk} x_{ijkl}(t) \leq z_{kl}(t), \tag{14}$$

$$\sum_{l=1}^{L} z_{kl}(t) \leq M_k(t), \tag{15}$$

$$LB_l \leq \sum_k^K z_{kl}(t) \leq UB_l. \tag{16}$$

In the dynamic model there are added constraint functions to relate the dynamic features of the problem. One of them is that the number of machines in different

time t in the cells may change with changing production demand. This can be expressed by the following coupling constraint relating the number of machines in cell l at times t and $t+1$:

$$z_{kl}(t+1) = z_{kl}(t) + [y_{kl}^{+}(t) - y_{kl}^{-}(t)]. \tag{17}$$

Summarizing the objective and constraint functions, we can present the integer programming model of the dynamic cell formation problem as follows:
DCF-1

$$\begin{aligned}\min M(\mathbf{x}(t), \mathbf{y}(t), \mathbf{z}(t)) = & \sum_{t=1}^{T}\sum_{i=1}^{I(t)} f_i(t)P_i(t)\sum_{j}\sum_{k}\sum_{l} D_{ll'}x_{ijkl}(t)x_{i(j+1)k'l'}(t) \\ & + \sum_{t=1}^{T}\sum_{k=1}^{K(t)} H_k(t)\sum_{l=1}^{L} z_{kl}(t) \\ & + \sum_{t=1}^{T-1}\sum_{k=1}^{K(t)}\left[I_k^{+}\sum_{l=1}^{L} y_{kl}^{+}(t) + I_k^{-}\sum_{l=1}^{L} y_{kl}^{-}(t)\right],\end{aligned} \tag{18}$$

such that

$$\sum_{l=1}^{L} x_{ijkl}(t) = 1, \tag{19}$$

$$\sum_{i}\sum_{j} A_{ijk}x_{ijkl}(t) \le z_{kl}(t), \tag{20}$$

$$\sum_{l=1}^{L} z_{kl}(t) \le M_k(t), \tag{21}$$

$$LB_l \le \sum_{k}^{K} z_{kl}(t) \le UB_l, \tag{22}$$

$$z_{kl}(t+1) = z_{kl}(t) + [y_{kl}^{+}(t) - y_{kl}^{-}(t)],\ t = 1, \cdots, T-1, \tag{23}$$

$$\text{where } x_{ijkl}(t) \text{ are 0-1 variables, and} \tag{24}$$

$$z_{kl}(t), y_{kl}^{+}(t), y_{kl}^{-}(t) \text{ are the general integer variables.} \tag{25}$$

In the above presented model, DCF-1 is similar to its static counterpart SCF-1 in the previous section. The main differences are the additional time dimension t and the corresponding items in the objective and constraint functions. If we assume the same material travel distance among the cells, then the non-linear items in the objective function can be replaced by linear ones. This will be similar to the static model SCF-2. If we further assume unlimited machine capacities, the model can be reduced again and eventually becomes similar to SCF-3.

Various features can also be added into the dynamic cell formation models. The models with such added features will become similar to the static models as discussed in the previous section.

4.2. *Solution methods*

Some of the solution methods developed to solve static cell formation problems can also be used to solve dynamic problems. However, since dynamic models involve a larger number of integer variables as well as more complicated functions, some of these methods may not be effective in solving dynamic problems.

4.2.1. *Direct methods*

The DCF-1 model has all the complicating features similar to the SCF-1. Furthermore, it has extra dimensions for most of its variables, more objective function items and additional constraint functions coupling different time periods. In the search for solution methods for the SCF-1, we found that it is almost impossible to find optimal solutions by just using widely available computers to solve practical problems. Since the DCF-1 is much more complicated than the SCF-1, we do not expect the DCF-1 to be effectively solved by direct methods. We have investigated the possibility of using direct methods to solve a dynamic model similar to SCF-3, which is the simplest among the three static models. We have limited our problems to a very small size but still we could not reach the optimal solution after performing long calculations. It seems that it is much more difficult to use direct methods to solve any dynamic cell formation optimization models. One may have to rely on decomposition, discrete simulation, heuristics, etc. to solve such problems.

4.2.2. *Decomposition-based methods*

If the time-varying items in the objective function and the corresponding constraint functions are removed from the dynamic model, it may break up into a number of static models of smaller sizes. If these smaller sub-models can be solved efficiently using any of the methods discussed in the previous section, the solution of the original dynamic model may also be found through proper composition of the sub-problem solutions. In this section, we introduce one such method based on the following dynamic cell formation model. This model is similar to the SCF-2, its static counter-part. The complete model presentation is given below.

DCF-2

$$\min \sum_{t=1}^{T} \sum_{i=1}^{I(t)} CT_i P_i(t) \left[\sum_{l} s_{il}(t) - 1 \right] + \sum_{t=1}^{T} \sum_{k=1}^{K(t)} H_k(t) \sum_{l=1}^{L} z_{kl}(t) + \sum_{t=1}^{T-1} \sum_{k=1}^{K(t)} \left[I_k^+ \sum_{l=1}^{L} y_{kl}^+(t) + I_k^- \sum_{l=1}^{L} y_{kl}^-(t) \right], \tag{26}$$

such that

$$\sum_{l=1}^{L} x_{ijkl}(t) = 1, \tag{27}$$

$$\sum_{i}\sum_{j} A_{ijk}x_{ijkl}(t) \leq z_{kl}(t), \tag{28}$$

$$\sum_{l=1}^{L} z_{kl}(t) \leq M_k(t), \tag{29}$$

$$LB_l \leq \sum_{k}^{K} z_{kl}(t) \leq UB_l, \tag{30}$$

$$z_{kl}(t+1) = z_{kl}(t) + [y_{kl}^{+}(t) - y_{kl}^{-}(t)],\ t = 1, \ldots, T-1, \tag{31}$$

$$\text{where } x_{ijkl}(t), \text{ are 0-1 variables, and} \tag{32}$$

$$z_{kl}(t), y_{kl}^{+}(t), y_{kl}^{-}(t) \text{ are the general integer variables.} \tag{33}$$

If we remove the third item in the objective function and the coupling constraint of Eq. 31, the model will be fully decomposed for each t into static sub-problems which are similar to SCF-2. These sub-models can be written as below:
DCF-2-D(t)($t = 1, \cdots, T$):

$$\sum_{i=1}^{I(t)} CT_i P_i(t) \left[\sum_{l} s_{il}(t) - 1\right] + \sum_{k=1}^{K(t)} H_k(t) \sum_{l=1}^{L} z_{kl}(t), \tag{34}$$

such that

$$\sum_{l=1}^{L} x_{ijkl}(t) = 1, \tag{35}$$

$$\sum_{i}\sum_{j} A_{ijk}x_{ijkl}(t) \leq z_{kl}(t), \tag{36}$$

$$\sum_{l=1}^{L} z_{kl}(t) \leq M_k(t), \tag{37}$$

$$LB_l \leq \sum_{k}^{K} z_{kl}(t) \leq UB_l, \tag{38}$$

$$x_{ijkl}(t), z_{kl}(t) = 0, 1. \tag{39}$$

As we have discussed before, the above static sub-models may be solved by different methods such as direct method, heuristics, discrete simulation, etc. After we obtain optimal or near-optimal solutions for each of the above static sub-problems, we need to develop a certain procedure to compose the sub-problem solutions into the solution for the original dynamic problem.

Assuming that all these sub-problems have feasible solutions and we let $\mathbf{f}(t)$ and $\mathbf{z}^*(t)$ be the set of production demand and the best cell configuration, respectively, of time period t. It is clear that $\mathbf{z}^*(t)$ for different t are not necessarily the same since $\mathbf{f}(t)$ and inter-cell travel costs may change with t. Let $S[\mathbf{f}(t), \mathbf{z}^*(t)]$ be the inter-cell travel cost and machine cost found from the solution of DCF-2-D(t). If it is feasible to meet the demands of time period t using the best cell configuration of a different time period t', then a corresponding cost value $S[\mathbf{f}(t), \mathbf{z}^*(t')], t' \neq t$, can be calculated. In this case, it is possible that the overall cost without changing the system is lower than the cost incurred in a changed system. For instance, let $S[\mathbf{f}(1), \mathbf{z}^*(1)]$ and $S[\mathbf{f}(2), \mathbf{z}^*(2)]$ be the minimum inter-cell part travel and machine costs at times $t = 1$ and $t = 2$, respectively. If both the combinations $[\mathbf{f}(2), \mathbf{z}^*(1)]$ and $[\mathbf{f}(1), \mathbf{z}^*(2)]$ are feasible, then there is a choice among $S[\mathbf{f}(1), \mathbf{z}^*(1)] + S[\mathbf{f}(2), \mathbf{z}^*(1)]$, $S[\mathbf{f}(1), \mathbf{z}^*(1)] + Q(1,2) + S[\mathbf{f}(2), \mathbf{z}^*(2)]$ and $S[\mathbf{f}(1), \mathbf{z}^*(2)] + S[\mathbf{f}(2), \mathbf{z}^*(2)]$ for the first two-time periods, where $Q(1,2)$ is the cost required to change the cell configuration from $\mathbf{z}^*(1)$ to $\mathbf{z}^*(2)$. In fact, making the best decision from a number of such combinations for the entire time horizon T is equivalent to solving a dynamic programming problem. The network form of such dynamic programming problem with $T = 3$ is shown in Fig. 2.

In the network shown in Fig. 2, $S(t, t')$ corresponds to the inter-cell travel and machine costs for production at time period t with the optimal cell configuration designed for time period t'. $Q(t', t'')$ is the reconfiguration cost required to change the system from t' to t'', where $t', t'' = 1, 2, \cdots, T$. $Q(t', t'')$ can be calculated as

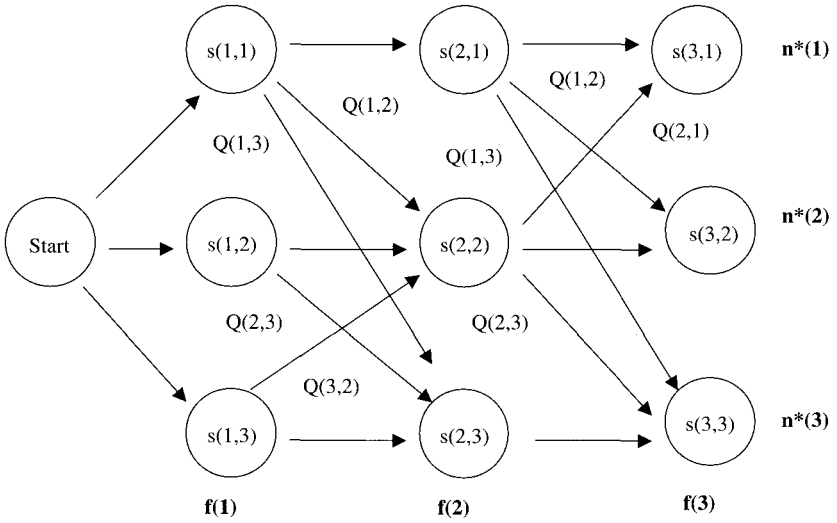


Fig. 2. Dynamic programming network.

follows:

$$Q(t',t'') = \sum_{k=1}^{\bar{K}(t',t'')} I_k^+ \sum_{l=1}^{L} \max\{z_{kl}(t'') - z_{kl}(t'), 0\} + \sum_{k=1}^{\bar{K}(t',t'')} I_k^- \sum_{l=1}^{L} \max\{z_{kl}(t') - z_{kl}(t''), 0\},$$

where $\bar{K}(t',t'') = \max\{K(t'), K(t'')\}$. Note that on all the horizontal line segments in the network shown in Fig. 2, $Q(t',t'') = Q(t',t') = 0$. The sub-network corresponding to any two immediately linked stages (t and $t+1$) of the dynamic programming problem is fully connected with two exceptions. The first is that at the beginning, there is only one link from the first stage to $S(2,1)$ since $S(1,1)$ is the cost function value of the best solution at stage 1. The other exception is that at the end, there is only one link pointing out from $S(T-1,T)$ since $S(T,T)$ corresponds to the best solution at the final stage T ($T = 3$ in Fig. 2). Such network characteristics can be viewed more clearly in a 4-stage ($T = 4$) network as shown in Fig. 3.

The general dynamic programming formulation can easily be developed. Let $W^*(t,t')$ be the optimal objective function value corresponding to node (t,t') in the network at stage t. The solution of this optimization problem can then be found by recursively solving the following equation:

$$W^*(t,t') = \min_{\tau \in \{1,\ldots,T\}} \{W^*(t-1,\tau) + Q(\tau,t')\} + S(t,t'). \tag{40}$$

One may notice that some of the combinations $[\mathbf{f}(t), \mathbf{z}^*(t')], t' \neq t$, may not be feasible. Since we assume that the sub-problems do have optimal solutions, there exists at least one feasible solution for the original problem. This feasible solution corresponds to the diagonal line in the network in Figs. 2 or 3 from $S(1,1), S(2,2), \ldots,$

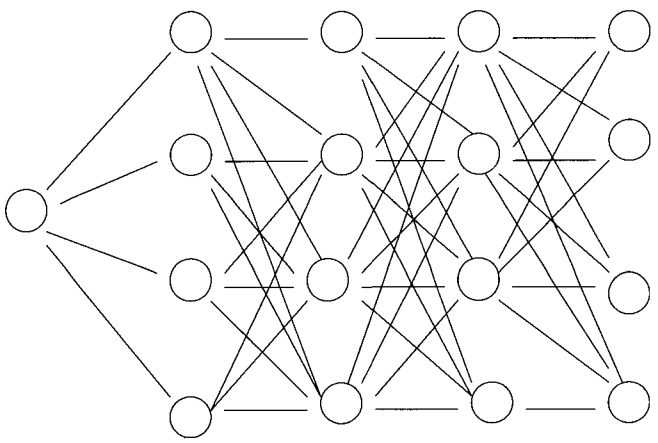


Fig. 3. General network connectivity.

to $S(T,T)$ with the total cost being:

$$\sum_{t=1}^{T} S[\mathbf{f}(t), \mathbf{z}^*(t)] + \sum_{t=1}^{T-1} Q(t, t+1) = UB_s,$$

where UB_s is an upper bound of the objective function of the original problem. If the solution $\mathbf{z}^*(t)$ for each sub-problem of time t is optimal, then a lower bound of the original problem can also be identified as:

$$LB_s = \sum_{t=1}^{T} \left\{ S[\mathbf{f}(t), \mathbf{z}^*(t)] + \sum_{k \in L_k^{\mp}(t,t')} \min \left[\sum_{\tau=t}^{t'} H_k(\tau), I_k^- y_k^-(t) + I_k^+ y_k^+(t') \right] \right.$$
$$\left. + \sum_{k \in L_k^-(t)} \min \left[\sum_{\tau=t}^{T} H_k(\tau), I_k^- y_k^-(t) \right] \right\}. \tag{41}$$

In the above equation, $L_k^{\mp}(t,t')$ is the set of cells where machine k will be removed at the end of time t and restored at the end of time t'. $L_k^-(t)$ is the set of cells where machine k becomes an extra machine from time t to T. This lower bound is the sum of the minimum inter-cell travel cost and the machine cost corresponding to the best solutions of the sub-problems. It also includes the cost to remove temporary and permanent extra machines from the system if the removal and re-installation costs are lower than machine holding cost. The step-by-step procedure of the above solution approach can be outlined below:

Step 1: Using any method discussed in Sec. 3.3 to solve sub-problem DCF(2)-D(t) for each t and find the minimum inter-cell travel and machine costs $S[\mathbf{f}(t), \mathbf{z}^*(t)], t = 1, \ldots, T$. Calculate the system reconfiguration costs $Q(t', t'')$ for changing the cells from $\mathbf{z}^*(t')$ to $\mathbf{z}^*(t'')$, where $t', t'' = 1, \ldots, T$, $t' \neq t''$.

Step 2: Calculate $S(t,t') = S[\mathbf{f}(t), \mathbf{z}^*(t')]$, where $t, t' = 1, \ldots, T, t \neq t'$. Discard any infeasible combinations and let the corresponding $S(t,t') = \infty$.

Step 3: Solve the dynamic programming problem as shown in Eq. 40.

Step 4: Use the solution from the dynamic programming problem from Step 3 as the solution of DCF(2), stop.

4.2.3. *Other solution methods*

Different solution methods, except for the direct method, as discussed in the previous section for solving static cell formation problems may also be used to solve dynamic cell formation problems. The general principles of those methods discussed in the previous section are the same and it can be more challenging to apply them

in solving dynamic problems. These are briefly discussed below:

(i) Relaxation-based methods — These methods are similar to those discussed in the previous section such as the relaxation on integer requirement or the relaxation on certain difficult constraint functions (Lagrangian relaxation). In fact, the decomposition method presented above for solving dynamic problems is also a type of relaxation. In the case that the integer variables are relaxed to continuous variables, the dynamic cell formation models, even in its more complicated formats, will be solvable using widely available PC computers. A branch-and-bound procedure may have to be used in order to find the optimal integer solutions of the real cell formation problem. Lagrangian relaxation can also be used to solve dynamic model problems. In doing so, one must carefully select the constraint functions to be relaxed. The relaxed models should not be too difficult to solve so that a lower bound of the original problem can be found easily.

(ii) Optimization-based heuristic search — These methods such as simulated annealing, Tabu search and genetic search are also similar to those discussed in the previous section. The only difference is that the dynamic problem sizes are larger than the static ones and the search procedures have to be designed in a more careful and systematic way.

(iii) Discrete simulation — This is a general descriptive approach used to solve many practical engineering analysis problems. Since the dynamic cell formation problems are quite complicated and the corresponding mathematical programming models normally have a large number of variables, discrete simulation may be the last resort for detailed system analysis. A preliminary work on cellular manufacturing system design with time-varying demands can be found in Chen and Mensah.[5]

In summary, one may be more cautious in using those methods which are successful in solving static cell formation problems. If it is difficult to use certain methods, such as the direct method, to solve certain static problems, then it is very unlikely that such methods can be used to solve a similar dynamic model.

4.3. *Other modeling approaches*

There are other modeling approaches in solving cell formation problems with dynamic aspects in addition to those discussed in this section. These models address various dynamic aspects of the problem as discussed in the following:

(i) Maximizing cell flexibility — If manufacturing cells are designed with higher flexibility, then it is anticipated that the demand changes in the future can be handled by the current system. One way of doing this is to assign multiple units of the same machine, if needed, to different cells. When part processing demand changes to different combinations and at different levels, the current

system can still be used without much modification due to the diversified types of machines in each cell. Dahel and Smith[7] proposed a bi-criteria mathematical programming model to minimize inter-cell material handling cost and to maximize the total number of machines in the cells. For a model similar to the SCF-1 or the SCF-2, the second objective would be:

$$\max \sum_k \sum_l z_{kl}.$$

Trade-off between the two non-compatible functions must be sought through a series of solutions of the problem. The non-dominated solutions will result in similar cells by having different types of machines in each cell. One assumption in this bi-criteria model is the unavailability of future production demand and data.

(ii) Cell formation for probabilistic demands — If the probability of having certain product mix can be estimated, then a static model to minimize the expected total travel cost can be developed to address the changing production demand. Mathematical programming models along this direction can be found in Seifoddini[20] and Harhalakis *et al.*[9]

(iii) Cell formation and expansion — If different types of machines in the system under consideration are classified as current machines and additional machines, a single model can be constructed to reflect demand changes in the system design. Vakharia and Kaku[27] proposed a mathematical model, similar to the SCF-2, for solving system expansion problems. Machines in the current system and those to be added into the system are explicitly expressed and the solution of the model is used to find the best cell formation for the increased demand.

4.4. *Summary*

Dynamic cell formation problems may be solved using different approaches. One of these approaches involves having the dynamic demands treated explicitly similar to that in models DCF-1 or DCF-2 as discussed extensively in this section. The other approach is to use the aggregate value with certain probability aspects and expectations. The solution of the latter models will be similar to solving the static models while the dynamic features are addressed implicitly. It is apparent that the former approach may produce more detailed results but solving the problem and the corresponding models will be more complicated.

5. Examples

In this section, two examples are discussed to illustrate the modeling and solution approaches presented in the previous sections. Example 1 has four similar problems of different sizes and they are presented to illustrate the models SCF-1, SCF-2 and SCF-3. Example 2 is presented to illustrate the dynamic model. We have limited

the size of the problems in Example 1 so that we are able to solve most of them by the direct optimization method.

Example 1: The three models SCF-1, SCF-2 and SCF-3 were first used to solve a very small problem. In this problem, four different types of machines in three manufacturing cells are used to process five types of parts. Data used in this first example are presented in Tables 1–4. In this 5-part example problem, each part has two operations except that Part 3 has three operations. Table 1 presents the machine requirement for part processing, the machine cost and the maximum number of machines available. Table 2 shows the distances between the three cells in the system and the maximum and minimum number of machines each cell can have. The corresponding LINGO (a user-friendly LINDO-based solution package) code of the SCF-1 model for this example problem is presented in the Appendix for the three examples. This LINGO model is shown after linearization. Table 3 is similar to Table 1 and the material handling costs for different parts are shown in this table. This set of information is required by the SCF-2 model. The LINGO code for

Table 1. Machine and process data for SCF-1 (Example 1).

Machine Number	Machine Cost	Available Number of Machines	Operation-Machine Capacity Requirement				
			Part 1	Part 2	Part 3	Part 4	Part 5
M1	6	10	0.8	0	1.2	0	0.6
M2	9	7	1.1	0.8	0.5	0	0
M3	8	6	0	0	0.0	0.2	0.5
M4	6	8	0	0.6	0.4	0.5	0

Table 2. Cell data for SCF-1 (Example 1).

Cell Distance	Cell 1	Cell 2	Cell 3
Cell 1	0	2	4
Cell 2	2	0	3
Cell 3	4	3	0
Maximum Number Of Machines in Cell	4	4	4
Minimum Number Of Machines in Cell	1	1	1

Table 3. Machine and process data for SCF-2 (Example 1).

Machine Number	Machine Cost	Available Number of Machines	Operation-Machine Capacity Requirement				
			Part 1	Part 2	Part 3	Part 4	Part 5
M1	6	10	0.8	0	1.2	0	0.6
M2	9	7	1.1	0.8	0.5	0	0
M3	8	6	0	0	0.0	0.2	0.5
M4	6	8	0	0.6	0.4	0.5	0
Material Handling Cost			9	7	6	8	8

Table 4. Machine and process data for SCF-3 (Example 1).

Machine Number	Machine Cost	Available Number of Machines	Operation-Machine Capacity Requirement				
			Part 1	Part 2	Part 3	Part 4	Part 5
M1	6	10	1	0	1	0	1
M2	9	7	1	1	1	0	0
M3	8	6	0	0	0	1	1
M4	6	8	0	1	1	1	0
Material Handling Cost			9	7	6	8	8

this model and the example problem are also presented in the Appendix. Table 4 presents the information for the SCF-3 model. Since no machine capacity limitation is considered, the corresponding matrix becomes a 0-1 incidence matrix. We have also reduced the maximum number of machines in a cell from four to two. These small problems can be solved easily using a widely available IBM-PC computer. We then made a slight change to this 5-part problem by adding one additional part with two operations. We have also increased the number of machine types from four to seven, located in four cells. We found that it became much more difficult to use the SCF-1 or the SCF-2 model to solve this slightly larger problem. In fact, we could not find the optimal solution of the SCF-1 model for the 6-part problem after many hours of calculation. It also took a much longer computational time for the SCF-2 to solve the 6-part problem. The increased CPU time for solving the 6-part problem using the SCF-3 model is at a minimum. We have further tested these models using a problem with ten parts. Two of the ten parts have three operations and all the others have two operations. These parts are processed by seven different types of machines in four cells. Finally, we used a 20-part problem consisting of seven machines and four cells. It was modeled and solved by the SCF-3 model, taking more than 90 minutes to reach the optimal solution. The number of integer variables and number of constraints in the SCF-1, SCF-2 and SCF-3 models for these problems are presented in Table 5. Table 5 also shows the computational time required to solve these problems. Computations were conducted using an IBM-PC (Pentium 133) compatible computer. Our experience as well as the results in Table 5 show that the feasibility of using brute-force method to solve such integer programming problems is very limited. For the models involving more complicated objectives or constraint functions such as the SCF-1 and the SCF-2, it is computationally intolerable to use widely available computing equipment to find optimal solutions even for very small problems. Problems of about 20 parts, having two or three operations for each part and involving seven to ten different machines, may be difficult to solve even with the SCF-3. Computational times could vary from a few minutes to many hours and are unpredictable.

In summary, the following points are observed regarding the use of the direct methods and conventional computational facilities for solving typical cell formation

Table 5. Problem and computational features for SCF-1, SCF-2 and SCF-3 (Example 1).

Examples	Key Problem Features	Model		
		SCF-1	SCF-2	SCF-3
5-Part Problem	No. of Int. Var.	87	60	60
	No. of Const.	118	67	88
	CPU (seconds)	410	40	0.02
6-Part Problem	No. of Int. Var.	99	69	69
	No. of Const.	132	75	102
	CPU (seconds)	N/A	987	5
10-Part Problem	No. of Int. Var.	171	117	117
	No. of Const.	225	123	168
	CPU (seconds)	N/A	3220	10
20-Part Problem	No. of Int. Var.	N/A	N/A	316
	No. of Const.	N/A	N/A	484
	CPU (seconds)	N/A	N/A	5668

problems:

(i) One may not rely on the direct method to solve the problems if the modeling and solution approaches are aimed at solving practical problems.
(ii) Computational times required to solve the simplest models such as the SCF-3 of reasonable sizes may be tolerable. Larger sizes and more complex problems will be solvable only with the fast development of computer hardware and software technology.
(iii) The constraints resulted from the linearization of the non-linear integer variables are the complicating ones. They should be avoided, if possible, in the modeling of the problems.
(iv) Machine capacity requirement constraints are also complicating constraints. They should also be avoided in the models or should be the primary targets of relaxation in solving the models.

Example 2: In this example, we consider a dynamic cell formation problem in three time periods. We follow the decomposition approach as discussed in the previous section to solve this problem. First, the main problem is decomposed into three sub-problems corresponding to each time period. These sub-problems can be optimally solved by LINGO on a PC computer for less than a minute of CPU time. In this example, we consider three cells in three time periods ($t = 1, 2, 3$) and seven different types of machines. The minimum and maximum numbers of machines in each cell are 1 and 5, respectively. We now assume that at $t = 1, 2$ and 3, there are 10, 11 and 12 different types of parts to be processed by the set of machines. Part operations and machine requirements for this three time periods are shown in Tables 6, 7 and 8. Machine holding and operating costs are also shown in Tables 6, 7 and 8. For brevity, we assume that the machine costs do not change over different time periods. Material handling costs for different parts among the cells are shown in Table 9.

Table 6. Part operation requirement and machine cost (Example 2, $t = 1$).

Mach. Num.	Mach. Cost	Parts 1	2	3	4	5	6	7	8	9	10
1	6	2					2				1
2	9	1	1			2			2		
3	8		2	2		1			1		2
4	6			3	2			2		2	
5	5						1				
6	7			1							3
7	9				1			1		1	

Table 7. Part operation requirement and machine cost (Example 2, $t = 2$).

Mach. Num.	Mach. Cost	Parts 1	2	3	4	5	6	7	8	9	10	11
1	6	2					2				1	
2	9	1	1			2			2			1
3	8		2	2		1			1		2	
4	6			3	2			2		2		2
5	5						1					
6	7			1							3	
7	9				1			1		1		3

Table 8. Part operation requirement and machine cost (Example 2, $t = 3$).

Mach. Num.	Mach. Cost	Parts 1	2	3	4	5	6	7	8	9	10	11	12
1	6		1			1					1		1
2	9			2					2			1	
3	8	2					2		1		2		
4	6	1		3			1	2		2		2	
5	5			1									2
6	7		2		1						3		3
7	9				2	2		1		1		3	

Table 9. Material handling cost (Example 2).

Time Period	Parts 1	2	3	4	5	6	7	8	9	10	11	12
$t = 1$	3	5	4	7	2	9	3	5	6	2		
$t = 2$	3	5	4	7	2	9	3	5	6	2	12	
$t = 4$	6	9	4	7	3	5	8	2	6	2	10	12

The individual cell formation sub-problems were solved by LINGO in a way similar to solving the problems in Example 1 by SCF-1. The result indicates the part families and the machine groups as shown in Tables 10, 11 and 12 for time periods 1, 2 and 3, respectively. Some of the computational features related to the sub-problems are presented in Table 13. The original problem without decomposition was also coded in LINGO but the optimal solution was not identified after more

Table 10. Part operation requirement and machine cost (Example 2, $t = 1$).

Cell Num.	Mach. Num.	Parts									
		2	3	5	8	4	7	9	1	6	10
	2	1		2	2				1		
1	3	2	2	1	1						2
	6		1								3
2	4		3			2	2	2			
	7					1	1	1			
3	1								2	2	1
	5									1	

Table 11. Part operation requirement and machine cost (Example 2, $t = 2$).

Cell Num.	Mach. Num.	Parts										
		2	3	5	8	4	7	9	11	1	6	10
	2	1		2	2					1		
1	3	2	2	1	1							2
	6		1									3
	4		3			2	2	2				
2	5								2			
	7					1	1	1	3			
3	1									2	2	1
	5										1	

Table 12. Part operation requirement and machine cost (Example 2, $t = 3$).

Mach. Num.	Mach. Cost	Parts											
		3	5	8	11	1	4	6	7	9	2	12	10
	2	2		2	1								
1	5	1	1		2								
	7		2		3								
	3			1		2		2					2
2	4	3				1		1	2	2			
	7						2		1	1			
	1										1	1	1
3	5											2	
	6						1				2	3	3

Table 13. Problem features (Example 2).

Problem	Sub-Problem 1	Sub-Problem 2	Sub-Problem 3	Original Prob.
Num. of Parts	10	11	12	33
Num. of Const.	167	182	203	586
Num. of Var.	124	136	148	529
Num. of Integ.	87	90	105	372
CPU (seconds)	14	38	52	>7200

than two hours of search on the PC computer. Some features of the original problem are presented in the last column of Table 13.

A schematic illustration of the machine cells is shown in Fig. 4. Figure 4 also indicates that there is one machine to be added to change the system from $t = 1$ to $t = 2$. There are seven and six machines to be removed/installed to change the system from times 2 to 3 and from 1 to 3, respectively. Figures 5, 6 and 7 show the dynamic programming networks for solving the original problem with unit machine installation/removal costs being 1, 4 and 5, respectively. Corresponding inter-cell travel costs and machine costs $S(t, t')$ are presented besides the network nodes in these figures. These simple dynamic programming problems can be solved using any standard procedure. The solutions of our 3×3 network problems were found by manual calculations. In the networks shown in Figs. 5, 6 and 7, thick lines show the network paths leading to the overall best solutions of the original problem. The result shown in Fig. 5 requires that the system is changed after every time period when the reconfiguration cost is relatively low ($I^+ = I^- \leq 1$). With the increase

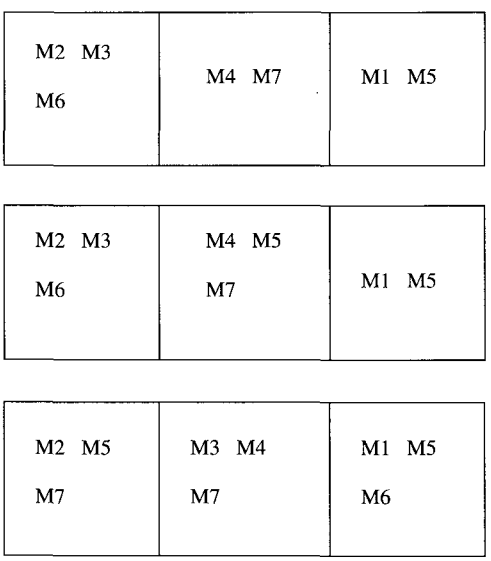


Fig. 4. Cell formation for sub-problems of Example 2.

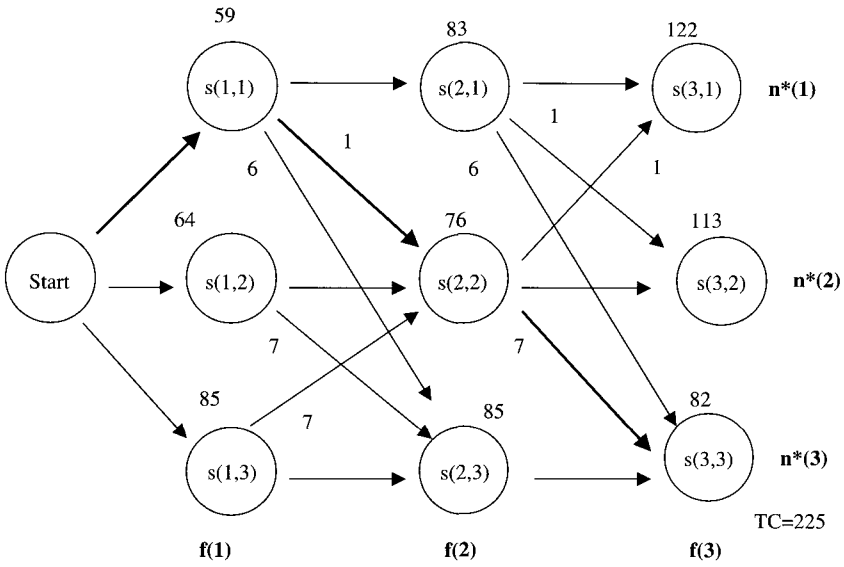


Fig. 5. Solution of Example 2 (unit machine moving cost = 1).

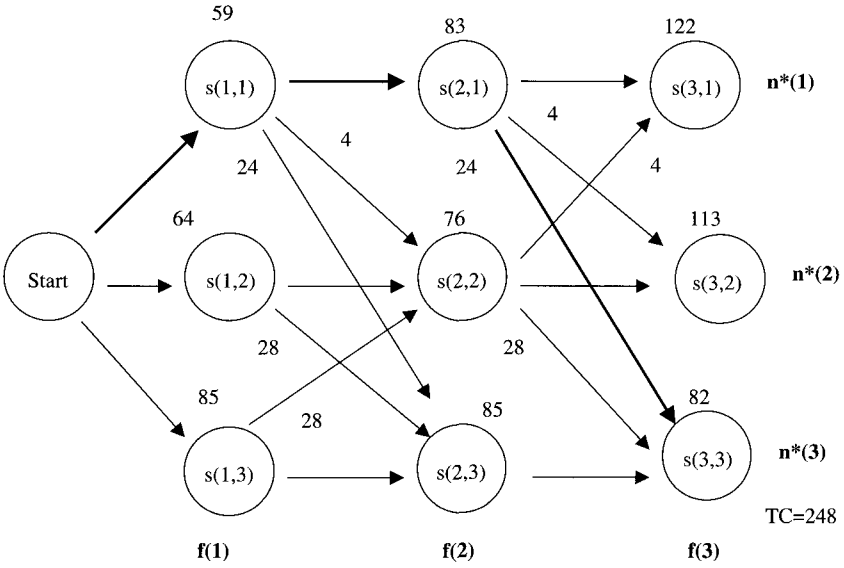


Fig. 6. Solution of Example 2 (unit machine moving cost = 4).

of machine installation/removal cost, a more stable system is a better approach. The system should be re-configured once if $I^+ = I^- = 4$ as shown in Fig. 6. The system should not be changed if $I^+ = I^- \geq 5$, as shown in Fig. 7. The overall costs are 225, 248 and 252 when unit machine removal/installation costs are 1, 4 and 5,

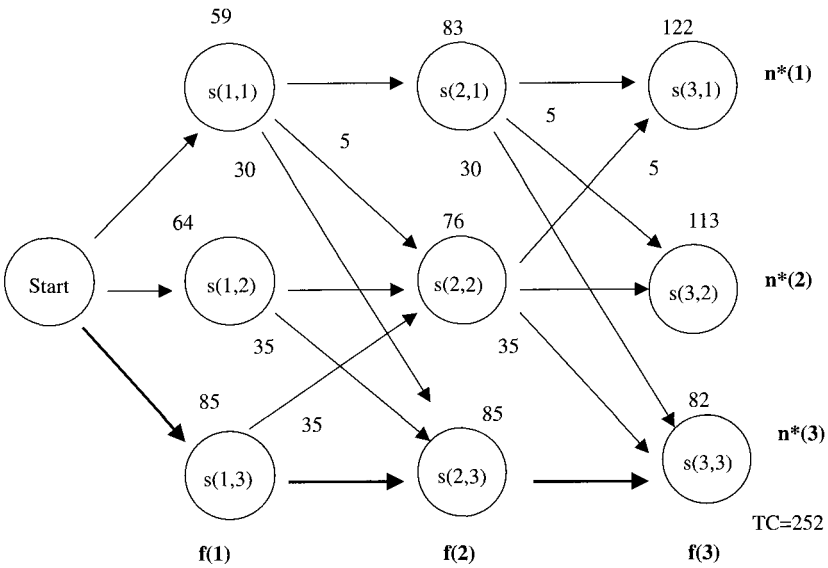


Fig. 7. Solution of Example 2 (unit machine moving cost = 5).

respectively. Since there are no extra machines to be removed from the system, the lower bound is calculated by:

$$\sum_{t=1}^{3} S[\mathbf{f}(t), \mathbf{z}^*(t)] = 59 + 76 + 82 = 217.$$

The real optimal objective values could be larger than this lower bound and the maximum relative errors are 3.7%, 14.3% and 16.1% with the unit machine removal/installation costs being 1, 4 and 5, respectively.

6. Conclusions

In this chapter, we discussed different computer techniques commonly used to solve manufacturing cell formation problems in the academic fields and the industries. These techniques include different mathematical programming models and computer simulation models, among others. This chapter introduces different types of mathematical programming models for solving basic cell formation problems and their various extensions. A cell formation model with dynamic and time-varying aspects is also introduced in this chapter. The dynamic problem, the model, and the solution technique presented in this chapter are relatively new and can interest the readers greatly. The static cell formation models and the direct method used to solve them are illustrated with examples. A more complicated example is also presented to illustrate the decomposition and the dynamic programming based method. Over the years, many researchers have developed methods to solve different static cell formation problems to overcome the NP-hardness associated with

most of the problem formulations. In this chapter we have demonstrated that if the methods based on relaxation or decomposition are to be used, the problems after relaxation should be easy to solve using widely available computing equipment such as a PC. We have also demonstrated that some simple cell formation models can be solved by general direct methods. However, one may be cautious when direct methods are used to solve even the simplest models in real industrial applications, since the required amount of computation could be very excessive.

Appendix

LINGO PROGRAM FILES AND DATA FILES

```
!SCF-1
MODEL:
SETS:
        PART/@FILE(AA1.LDT)/;
        MACHINE/@FILE(AA1.LDT)/:MCOST,NOM;
        CELL/@FILE(AA1.LDT)/: UB,LB;
        OPERATION/@FILE(AA1.LDT)/ ;
        OPERATION_SEQ(PART,OPERATION)/@FILE(AA1.LDT)/;
        PART_OPT_OPT(PART,OPERATION,OPERATION)/@FILE(AA1.LDT)/;
        OPERATION_SEQ_MN(OPERATION_SEQ,MACHINE)/@FILE(AA1.LDT)/;
        PROCESS(OPERATION_SEQ,CELL): X;
        PART_MACHINE(PART,MACHINE): F;
        MACHINE_CELL(MACHINE,CELL): Z;
        DISTANCE(CELL,CELL): D;
        AU_VARIA(PART_OPT_OPT,CELL,CELL):W;
ENDSETS

        MIN = @SUM(AU_VARIA(I,J,J1,L,L1)| L #NE# L1
                :D(L,L1)*W(I,J,J1,L,L1))
                + @SUM(MACHINE_CELL(K,L):MCOST(K)*Z(K,L));

        @FOR(AU_VARIA(I,J,J1,L,L1) — L #NE# L1:
          W(I,J,J1,L,L1) >= X(I,J,L)+X(I,J1,L1)-1;
          W(I,J,J1,L,L1) <= 0.5*(X(I,J,L)+X(I,J1,L1));
          @BIN(W(I,J,J1,L,L1)) );

        @FOR(OPERATION_SEQ(I,J):
          @SUM(PROCESS(I,J,L): X(I,J,L)) = 1 );

        @FOR(PROCESS(I,J,L):
          @BIN (X(I,J,L)) );
```

```
        @FOR(MACHINE_CELL(K,L):
          @SUM (OPERATION_SEQ_MN(I,J,K):F(I,K)*X(I,J,L)) <= Z(K,L);
          @GIN (Z(K,L)) );

        @FOR(MACHINE(K):
          @SUM(MACHINE_CELL(K,L):Z(K,L))<= NOM(K) );

        @FOR(CELL(L):
          @SUM(MACHINE_CELL(K,L):Z(K,L))<= UB(L);
          @SUM(MACHINE_CELL(K,L):Z(K,L))>= LB(L) );


DATA:
        MCOST = 6, 9, 8, 6;
        NOM = 10, 7, 6 8;
        UB = 4, 4, 4;
        LB = 1, 1, 1;
        D = 0, 2 4
              2, 0 3
              4, 3, 0;

        F = 0.8, 1.1, 0.0, 0.0
              0.0, 0.8, 0.0, 0.6
              1.2, 0.5, 0.0, 0.4
              0.0, 0.0, 0.2, 0.5
              0.6, 0.0, 0.5, 0.0;
ENDDATA
END


Data file AA1.LDT
        !NUMBER OF PARTS;
        1..5 ~

        !MACHINES;
        1..4 ~

        !CELLS;
        C1,C2,C3 ~

        !OPERATIONS;
        1..3 ~
```

```
        !OPERATIONS ON EACH PART;
        1,1 1,2 2,1 2,2 3,1 3,2 3,3 4,1 4,2
        5,1 5,2 ~

        !OPERATIONS ON EACH PART (DOUBLE);
        1,1,2 2,1,2 3,1,2 3,1,3 3,2,3 4,1,2 5,1,2 ~

        !PART-OPERATION-MACHINE;
        1,1,2 1,2,1 2,1,2 2,2,4 3,1,1 3,2,4 3,3,2
        4,1,4 4,2,3 5,1,1 5,2,3 ~


!SCF-2
MODEL:
SETS:
        PART/@FILE(AA2.LDT)/:TCOST;
        MACHINE/@FILE(AA2.LDT)/:MCOST,NOM;
        CELL/@FILE(AA2.LDT)/: UB,LB;
        OPERATION/@FILE(AA2.LDT)/ ;
        OPERATION_SEQ(PART,OPERATION)/@FILE(AA2.LDT)/;
        OPERATION_SEQ_MN(OPERATION_SEQ,MACHINE)/@FILE(AA2.LDT)/;
        PROCESS(OPERATION_SEQ,CELL): X;
        PART_CELL(PART,CELL):S;
        PART_MACHINE(PART,MACHINE): F;
        MACHINE_CELL(MACHINE,CELL): Z;
ENDSETS

        MIN = @SUM(PART_CELL(I,L):TCOST(I)*S(I,L))
              +@SUM(MACHINE_CELL(K,L):MCOST(K)*Z(K,L));

        @FOR (OPERATION_SEQ(I,J):
        @SUM (PROCESS(I,J,L): X(I,J,L)) = 1);

        @FOR (PROCESS(I,J,L): X(I,J,L) <= S(I,L);
        @BIN(X(I,J,L)));

        @FOR (MACHINE_CELL(K,L):
        @SUM (OPERATION_SEQ_MN(I,J,K):F(I,K)*X(I,J,L)) <= Z(K,L);
        @GIN (Z(K,L)));

        @FOR(MACHINE(K):
        @SUM(MACHINE_CELL(K,L):Z(K,L)) <= NOM(K));
```

```
        @FOR(CELL(L):
        @SUM(MACHINE_CELL(K,L):Z(K,L)) <= UB(L);
        @SUM(MACHINE_CELL(K,L):Z(K,L)) >= LB(L));

        @FOR(PART_CELL(I,L):
        @BIN(S(I,L)));


DATA:
        TCOST = 9, 7, 6, 8, 8;
        MCOST = 6, 9, 8, 6;
        NOM = 10, 7, 6 8;
        UB = 4, 4, 4;
        LB = 1, 1, 1;
        F = 0.8, 1.1, 0.0, 0.0
            0.0, 0.8, 0.0, 0.6
            1.2, 0.5, 0.0, 0.4
            0.0, 0.0, 0.2, 0.5
            0.6, 0.0, 0.5, 0.0;
ENDDATA
END


Data File AA2.LDT
        !NUMBER OF PARTS;
        1..5 ~

        !MACHINES;
        1..4 ~

        !CELLS;
        C1,C2,C3 ~

        !OPERATIONS;
        1..3 ~

        !OPERATIONS ON EACH PART;
        1,1 1,2 2,1 2,2 3,1 3,2 3,3 4,1 4,2
        5,1 5,2 ~

        !PART-OPERATION-MACHINE;
        1,1,2 1,2,1 2,1,2 2,2,4 3,1,1 3,2,4 3,3,2
        4,1,4 4,2,3 5,1,1 5,2,3 ~
```

```
!SCF-3
MODEL:
SETS:
        PART/@FILE(AA3.LDT)/:TCOST;
        MACHINE/@FILE(AA3.LDT)/:MCOST,NOM;
        CELL/@FILE(AA3.LDT)/: UB,LB;
        OPERATION/@FILE(AA3.LDT)/ ;
        OPERATION_SEQ(PART,OPERATION)/@FILE(AA3.LDT)/;
        OPERATION_SEQ_MN(OPERATION_SEQ,MACHINE)/@FILE(AA3.LDT)/;
        PROCESS(OPERATION_SEQ,CELL): X;
        PART_CELL(PART,CELL):S;
        PART_MACHINE(PART,MACHINE): F;
        MACHINE_CELL(MACHINE,CELL): Z;
ENDSETS
        MIN = @SUM(PART_CELL(I,L):TCOST(I)*S(I,L))
                + @SUM(MACHINE_CELL(K,L):MCOST(K)*Z(K,L));

        @FOR (OPERATION_SEQ(I,J):
        @SUM (PROCESS(I,J,L): X(I,J,L)) = 1);

        @FOR (PROCESS(I,J,L): X(I,J,L) <= S(I,L);
        @BIN(X(I,J,L)));

        @FOR(MACHINE_CELL(K,L):
        @FOR(PROCESS(I,J,L):
        @FOR(OPERATION_SEQ_MN(I,J,K):
        F(I,K)*X(I,J,L) <= Z(K,L));
        @BIN(Z(K,L))));

        @FOR(MACHINE(K):
        @SUM(MACHINE_CELL(K,L):Z(K,L)) <= NOM(K));

        @FOR(CELL(L):
        @SUM(MACHINE_CELL(K,L):Z(K,L)) <= UB(L);
        @SUM(MACHINE_CELL(K,L):Z(K,L)) >= LB(L));

        @FOR(PART_CELL(I,L):
        @BIN(S(I,L)));
```

```
DATA:
        TCOST = 9, 7, 6, 8, 6;
        MCOST = 6, 9, 8, 6;
        NOM = 10, 7, 6 8;
        UB = 2, 2, 2;
        LB = 1, 1, 1;
        F = 1.0, 1.0, 0.0, 0.0
            0.0, 1.0, 0.0, 1.0
            1.0, 1.0, 0.0, 1.0
            0.0, 0.0, 1.0, 1.0
            1.0, 0.0, 1.0, 0.0;
ENDDATA
END
```

Data file AA3.LDT

```
        !NUMBER OF PARTS;
        1..5 ~

        !MACHINES;
        1..4 ~

        !CELLS;
        C1,C2,C3 ~

        !OPERATIONS;
        1..3 ~

        !OPERATIONS ON EACH PART;
        1,1 1,2 2,1 2,2 3,1 3,2 3,3 4,1 4,2
        5,1 5,2 ~

        !PART-OPERATION-MACHINE;
        1,1,2 1,2,1 2,1,2 2,2,4 3,1,1 3,2,4 3,3,2
        4,1,4 4,2,3 5,1,1 5,2,3 ~
```

References

1. A. S. Alfa, M. Chen, and S. S. Heragu, A 3-opt based simulated annealing algorithm for vehicle routing problems, *Computers and Industrial Engineering* **21** (1991) 635–641.
2. A. Atmani, R. S. Lashkari, and R. J. Caron, A mathematical programming approach to joint cell formation and operation allocation in cellular manufacturing, *International Journal of Production Research* **33** (1995) 1–15.

3. D. D. Bedworth, M. R. Henderson, and P. M. Wolfe, *Computer-Integrated Design and Manufacturing* (McGraw-Hill, New York, NY, 1991).
4. W. E. Biles, A. S. Elmaghraby, and I. Zahran, A simulation study of hierarchical clustering techniques for the design of cellular manufacturing systems, *Computers and Industrial Engineering* **21** (1991) 267–272.
5. M. Chen and D. Mensah, A simulation study on cellular manufacturing system design and reconfiguration, *Proceedings of the IASTED International Conference on Applied Modeling and Simulation*, Banff, Canada (1997) 92–93.
6. N.-E. Dahel, Design of cellular manufacturing systems in tandem configuration, *International Journal of Production Research* **33** (1995) 2079–2095.
7. N.-E. Dahel and S. B. Smith, Designing flexibility into cellular manufacturing systems, *International Journal of Production Research* **31** (1993) 933–945.
8. G. Harhalakis, R. Nagi, and J. M. Proth, An efficient heuristic in manufacturing cell formation for group technology applications, *International Journal of Production Research* **28** (1990) 185–198.
9. G. Harhalakis, G. Ioannou, I. Minis, and R. Nagi, Manufacturing cell formation under random product demand, *International Journal of Production Research* **32** (1994) 47–64.
10. S. Heragu, *Facilities Design* (PWS, Boston, MA., 1997).
11. N. L. Hyer and U. Wemmerlov, Group technology in the US manufacturing industry: A survey of the current practices, *International Journal of Production Research* **27** (1989) 1287–1304.
12. A. M. M. Jamal, Neural network and cellular manufacturing, *Industrial Management and Data Systems* **93** (1993) 21–25.
13. J. A. Joines, C. T. Culbreth, and R. E. King, Manufacturing cell design: An integer programming model employing genetic algorithms, *IIE Transactions* **28** (1996) 69–85.
14. F. Kolahan, *Modelling and Analysis of Integrated Machine-level Planning Problems for Automated Manufacturing*, PhD Dissertation, Department of Mechanical Engineering University of Ottawa, Ottawa, Ontario, Canada, 1999.
15. S. Lozano, B. Adenso-Diaz, and L. Onieva, A one-step Tabu search algorithm for manufacturing cell design, *OR; the Journal of the Operational Research Society* **50** (1999) 509–516.
16. G. L. Nemhauser and L. A. Wolsey, *Integer and Combinatorial Optimization* (Wiley, New York, NY., 1988).
17. A. Prakash and M. Chen, A simulation study of flexible manufacturing systems, *Computers and Industrial Engineering* **28** (1995) 191–199.
18. H. A. Rao and P. Gu, A multi-constraint neural network for the pragmatic design of cellular manufacturing systems, *International Journal of Production Research* **33**, 4 (1995) 1049–1070.
19. A. Ravindran, D. T. Phillips, and J. J. Solberg, *Operations Research: Principles and Practice* (Wiley, New York, NY., 1987).
20. H. Seifoddini, A probabilistic model for machine cell formation, *Journal of Manufacturing Systems* **9** (1990) 69–75.
21. S. M. Shafer and J. M. Charnes, A simulation analysis of factors influencing loading practices in cellular manufacturing, *International Journal of Production Research* **33** (1995) 279–292.
22. J. S. Shang and P. R. Tadikamalla, Multicriteria design and control of a cellular manufacturing system through simulation and optimization, *International Journal of Production Research* **36** (1998) 1515–1529.

23. A. Shinn and T. Williams, A stitch in time: A simulation of cellular manufacturing, *Production and Inventory Management Journal* **39** (1998) 72–77.
24. N. Singh, *Systems Approach to Computer-Integrated Design and Manufacturing* (Wiley, New York, NY, 1996).
25. S. Sofianopoulou, Application of simulated annealing to a linear model for the formulation of machine cells in group technology, *International Journal of Production Research* **35** (1997) 501–513.
26. University of Wollongong, University of New South Wales and IE Management Consultants, *Survey of Cellular Manufacture: Cellular Manufacturing's Impact on Australian Industry*, Technical Report (1996).
27. A. J. Vakharia and B. K. Kaku, Redesigning a cellular manufacturing system to handle long-term demand changes: A methodology and investigation, *Decision Sciences* **24** (1993) 909–923.
28. S. Viswanathan, Configuring cellular manufacturing systems: A quadratic integer programming formulation and a simple interchange heuristic, *International Journal of Production Research* **33** (1995) 361–376.
29. U. Wemmerlov and N. L. Hyer, Cellular manufacturing practices, *Manufacturing Engineering* **102**, 3 (1995) 79–82.
30. U. Wemmerlov and N. L. Hyer, Procedures for the part family/machine group identification problem in cellular manufacturing, *Journal of Operations Management* **6** (1986) 125–147.
31. U. Wemmerlov and D. J. Johnson, Cellular manufacturing at 46 user plants: Implementation experiences and performance improvements, *International Journal of Production Research* **35** (1997) 29–49.
32. S. Zolfaghari and M. Liang, An objective-guided ortho-synapse Hopfield network approach to machine grouping problems, *International Journal of Production Research* **35** (1997) 2773–2780.

This page is intentionally left blank

CHAPTER 3

OPTIMAL COMPUTER AIDED DESIGN (CAD) METHODS AND APPLICATIONS FOR INJECTION MOLDING PROCESSES IN MANUFACTURING SYSTEMS

SEONG JIN PARK

CAE Center,
LG Production Engineering Research Center,
LG-PRC, 19-1, Cheongho-ri, Jinwuy-myun,
Pyungtaek-si, Kyunggi-do, 451-713, South Korea
E-mail: steelers@lge.co.kr

TAI HUN KWON

Department of Mechanical Engineering,
Pohang University of Science and Technology,
San 31 Hyoha-dong, Nam-ku,
Pohang 790-784, South Korea
E-mail: thkwon@postech.ac.kr

In recent years, increased attention has been shifted to the design of cooling systems in the injection molding process, as it becomes clear that the cooling systems affect significantly both productivity and the part quality. In order to systematically improve the performance of a cooling system the mold designer may need a computer aided optimal design system for designing the injection mold cooling systems and determining the process conditions during the cooling stage. In this chapter, an efficient optimization procedure for this problem is proposed utilizing (i) the special boundary element analysis, (ii) the corresponding design sensitivity analysis using the direct differentiation approach, and (iii) the optimization algorithm. For this optimal design, an objective function is proposed to minimize a weighted combination of the cooling time and the temperature nonuniformity over the part surface. The former has to do with the warpage in the final part, while the latter is directly related to the overall productivity of the injection molding process. In this optimization program, various design variables are considered as follows: (i) (design variables related to processing conditions) the inlet coolant bulk temperature and inlet coolant volumetric flow rate of each cooling channel and (ii) (design variables related to mold cooling system design) the radius and location of each cooling channel. Each step of the proposed optimization procedure will be briefly explained below. First, in thermal analysis, mold heat transfer is considered as a cyclic-steady, three-dimensional conduction; heat transfer within the melt region is treated as a transient, one-dimensional conduction; heat exchange between the cooling channel surfaces and the coolant is considered to be steady; heat exchange between ambient air and mold exterior

surfaces is also considered steady. Numerical implementation includes the application of a hybrid scheme consisting of a modified three-dimensional boundary element method for the mold region and a finite difference method with a variable mesh for the melt region. However, it was found that seemingly negligible inaccuracy in the thermal analysis result sometimes lead to a meaningless sensitivity analysis result. In this study, the thermal analysis system based on the above-mentioned modified boundary element method has been improved and rigorous treatments of boundary conditions appropriate for sensitivity analysis have been developed by considering the following issues: (i) numerical convergency, (ii) the series solution in part thermal analysis, (iii) the treatment of tip surface of line elements, (iv) the treatment of coolant, and (v) the treatment of mold exterior surface. Using an example, the importance of these issue is amply demonstrated. Next, the sensitivity analysis program developed in the present studies utilizes the implicit differentiation of the boundary integral equations and the boundary conditions presented in thermal analysis with respect to all design variables to obtain the sensitivity equations. A sample problem is solved to demonstrate the accuracy and the efficiency of the present sensitivity analysis formulation as well as to discuss the characteristics of each design variable. Finally, the CONMIN algorithm is applied for the optimization program with the help of the above thermal analysis and corresponding design sensitivity analysis. In this optimization program, the proper constraints imposed upon the design variables are considered to maintain design reality. The CONMIN algorithm employs the augmented Lagrangian multiplier method to deal with the equality constraints and the Davidon–Fletcher–Powell method for the unconstrained minimization during the successive unconstrained minimization procedure. Two sample problems were solved to demonstrate the efficiency and the usefulness of the objective function. The developed computer aided optimal design system would be very useful for injection mold designers in obtaining an optimal configuration of an injection mold cooling system in terms of radii and locations of cooling channels, as well as determining the optimal processing conditions of the cooling stage in terms of the inlet coolant bulk temperature and the inlet coolant volumetric flow rate of each cooling channel by minimizing certain objective functions related to the part quality and/or the productivity in the injection molding processes.

Keywords: Injection molding; cooling system design; optimization; sensitivity analysis; boundary element method.

1. Introduction

1.1. *Injection molding*

The improvement of high-performance plastic resins and appropriate processing techniques has caused plastic materials to be increasingly important in many industries (such as automotive, major appliances, aerospace, and electronics) as a substitute for conventional materials for structural parts. The advantages of using plastic materials are many: for instance, the weight of part is reduced, the manufacturing time and cost are also reduced and sometimes certain material properties like corrosion-resistance can be increased.[1,2]

Among the many processing techniques for plastic materials, injection molding is recognized as one of the most efficient processing techniques for producing precision plastic parts of complex shapes at a low cost. The schematic diagram of an injection mold is shown in Fig. 1. The mold cavity is surrounded by several cooling channels, and the mold exterior surfaces are in contact with the ambient air or platens. Typically, an injection molding process starts from filling the cavity with hot polymer melt at an injection temperature (filling stage). After the cavity is filled, additional polymer melt is packed into the cavity at a high pressure to compensate for the shrinkage (post-filling or packing stage). This is followed by cooling until a specific ejection criterion is reached when the part can be ejected without any deformation (cooling stage). After the part is ejected, the mold is closed and the next injection cycle begins. Thus a typical cycle in the process includes filling, packing, and cooling stages. Figure 2 shows the pressure–time plot during one cycle of the injection molding process. After these stages are repeated a few times, the initial transients die out and all reactions within the mold become periodic.[3,4]

Until recently, design in this area was seen as a work of art rather than a science subject, and it has been a specialty of few mold designers. Therefore, in the past, injection molding industries relied mostly on past experience and/or trial-and-error in mold design because of the inherently complicated nature of the process. However,

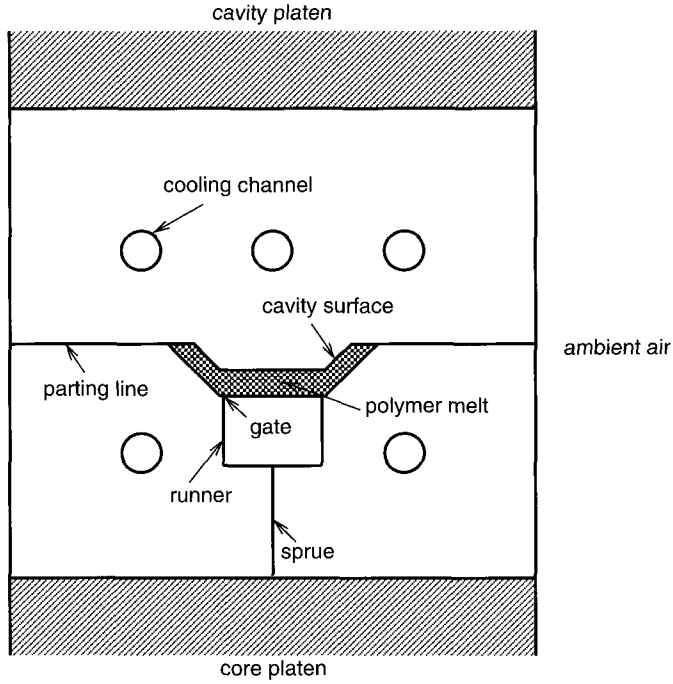


Fig. 1. Schematic diagram for an injection mold.

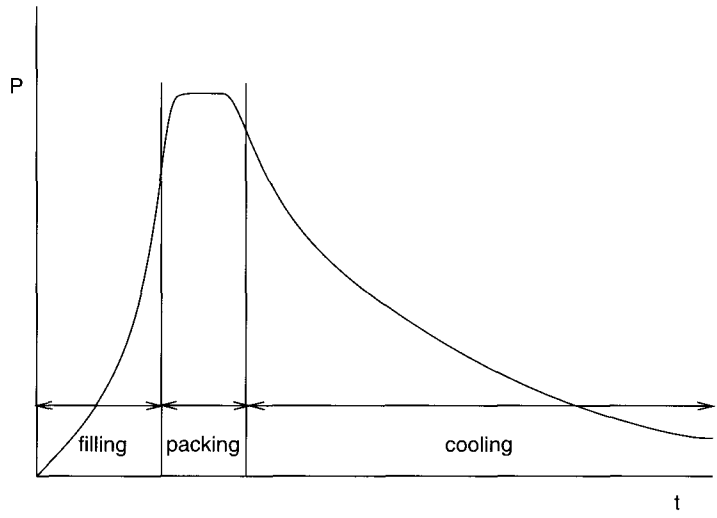


Fig. 2. Pressure versus time plot during one cycle.

there have been extensive research efforts in the academic community as well as in the related industries to develop a scientific base for the injection molding process and to introduce modern computer techniques for mold design and manufacturing. The computer aided design for injection molding might be divided into three major design procedures as follows: (i) the moldbase selection to determine the size of a mold, the locations and sizes of leader pins and ejector pins, etc.; (ii) the runner-gate-cavity system design to determine the location and size of runners and gates for balancing flows of polymer melts in the filling stage of the whole process; and (iii) the cooling system design to determine an optimum layout of cooling channels and the cooling processing conditions for the cooling stage of the process. In the present study, the cooling system design will be the one discussed among the three design procedures.[3]

1.2. *Research objective*

Cooling system design in injection molding industries is of great importance because it significantly affects productivity and the quality of the final part. It is well known that more than 75% of the cycle time in the injection molding process is spent on cooling the hot polymer melt sufficiently so that the part can be ejected without any significant deformation. An efficient cooling line design can considerably reduce the cooling time, and in turn increase the productivity of the injection molding process. On the other hand, severe warpage and thermal residual stress in the final product may result from non-uniform cooling. The warpage and sink marks are sometimes very critical in the final part quality, especially in terms of appearance and precision.[3,4]

Accordingly, there are at least two important concepts to keep in mind considering cooling system design: (a) minimizing cycle time and (b) achieving uniform cooling. There are many design variables to be considered in these goals, such as the size, location and arrangement of cooling channels; the temperature, flow rate and thermal properties of the coolant; the mold material properties; the injection melt temperature, etc. With so many design parameters involved, design work in determining the optimum cooling system and processing conditions is made extremely difficult.

With the above need in mind, the objective of the present study is to developed a computer aided optimal design system for designing an injection mold cooling system and determining the process conditions during the cooling stage in order to systematically improve the performance of a cooling system.

1.3. *Problem definition*

In order to increase the part quality (uniformity) and the productivity of the fixed injection molded part, the cooling system designer can adjust the mold cooling system design and the process conditions for the cooling stage of the injection molding process. Therefore, various design variables are considered as follows:

- Design variables related to processing conditions:
  - * inlet coolant bulk temperature of each cooling channel,
  - * inlet coolant volumetric flow rate of each cooling channel.
- Design variables related to mold cooling system design:
  - * radius of each cooling channel,
  - * location of each cooling channel.

The optimal design procedure of a cycle-averaged conduction heat transfer in the cooling stage of the injection molding process is defined as follows: given a part geometry and number of cooling channels, we find the optimal values of all design variables to minimize the weighted combination of the non-uniformity in the temperature distribution on the part surface and the cooling time (productivity), with side constraints (i.e. realistic interval of each design variable).

1.4. *Chapter organization*

In this study, an efficient optimization procedure for this problem is proposed utilizing (i) the special boundary element analysis, (ii) the corresponding design sensitivity analysis using the direct differentiation approach, and (iii) the optimization algorithm. Therefore, the sequence of this chapter is organized in accordance to above order.

In Sec. 2, the thermal analysis for the cooling stage of the injection molding process is presented. Governing equations with initial and boundary conditions, numerical implementation combining the modified boundary element method for

the mold analysis and analytical series solution for the part analysis, as well as several representative results will be highlighted in this section.

In Sec. 3, the design sensitivity analysis is explained by the following: boundary integral formulation using the direct differentiation approach; numerical results by favorable comparisons between the direct differentiation approach and a numerical derivative using the forward finite difference method; and understanding the characteristics of each design variable.

In Sec. 4, the optimization algorithm is presented. The proposed objective function, the side constraints (interval of each design variable), the CONMIN algorithm, as well as two numerical results will be used to explain the algorithm.

In Sec. 5, an overall summary of this study as well as future work will be discussed.

2. Thermal Analysis

2.1. *Introduction*

For the optimal design system, the designer needs a thermal analysis tool for the three-dimensional mold heat transfer during the cooling stage of an injection molding process. This thermal analysis tool should be able to predict the cooling time (and thus cycle time), the temperature and the temperature gradients on the mold surface, and so on. Several thermal analysis tools have been developed by many researchers using the modified boundary element method under the cycle-averaged concept: POYCOOL2 of SDRC,[5] C-MOLD of CIMP,[4] MF/COOL of MOLDFLOW and so on. In these simulation packages, mold heat transfer is considered as a cyclic-steady, three-dimensional conduction process; heat transfer within the melt region is treated as a transient, one-dimensional conduction process; heat exchange between the cooling channel surfaces and coolant is considered to be steady, and so is the heat exchange between the ambient air and the mold exterior surfaces.[3–5] Such numerical simulation packages are validated and are being used by many relevant industries with a reasonable satisfaction.

However, it was found that seemingly negligible inaccuracy in the thermal analysis result sometimes gives rise to meaningless sensitivity analysis results. Thus an accurate thermal analysis is critically essential in obtaining precise sensitivity analysis results. Developing a precise sensitivity analysis tool is one of the ultimate goals of this study. In view of this the accuracy of the thermal analysis system has been improved based on the modified boundary element method and rigorous treatments of the boundary conditions appropriate for the sensitivity analysis have been developed. The following issues have been considered: (i) the numerical convergency for obtaining a temperature distribution accurate enough to warrant a meaningful design sensitivity analysis; (ii) the series solution in part analysis for yielding appropriate equations to determine the cooling time and the cycle-averaged heat flux on the cavity surface; (iii) the treatment of the tip surface of line elements for satisfying the basic assumptions of boundary integral equations; (iv) the treatment of coolant

for improving the cooling process condition; and (v) the treatment of mold exterior surface associated with an algorithm to determine the temperature on the mold exterior surface. All these issues contribute greatly towards overcoming the difficulties in obtaining successfully the corresponding sensitivity analysis. It may be worth mentioning that most of them are closely related to the rigorous treatment of boundary conditions and that all design variables considered in the sensitivity analysis are defined only on boundary surfaces. (See Secs. 2.3.2 and 3.3.2.) Using an example, the importance of these issues can be amply demonstrated.

2.2. *Physical modeling*

2.2.1. *Cycle-average approach*

Some comparative studies have been performed to identify a methodology for the analysis of heat transfer within the injection mold. This should be simple, yet computationally efficient and sufficiently accurate for mold design purposes. Three methods were considered for comparison: a fully transient technique, a periodic analysis, and a cycle-average approach.

The temperature field during such a fully continuous operation may be separated into two parts: the cycle-averaged temperature field during one cycle and the temperature fluctuation field. Based on these results obtained for several typical molds, polymer-melt materials and processing conditions, it is known that the average temperature field varies very little in time during the fully continuous molding operation. The relatively small temperature fluctuation is at its maximum near the cavity surface and diminishes away from the cavity surface. The region of fluctuation is localized near the cavity surface and the temperature fluctuation near the cooling channels is very small. The effects of the fluctuating mold temperature on the transient melt temperature and the heat exchange between the mold and cooling channels was found to be negligibly small. Thus, a cycle-average approach for the mold region is accurate enough to predict the mold temperature distributions, the transient and non-uniform melt temperature and their effects on heat removal by the coolant.[3,4] In addition, a cycle-average approach is computationally more efficient than the other two methods, namely, a fully transient technique and a periodic analysis.

With the above rationality in mind, a three-dimensional, cycle-average approach was adopted for the mold thermal analysis to determine the cycle-averaged temperature field and its effects on the plastic part and cooling system. It should be noted, however, that the subsequent approach will be applicable only to the thin injection molded parts.[3–5]

2.2.2. *Governing equation*

The cross-sectional diagram of an injection mold is shown in Fig. 3. Here the mold cavity is represented by the part mid-surface. The part can be described by only

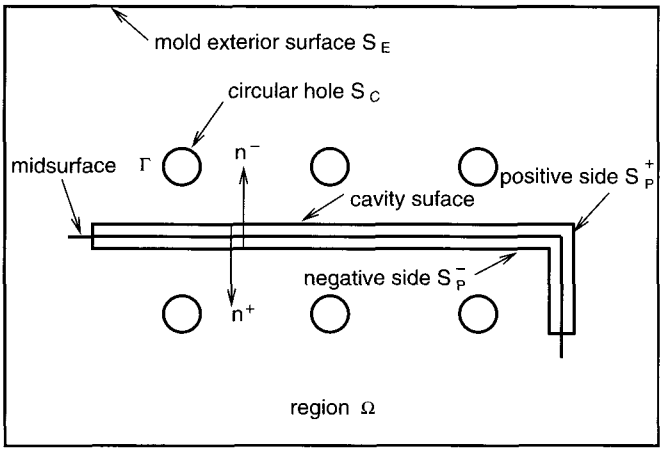


Fig. 3. Notations for injection mold surfaces.

the mid-surface (Γ) of the part instead of the whole surface (S_P^+ and S_P^-) because of its thin width.[4,5]

The cycle-averaged temperature field can be represented by the steady-state heat conduction. Thus under this cycle-average concept, the governing differential equation of the heat transfer for the injection mold cooling system can be written as:

$$\nabla^2 T = 0 \quad (\text{in } \Omega). \tag{1}$$

2.2.3. *Boundary conditions*

In order to obtain a meaningful cycle-averaged temperature field, it is important to introduce proper boundary conditions on the cavity surface, the cooling channel surface, and the mold exterior surface, which are consistent with the cycle-average concept, especially over the cavity surface.

First, on the cavity surface, a cycle-averaged heat flux is imposed as follows:

$$-k_m \frac{\partial T}{\partial n} = \bar{q}, \tag{2}$$

where $\bar{q}$ is the cycle-averaged heat flux given by:

$$\bar{q} = \frac{1}{t_f + t_c + t_o} \left[\int_0^{t_f} q_f \, dt + \int_{t_f}^{t_f + t_c} q_c \, dt + \int_{t_f + t_c}^{t_f + t_c + t_o} q_o \, dt \right]. \tag{3}$$

Thus, the cycle time is simply the total time spent on the filling, the cooling and the mold open stages. The cooling time is the time required to meet a certain ejection criterion in order for the part to be ejected without any deformation. The cooling time may be determined by three different ways as follows: (i) finding the time required for the maximum polymer melt temperature to reach the ejection

temperature (Type I); (ii) finding the time required for the maximum average-temperature in the thickness direction to reach the ejection temperature (Type II); (iii) or finding a desirable cooling time as suggested by the mold designer (Type III). In order to evaluate the above cycle-averaged heat flux and cooling time, one should know the temperature distribution of polymer melt as a function of time during the filling and cooling stages. For the thermal analysis of the plastic part, a local one-dimensional transient analysis is adequate since it is usually thin.[4,5] With an assumption of constant thermal properties of plastic part material, the governing equation for the heat transfer in the plastic part is given by:

$$\frac{\partial T}{\partial t} = \frac{1}{\alpha}\frac{\partial^2 T}{\partial z^2}. \tag{4}$$

As an initial condition of this problem, one might either use a uniform injection temperature or the temperature distribution at the end of the filling stage determined from the numerical filling simulation. Furthermore, a perfect thermal contact between the polymer melt and the mold is assumed and a cycle-averaged mold-melt interfacial temperature is used as the boundary condition. Hence, Eqs. 1 and 4 are coupled with each other through the boundary condition on the cavity surface using an iterative solution technique. One can evaluate the cooling time and the cycle-averaged heat flux from the coupled analysis results, neglecting in most cases the relatively small q_o (and even q_f when numerical results of the filling stage are not available). It is noted that the sprue-and-runner system can be treated in the same manner in terms of the cylindrical coordinate system.

Secondly, on the cooling channel surface, a mixed boundary condition is imposed as follows:

$$-k_m \frac{\partial T}{\partial n} = h_c(T - T_b). \tag{5}$$

In this equation, heat transfer coefficient, h_c, is evaluated via the Dittus–Boetler correlation for forced convective heat transfer in a turbulent pipe flow.[6] Obviously, to calculate the heat transfer coefficient, one may need a volumetric flow rate of coolant for each cooling channel. One can use the relationship between the pressure and the flow rate in turbulent pipe flow to determine the flow rate.[7] Furthermore, the log mean temperature difference concept has been used to evaluate the coolant bulk temperature.[6]

Finally, there are basically three different approaches for the mold exterior surface treatment. These approaches are described as follows:

Type A: normal treatment — In this approach, the real mold exterior surface is to be included in the analysis. For the mold exterior surfaces in contact with the ambient air, the ambient air temperature and heat transfer coefficients are prescribed based on the correlations available in the literature for proper steady-free convective heat transfer on each mold exterior surface.[6] For the surfaces in contact with the metal platens, a given pseudo heat transfer coefficient (that is, an inverse of the metal thermal contact resistance) and a given platen temperature are

used to maintain consistency throughout the mold exterior surface having mixed boundary conditions.

Type B: infinite adiabatic sphere — It may be noted that in most injection molding applications, heat loss through the mold exterior surfaces is very small (typically less than 5%). In such cases, the mold exterior surface may be approximated as an infinite adiabatic sphere. This approximation does not require the modeling of the mold exterior surface. Thus this approach leads to substantial savings in the memory requirement as well as the reduction in CPU time.

Type C: convective sphere of equivalent radius — A more accurate alternative to the above-mentioned infinite adiabatic formulation is to impose a mixed boundary condition on the mold exterior surface approximated as a sphere with an equivalent radius. This preserves the actual exterior surface area.

2.3. *Boundary integral formulation for analysis*

2.3.1. *Governing equation*

A standard boundary element formulation for the three-dimensional Laplace's equation given by Eq. 1, governing the steady conduction based on Green's second identity, leads to[8]:

$$\alpha T(\vec{x}) = \int_S \left[\frac{1}{r} \left(\frac{\partial T(\vec{\xi})}{\partial n} \right) - T(\vec{\xi}) \frac{\partial}{\partial n} \left(\frac{1}{r} \right) \right] dS(\vec{\xi}). \tag{6}$$

Here $\vec{x}$ and $\vec{\xi}$ are points in space, $r = |\vec{\xi} - \vec{x}|$, and α denotes a solid angle formed by the boundary surface. Note that $\alpha = 2\pi$ at the smooth boundary surface, and $\alpha = 4\pi$ or 0 at the internal points or external points of the domain, respectively.

For any two closely spaced surfaces such as the part surfaces in the geometries of injection molds (as Eq. 6 leads to a redundancy in the final system of linear algebraic equations), a modified procedure as described in Ref. 5 is used. According to this modification, the mid-surface, Γ, is considered rather than two closely spaced surfaces, S_P^+ and S_P^- as depicted in Fig. 3. For each mid-surface element, in order to derive the extra equation corresponding to the additional degree of freedom, a derivative of Eq. 6 with respect to the normal direction, $\hat{\nu}$ ($\hat{n}^+$ in Fig. 3), at the mid-surface element, is taken. For circular hole surfaces (sprue, runner or cooling channel), a special formulation based on the line-sink approximation is used. This formulation avoids the discretization of the circular channels along the circumference and thus saves a large amount of computer memory and time (see papers by Rezayat and Burton[5] and Park and Kwon[9] for further details of this modified approach for part surfaces, circular holes, and exterior surface). The present study pays special attention to the tip surface of circular holes like a sprue, a runner and a cooling channel. Equation 6, based on Green's second identity, can be applied to a region bounded by closed surfaces. The ignorance of tip surfaces of the circular holes causes the critical disturbance of the temperature field near the gates attached to a sprue.

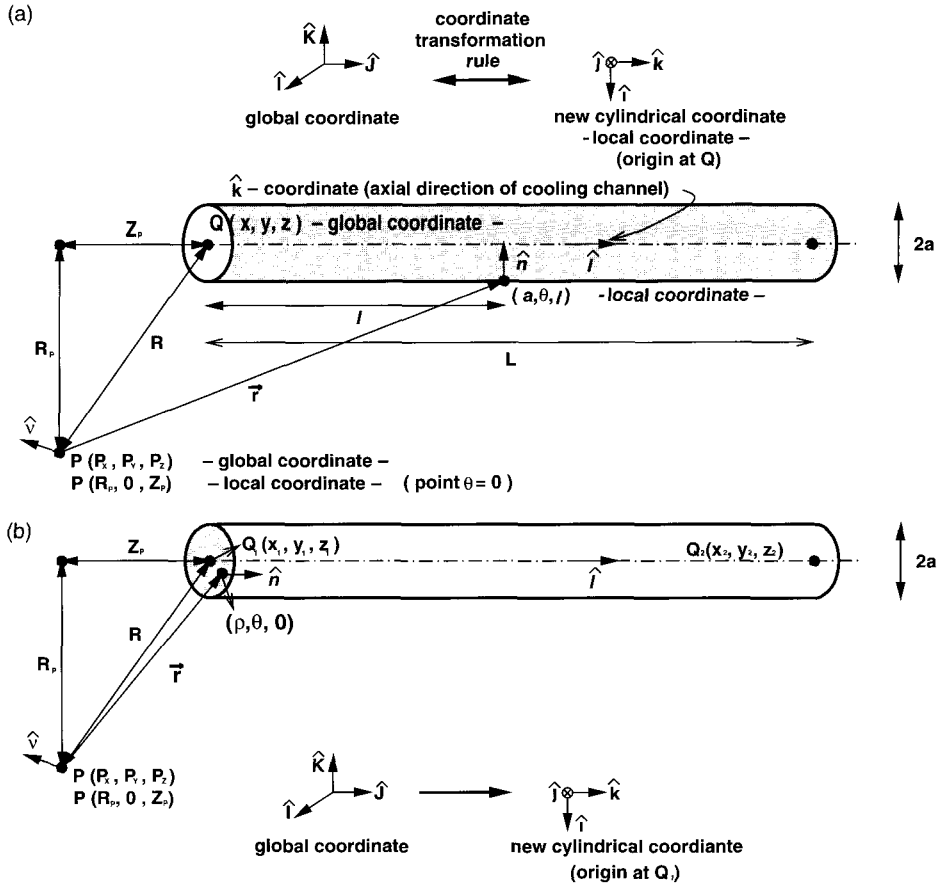


Fig. 4. Notations of line element for (a) circumferential surface and (b) tip surface.

This will be described in Sec. 2.4. Therefore, one should consider not only the circumferential surfaces (Fig. 4a) but also the tip surfaces (Fig. 4b).

The following are the final BEM formulae with modifications including the tip surface treatment: (i) for a point $\vec{x}$ on the mid-surface of the part, Γ:

$$
\begin{aligned}
&\alpha^- T^+(\vec{x}) + \alpha^+ T^-(\vec{x}) \\
&\quad = \int_\Gamma \left[\frac{1}{r} \left(\frac{\partial T^+}{\partial n^+} + \frac{\partial T^-}{\partial n^-} \right) - \frac{\partial}{\partial n^+} \left(\frac{1}{r} \right) \left(T^+ - T^- \right) \right] dS(\vec{\xi}) \\
&\qquad + \int_\lambda \left[\left(\frac{\partial T}{\partial n} \right) \int_0^{2\pi} \left(\frac{1}{r} \right) d\theta - T \int_0^{2\pi} \frac{\partial}{\partial n} \left(\frac{1}{r} \right) d\theta \right] a \, dl(\vec{\xi}) \\
&\qquad + \int_\varrho \left[\left(\frac{\partial T}{\partial n} \right) \int_0^{2\pi} \left(\frac{1}{r} \right) d\theta - T \int_0^{2\pi} \frac{\partial}{\partial n} \left(\frac{1}{r} \right) d\theta \right] \rho \, d\rho(\vec{\xi})
\end{aligned}
$$

$$+\sum_{k=1}^{N}\int_{l_k}\left[\left(\frac{\partial T}{\partial n}\right)\int_0^{2\pi}\left(\frac{1}{r}\right)d\theta - T\int_0^{2\pi}\frac{\partial}{\partial n}\left(\frac{1}{r}\right)d\theta\right]a_k\,dl(\vec{\xi})$$
$$+\sum_{k=1}^{N}\int_{\rho_k}\left[\left(\frac{\partial T}{\partial n}\right)\int_0^{2\pi}\left(\frac{1}{r}\right)d\theta - T\int_0^{2\pi}\frac{\partial}{\partial n}\left(\frac{1}{r}\right)d\theta\right]\rho\,d\rho(\vec{\xi})$$
$$+\int_{S_E}\left[\frac{1}{r}\frac{\partial T}{\partial n} - \frac{\partial}{\partial n}\left(\frac{1}{r}\right)T\right]dS(\vec{\xi}); \tag{7}$$

$$\alpha^-\frac{\partial T^+(\vec{x})}{\partial \nu} - \alpha^+\frac{\partial T^-(\vec{x})}{\partial \nu}$$
$$= \int_\Gamma\left[\frac{\partial}{\partial \nu}\left(\frac{1}{r}\right)\left(\frac{\partial T^+}{\partial n^+} + \frac{\partial T^-}{\partial n^-}\right) - \frac{\partial}{\partial \nu}\left(\frac{\partial}{\partial n^+}\left(\frac{1}{r}\right)\right)(T^+ - T^-)\right]dS(\vec{\xi})$$
$$+\int_\lambda\left[\left(\frac{\partial T}{\partial n}\right)\int_0^{2\pi}\frac{\partial}{\partial \nu}\left(\frac{1}{r}\right)d\theta - T\int_0^{2\pi}\frac{\partial}{\partial \nu}\left\{\frac{\partial}{\partial n}\left(\frac{1}{r}\right)\right\}d\theta\right]a\,dl(\vec{\xi})$$
$$+\int_\varrho\left[\left(\frac{\partial T}{\partial n}\right)\int_0^{2\pi}\frac{\partial}{\partial \nu}\left(\frac{1}{r}\right)d\theta - T\int_0^{2\pi}\frac{\partial}{\partial \nu}\left\{\frac{\partial}{\partial n}\left(\frac{1}{r}\right)\right\}d\theta\right]\rho\,d\rho(\vec{\xi})$$
$$+\sum_{k=1}^{N}\int_{l_k}\left[\left(\frac{\partial T}{\partial n}\right)\int_0^{2\pi}\frac{\partial}{\partial \nu}\left(\frac{1}{r}\right)d\theta - T\int_0^{2\pi}\frac{\partial}{\partial \nu}\left(\frac{\partial}{\partial n}\left(\frac{1}{r}\right)\right)d\theta\right]a_k\,dl(\vec{\xi})$$
$$+\sum_{k=1}^{N}\int_{\rho_k}\left[\left(\frac{\partial T}{\partial n}\right)\int_0^{2\pi}\frac{\partial}{\partial \nu}\left(\frac{1}{r}\right)d\theta - T\int_0^{2\pi}\frac{\partial}{\partial \nu}\left\{\frac{\partial}{\partial n}\left(\frac{1}{r}\right)\right\}d\theta\right]\rho\,d\rho(\vec{\xi})$$
$$+\int_{S_E}\left[\frac{\partial}{\partial \nu}\left(\frac{1}{r}\right)\frac{\partial T}{\partial n} - \frac{\partial}{\partial \nu}\left(\frac{\partial}{\partial n}\left(\frac{1}{r}\right)\right)T\right]dS(\vec{\xi}); \tag{8}$$

(ii) for a point on the axis of the cylindrical segment of the circular channels (sprue, runner or cooling channel) one obtains the following integral equation:

$$0 = \int_\Gamma\left[\frac{1}{r}\left(\frac{\partial T^+}{\partial n^+} + \frac{\partial T^-}{\partial n^-}\right) - \frac{\partial}{\partial n^+}\left(\frac{1}{r}\right)(T^+ - T^-)\right]dS(\vec{\xi})$$
$$+\int_\lambda\left[\left(\frac{\partial T}{\partial n}\right)\int_0^{2\pi}\left(\frac{1}{r}\right)d\theta - T\int_0^{2\pi}\frac{\partial}{\partial n}\left(\frac{1}{r}\right)d\theta\right]a\,dl(\vec{\xi})$$
$$+\int_\varrho\left[\left(\frac{\partial T}{\partial n}\right)\int_0^{2\pi}\left(\frac{1}{r}\right)d\theta - T\int_0^{2\pi}\frac{\partial}{\partial n}\left(\frac{1}{r}\right)d\theta\right]\rho\,d\rho(\vec{\xi})$$
$$+\sum_{k=1}^{N}\int_{l_k}\left[\left(\frac{\partial T}{\partial n}\right)\int_0^{2\pi}\left(\frac{1}{r}\right)d\theta - T\int_0^{2\pi}\frac{\partial}{\partial n}\left(\frac{1}{r}\right)d\theta\right]a_k\,dl(\vec{\xi})$$
$$+\sum_{k=1}^{N}\int_{\rho_k}\left[\left(\frac{\partial T}{\partial n}\right)\int_0^{2\pi}\left(\frac{1}{r}\right)d\theta - T\int_0^{2\pi}\frac{\partial}{\partial n}\left(\frac{1}{r}\right)d\theta\right]\rho\,d\rho(\vec{\xi})$$
$$+\int_{S_E}\left[\frac{1}{r}\frac{\partial T}{\partial n} - \frac{\partial}{\partial n}\left(\frac{1}{r}\right)T\right]dS(\vec{\xi}); \tag{9}$$

and (iii) for a point on the mold exterior surfaces one obtains the following integral equation:

$$\begin{aligned}
\alpha T(\vec{x}) = & \int_\Gamma \left[\frac{1}{r}\left(\frac{\partial T^+}{\partial n^+} + \frac{\partial T^-}{\partial n^-}\right) - \frac{\partial}{\partial n^+}\left(\frac{1}{r}\right)(T^+ - T^-)\right] dS(\vec{\xi}) \\
& + \int_\lambda \left[\left(\frac{\partial T}{\partial n}\right)\int_0^{2\pi}\left(\frac{1}{r}\right)d\theta - T\int_0^{2\pi}\frac{\partial}{\partial n}\left(\frac{1}{r}\right)d\theta\right] a\, dl(\vec{\xi}) \\
& + \int_\varrho \left[\left(\frac{\partial T}{\partial n}\right)\int_0^{2\pi}\left(\frac{1}{r}\right)d\theta - T\int_0^{2\pi}\frac{\partial}{\partial n}\left(\frac{1}{r}\right)d\theta\right] \rho\, d\rho(\vec{\xi}) \\
& + \sum_{k=1}^{N}\int_{l_k} \left[\left(\frac{\partial T}{\partial n}\right)\int_0^{2\pi}\left(\frac{1}{r}\right)d\theta - T\int_0^{2\pi}\frac{\partial}{\partial n}\left(\frac{1}{r}\right)d\theta\right] a\, dl(\vec{\xi}) \\
& + \sum_{k=1}^{N}\int_{\rho_k} \left[\left(\frac{\partial T}{\partial n}\right)\int_0^{2\pi}\left(\frac{1}{r}\right)d\theta - T\int_0^{2\pi}\frac{\partial}{\partial n}\left(\frac{1}{r}\right)d\theta\right] \rho\, d\rho(\vec{\xi}) \\
& + \int_{S_E}\left[\frac{1}{r}\frac{\partial T}{\partial n} - \frac{\partial}{\partial n}\left(\frac{1}{r}\right)T\right] dS(\vec{\xi}). \qquad (10)
\end{aligned}$$

It should be noted that the third and fifth terms in Eqs. 7–10 are due to the special treatment of the tip surfaces of runner (or sprue) and cooling channel, respectively. For boundary element analysis, triangular elements were used for the part surface and the mold exterior surfaces and line elements were used for the sprue, runner and cooling channel. It might be mentioned that each element has a constant temperature and a constant heat flux in the boundary element analysis in the present study.

2.3.2. *Boundary conditions*

The following rigorous numerical treatments of boundary conditions are of great importance not only for the thermal analysis itself but also for the corresponding sensitivity analysis since all design variables considered in the sensitivity analysis are defined only on boundary surfaces. (See Sec. 3.2.)

Part surface

One should analyze Eq. 4 as described in the previous section. Two methods to solve this equation might be available: a finite difference method (FDM) and an analytical series solution. These methods are elaborated as follows:

(i) Finite difference method — The Crank–Nicholson method was used with a variable grid system using the one-dimensional stretching function presented by Roberts and Eiseman.[10] Note that the variable grid is introduced to obtain an accurate evaluation of the cycle-averaged heat flux as a boundary condition. The resulting tri-diagonal system of equations is solved by the Thomas algorithm. One can determine the cooling time of Type I by checking the maximum temperature at each time

step and grid as well as the maximum temperature in each element. Also the cooling time of Type II is determined by checking the maximum average temperature in the thickness direction at each time step of each element. The cycle-averaged heat flux can be evaluated by the numerical integration of instantaneous heat flux combining the Newton–Cotes rule (open type for improper integration at $t = 0$ in the case of uniform temperature distribution as an initial condition) and the Simpson's rule after the evaluation of instantaneous heat flux using the three-point rule. For sprue and runner, the same treatment as the part analysis was used except for the use of the cylindrical coordinate system. The advantages of this method over the analytical series solution are as follows: (i) it is good with using filling analysis results as an initial condition and (ii) it is good for considering temperature dependent thermal properties such as thermal conductivity, heat capacity, etc.

(ii) Analytical series solution — One may also use the series solution with a uniform injection temperature (T_m) as an initial condition. Figure 5 schematically shows the geometry and temperature response in the melt region. The analytical series solution is available as follows:

$$T(z,t) = T^- + (T^+ - T^-)\frac{z}{b} + \sum_{n=1}^{\infty} a_n \exp\left(-\frac{n^2\pi^2\alpha t}{b^2}\right) \sin\left(\frac{n\pi z}{b}\right), \tag{11}$$

$$\text{where} \quad a_n = \frac{2}{n\pi}\left[\{1-(-1)^n\}T_m + (-1)^n T^+ - T^-\right].$$

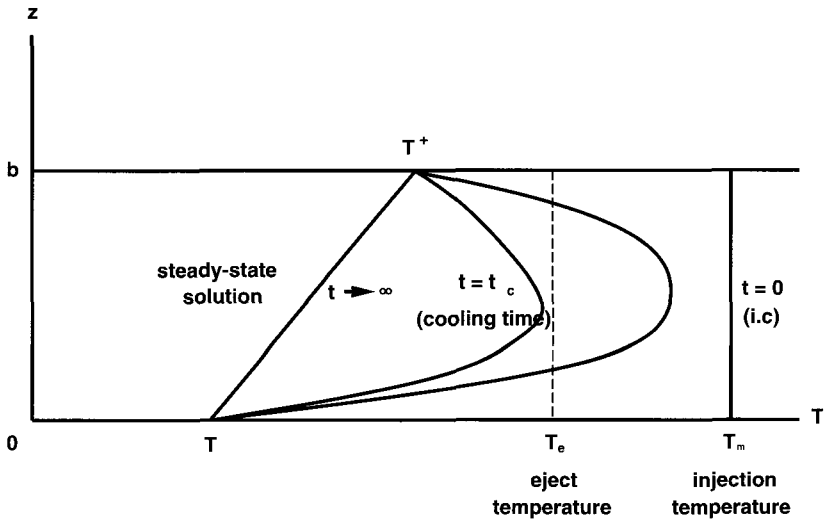


Fig. 5. Notations and temperature response in part.

Then one can obtain the cooling time of each type as follows:

(i) Type I

for each element, find $\bar{t}_c$ and $\bar{z}_c$ by solving following equation:

$$\frac{\partial T(z,t)}{\partial z} = 0, \qquad T(z,t) = T_e \qquad \text{at } t = \bar{t}_c,\ z = \bar{z}_c, \tag{12}$$

then find t_c and z_c as follows:

$$\begin{cases} t_c = \max \bar{t}_c & \text{for all elements,} \\ z_c = \bar{z}_c & \text{at the element which has the cooling time } t_c. \end{cases}$$

(ii) Type II

for each element, find $\bar{t}_c$ by solving following equation:

$$\frac{1}{b}\int_0^b T(z,t)\,dz = T_e \qquad \text{at } t = \bar{t}_c, \tag{13}$$

then find t_c as follows:

$$t_c = \max \bar{t}_c \quad \text{for all elements.}$$

One can solve Eqs. 12 and 13 to find $\bar{t}_c$ or $\bar{z}_c$ using the generalized Newton's method with an initial guess of value at $n = 1$. Equations 12 and 13 give us the cooling time which is very important since it is directly related to the overall productivity of the injection molding process. Moreover, these equations enable us to calculate the design sensitivity of the cooling time, which will be explained in detail in Sec. 3 of the present chapter. Finally, one can obtain the cycle-averaged heat flux on the minus plane neglecting the effects during filling and open times. This is shown in the equation below:

$$\frac{\partial T^-}{\partial n^-} = \frac{k_p}{k_m}\left[\frac{1}{b}\left(T^+ - T^-\right) - \frac{1}{t_c}\sum_{n=1}^{\infty}\frac{b}{n\pi\alpha}a_n\left\{\exp\left(-\frac{n^2\pi^2\alpha t_c}{b^2}\right) - 1\right\}\right]. \tag{14}$$

One can also determine the cycle-averaged heat flux on the plus plane by interchanging + and − in Eq. 14. For sprue-and-runner, one can use the same treatment similar to the part analysis except that the Bessel series in the cylindrical coordinate system is not used. The advantages of this method over the finite difference method are as follows: (i) there is greater accuracy and faster part analysis (including sprue-and-runner) as compared to the finite difference method and (ii) it is more suitable for sensitivity analysis.

Cooling channel surface

One should evaluate the heat transfer coefficient and the coolant bulk temperature. First of all one needs to evaluate the volumetric flow rate in each element for

branched cooling channels. To do this, a nonlinear one-dimensional finite element method was used based on the following relationship between the pressure and the flow rate in a turbulent pipe flow with an inlet volumetric flow rate or a relative pressure difference between the inlet and outlet prescribed: for an ith element,

$$P_{I,i} - P_{O,i} = 0.241 L_i \rho^{0.75} \mu^{0.25} D_i^{-4.75} Q_i^{1.75}, \tag{15}$$

$$\text{with} \quad \begin{cases} P = \bar{P} \text{ or } Q = \bar{Q} & \text{at inlet node,} \\ P = 0 & \text{at outlet node,} \\ \sum_{k=1}^{n} Q_k = 0 & \text{at other nodes.} \end{cases}$$

In Eq. 15, n in the summation symbol represents the total number of elements connected with each node. As an initial guess, one can use the analysis results of a laminar pipe flow via a linear one-dimensional finite element method.

After solving for the pressure at each node and subsequently evaluating the flow rate at each channel element, one can then evaluate the heat transfer coefficient of each cooling channel element using the following Dittus–Boetler correlation:

$$h_c = 0.023 \frac{k_c}{D} Re_D^{0.8} Pr^{0.4}, \tag{16}$$

where Re_D is the Reynolds number based on D and Pr is the Prandtl number.

Next, for the coolant bulk temperature in each element, the log mean difference concept was introduced (based on the simple energy balance under constant surface temperature) in each cooling channel element since a constant element was used. In particular, for the case of branched cooling channels, the average outlet temperature at the junction node was determined with the volumetric flow rate as a weighting factor as follows:

We first determine the outlet temperature at each element by the following equation for the ith element:

$$T_{O,i} = T_{S,i} - (T_{S,i} - T_{I,i}) \exp\left[-\frac{P_i L_i h_{c,i}}{\rho c_p Q_i}\right] \quad \text{if } P_{I,i} \geq P_{O,i}, \tag{17}$$

then the average outlet temperature at the junction node is determined by

$$T_O = \left(\sum_{k=1, P_{I,k} \geq P_{O,k}}^{n} T_{O,k} Q_k \right) \Bigg/ \left(\sum_{k=1, P_{I,k} \geq P_{O,k}}^{n} Q_k \right). \tag{18}$$

Then the coolant bulk temperature for each cooling channel element is obtained from

$$T_b = T_S + \frac{T_O - T_I}{\ln(T_S - T_O) - \ln(T_S - T_I)}. \tag{19}$$

It may be noted that this rigorous treatment of the boundary condition on cooling channels enables us to carry out sensitivity analysis with respect to process conditions such as inlet coolant bulk temperature and inlet volumetric flow rate which are part of the design variables.

Mold exterior surface

The boundary condition on the mold exterior surface is described below, depending on the types of mold exterior surface treatment.

(i) Type A — One can evaluate the heat transfer coefficient using the free convection correlation with a given temperature distribution on the mold exterior surface exposed to air (thus an iteration is required during the solution procedure). For the exterior surface in contact with the platens, one needs the thermal contact resistance and platen temperature.

(ii) Type B — In this case, the last terms on the right hand side of Eqs. 7–9 need not be evaluated because the integrals over the exterior surface are readily available as follows:

$$\int_{S_E} \left[\frac{1}{r} \frac{\partial T}{\partial n} - \frac{\partial}{\partial n} \left(\frac{1}{r} \right) T \right] dS = 4\pi T_\infty, \tag{20}$$

$$\int_{S_E} \left[\frac{\partial}{\partial \nu} \left(\frac{1}{r} \right) \frac{\partial T}{\partial n} - \frac{\partial}{\partial \nu} \left(\frac{\partial}{\partial n} \left(\frac{1}{r} \right) \right) T \right] dS = 0. \tag{21}$$

Therefore one should obtain T_∞ which represents the temperature on the mold exterior surface. In order to evaluate this value, the following iterative algorithm is introduced based on the requirement of satisfying the energy balance. One may first introduce some definitions to derive the iterative algorithm as follows:

$$HG = k_m \int_\Gamma \left(\frac{\partial T^+}{\partial n^+} + \frac{\partial T^-}{\partial n^-} \right) dS + k_m \int_\lambda \int_0^{2\pi} \frac{\partial T}{\partial n} \, a \, d\theta \, dl, \tag{22}$$

$$HL = k_m \sum_{k=1}^{N} \int_{l_k} \int_0^{2\pi} \left(-\frac{\partial T}{\partial n} \right) a_k \, d\theta \, dl, \tag{23}$$

$$HBE(\%) = \frac{HG - HL}{HG + HL} \times 100. \tag{24}$$

In the above equations HG, HL, and HBE denote the total heat gain from the polymer melt, the total heat loss through the cooling channels, and the heat-balanced error, respectively. Note that the heat flux from the polymer melt to the mold and the heat flux from the mold to the cooling channels are both positive. Now T_∞ can be determined by the following iterative algorithm,

$$(T_\infty)^{\text{new}} = (T_\infty)^{\text{old}} + T_\alpha \cdot HBE, \tag{25}$$

until HBE approaches zero to satisfy the energy balance for the steady state heat transfer. In Eq. 25, T_α is a relaxation parameter of which the unit is the temperature. This iterative algorithm will eventually enable us to evaluate the design sensitivity analysis as will be described in Sec. 3 of the present chapter.

(iii) Type C — For this case, the last term of Eqs. 7–9 are also readily available:

$$\int_{S_E} \left[\frac{1}{r} \frac{\partial T}{\partial n} - \frac{\partial}{\partial n} \left(\frac{1}{r} \right) T \right] dS = 4\pi T_{om} + 4\pi R \left(\frac{\partial T}{\partial n} \right)_{om}, \text{ and} \tag{26}$$

$$\int_{S_E} \left[\frac{\partial}{\partial \nu} \left(\frac{1}{r} \right) \frac{\partial T}{\partial n} - \frac{\partial}{\partial \nu} \left(\frac{\partial}{\partial n} \left(\frac{1}{r} \right) \right) T \right] dS = 0, \tag{27}$$

which are similar to Eqs. 20 and 21. Now T_{om} and $(\partial T/\partial n)_{om}$ on the exterior mold surface can be determined by the following iterative algorithm similar to that in Type B:

$$(T_{om})^{\text{new}} = (T_{om})^{\text{old}} + T_\alpha \cdot HBE, \tag{28}$$

$$-k_m \left(\frac{\partial T}{\partial n} \right)_{om} = h_{air}(T_{om} - T_{air}). \tag{29}$$

2.3.3. *Solution procedure*

Once the integrals on each element are calculated from Eqs. 7–10, the discretized boundary element formulae for the analysis can be manipulated to the following matrix equation form:

$$[H_{ij}] \{T_j\} = [G_{ij}] \left\{ \left. \frac{\partial T}{\partial n} \right|_j \right\}, \tag{30}$$

where $[H_{ij}]$ and $[G_{ij}]$ are functions of geometry of boundary surfaces. Some of $[H_{ij}]$ and $[G_{ij}]$ are singular integrals, which can be evaluated analytically in a simple manner as suggested by Rezayat and Burton[5] (see Appendix B). Also, the integrals over θ in Eqs. 7–10 both on the circumferential surfaces and on the tip surfaces, are evaluated in a closed analytical form using the complete elliptic integrals as proposed by Park and Kwon[11] (see Appendix B). Other integral terms are evaluated by the Gaussian quadrature integration.

Next, the boundary conditions can be introduced to obtain a system of linear algebraic equations:

$$[A_{ij}] \{T_j\} = \{f_i\}, \tag{31}$$

where $[A_{ij}]$ and $\{f_i\}$ reflect the boundary conditions. For this problem, temperature T, is taken as an unknown on each element with $\partial T/\partial n$ being eliminated with the help of a mixed boundary condition. This system of equations can be solved either by LU-decomposition through Gaussian elimination or by an under-relaxation iterative method. It should be noted that it is very easy to incorporate the iteration algorithm described in Eqs. 25, 28 and 29 into the solution procedure for Eq. 31 because only the forcing term, $\{f_i\}$, changes in each iteration.

2.4. *Results and discussions*

A simple injection mold cooling problem as depicted in Fig. 6 is presented as an example to demonstrate the importance of several issues in the present study. The mold has two cooling channels, and the dimension of the rectangular part is 200 × 200 × 2 (mm).

The radius of each cooling channel is 0.4 cm for cooling channel #1 and 0.6 cm for cooling channel #2, respectively. The global coordinate system was chosen such that the x-coordinate is in the axial direction of the cooling channels, the y-coordinate in the thickness direction of the part, and the z-coordinate in the spanwise direction of the part, as shown in Fig. 6. The discretization for the boundary element analysis is shown in Fig. 7 where each asterisk represents a node. The number of elements in the part, the sprue and the two cooling channels are 558, 5 and 30, respectively. HR 750 is used as a mold material whose thermal conductivity is 105 W/m·K. Polyethylene

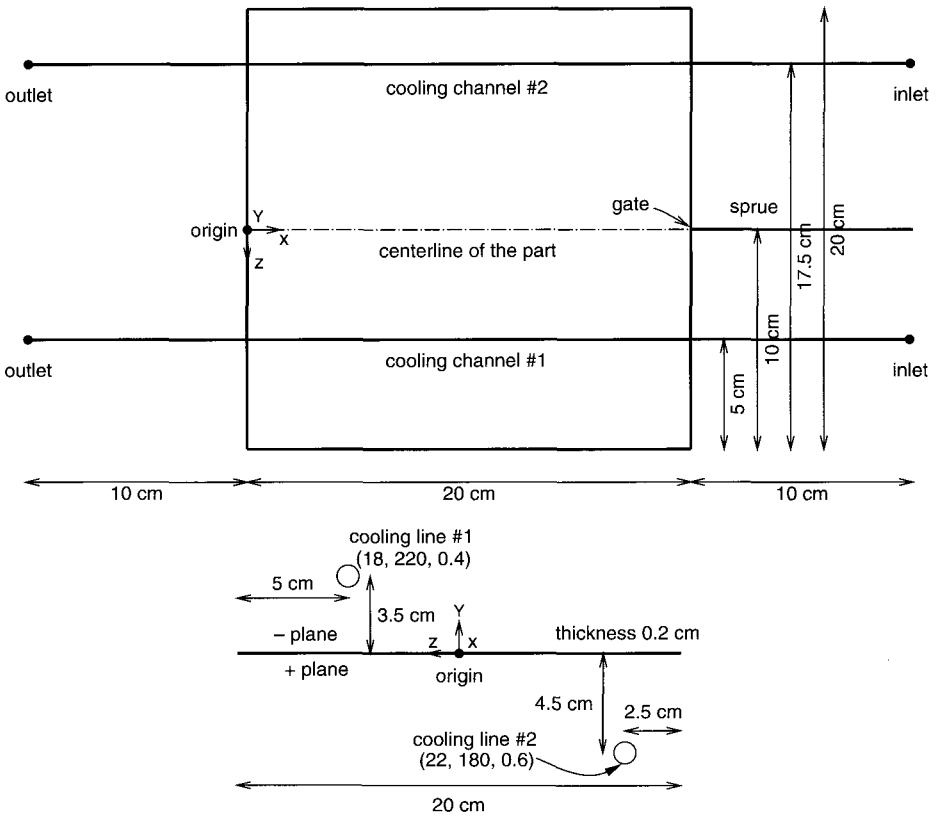


Fig. 6. Mold geometry of an example problem.

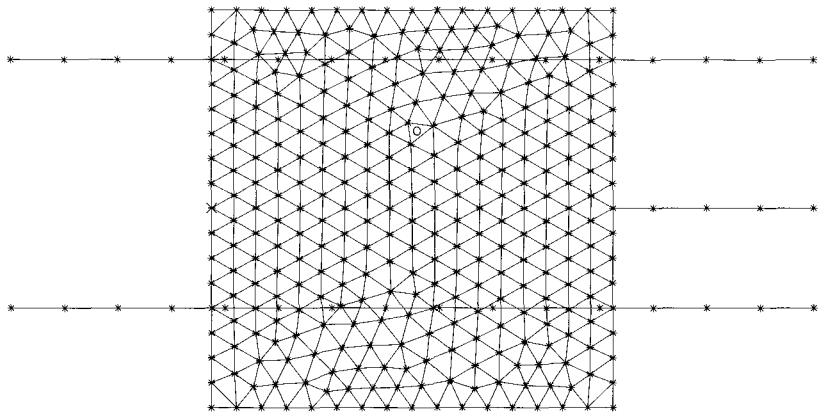


Fig. 7. Boundary element mesh for the example.

and water are used as part material and coolant, respectively. For this example, the following process conditions are introduced:

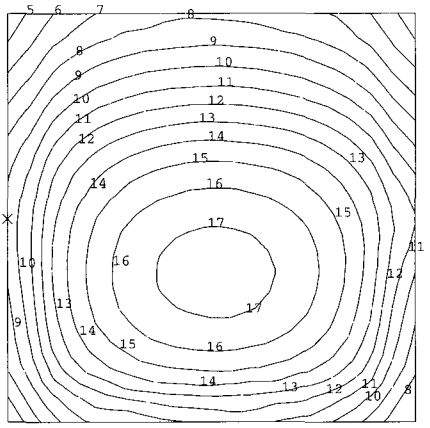
- The injection and ejection temperatures of part material are 250°C and 110°C, respectively;
- The inlet coolant temperature and the volumetric flow rate are 18°C and 220 cm^3/s respectively for cooling channel #1, and 22°C and 180 cm^3/s respectively for cooling channel #2;
- The ambient air temperature is 20°C.

The sizes, the locations and the process conditions of cooling channel #1 and cooling channel #2 are made unsymmetrical intentionally to show their effects on the cooling analysis.

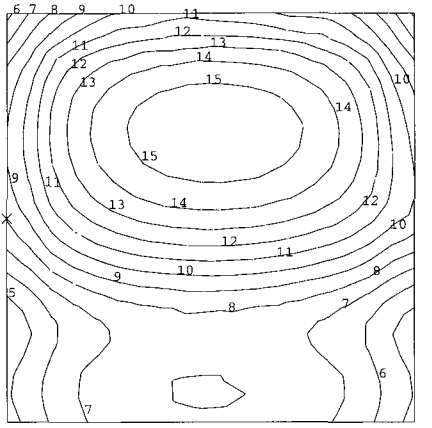
Convergency

Figure 8 illustrates the analysis results of the part surface temperature and the heat flux distribution in each plane. These analysis results were obtained when Type I for the cooling time and Type B for the mold exterior surface treatment are used together with the series solution, Eq. 11. The elapsed CPU time for this analysis is 0:22:19 (hr:min:s) in SUN SPARC 10 (22.9 MFLOPS/102 mips). The maximum surface temperature in each plane is located in the position opposite to two cooling channels in the direction of the y-coordinate and the z-coordinate, respectively as indicated in Fig. 8. Moreover, the maximum temperature in the plus plane is higher than that in the minus plane, thus cooling channel #1 has a more cooling effect than cooling channel #2, as expected. In this calculation, the maximum norm convergence criterion is used in the following iterative solution method:

$$\max_i \left| \frac{T_i^{\text{new}} - T_i^{\text{old}}}{T_i^{\text{old}}} \right| \leq 10^{-n}, \tag{32}$$

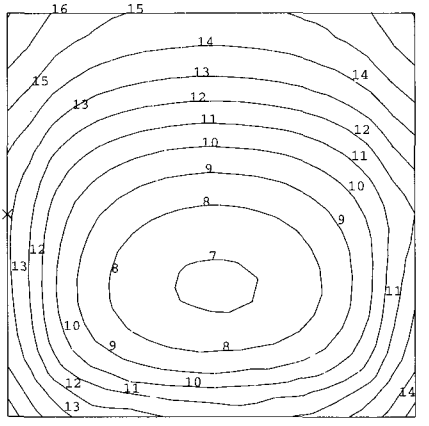


min: 61.48 max: 87.37
plus plane

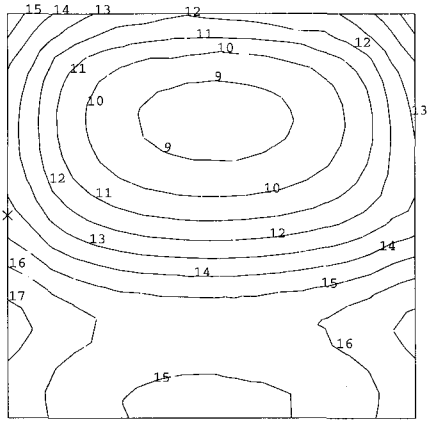


min: 61.80 max: 84.71
minus plane

(a) temperature (unit: °C)



min: 5.61×10^7 max: 6.61×10^7
plus plane



min: 5.76×10^7 max: 6.63×10^7
minus plane

(b) heat flux (unit: g/s^3)

Fig. 8. Distributions of temperature and heat flux on the part surface: (a) temperature [level value = 75 + (level number − 11) × 2] and (b) flux distribution [level value = {6 + (level number − 11)/10} × 10^7].

where i changes from 1 to the number of unknowns. In Fig. 9, the abscissa of both plots (a) and (b) represents the x-coordinate of the centerline of the part with their ordinates representing, respectively, the temperature on both planes and the temperature difference between the plus and minus plane, $T^+ - T^-$ (the number in legend

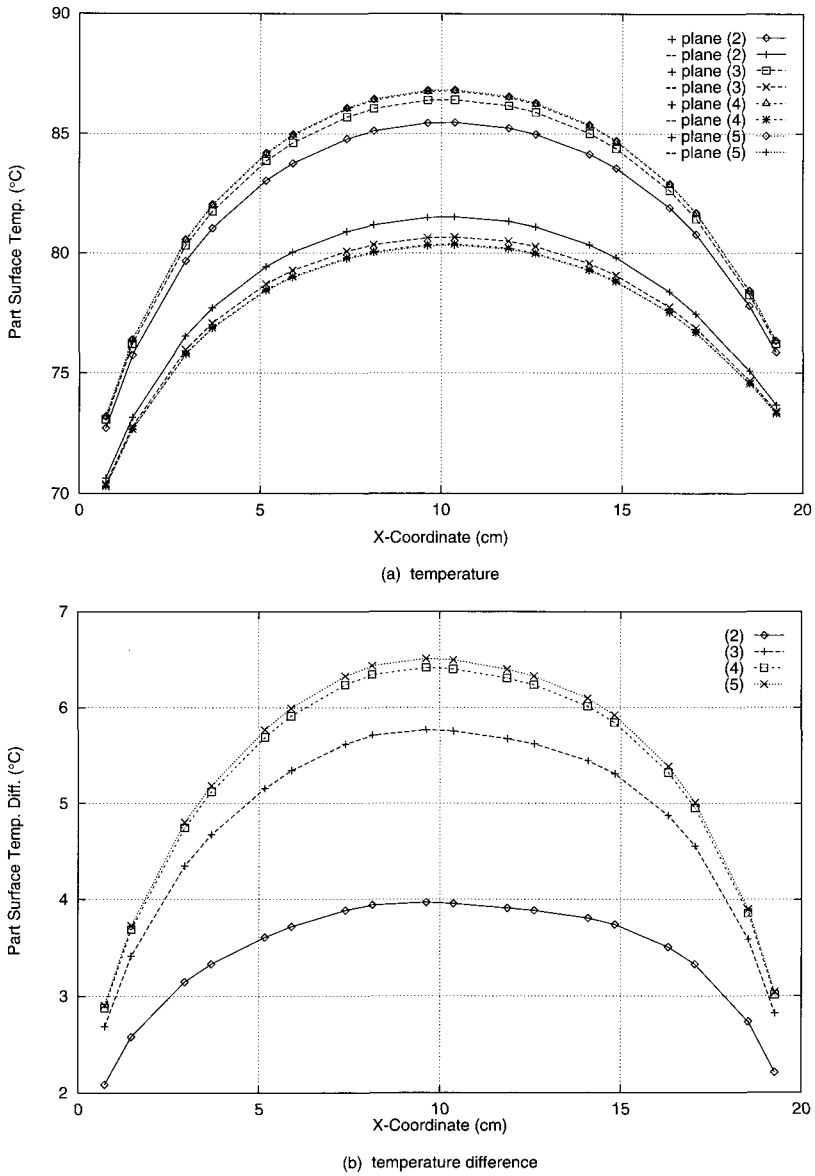


Fig. 9. Effect of convergence criterion on (a) temperature and (b) temperature difference on the part surface along the centerline.

denotes n in the above convergence criterion expression). This result may indicate that the accuracy in the temperature distribution is good enough when $n \geq 3$, but the accuracy in the temperature difference can be obtained when $n \geq 4$. Therefore, $n \geq 4$ is used in this study for thermal analysis and sensitivity analysis via the direct differentiation approach which will be discussed in Sec. 3. It may be further

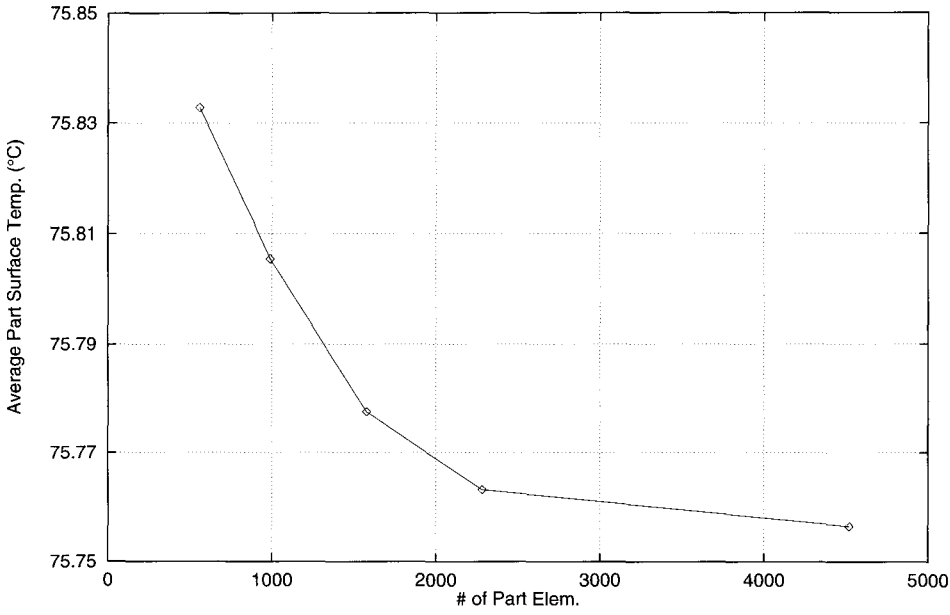


Fig. 10. A representative effect of boundary element discretization on average part surface temperature.

noted that the highly accurate sensitivity analysis via the finite difference method requires, surprisingly, quite an accurate thermal analysis with $n \geq 10$ (otherwise, the finite difference method sensitivity analysis result becomes meaningless), which will be discussed in detail in Sec. 3. Accuracy is also checked in another way, that is, by means of the discretization effect as indicated in Fig. 10. Figure 10 shows that the average part surface temperature converges as the number of elements increases, which verifies the convergent nature of the boundary element analysis. It was found that the average part surface temperature is slightly overestimated when a small number of elements are used.

Series solution

The evaluated cooling time is 6.46 and 5.63 seconds for Type I and Type II, respectively. In Fig. 7, O indicates the position where the cooling time is determined for both Type I and Type II. This corresponds to the element which has the maximum value of the sum of temperature on both surfaces. Using Type I, Fig. 11 shows the comparison between the series solution and the finite difference method in solving Eq. 4. In Fig. 11, the abscissa of both plots (a) and (b) represents the x-coordinate of the centerline of the part with their ordinates representing, respectively, the temperature on both planes and the temperature difference between the plus and minus plane, $T^+ - T^-$. According to these results, the time step should be at least 10^{-3} seconds in order for the finite difference method to obtain almost the same result

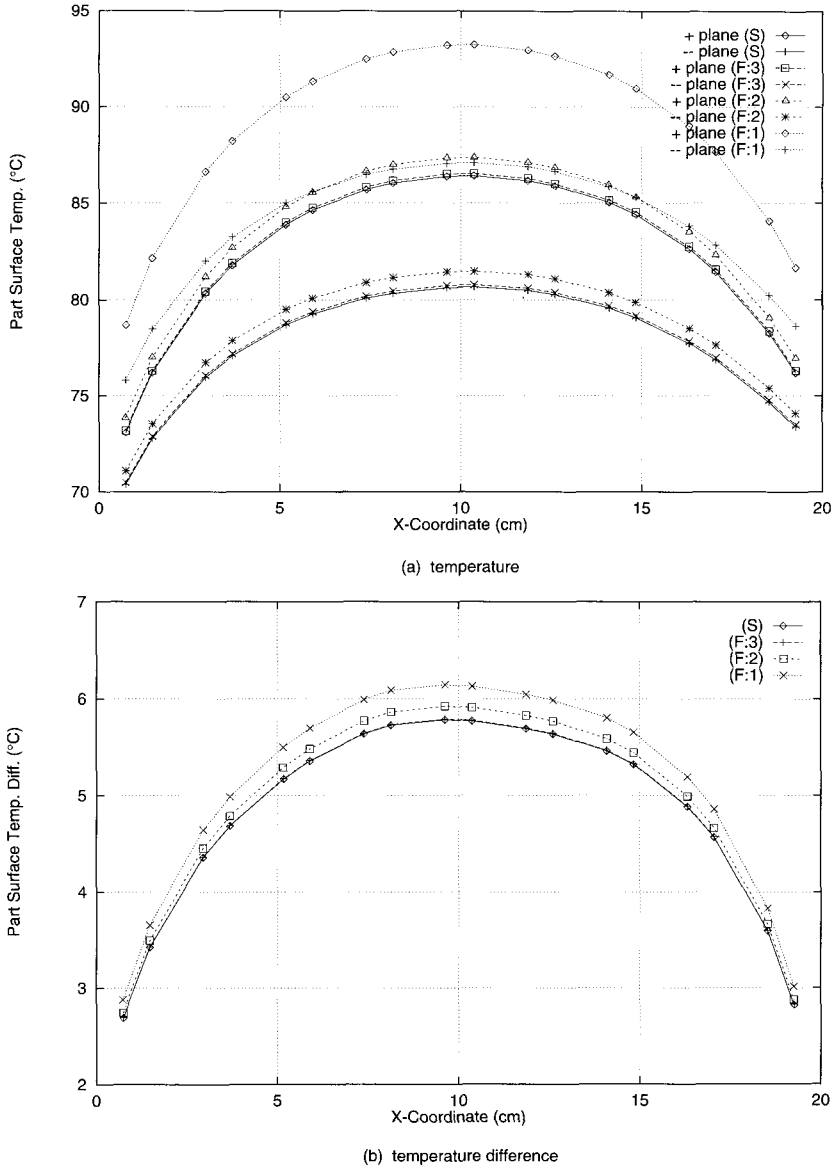


Fig. 11. Comparisons between the analytical series solution and the FDM for the part analysis: (a) temperature and (b) temperature difference along the centerline (S: series solution, F: FDM, number: n in $\Delta t = 10^{-n}$).

as that by the series solution. With the time step of 10^{-3} seconds, the cooling time is found to be 6.47 seconds in comparison with 6.46 seconds by the series solution. In this case, the total elapsed CPU time is about ten times as long as that of the series solution case. It should be emphasized that the series solution is much faster than the finite difference method for the same accuracy. In the sprue, one can

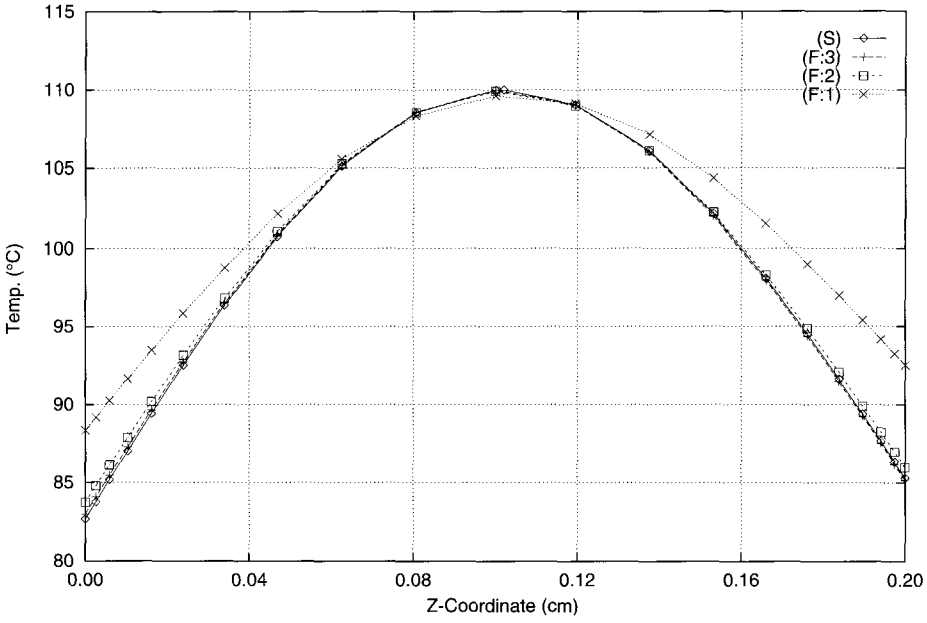
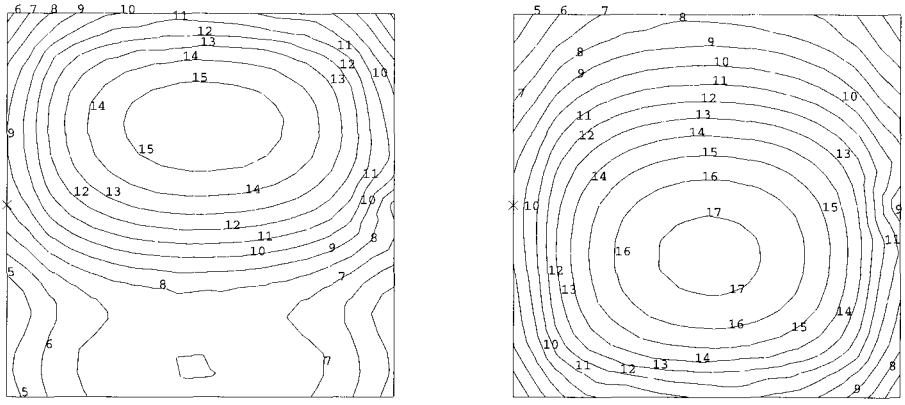


Fig. 12. The temperature distribution in the thickness direction at ejection time in the element where the cooling time is determined (meanings of legends are the same as in Fig. 11).

observe the trends similar to the part case. Under Type I, Fig. 12 shows the temperature distribution in the thickness direction at the location indicated by O in Fig. 7 at the ejection time using both the series solution and the finite difference methods. One can examine the asymmetric temperature distribution from this result, which causes the value of z_c to be anything but $b/2$. It may be worth mentioning that this asymmetry needs to be taken into account in the design sensitivity analysis.

Tip surface

When directly compared with Fig. 8a, Fig. 13 shows the temperature distribution of part surface on both planes when the tip surface of the sprue is not taken into account (that is, when the third and fifth terms in Eqs. 7–9 are omitted). One can observe the cooling effect near the gate since the tip surface, which is a greater heat source, is excluded in this calculation. This may pose a problem in the cooling analysis because the basic assumption on the Green's second identity is not satisfied. This phenomenon was observed not only in the part surface but also in the sprue. Figure 14 shows the effect on the surface temperature in the sprue when the sprue tip surface is not considered. In the figure, the abscissa represents the x-coordinate of line element of sprue with their ordinates representing the surface temperature ('YES' and 'NO' in the legend denote the case when the sprue tip surface is considered, and the case when it is not considered, respectively). The result without



min: 61.34 max: 87.70
plus plane

min: 61.32 max: 84.54
minus plane

(unit: °C)

Fig. 13. Tip surface effect: surface temperature distribution when the tip surface is excluded [level value = 75 + (level number − 11) × 2].

the sprue tip surface included gives us unrealistic temperature distributions. In the cooling channels, this effect is observed but it is not as significant as that for the sprue tip. This comparison illustrates the importance of taking into account the tip surface of the sprue and the cooling channels. It may be worth noting that the sensitivity analysis is also affected by the sprue tip surface.

Coolant bulk temperature

In Fig. 15, the abscissa of all plots represents the x-coordinate of the cooling channel with the ordinate of plots (a), (b) and (c) representing the coolant bulk temperature, the channel surface temperature, and the heat flux, respectively. Figure 15a shows the bulk temperature at each element center and its respective node. Smooth curves in this figure indicate that the log mean temperature concept enables us to consider the cooling-related processing conditions as design variables in the optimum cooling system design. Also this result gives us the total increase of coolant bulk temperature from the inlet to the outlet, which is quite an important information for setting up the processing conditions of the cooling stage. Figures 15b and 15c show the surface temperature and the heat flux. These results will be needed to explain the characteristics of each design variable in Sec. 3 (see Sec. 3.4).

Mold exterior surface

Three different ways of treating the mold exterior surface were applied to this problem to assess the approximate methods, Types B and C. The analysis results are

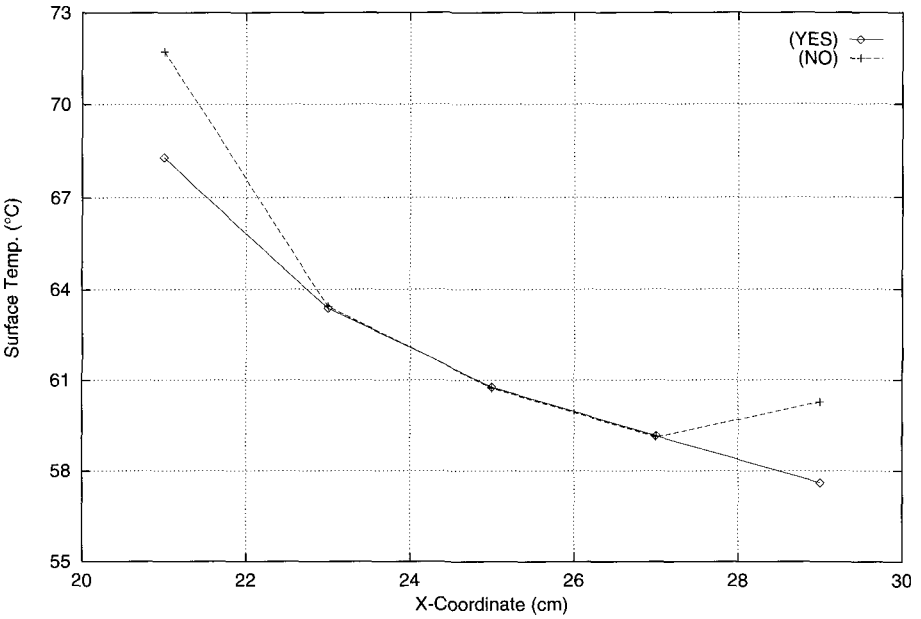


Fig. 14. The surface temperature distribution along the sprue.

summarized below: For Type A the used platen temperature is 40°C and the used thermal contact resistance is $1.5 \times 10^{-2}\,\mathrm{m^2{\cdot}K/W}$. The number of elements on the mold exterior surface used in this analysis was 342. As a result, the averaged mold exterior surface temperature is 50.1°C, the averaged part surface temperature is 75.4°C, and the percentage of total heat loss through the mold exterior surface is 2.22%. For Type B, T_∞ is 53.2°C, and the averaged part surface temperature is 75.8°C. For Type C the used h_{air} is $4\,\mathrm{W/m^2{\cdot}K}$ and the used R is 0.276 m. As a result, T_{om} is 52.2°C, the averaged part surface temperature is 75.4 °C, and the percentage of total heat loss through the mold exterior surface is 2.31%. From the above comparisons, the approximations of Types B and C are good enough in view of the extra human labor of discretization on the mold exterior surface and the additional computational time for the Type A approach.

2.5. *Concluding remarks*

For an optimum design of the injection mold cooling system (to achieve fast and uniform cooling), mold designers need a thermal analysis tool which is predicting heat transfer within the mold and the part melt. It has been found that a seemingly negligible inaccuracy in the thermal analysis result can sometimes become detrimental enough to give rise to meaningless sensitivity analysis results. In this regard, the possible origins of the inaccuracy in the thermal analysis have been thoroughly examined, however minor they might be in the sense of the analysis result

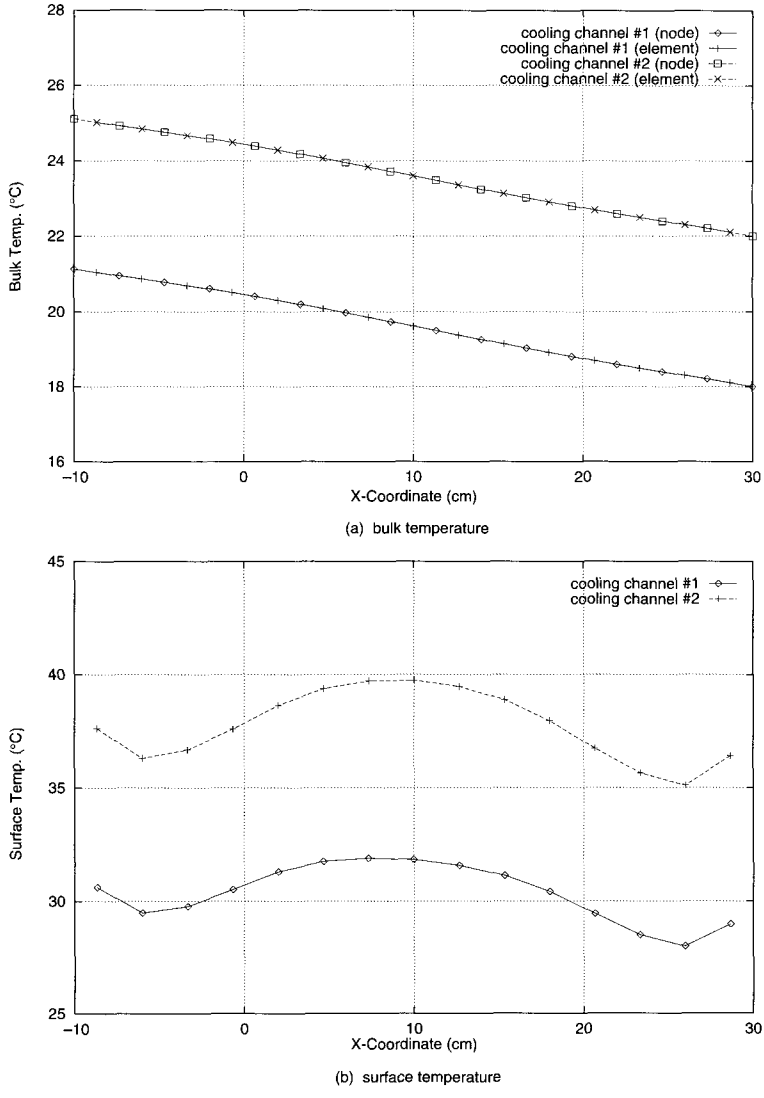


Fig. 15. The distribution along the cooling channel: (a) bulk temperature, (b) surface temperature and (c) heat flux.

itself, and the analysis tool improved accordingly in order to obtain an accurate sensitivity analysis result.

With this in mind, the thermal analysis system based on the modified boundary element method has been improved by considering the following issues:

(i) the convergence criteria for obtaining an accurate temperature distribution on the mold surface, which also guarantees the accuracy of design sensitivity analysis;

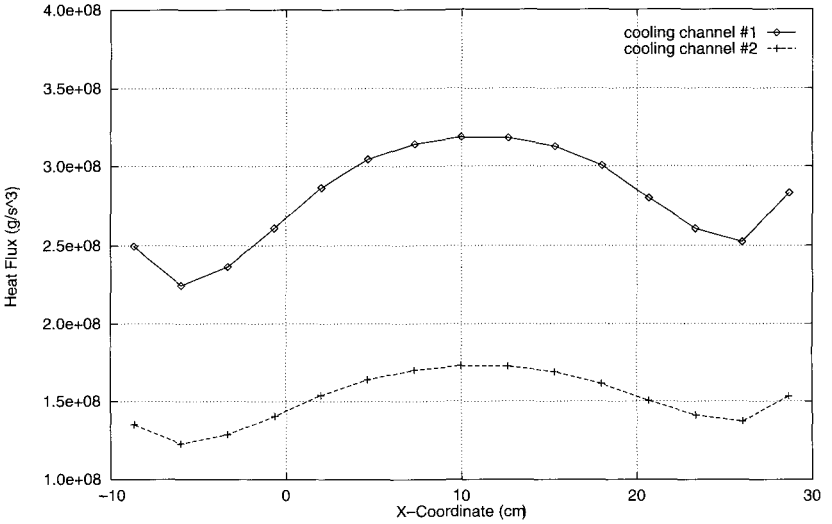


Fig. 15. *Continued*

(ii) the analytical series solution in the plastic part analysis for determining the cooling time and the cycle-averaged heat flux on each element of cavity surface;
(iii) the tip surface of line elements with circular cross section for satisfying the basic assumption of boundary integral equation;
(iv) the treatment of coolant in consideration of the cooling process condition, and
(v) the treatment of mold exterior surface associated with an algorithm to determine the accurate temperature on the mold exterior surface.

All these items contribute greatly towards overcoming various numerical difficulties in obtaining the corresponding design sensitivity analysis.

3. Design Sensitivity Analysis

3.1. *Introduction*

Once the thermal analysis has been successfully achieved, as described in Sec. 2, the design sensitivity analysis is naturally the next target towards the optimal design of the injection mold cooling system. Design sensitivity coefficients (representing gradients of the variables of interest with respect to design variables) are required by the nonlinear optimization algorithms with first order methods such as the steepest descent method or the CONMIN algorithm to determine the search direction for better designs during the iterative algorithm of optimization methods. The determination of accurate design sensitivities is of crucial significance since inaccurate evaluations of design sensitivities may lead to an increased number of iterations and may even render the effort unsuccessful.

In recent years, the boundary element method has gained importance not only as an analysis tool but also as a sensitivity analysis tool. Two methods, namely the direct differentiation approach (DDA) and the adjoint structure approach (ASA), have emerged as the most significant ones for the evaluation of design sensitivity coefficient using the boundary element method. Both the approaches have recently been developed in the context of boundary element method (BEM) for elastic systems. Examples of such approaches can be found in the papers by Barone and Yang[12] for two-dimensional systems, and Saigal *et al.*[13] for axisymmetric systems, etc. These approaches have more recently been applied to several thermal systems such as the steady-state diffusion problems. Examples of such approaches can be found in the papers by Saigal and Chandra[14] for two-dimensional and axisymmetric heat diffusion problems, Prasad and Kane[15] for three-dimensional heat conduction problems, as well as Park and Kwon[16] for three-dimensional heat conduction problems in special geometries, etc.

In this section, an efficient and accurate approach for the design sensitivity analysis for the injection mold cooling system is proposed using the direct differentiation approach based on the modified special boundary integral formulation as presented in Sec. 2. The major advantage of this approach lies in the fact that the accurate sensitivity analysis results can be obtained simultaneously for all design variables, thus saving computational time. To maximize this advantage, a special rearrangement of evaluated matrices and some definitions about the sensitivities of the boundary conditions are introduced for the use of the solution procedure similar to that of the thermal analysis case in Sec. 2. In this sensitivity analysis, several design variables have been considered as follows: (i) (design variables related to processing conditions) the inlet coolant bulk temperature and the inlet coolant volumetric flow rate of each cooling channel, and (ii) (design variables related to mold cooling system design) the radius and location of each cooling channel. Using a simple and yet representative example, the numerical results of the sensitivity analysis simulation developed in the present study are discussed in terms of accuracy and computational efficiency. The accuracy of the present formulation is demonstrated through favorable comparisons between the present direct differentiation approach and a numerical derivative approach using the forward finite difference method (FDM). Furthermore, the characteristics of design variables are also discussed in detail to enhance the understanding of the nature of the cooling system design.

3.2. *Design variables and basic assumption*

In the injection mold cooling system design, with the moldbase and part geometry being fixed, the size and arrangement of the cooling channels can be changed to achieve a certain design objective related to the productivity and/or the uniformity. Furthermore, one can also change the processing conditions of the cooling stage in the injection molding process. Therefore, in the present sensitivity analysis, several design variables are considered as follows: (i) (design variables related to processing

conditions) the inlet coolant bulk temperature and the inlet coolant volumetric flow rate of each cooling channel, and (ii) (design variables related to mold cooling system design) the radius and location of each cooling channel.

In the present sensitivity analysis, the following options among those described in Sec. 2 of this chapter have been used:

(i) Part analysis — the analytical series solution with a uniform injection temperature as an initial condition;
(ii) Cooling time — Type I (maximum temperature criterion);
(iii) Mold exterior surface treatment — Type B (adiabatic infinite sphere approximation); and
(iv) Tip surfaces of circular hole segments — included.

3.3. *Formulation for sensitivity analysis*

3.3.1. *Boundary integral formulation for sensitivity analysis*

The boundary element method is also a strong tool in design sensitivity analyzes when the problem can be treated without the domain discretization and the design variables can be defined on the boundary. It is interesting to note that all the design variables mentioned above are defined on the boundary. Therefore, the boundary element method can be effectively applied to the sensitivity analysis of this kind of heat transfer problem.

The design sensitivity analysis is essential to determine the variation rate of an objective function with respect to the variations of design variables, which is required in the design optimization algorithm. Thus the sensitivity values of the cooling time, the temperature and the heat flux with respect to these design variables are to be evaluated for this purpose. In the present study, to determine such sensitivity values, the implicit differentiation of Eqs. 7–9 with respect to each design variable was introduced resulting in a system of design sensitivity equations for all design variables, as will be described below.

First, the design variables related to processing conditions are considered: (i) for a point $\vec{x}$ on the mid-surface of the part, Γ, the implicit derivatives of Eqs. 7 and 8 with respect to a design variable, X, give us the following pair of integral equations for the sensitivity analysis,

$$\begin{aligned}
&\alpha^{-}\frac{\partial T^{+}(\vec{x})}{\partial X}+\alpha^{+}\frac{\partial T^{-}(\vec{x})}{\partial X}\\
&\quad=\int_{\Gamma}\left[\frac{1}{r}\left\{\frac{\partial}{\partial X}\left(\frac{\partial T^{+}}{\partial n^{+}}\right)+\frac{\partial}{\partial X}\left(\frac{\partial T^{-}}{\partial n^{-}}\right)\right\}\right.\\
&\qquad\left.-\frac{\partial}{\partial n^{+}}\left(\frac{1}{r}\right)\left(\frac{\partial T^{+}}{\partial X}-\frac{\partial T^{-}}{\partial X}\right)\right]dS(\vec{\xi})\\
&\qquad+\int_{\lambda}\left[\frac{\partial}{\partial X}\left(\frac{\partial T}{\partial n}\right)\int_{0}^{2\pi}\left(\frac{1}{r}\right)d\theta-\frac{\partial T}{\partial X}\int_{0}^{2\pi}\frac{\partial}{\partial n}\left(\frac{1}{r}\right)d\theta\right]a\,dl(\vec{\xi})
\end{aligned}$$

$$
\begin{aligned}
&+\int_{\varrho}\left[\frac{\partial}{\partial X}\left(\frac{\partial T}{\partial n}\right)\int_0^{2\pi}\left(\frac{1}{r}\right)d\theta-\frac{\partial T}{\partial X}\int_0^{2\pi}\frac{\partial}{\partial n}\left(\frac{1}{r}\right)d\theta\right]\rho\,d\rho(\vec{\xi})\\
&+\sum_{k=1}^{N}\int_{l_k}\left[\frac{\partial}{\partial X}\left(\frac{\partial T}{\partial n}\right)\int_0^{2\pi}\left(\frac{1}{r}\right)d\theta-\frac{\partial T}{\partial X}\int_0^{2\pi}\frac{\partial}{\partial n}\left(\frac{1}{r}\right)d\theta\right]a_k\,dl(\vec{\xi})\\
&+\sum_{k=1}^{N}\int_{\rho_k}\left[\frac{\partial}{\partial X}\left(\frac{\partial T}{\partial n}\right)\int_0^{2\pi}\left(\frac{1}{r}\right)d\theta-\frac{\partial T}{\partial X}\int_0^{2\pi}\frac{\partial}{\partial n}\left(\frac{1}{r}\right)d\theta\right]\rho\,d\rho(\vec{\xi})\\
&+4\pi\frac{\partial T_\infty}{\partial X},
\end{aligned}
\tag{33}
$$

$$
\begin{aligned}
&\alpha^-\frac{\partial}{\partial X}\left(\frac{\partial T^+(\vec{x})}{\partial \nu}\right)-\alpha^+\frac{\partial}{\partial X}\left(\frac{\partial T^-(\vec{x})}{\partial \nu}\right)\\
&=\int_\Gamma\left[\frac{\partial}{\partial \nu}\left(\frac{1}{r}\right)\left\{\frac{\partial}{\partial X}\left(\frac{\partial T^+}{\partial n^+}\right)+\frac{\partial}{\partial X}\left(\frac{\partial T^-}{\partial n^-}\right)\right\}\right.\\
&\quad\left.-\frac{\partial}{\partial \nu}\left(\frac{\partial}{\partial n^+}\left(\frac{1}{r}\right)\right)\left(\frac{\partial T^+}{\partial X}-\frac{\partial T^-}{\partial X}\right)\right]dS(\vec{\xi})\\
&\quad+\int_\lambda\left[\frac{\partial}{\partial X}\left(\frac{\partial T}{\partial n}\right)\int_0^{2\pi}\frac{\partial}{\partial \nu}\left(\frac{1}{r}\right)d\theta-\frac{\partial T}{\partial X}\int_0^{2\pi}\frac{\partial}{\partial \nu}\left(\frac{\partial}{\partial n}\left(\frac{1}{r}\right)\right)d\theta\right]a\,dl(\vec{\xi})\\
&\quad+\int_{\varrho}\left[\frac{\partial}{\partial X}\left(\frac{\partial T}{\partial n}\right)\int_0^{2\pi}\frac{\partial}{\partial \nu}\left(\frac{1}{r}\right)d\theta-\frac{\partial T}{\partial X}\int_0^{2\pi}\frac{\partial}{\partial \nu}\left(\frac{\partial}{\partial n}\left(\frac{1}{r}\right)\right)d\theta\right]\rho\,d\rho(\vec{\xi})\\
&\quad+\sum_{k=1}^{N}\int_{l_k}\left[\frac{\partial}{\partial X}\left(\frac{\partial T}{\partial n}\right)\int_0^{2\pi}\frac{\partial}{\partial \nu}\left(\frac{1}{r}\right)d\theta\right.\\
&\quad\left.-\frac{\partial T}{\partial X}\int_0^{2\pi}\frac{\partial}{\partial \nu}\left(\frac{\partial}{\partial n}\left(\frac{1}{r}\right)\right)d\theta\right]a_k\,dl(\vec{\xi})\\
&\quad+\sum_{k=1}^{N}\int_{\rho_k}\left[\frac{\partial}{\partial X}\left(\frac{\partial T}{\partial n}\right)\int_0^{2\pi}\frac{\partial}{\partial \nu}\left(\frac{1}{r}\right)d\theta\right.\\
&\quad\left.-\frac{\partial T}{\partial X}\int_0^{2\pi}\frac{\partial}{\partial \nu}\left(\frac{\partial}{\partial n}\left(\frac{1}{r}\right)\right)d\theta\right]\rho\,d\rho(\vec{\xi}),
\end{aligned}
\tag{34}
$$

and (ii) for a point on the axis of the circular hole (sprue, runner or cooling channel), the implicit derivative of Eq. 9 with respect to design variable, X, gives us the following integral equation for the sensitivity analysis,

$$
\begin{aligned}
0=&\int_\Gamma\left[\frac{1}{r}\left\{\frac{\partial}{\partial X}\left(\frac{\partial T^+}{\partial n^+}\right)+\frac{\partial}{\partial X}\left(\frac{\partial T^-}{\partial n^-}\right)\right\}\right.\\
&\left.-\frac{\partial}{\partial n^+}\left(\frac{1}{r}\right)\left(\frac{\partial T^+}{\partial X}-\frac{\partial T^-}{\partial X}\right)\right]dS(\vec{\xi})\\
&+\int_\lambda\left[\frac{\partial}{\partial X}\left(\frac{\partial T}{\partial n}\right)\int_0^{2\pi}\left(\frac{1}{r}\right)d\theta-\frac{\partial T}{\partial X}\int_0^{2\pi}\frac{\partial}{\partial n}\left(\frac{1}{r}\right)d\theta\right]a_k\,dl(\vec{\xi})
\end{aligned}
$$

$$+\int_{\varrho}\left[\frac{\partial}{\partial X}\left(\frac{\partial T}{\partial n}\right)\int_0^{2\pi}\left(\frac{1}{r}\right)d\theta-\frac{\partial T}{\partial X}\int_0^{2\pi}\frac{\partial}{\partial n}\left(\frac{1}{r}\right)d\theta\right]\rho\,d\rho(\vec{\xi})$$
$$+\sum_{k=1}^{N}\int_{l_k}\left[\frac{\partial}{\partial X}\left(\frac{\partial T}{\partial n}\right)\int_0^{2\pi}\left(\frac{1}{r}\right)d\theta-\frac{\partial T}{\partial X}\int_0^{2\pi}\frac{\partial}{\partial n}\left(\frac{1}{r}\right)d\theta\right]a_k\,dl(\vec{\xi})$$
$$+\sum_{k=1}^{N}\int_{\rho_k}\left[\frac{\partial}{\partial X}\left(\frac{\partial T}{\partial n}\right)\int_0^{2\pi}\left(\frac{1}{r}\right)d\theta-\frac{\partial T}{\partial X}\int_0^{2\pi}\frac{\partial}{\partial n}\left(\frac{1}{r}\right)d\theta\right]\rho\,d\rho(\vec{\xi})$$
$$+4\pi\frac{\partial T_\infty}{\partial X}. \tag{35}$$

In Eqs. 33–35, the design variable X can either be the inlet coolant bulk temperature or the inlet coolant volumetric flow rate of each cooling channel. These design variables does not change the mold geometry, and thus these sensitivity equations look similar to the thermal analysis equations, Eqs. 7–9, except for the partial derivatives involved.

Next, the design variables related to the mold cooling system are considered, that is, the radius and the location of each cooling channel, on which the mold geometry depends. Thus the implicit derivatives of Eqs. 7–9 with respect to such a design variable will include extra boundary integral terms in addition to those terms in Eqs. 33–35. First, the radius of the jth cooling channel, a_j, is considered as a design variable: (i) for the part surface, the implicit derivatives of Eqs. 7 and 8 with respect to the radius of the jth cooling channel, a_j, give us the following pair of integral equations for the sensitivity analysis,

$$\alpha^-\frac{\partial T^+(\vec{x})}{\partial a_j}+\alpha^+\frac{\partial T^-(\vec{x})}{\partial a_j}$$
$$=\int_\Gamma\left[\frac{1}{r}\left\{\frac{\partial}{\partial a_j}\left(\frac{\partial T^+}{\partial n^+}\right)+\frac{\partial}{\partial a_j}\left(\frac{\partial T^-}{\partial n^-}\right)\right\}\right.$$
$$\left.-\frac{\partial}{\partial n^+}\left(\frac{1}{r}\right)\left(\frac{\partial T^+}{\partial a_j}-\frac{\partial T^-}{\partial a_j}\right)\right]dS(\vec{\xi})$$
$$+\int_\lambda\left[\frac{\partial}{\partial a_j}\left(\frac{\partial T}{\partial n}\right)\int_0^{2\pi}\left(\frac{1}{r}\right)d\theta-\frac{\partial T}{\partial a_j}\int_0^{2\pi}\frac{\partial}{\partial n}\left(\frac{1}{r}\right)d\theta\right]a\,dl(\vec{\xi})$$
$$+\int_{\varrho}\left[\frac{\partial}{\partial a_j}\left(\frac{\partial T}{\partial n}\right)\int_0^{2\pi}\left(\frac{1}{r}\right)d\theta-\frac{\partial T}{\partial a_j}\int_0^{2\pi}\frac{\partial}{\partial n}\left(\frac{1}{r}\right)d\theta\right]\rho\,d\rho(\vec{\xi})$$
$$+\sum_{k=1}^{N}\int_{l_k}\left[\frac{\partial}{\partial a_j}\left(\frac{\partial T}{\partial n}\right)\int_0^{2\pi}\left(\frac{1}{r}\right)d\theta-\frac{\partial T}{\partial a_j}\int_0^{2\pi}\frac{\partial}{\partial n}\left(\frac{1}{r}\right)d\theta\right]a_k\,dl(\vec{\xi})$$
$$+\int_{l_j}\left[\frac{\partial T}{\partial n}\frac{\partial}{\partial a_j}\left\{a_j\int_0^{2\pi}\left(\frac{1}{r}\right)d\theta\right\}-T\frac{\partial}{\partial a_j}\left\{a_j\int_0^{2\pi}\frac{\partial}{\partial n}\left(\frac{1}{r}\right)d\theta\right\}\right]dl(\vec{\xi})$$
$$+\sum_{k=1}^{N}\int_{\rho_k}\left[\frac{\partial}{\partial a_j}\left(\frac{\partial T}{\partial n}\right)\int_0^{2\pi}\left(\frac{1}{r}\right)d\theta-\frac{\partial T}{\partial a_j}\int_0^{2\pi}\frac{\partial}{\partial n}\left(\frac{1}{r}\right)d\theta\right]\rho\,d\rho(\vec{\xi})$$

$$+ a_j \left[\frac{\partial T}{\partial n} \int_0^{2\pi} \left[\frac{1}{r} \right]_{\rho=a_i} d\theta - T \int_0^{2\pi} \left[\frac{\partial}{\partial n} \left(\frac{1}{r} \right) \right]_{\rho=a_i} d\theta \right] + 4\pi \frac{\partial T_\infty}{\partial a_j}, \tag{36}$$

$$\begin{aligned}
&\alpha^- \frac{\partial}{\partial a_j} \left(\frac{\partial T^+(\vec{x})}{\partial \nu} \right) - \alpha^+ \frac{\partial}{\partial a_j} \left(\frac{\partial T^-(\vec{x})}{\partial \nu} \right) \\
&= \int_\Gamma \left[\frac{\partial}{\partial \nu} \left(\frac{1}{r} \right) \left\{ \frac{\partial}{\partial a_j} \left(\frac{\partial T^+}{\partial n^+} \right) + \frac{\partial}{\partial a_j} \left(\frac{\partial T^-}{\partial n^-} \right) \right\} \right. \\
&\quad \left. - \frac{\partial}{\partial \nu} \left(\frac{\partial}{\partial n^+} \left(\frac{1}{r} \right) \right) \left(\frac{\partial T^+}{\partial a_j} - \frac{\partial T^-}{\partial a_j} \right) \right] dS(\vec{\xi}) \\
&\quad + \int_\lambda \left[\frac{\partial}{\partial a_j} \left(\frac{\partial T}{\partial n} \right) \int_0^{2\pi} \frac{\partial}{\partial \nu} \left(\frac{1}{r} \right) d\theta - \frac{\partial T}{\partial a_j} \int_0^{2\pi} \frac{\partial}{\partial \nu} \left(\frac{\partial}{\partial n} \left(\frac{1}{r} \right) \right) d\theta \right] a \, dl(\vec{\xi}) \\
&\quad + \int_\varrho \left[\frac{\partial}{\partial a_j} \left(\frac{\partial T}{\partial n} \right) \int_0^{2\pi} \frac{\partial}{\partial \nu} \left(\frac{1}{r} \right) d\theta - \frac{\partial T}{\partial a_j} \int_0^{2\pi} \frac{\partial}{\partial \nu} \left(\frac{\partial}{\partial n} \left(\frac{1}{r} \right) \right) d\theta \right] \rho \, d\rho(\vec{\xi}) \\
&\quad + \sum_{k=1}^{N} \int_{l_k} \left[\frac{\partial}{\partial a_j} \left(\frac{\partial T}{\partial n} \right) \int_0^{2\pi} \frac{\partial}{\partial \nu} \left(\frac{1}{r} \right) d\theta \right. \\
&\qquad \left. - \frac{\partial T}{\partial a_j} \int_0^{2\pi} \frac{\partial}{\partial \nu} \left(\frac{\partial}{\partial n} \left(\frac{1}{r} \right) \right) d\theta \right] a_k \, dl(\vec{\xi}) \\
&\quad + \int_{l_j} \left[\frac{\partial T}{\partial n} \frac{\partial}{\partial a_j} \left\{ a_j \int_0^{2\pi} \frac{\partial}{\partial \nu} \left(\frac{1}{r} \right) d\theta \right\} \right. \\
&\quad \left. - T \frac{\partial}{\partial a_j} \left\{ a_j \int_0^{2\pi} \frac{\partial}{\partial \nu} \left(\frac{\partial}{\partial n} \left(\frac{1}{r} \right) \right) d\theta \right\} \right] dl(\vec{\xi}) \\
&\quad + \sum_{k=1}^{N} \int_{\rho_k} \left[\frac{\partial}{\partial a_j} \left(\frac{\partial T}{\partial n} \right) \int_0^{2\pi} \frac{\partial}{\partial \nu} \left(\frac{1}{r} \right) d\theta \right. \\
&\quad \left. - \frac{\partial T}{\partial a_j} \int_0^{2\pi} \frac{\partial}{\partial \nu} \left(\frac{\partial}{\partial n} \left(\frac{1}{r} \right) \right) d\theta \right] \rho \, d\rho(\vec{\xi}) \\
&\quad + a_j \left[\frac{\partial T}{\partial n} \int_0^{2\pi} \left[\frac{\partial}{\partial \nu} \left(\frac{1}{r} \right) \right]_{\rho=a_i} d\theta - T \int_0^{2\pi} \left[\frac{\partial}{\partial \nu} \left(\frac{\partial}{\partial n} \left(\frac{1}{r} \right) \right) \right]_{\rho=a_i} d\theta \right];
\end{aligned} \tag{37}$$

(ii) for the ith cooling channel surface, the implicit derivative of Eq. 9 with respect to the radius of the jth cooling channel, a_j, gives us the following integral equation for the sensitivity analysis,

$$\begin{aligned}
0 = &\int_\Gamma \left[\frac{1}{r} \left\{ \frac{\partial}{\partial a_j} \left(\frac{\partial T^+}{\partial n^+} \right) + \frac{\partial}{\partial a_j} \left(\frac{\partial T^-}{\partial n^-} \right) \right\} \right. \\
&\left. - \frac{\partial}{\partial n^+} \left(\frac{1}{r} \right) \left(\frac{\partial T^+}{\partial a_j} - \frac{\partial T^-}{\partial a_j} \right) \right] dS(\vec{\xi})
\end{aligned}$$

$$
\begin{aligned}
&+ \int_{\lambda} \left[\frac{\partial}{\partial a_j} \left(\frac{\partial T}{\partial n} \right) \int_0^{2\pi} \left(\frac{1}{r} \right) d\theta - \frac{\partial T}{\partial a_j} \int_0^{2\pi} \frac{\partial}{\partial n} \left(\frac{1}{r} \right) d\theta \right] a \, dl(\vec{\xi}) \\
&+ \int_{\varrho} \left[\frac{\partial}{\partial a_j} \left(\frac{\partial T}{\partial n} \right) \int_0^{2\pi} \left(\frac{1}{r} \right) d\theta - \frac{\partial T}{\partial a_j} \int_0^{2\pi} \frac{\partial}{\partial n} \left(\frac{1}{r} \right) d\theta \right] \rho \, d\rho(\vec{\xi}) \\
&+ \sum_{k=1}^{N} \int_{l_k} \left[\frac{\partial}{\partial a_j} \left(\frac{\partial T}{\partial n} \right) \int_0^{2\pi} \left(\frac{1}{r} \right) d\theta - \frac{\partial T}{\partial a_j} \int_0^{2\pi} \frac{\partial}{\partial n} \left(\frac{1}{r} \right) d\theta \right] a_k \, dl(\vec{\xi}) \\
&+ \int_{l_j} \left[\frac{\partial T}{\partial n} \frac{\partial}{\partial a_j} \left\{ a_j \int_0^{2\pi} \left(\frac{1}{r} \right) d\theta \right\} - T \frac{\partial}{\partial a_j} \left\{ a_j \int_0^{2\pi} \frac{\partial}{\partial n} \left(\frac{1}{r} \right) d\theta \right\} \right] dl(\vec{\xi}) \\
&+ \sum_{k=1}^{N} \int_{\rho_k} \left[\frac{\partial}{\partial a_j} \left(\frac{\partial T}{\partial n} \right) \int_0^{2\pi} \left(\frac{1}{r} \right) d\theta - \frac{\partial T}{\partial a_j} \int_0^{2\pi} \frac{\partial}{\partial n} \left(\frac{1}{r} \right) d\theta \right] \rho \, d\rho(\vec{\xi}) \\
&+ a_j \left[\frac{\partial T}{\partial n} \int_0^{2\pi} \left[\frac{1}{r} \right]_{\rho = a_i} d\theta - T \int_0^{2\pi} \left[\frac{\partial}{\partial n} \left(\frac{1}{r} \right) \right]_{\rho = a_i} d\theta \right] \\
&+ 4\pi \frac{\partial T_\infty}{\partial a_j}. \qquad (38)
\end{aligned}
$$

In Eqs. 36–38, the fifth and seventh integral terms appear because r is the function of the design variable a_j and the domain of integral (the surface of jth cooling channel) also changes. Next, the location of jth cooling channel is considered as a design variable (a vector): (i) for the part surface, the implicit gradient of Eqs. 7 and 8 with respect to the location of the jth cooling channel gives us a pair of integral equations for the sensitivity analysis as follows in a gradient form,

$$
\begin{aligned}
&\alpha^- \nabla T^+(\vec{x}) + \alpha^+ \nabla T^-(\vec{x}) \\
&= \int_{\Gamma} \left[\frac{1}{r} \left\{ \nabla \left(\frac{\partial T^+}{\partial n^+} \right) + \nabla \left(\frac{\partial T^-}{\partial n^-} \right) \right\} \right. \\
&\quad \left. - \frac{\partial}{\partial n^+} \left(\frac{1}{r} \right) \left(\nabla T^+ - \nabla T^- \right) \right] dS(\vec{\xi}) \\
&\quad + \int_{\lambda} \left\{ \nabla \left(\frac{\partial T}{\partial n} \right) \int_0^{2\pi} \left(\frac{1}{r} \right) d\theta - \nabla T \int_0^{2\pi} \frac{\partial}{\partial n} \left(\frac{1}{r} \right) d\theta \right\} a \, dl(\vec{\xi}) \\
&\quad + \int_{\varrho} \left\{ \nabla \left(\frac{\partial T}{\partial n} \right) \int_0^{2\pi} \left(\frac{1}{r} \right) d\theta - \nabla T \int_0^{2\pi} \frac{\partial}{\partial n} \left(\frac{1}{r} \right) d\theta \right\} \rho \, d\rho(\vec{\xi}) \\
&\quad + \sum_{k=1}^{N} \int_{l_k} \left\{ \nabla \left(\frac{\partial T}{\partial n} \right) \int_0^{2\pi} \left(\frac{1}{r} \right) d\theta - \nabla T \int_0^{2\pi} \frac{\partial}{\partial n} \left(\frac{1}{r} \right) d\theta \right\} a_k \, dl(\vec{\xi}) \\
&\quad + \int_{l_j} \left[\frac{\partial T}{\partial n} \int_0^{2\pi} \nabla_\xi \left(\frac{1}{r} \right) d\theta - T \int_0^{2\pi} \nabla_\xi \left\{ \frac{\partial}{\partial n} \left(\frac{1}{r} \right) \right\} d\theta \right] a_j \, dl(\vec{\xi}) \\
&\quad + \sum_{k=1}^{N} \int_{\rho_k} \left\{ \nabla \left(\frac{\partial T}{\partial n} \right) \int_0^{2\pi} \left(\frac{1}{r} \right) d\theta - \nabla T \int_0^{2\pi} \frac{\partial}{\partial n} \left(\frac{1}{r} \right) d\theta \right\} \rho \, d\rho(\vec{\xi})
\end{aligned}
$$

$$+\int_{\rho_j}\left[\frac{\partial T}{\partial n}\int_0^{2\pi}\nabla_\xi\left(\frac{1}{r}\right)d\theta - T\int_0^{2\pi}\nabla_\xi\left\{\frac{\partial}{\partial n}\left(\frac{1}{r}\right)\right\}d\theta\right]\rho\, d\rho(\vec{\xi})$$
$$+4\pi\nabla T_\infty, \tag{39}$$

$$\alpha^-\nabla\left(\frac{\partial T^+(\vec{x})}{\partial\nu}\right)-\alpha^+\nabla\left(\frac{\partial T^-(\vec{x})}{\partial\nu}\right)$$
$$=\int_\Gamma\left[\frac{\partial}{\partial\nu}\left(\frac{1}{r}\right)\left\{\nabla\left(\frac{\partial T^+}{\partial n^+}\right)+\nabla\left(\frac{\partial T^-}{\partial n^-}\right)\right\}\right.$$
$$\left.-\frac{\partial}{\partial\nu}\left(\frac{\partial}{\partial n^+}\left(\frac{1}{r}\right)\right)(\nabla T^+-\nabla T^-)\right]dS(\vec{\xi})$$
$$+\int_\lambda\left[\nabla\left(\frac{\partial T}{\partial n}\right)\int_0^{2\pi}\frac{\partial}{\partial\nu}\left(\frac{1}{r}\right)d\theta-\nabla T\int_0^{2\pi}\frac{\partial}{\partial\nu}\left(\frac{\partial}{\partial n}\left(\frac{1}{r}\right)\right)d\theta\right]a\, dl(\vec{\xi})$$
$$+\int_\varrho\left[\nabla\left(\frac{\partial T}{\partial n}\right)\int_0^{2\pi}\frac{\partial}{\partial\nu}\left(\frac{1}{r}\right)d\theta-\nabla T\int_0^{2\pi}\frac{\partial}{\partial\nu}\left(\frac{\partial}{\partial n}\left(\frac{1}{r}\right)\right)d\theta\right]\rho\, d\rho(\vec{\xi})$$
$$+\sum_{i=1}^N\int_{l_k}\left[\nabla\left(\frac{\partial T}{\partial n}\right)\int_0^{2\pi}\frac{\partial}{\partial\nu}\left(\frac{1}{r}\right)d\theta\right.$$
$$\left.-\nabla T\int_0^{2\pi}\frac{\partial}{\partial\nu}\left(\frac{\partial}{\partial n}\left(\frac{1}{r}\right)\right)d\theta\right]a_k\, dl(\vec{\xi})$$
$$+\int_{l_j}\left[\frac{\partial T}{\partial n}\int_0^{2\pi}\nabla_\xi\left\{\frac{\partial}{\partial\nu}\left(\frac{1}{r}\right)\right\}d\theta\right.$$
$$\left.-T\int_0^{2\pi}\nabla_\xi\left\{\frac{\partial}{\partial\nu}\left(\frac{\partial}{\partial n}\left(\frac{1}{r}\right)\right)\right\}d\theta\right]a_j\, dl(\vec{\xi})$$
$$+\sum_{i=1}^N\int_{\rho_k}\left[\nabla\left(\frac{\partial T}{\partial n}\right)\int_0^{2\pi}\frac{\partial}{\partial\nu}\left(\frac{1}{r}\right)d\theta\right.$$
$$\left.-\nabla T\int_0^{2\pi}\frac{\partial}{\partial\nu}\left(\frac{\partial}{\partial n}\left(\frac{1}{r}\right)\right)d\theta\right]\rho\, d\rho(\vec{\xi})$$
$$+\int_{\rho_j}\left[\frac{\partial T}{\partial n}\int_0^{2\pi}\nabla_\xi\left\{\frac{\partial}{\partial\nu}\left(\frac{1}{r}\right)\right\}d\theta\right.$$
$$\left.-T\int_0^{2\pi}\nabla_\xi\left\{\frac{\partial}{\partial\nu}\left(\frac{\partial}{\partial n}\left(\frac{1}{r}\right)\right)\right\}d\theta\right]\rho\, d\rho(\vec{\xi}); \tag{40}$$

(ii) for the ith cooling channel surface, the implicit gradient of Eq. 9 with respect to the location of the jth cooling channel gives us the following integral equation for the sensitivity analysis, depending on the value of i,

when $i \neq j$,

$$0=\int_\Gamma\left[\frac{1}{r}\left\{\nabla\left(\frac{\partial T^+}{\partial n^+}\right)+\nabla\left(\frac{\partial T^-}{\partial n^-}\right)\right\}\right.$$
$$\left.-\frac{\partial}{\partial n^+}\left(\frac{1}{r}\right)(\nabla T^+-\nabla T^-)\right]dS(\vec{\xi})$$

$$
\begin{aligned}
&+\int_{\lambda}\left\{\nabla\left(\frac{\partial T}{\partial n}\right)\int_0^{2\pi}\left(\frac{1}{r}\right)d\theta-\nabla T\int_0^{2\pi}\frac{\partial}{\partial n}\left(\frac{1}{r}\right)d\theta\right\}a\,dl(\vec{\xi})\\
&+\int_{\varrho}\left\{\nabla\left(\frac{\partial T}{\partial n}\right)\int_0^{2\pi}\left(\frac{1}{r}\right)d\theta-\nabla T\int_0^{2\pi}\frac{\partial}{\partial n}\left(\frac{1}{r}\right)d\theta\right\}\rho\,d\rho(\vec{\xi})\\
&+\sum_{k=1}^{N}\int_{l_k}\left\{\nabla\left(\frac{\partial T}{\partial n}\right)\int_0^{2\pi}\left(\frac{1}{r}\right)d\theta-\nabla T\int_0^{2\pi}\frac{\partial}{\partial n}\left(\frac{1}{r}\right)d\theta\right\}a_k\,dl(\vec{\xi})\\
&+\int_{l_j}\left[\frac{\partial T}{\partial n}\int_0^{2\pi}\nabla_{\xi}\left(\frac{1}{r}\right)d\theta-T\int_0^{2\pi}\nabla_{\xi}\left\{\frac{\partial}{\partial n}\left(\frac{1}{r}\right)\right\}d\theta\right]a_j\,dl(\vec{\xi})\\
&+\sum_{k=1}^{N}\int_{\rho_k}\left\{\nabla\left(\frac{\partial T}{\partial n}\right)\int_0^{2\pi}\left(\frac{1}{r}\right)d\theta-\nabla T\int_0^{2\pi}\frac{\partial}{\partial n}\left(\frac{1}{r}\right)d\theta\right\}\rho\,d\rho(\vec{\xi})\\
&+\int_{\rho_j}\left[\frac{\partial T}{\partial n}\int_0^{2\pi}\nabla_{\xi}\left(\frac{1}{r}\right)d\theta-T\int_0^{2\pi}\nabla_{\xi}\left\{\frac{\partial}{\partial n}\left(\frac{1}{r}\right)\right\}d\theta\right]\rho\,d\rho(\vec{\xi})\\
&+4\pi\nabla T_{\infty},
\end{aligned}
\tag{41}
$$

when $i=j$,

$$
\begin{aligned}
0=&\int_{\Gamma}\left[\frac{1}{r}\left\{\nabla\left(\frac{\partial T^{+}}{\partial n^{+}}\right)+\nabla\left(\frac{\partial T^{-}}{\partial n^{-}}\right)\right\}\right.\\
&\left.-\frac{\partial}{\partial n^{+}}\left(\frac{1}{r}\right)\left(\nabla T^{+}-\nabla T^{-}\right)\right]dS(\vec{\xi})\\
&+\int_{\Gamma}\left[\nabla_x\left(\frac{1}{r}\right)\left\{\left(\frac{\partial T^{+}}{\partial n}\right)+\left(\frac{\partial T^{-}}{\partial n}\right)\right\}\right.\\
&\left.-\nabla_x\left\{\frac{\partial}{\partial n}\left(\frac{1}{r}\right)\right\}\left(T^{+}-T^{-}\right)\right]dS(\vec{\xi})\\
&+\int_{\lambda}\left\{\nabla\left(\frac{\partial T}{\partial n}\right)\int_0^{2\pi}\left(\frac{1}{r}\right)d\theta-\nabla T\int_0^{2\pi}\frac{\partial}{\partial n}\left(\frac{1}{r}\right)d\theta\right\}a\,dl(\vec{\xi})\\
&+\int_{\lambda}\left[\frac{\partial T}{\partial n}\int_0^{2\pi}\nabla_x\left(\frac{1}{r}\right)d\theta-T\int_0^{2\pi}\nabla_x\left\{\frac{\partial}{\partial n}\left(\frac{1}{r}\right)\right\}d\theta\right]a\,dl(\vec{\xi})\\
&+\int_{\varrho}\left\{\nabla\left(\frac{\partial T}{\partial n}\right)\int_0^{2\pi}\left(\frac{1}{r}\right)d\theta-\nabla T\int_0^{2\pi}\frac{\partial}{\partial n}\left(\frac{1}{r}\right)d\theta\right\}\rho\,d\rho(\vec{\xi})\\
&+\int_{\varrho}\left[\frac{\partial T}{\partial n}\int_0^{2\pi}\nabla_x\left(\frac{1}{r}\right)d\theta-T\int_0^{2\pi}\nabla_x\left\{\frac{\partial}{\partial n}\left(\frac{1}{r}\right)\right\}d\theta\right]\rho\,d\rho(\vec{\xi})\\
&+\sum_{k=1}^{N}\int_{l_k}\left\{\nabla\left(\frac{\partial T}{\partial n}\right)\int_0^{2\pi}\left(\frac{1}{r}\right)d\theta-\nabla T\int_0^{2\pi}\frac{\partial}{\partial n}\left(\frac{1}{r}\right)d\theta\right\}a_k\,dl(\vec{\xi})\\
&+\sum_{k=1,k\neq i}^{N}\int_{l_k}\left[\frac{\partial T}{\partial n}\int_0^{2\pi}\nabla_x\left(\frac{1}{r}\right)d\theta-T\int_0^{2\pi}\nabla_x\left\{\frac{\partial}{\partial n}\left(\frac{1}{r}\right)\right\}d\theta\right]a_k\,dl(\vec{\xi})
\end{aligned}
$$

$$+\sum_{k=1}^{N}\int_{\rho_k}\left\{\nabla\left(\frac{\partial T}{\partial n}\right)\int_0^{2\pi}\left(\frac{1}{r}\right)d\theta-\nabla T\int_0^{2\pi}\frac{\partial}{\partial n}\left(\frac{1}{r}\right)d\theta\right\}\rho\,d\rho(\vec{\xi})$$

$$+\sum_{k=1,k\neq i}^{N}\int_{\rho_k}\left[\frac{\partial T}{\partial n}\int_0^{2\pi}\nabla_x\left(\frac{1}{r}\right)d\theta-T\int_0^{2\pi}\nabla_x\left\{\frac{\partial}{\partial n}\left(\frac{1}{r}\right)\right\}d\theta\right]\rho\,d\rho(\vec{\xi})$$

$$+4\pi\nabla T_\infty. \tag{42}$$

In Eqs. 39–42, ∇_ξ and ∇_x specifically denote gradient with respect to $\vec{\xi}$ and $\vec{x}$, respectively, depending on which is the design variable, where $r=|\vec{\xi}-\vec{x}|$. Since the location is a vector, the coordinate transformation rule should be introduced for the sensitivity analysis with respect to the location of the cooling channel. The coordinate transformation rule between the global coordinate system and the local coordinate system is illustrated in Fig. 4 and is briefly described in Appendix A.

3.3.2. *Sensitivities of boundary conditions*

In order to solve the above system of the boundary integral equations for the design sensitivity analysis, it is important to introduce proper sensitivities of boundary conditions on the part surface (including sprue and runner surface), the cooling channel surface, and the mold exterior surface into the sensitivity analysis equations.

Part surface

One has to determine the sensitivity of the cooling time and thereafter he can evaluate the sensitivity of the cycle-averaged heat flux on the part surface. We can obtain the sensitivity of the cooling time of Type I from the element which determines the cooling time. Here we find $\partial t_c/\partial X$ and $\partial z_c/\partial X$ by solving the following equation:

$$\frac{\partial}{\partial X}\left(\frac{\partial T(z,t)}{\partial z}\right)=0,\quad \frac{\partial T(z,t)}{\partial X}=0 \qquad \text{at } t=t_c \text{ and } z=z_c, \tag{43}$$

which is an implicit derivative of Eq. 12. The obtained value $\partial t_c/\partial X$, which is the sensitivity of the cooling time with respect to each design variable, is a very important information for the optimization process because this value is directly related to the overall productivity of the injection molding process. Then, one can also obtain the sensitivity of the cycle-averaged heat flux on the minus plane neglecting the effects during the filling and opening times as follows:

$$\frac{\partial}{\partial X}\left(\frac{\partial T^-}{\partial n^-}\right)=\frac{k_p}{k_m}\left[\frac{1}{b}\left(\frac{\partial T^+}{\partial X}-\frac{\partial T^-}{\partial X}\right)\right.$$

$$\left.+\frac{1}{t_c^2}\frac{\partial t_c}{\partial X}\sum_{n=1}^{\infty}\frac{b}{n\pi\alpha}a_n\left\{\exp\left(-\frac{n^2\pi^2\alpha t_c}{b^2}\right)-1\right\}\right.$$

$$- \frac{1}{t_c} \sum_{n=1}^{\infty} \frac{b}{n\pi\alpha} \frac{\partial a_n}{\partial X} \left\{ \exp\left(-\frac{n^2\pi^2\alpha t_c}{b^2} \right) - 1 \right\}$$
$$+ \frac{1}{t_c} \frac{\partial t_c}{\partial X} \sum_{n=1}^{\infty} \frac{n\pi}{b} a_n \exp\left(-\frac{n^2\pi^2\alpha t_c}{b^2} \right) \Bigg], \tag{44}$$

where $$\frac{\partial a_n}{\partial X} = \frac{2}{n\pi} \left\{ (-1)^n \frac{\partial T^+}{\partial X} - \frac{\partial T^-}{\partial X} \right\},$$

which is in fact an implicit derivative of Eq. 14. One can also determine the sensitivity of the cycle-averaged heat flux on the plus plane by interchanging + and − in Eq. 44. For sprue and runner, one can use the same treatment similar to that in the part analysis except for the use of the Bessel series in the cylindrical coordinate system.

Cooling channel surface

In order to consider the sensitivity of the mixed type of boundary condition (i.e. Eq. 5) on the cooling channel surface, one should also evaluate properly the sensitivities of both the heat transfer coefficient ($\partial h_c/\partial X$) and the coolant bulk temperature ($\partial T_b/\partial X$) with respect to all design variables. It may be noted that the heat transfer coefficient, h_c, depends on the coolant flow rate and the radius of the cooling channel, as indicated in Eq. 16. Thus, before evaluating $\partial h_c/\partial X$ (when the design variable, X, is one of them) one needs to determine the sensitivity of the volumetric flow rate in each channel element for branched cooling channels with respect to a design variable (i.e. either the inlet coolant volumetric flow rate of each cooling channel or the radius of each cooling channel). Governing equations of these values can be obtained from implicit derivatives of Eq. 15 for each element with respect to these design variables. These equations can be solved by the 1-dimensional linear finite element method, as was done for the flow rate in Sec. 2. After solving the sensitivity of the volumetric flow rate in each element, one can evaluate the sensitivity of the heat transfer coefficient of each cooling channel element using an implicit derivative of the Dittus–Boetler correlation, Eq. 16, with respect to these design variables. Next the sensitivities of coolant bulk temperature in each element can be obtained from implicit derivatives of Eqs. 17–19 with respect to all the design variables. Particularly, for the case of branched cooling channels, the average sensitivity of the outlet temperature at the junction node was determined with the volumetric flow rate as a weighting factor same manner as that described for the thermal analysis case in Sec. 2. For computational convenience, the modified sensitivity of coolant bulk temperature, $(\partial T_b/\partial X)_m$, is defined as follows:

$$\left(\frac{\partial T_b}{\partial X} \right)_m = \begin{cases} \dfrac{\partial T_b}{\partial X} - \dfrac{1}{h_c} \dfrac{\partial h_c}{\partial X} (T - T_b), & \text{for } X = a_i \text{ or } \bar{Q}_i \\ \dfrac{\partial T_b}{\partial X}, & \text{otherwise} \end{cases} \tag{45}$$

Now, having the sensitivity of the coolant bulk temperature as defined in Eq. 46, one can compute the sensitivity of the boundary condition as follows:

$$-k_m \frac{\partial}{\partial X}\left(\frac{\partial T}{\partial n}\right) = h_c \left[\frac{\partial T}{\partial X} - \left(\frac{\partial T_b}{\partial X}\right)_m\right]. \tag{46}$$

The reason for introducing the specially modified sensitivity of the coolant bulk temperature at the ith cooling channel when the design variable is either the radius or the coolant volumetric flow rate of the ith cooling channel is in fact to arrive at a simpler version of Eq. 46, which otherwise becomes too complicated to represent the sensitivity of the boundary condition on the cooling channel surface.

Mold exterior surface

On the mold exterior surface, one can evaluate the sensitivity of boundary condition with respect to each design variable (that is, $\partial T_\infty / \partial X$) by using the following iterative algorithm. First of all, for computational convenience the modified sensitivity of the heat flux on the cooling channel surface is defined as follows:

$$k_m \left\{\frac{\partial}{\partial X}\left(-\frac{\partial T}{\partial n}\right)\right\}_m = \begin{cases} k_m \left[\dfrac{\partial}{\partial a_i}\left(-\dfrac{\partial T}{\partial n}\right) + \dfrac{1}{a_i}\left(-\dfrac{\partial T}{\partial n}\right)\right] & \text{for } X = a_i \\ k_m \dfrac{\partial}{\partial X}\left(-\dfrac{\partial T}{\partial n}\right) & \text{otherwise} \end{cases}, \tag{47}$$

which is in fact needed to take care of the special effect (i.e. the change in the integral domain) of the change in the radius of the ith cooling channel as a design variable. Then some definitions are introduced to derive the iterative algorithm:

$$DHG = k_m \int_\Gamma \left[\frac{\partial}{\partial X}\left(\frac{\partial T^+}{\partial n^+}\right) + \frac{\partial}{\partial X}\left(\frac{\partial T^-}{\partial n^-}\right)\right] dS + k_m \int_\lambda \int_0^{2\pi} \frac{\partial}{\partial X}\left(\frac{\partial T}{\partial n}\right) a \, d\theta \, dl, \tag{48}$$

$$DHL = k_m \sum_{k=1}^{N} \int_{l_K} \int_0^{2\pi} \left\{\frac{\partial}{\partial X}\left(-\frac{\partial T}{\partial n}\right)\right\}_m a_k \, d\theta \, dl, \tag{49}$$

$$\frac{\partial HBE}{\partial X} = \frac{100 \cdot (DHG - DHL) - HBE \cdot (DHG + DHL)}{HG + HL}, \tag{50}$$

which are implicit derivative versions of Eqs. 22–24. In the above equations, DHG, DHL, and $DHBE$ denote the sensitivity of the total heat gain, that of the total heat loss, and that of the heat-balanced error, respectively. With those definitions, $\partial T_\infty / \partial X$ can be evaluated by the following iterative algorithm:

$$\left(\frac{\partial T_\infty}{\partial X}\right)^{\text{new}} = \left(\frac{\partial T_\infty}{\partial X}\right)^{\text{old}} + T_\alpha \cdot \frac{\partial HBE}{\partial X} \tag{51}$$

until $\partial HBE / \partial X$ approaches zero to satisfy the energy balance for the steady state heat transfer. Equation 51 is an implicit derivative of Eq. 25.

3.3.3. *Solution procedure*

Once the integrals on each element are calculated, the discretized boundary element formulae for the sensitivity analysis can be manipulated to the following form:

$$[H_{ij,X}]\ \{T_j\} + [H_{ij}]\ \{T_{j,X}\} = [G_{ij,X}] \left\{ \left.\frac{\partial T}{\partial n}\right|_j \right\} + [G_{ij}] \left\{ \left.\frac{\partial T}{\partial n}\right|_{j,X} \right\}, \tag{52}$$

where '$,_X$' represents a partial derivative with respect to a design variable X. It may be noted that in Eq. 52, $\{T_j\}$, $\{(\partial T/\partial n)_j\}$, $[H_{ij}]$ and $[G_{ij}]$ are already obtained in the thermal analysis. The integrals for the coefficients of the matrices of $[H_{ij,X}]$ and $[G_{ij,X}]$ can be evaluated in a manner similar to that in the thermal analysis case. See Appendix C and the paper by Park and Kwon[11] for details of evaluating these integrals. Next, the sensitivities of boundary conditions as described in the previous section can be introduced into Eq. 52 to obtain a system of linear algebraic equations as follows:

$$\begin{aligned} [A_{ij}]\ \{T_{j,X}\} &= \{f_{i,X}\} - [A_{ij,X}]\ \{T_j\} \\ &\equiv \{f'_{i,X}\}, \end{aligned} \tag{53}$$

where $T_{j,X}$ is taken to be an unknown on each element. It should be noted that $[A_{ij}]$ in Eq. 53 is exactly the same as that in Eq. 31 for the thermal analysis, and that only the forcing terms $\{f'_{i,X}\}$ are to be evaluated for each design variable. Therefore, once $\{f'_{i,X}\}$ are evaluated for all design variables, the sensitivity analysis results of interest, i.e. $\{T_{j,X}\}$, can be determined simultaneously with multiple forcing matrices for all design variables, thus significantly saving the computational time. This fact is the most fascinating advantage of the present sensitivity analysis approach over the numerical derivative approach, i.e. the finite difference method. It may be noted that the LU-factorization of the matrix $[A_{ij}]$, formed during the thermal analysis, can be reused many times in this sensitivity analysis. On the other hand, an iterative method, which takes up less computer memory than a direct solver using the LU-factorization, can also efficiently evaluate $\{T_{j,X}\}$ for all design variables with multiple forcing matrices (but, of course, at the cost of more CPU time). Thus it would be very useful when one needs a large number of boundary elements with which the direct solver cannot handle due to limits in the computer memory.

3.4. *Results and discussion*

The example introduced for the thermal analysis in Sec. 2 was also applied to the sensitivity analysis.

As far as the sensitivity analysis for this example is concerned, the inlet coolant bulk temperature, the inlet coolant volumetric flow rate, the radius, the y-coordinate and the z-coordinate of each cooling channel are considered as design variables of

importance because the effect of the x-coordinate is negligible (see Fig. 7). Therefore, ten design variables exist in reality (five design variables for each cooling channel) in this example even if the present direct differentiation approach deals with all the design variables including the x-coordinate. In solving Eq. 53 via an iterative method, the maximum norm convergence criterion is used as follows:

$$\max_i \left| \frac{T_{i,X}^{\text{new}} - T_{i,X}^{\text{old}}}{T_{i,X}^{\text{old}}} \right| \leq 10^{-m}, \tag{54}$$

where i changes from 1 to the number of unknowns. It may just be noted that $m \geq 3$ is required for an accurate sensitivity analysis according to our computational experience (thus, possible in a single precision calculation). It may be further mentioned that, for an accurate thermal and sensitivity analysis, four significant digits (i.e. $n = 4$ in Eq. 32) were used for the thermal analysis, of which the results are to be used in Eq. 52, and that the sensitivity analysis results were deteriorated if $n < 4$ even with $m \geq 3$. The computational efficiency and accuracy of the present approach will be discussed below in comparison with the finite difference method and finally the characteristics of each design variable will be highlighted.

3.5. *Efficiency and accuracy*

The elapsed CPU time for this sensitivity analysis using the direct differentiation approach with $m = 3$ is 2:25:03 (hr:min:s) using the same computer as in Sec. 2, thus the total elapsed CPU time is 2:47:22 (one analysis and one sensitivity analysis). In this kind of geometries, the exact solution does not exist. Therefore, in order to check the accuracy of these sensitivity analysis results, these results were directly compared with those from another numerical method, namely a forward finite difference method in two aspects: global convergency and local convergency.

Global convergency

To look into the accuracy of the forward finite difference method, the effects of the number of significant digits used in the thermal analysis (which is exactly n in Eq. 32) and the magnitude of difference rate in a design variable ($\Delta X/X$) on the sensitivity analysis results have been investigated. For instance, Fig. 16 illustrates those effects on the sensitivity analysis (global convergency) with respect to a design variable, the y-coordinate of the cooling channel #1, Y_1. In Fig. 16, the ordinates of both plots (a) and (b) represent the root-mean-square value of $\partial(T^+ - T^-)/\partial Y_1$, i.e.

$$\sqrt{\frac{\int_\Gamma \left(\frac{\partial T^+}{\partial Y_1} - \frac{\partial T^-}{\partial Y_1} \right)^2 dA}{\int_\Gamma dA}},$$

with the abscissa of plot (a) and (b) representing the number of significant digits and the magnitude of difference rate, respectively. The magnitude of the difference

rate is set to be 0.0001% (i.e. $\Delta X/X = 0.0000035\,\text{cm}/3.5\,\text{cm}$ in this particular case) to obtain results as shown in Fig. 16a, while the number of significant digits is set to 10 for obtaining the results in Fig. 16b. According to the global convergency of the finite difference method to the direct differentiation approach as shown in Fig. 16, it was found that at least a 10-digit significant figure and a 0.0001% difference rate are required for the accurate computations for the sensitivity analysis via the forward finite difference method. Therefore, a double precision calculation should be used in order to satisfy the above requirements when the forward finite difference method was used for the sensitivity analysis. The sensitivity analysis by the forward finite difference method requires the total elapsed CPU time of 29:09:11 (the elapsed CPU time is 2:39:01 for each run of the analysis program and a total of 11 runs of the analysis program are needed for 10 design variables), thus requiring about ten times as much as that for sensitivity analysis by the direct differentiation approach in this particular example. It should be noted that the ratio of the total elapsed CPU time by the forward finite difference method to that by the direct differentiation approach approximately linearly increases with the number of design variables (i.e. cooling channels) due to the advantages of the present approach as mentioned in the previous section.

Local convergency

Results between the present approach and the finite difference method have been compared and an excellent agreement has been found between them for all the design variables. The sensitivities of both part surface temperature and heat flux with respect to the y-coordinate of the cooling channel #1 as obtained from the direct differentiation approach and the forward finite difference method are shown in Figs. 17 and 18, respectively (the 10-digit significant figure and the 0.0001% difference rate in design variables are adopted in the finite difference calculations as stated before). These figures reveal that the computed sensitivity results are in an excellent agreement with each other over the whole surface (local convergency). The maximum relative error found in this example was about 1.27% (the relative error on each element is defined as a ratio of the absolute value of the difference between the sensitivity result by the forward finite difference method and that by the direct differentiation approach to that by the direct differentiation approach). The maximum error may be reduced by increasing the number of significant digits and then reducing the magnitude of difference rate in the forward finite difference method. It may be emphasized again that, according to our computation experiences, the sensitivity analysis via the finite difference method becomes absolutely meaningless if the significant figure for the thermal analysis is less than or equal to 6. The global and local convergencies as explained above (Figs. 16–18) indirectly, but quite concretely prove the accuracy of both approaches. Thus one can conclude that the direct differentiation approach is more accurate and faster than the forward finite difference method. Furthermore these accurate sensitivity analysis results obtained

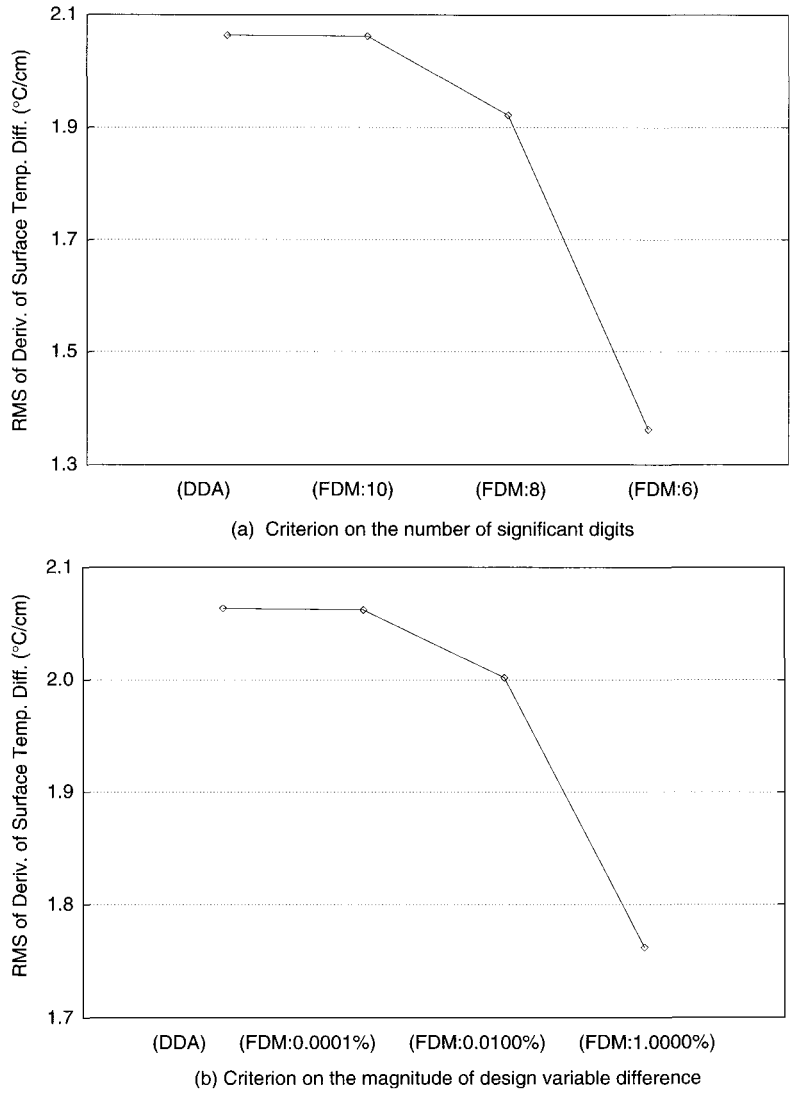
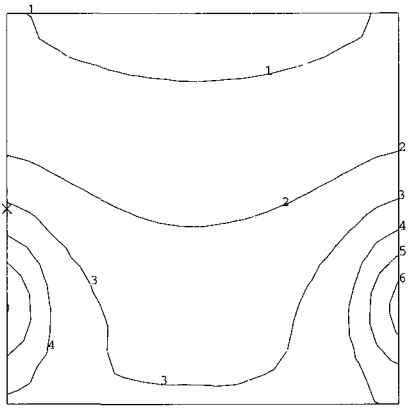


Fig. 16. Some criteria for finite difference method.

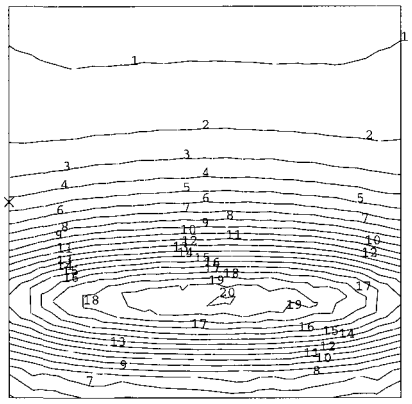
from the two approaches also confirm the accuracy of the thermal analysis formulation as mentioned in Sec. 2. In this chapter, a representative comparison for only one design variable is presented due to the size limit of the chapter.

3.6. *Characteristics of design variables*

From the sensitivity analysis result with respect to each design variable, one can improve our understanding of the characteristics of each design variable.

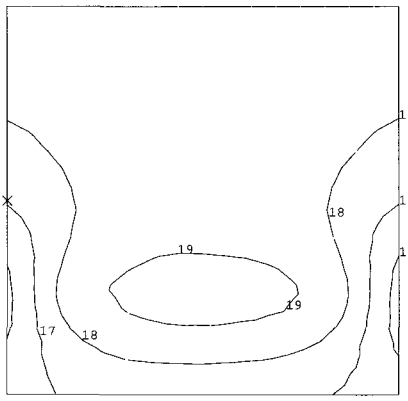


min:−0.3102 max: 1.6300
plus plane

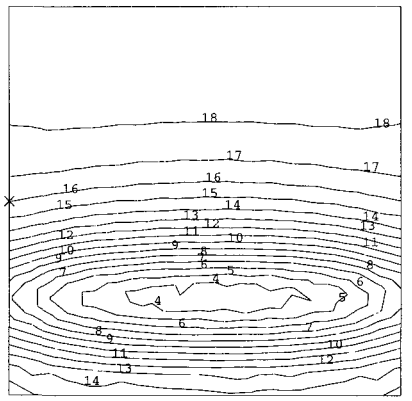


min:−0.0441 max: 5.7700
minus plane

(a) sensitivity of temperature (unit: °C/cm)



min:-0.810×10^6 max:-0.240×10^6
plus plane

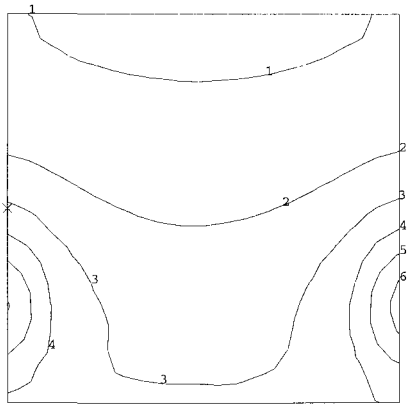


min:-2.680×10^6 max:-0.354×10^6
minus plane

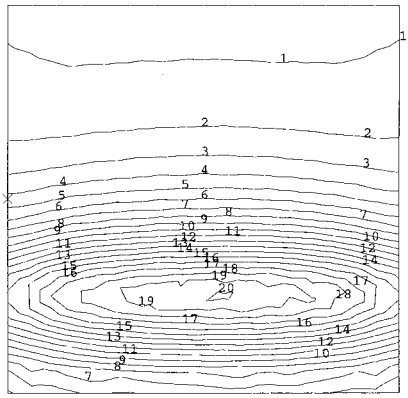
(b) sensitivity of heat flux (unit: $g/s^3 \cdot cm$)

Fig. 17. Sensitivity results of the temperature and the heat flux on part surface with respect to the y-coordinate of the cooling channel #1 by the DDA: (a) sensitivity of temperature [level value = (level number − 1) × 3/10] and (b) sensitivity of heat flux [level value = (level number − 21) × 1.5×10^5].

Furthermore, one can confirm that the design variables related to the mold cooling system design affect the temperature field much more significantly than those related to the process conditions. The characteristics of each design variable will be discussed using this example.

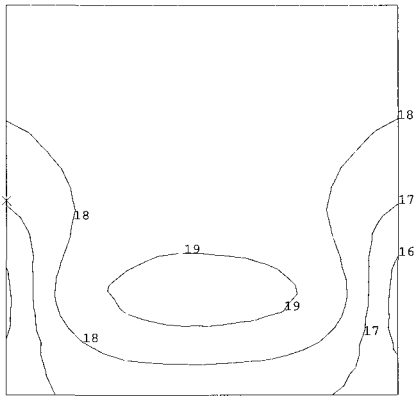


min:−0.3100 max: 1.6300
plus plane

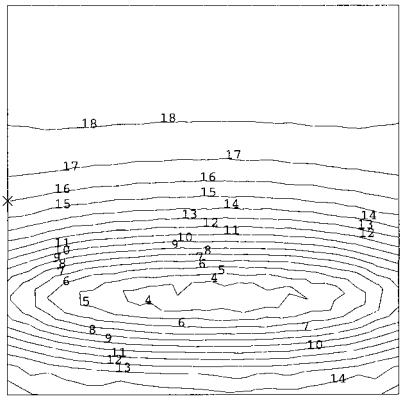


min:−0.0442 max: 5.7700
minus plane

(a) sensitivity of temperature (unit: °C/cm)



min:-0.810×10^6 max:-0.239×10^6
plus plane



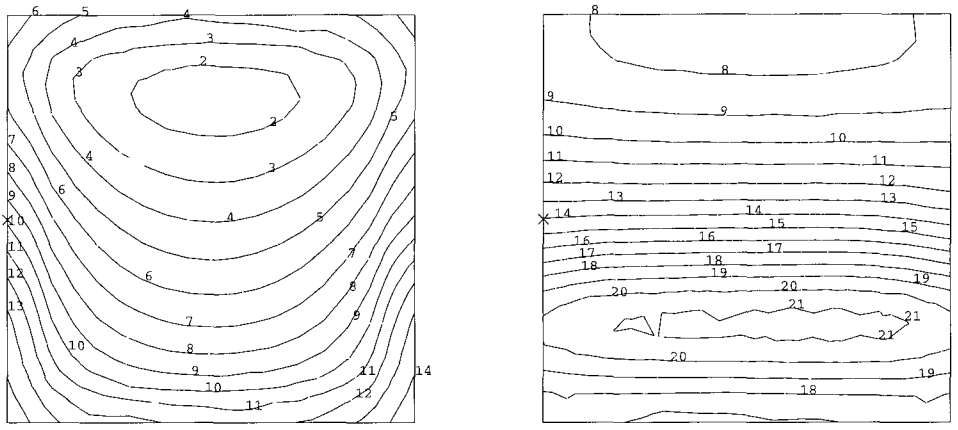
min:-2.680×10^6 max:-0.354×10^6
minus plane

(b) sensitivity of heat flux (unit: $g/s^3 \cdot cm$)

Fig. 18. Sensitivity results of the temperature and the heat flux on part surface with respect to the y-coordinate of the cooling channel #1 by the FDM: (a) sensitivity of temperature [level value = (level number − 1) × 3/10] and (b) sensitivity of heat flux [level value = (level number − 21) × 1.5×10^5].

Inlet coolant bulk temperature

Figure 19 shows the distribution of the part surface temperature sensitivity with respect to the inlet coolant bulk temperature in the cooling channel #1. The sensitivity of the cooling time is found to be 0.0234 s/°C in this case (and it is 0.0227 s/°C



min: 0.112 max: 0.379
plus plane

min: 0.228 max: 0.507
minus plane

(unit: °C/°C)

Fig. 19. Sensitivity results of the temperature on part surface with respect to the inlet coolant bulk temperature of the cooling channel #1 by the DDA [level value = 0.1 + (level number − 1) × 2/100].

in the case of the cooling channel #2). These results show that an increase in the inlet coolant bulk temperature obviously increase the cooling time and the part surface temperature (cooling reduction effect).

Inlet coolant volumetric flow rate

Figure 20 shows the distribution of the surface temperature sensitivity with respect to the inlet coolant volumetric flow rate in the cooling channel #1. The sensitivity of the cooling time is $-0.00110\,\mathrm{s/(cm^3/s)}$ in this case (and it is $-0.00166\,\mathrm{s/(cm^3/s)}$ in the case of the cooling channel #2). As expected, an increase of the inlet coolant volumetric flow rate decreases the cooling time and lowers the part surface temperature. This is mainly due to the increase of the heat transfer coefficient (cooling enhancement effect). The effect of the increase in the heat transfer coefficient can be understood through the comparison between the sensitivity of the coolant bulk temperature and the modified sensitivity of the coolant bulk temperature along the cooling channel #1 which are shown in Fig. 21 (note that the inlet and outlet positions of the cooling channel #1 correspond to $x = 30$ and $x = -10$, respectively in Fig. 21). The modified sensitivity of the coolant bulk temperature are much larger than the sensitivity of the coolant bulk temperature in absolute value, thus one can conclude that the very dominant effect is due to the increase of the heat transfer coefficient (i.e. due to the second term in the right hand side of Eq. 45).

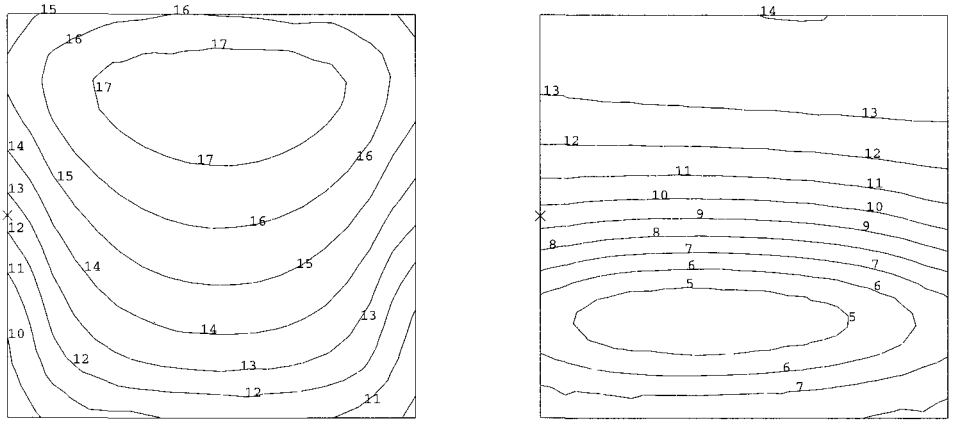


Fig. 20. Sensitivity results of the temperature on part surface with respect to the inlet volumetric flow rate of the cooling channel #1 by the DDA [level value = (level number − 21) × 3/2000].

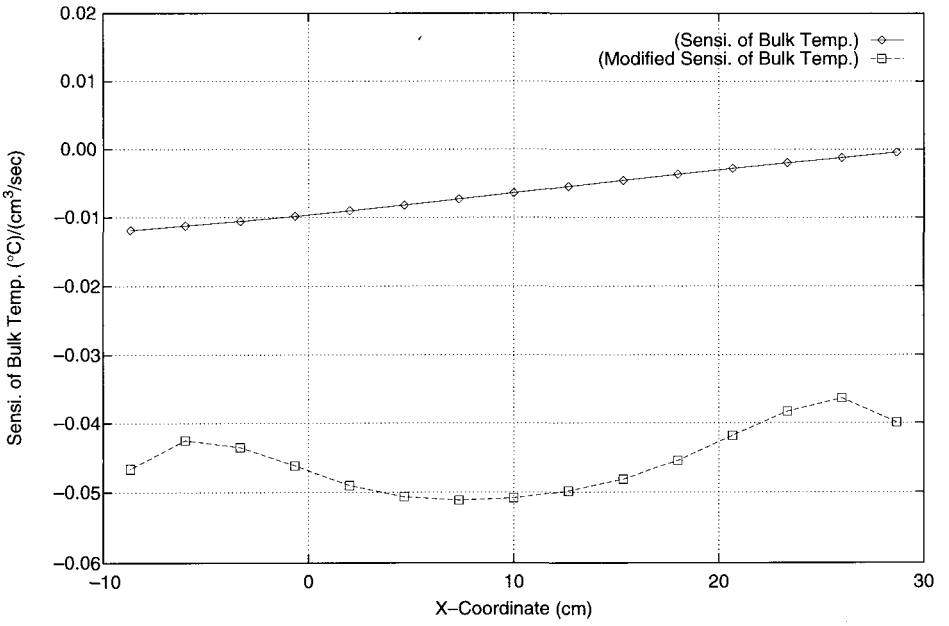


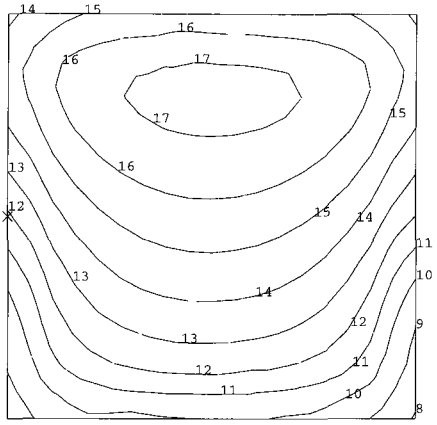
Fig. 21. Results of both the sensitivity and the modified sensitivity of the coolant bulk temperature in cooling channel #1 with respect to the inlet volumetric flow rate of cooling channel #1 by the DDA.

Radius

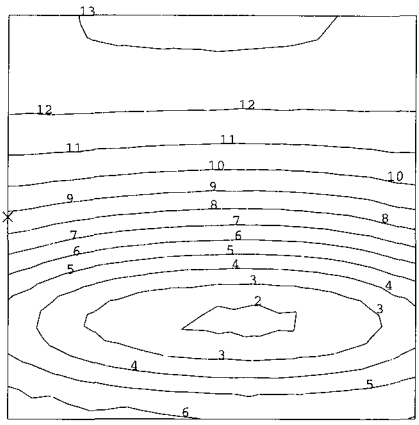
An increase of the radius of cooling channel may result in either the cooling enhancement effect (mainly due to the increase of surface area of cooling channel) or the cooling reduction effect (due to the decrease of the heat transfer coefficient), depending on which effect is more dominant. The other factors, which are the change of r ($|\vec{\xi} - \vec{x}|$ in Eqs. 36–38) and the change of the perimeter (P_i in Eq. 17) of cooling channel in computing the coolant bulk temperature, have relatively small effects. As shown in Fig. 22, the cooling enhancement effect is more dominant than the cooling reduction effect in the case of the cooling channel #1 resulting in negative sensitivity values, whereas the cooling reduction effect is more dominant in the case of the cooling channel #2, as compared to cooling channel #1. These opposing effects can be explained through the comparison between the sensitivity of coolant bulk temperature and the modified sensitivity of coolant bulk temperature as shown in Fig. 23, as well as the comparison between the sensitivity of heat flux (i.e. $-k_m \partial(\partial T/\partial n)/\partial a_i$) and the modified sensitivity of heat flux (i.e. $k_m \left[\partial(-\partial T/\partial n)/\partial a_i\right]_m$), as shown in Fig. 24. In Fig. 23, the modified sensitivity of the coolant bulk temperature is much larger than the sensitivity of the coolant bulk temperature, which implies that the second term in the right hand side of Eq. 45 is larger than the first term and that the decrease of the heat transfer coefficient is dominant in the modified sensitivity of the coolant bulk temperature. The positive value of the modified sensitivity of the coolant bulk temperature (about 40°C/cm), with $\partial T/\partial a_i$ in Eq. 46 being about 20°C/cm, results in the negative left hand side of Eq. 46 (i.e. the negative first term in right hand side of Eq. 47), which means the decrease of the heat flux on the cooling channel surface is mainly due to the decrease of the heat transfer coefficient (cooling reduction effect). Therefore, the sensitivity of the heat flux becomes negative as indicated in Fig. 24a. On the other hand, the second term on the right hand side of Eq. 47 has obviously a positive value due to the increase of the surface area of the cooling channel (cooling enhancement effect). Now, the cooling reduction or enhancement will be determined by the relative magnitude between the negative first term (cooling reduction effect) and the positive second term (cooling enhancement effect) in Eq. 47. For instance, in the case where the radius of cooling channel #1 is a design variable, the value of the modified sensitivity of heat flux is positive as shown in Fig. 24, which means that the cooling enhancement effect is more dominant than the cooling reduction effect. The opposite is true for the case of cooling channel #2.

Y-coordinate

Figure 17 shows the distribution of the surface temperature sensitivity with respect to the y-coordinate in cooling channel #1. The sensitivity of the cooling time is 0.0510 s/cm in this case. These results obviously show that an increase of the y-coordinate of cooling channel #1 (i.e. the cooling channel moving further away

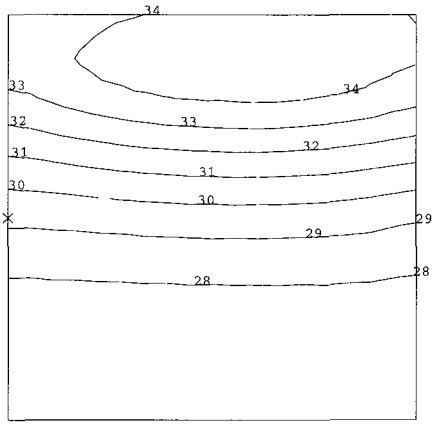


min:−1.96 max:−0.55
plus plane

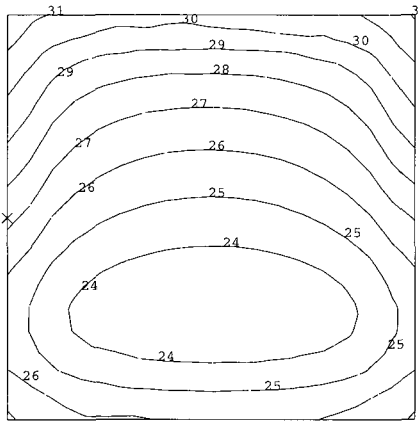


min:−2.89 max:−1.17
minus plane

(a) cooling channel #1 (unit: °C/cm)



min: 0.93 max: 2.07
plus plane



min: 0.33 max: 1.67
minus plane

(b) cooling channel #2 (unit: °C/cm)

Fig. 22. Sensitivity results of the temperature on part surface with respect to the radius of both (a) cooling channel #1 and (b) cooling channel #2 by DDA [level value = (level number − 21) × 3/20].

from the part surface) increases the cooling time and the part surface temperature (cooling reduction effect) due to the increase of the distance between the part and cooling channel #1. It may be worth mentioning that in the case of cooling channel #2, similar results to the case of cooling channel #1 were obtained.

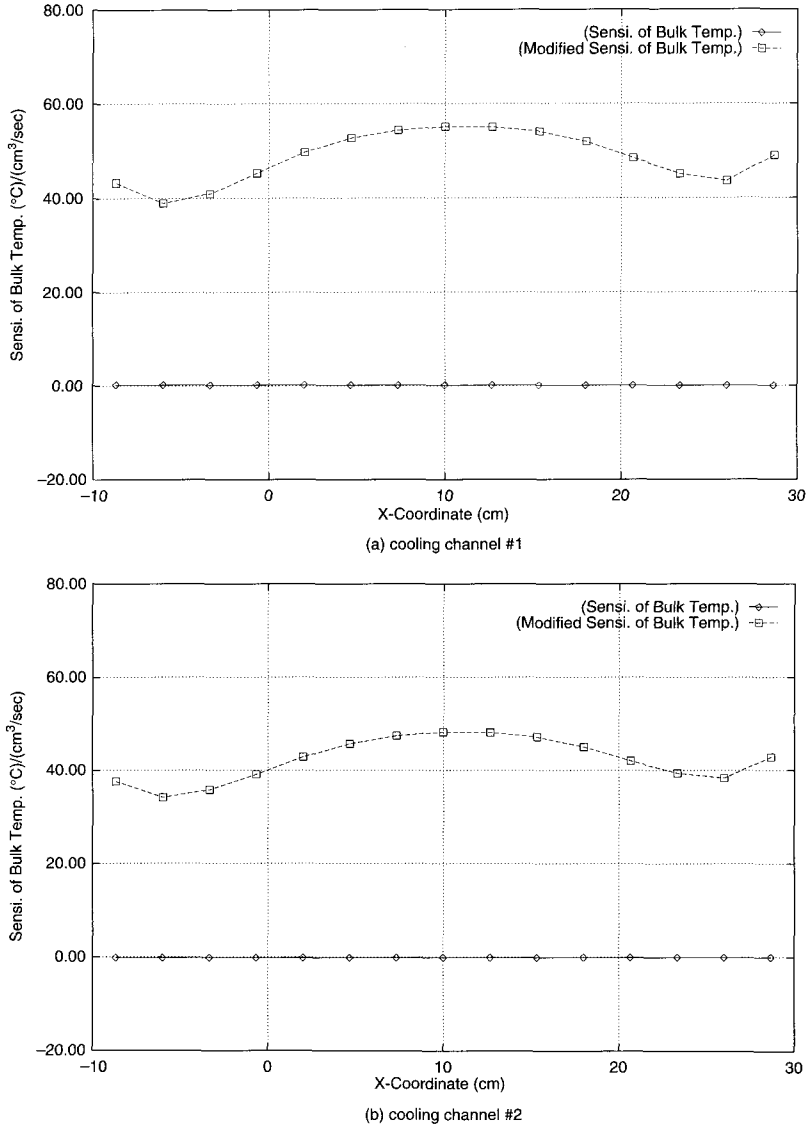


Fig. 23. Results of the sensitivity and the modified sensitivity of the coolant bulk temperature by DDA: (a) at cooling channel #1 with respect to the radius of the cooling channel #1 and (b) at cooling channel #2 with respect to the radius of the cooling channel #2.

Z-coordinate

As shown in Fig. 25, as the z-coordinate of cooling channel #1 increases (i.e. moving in the downward direction as in Fig. 7), the part surface temperature increases in the portion of the part where the distance between the part position and cooling channel #1 increases. At the same time, the part surface temperature decreases in

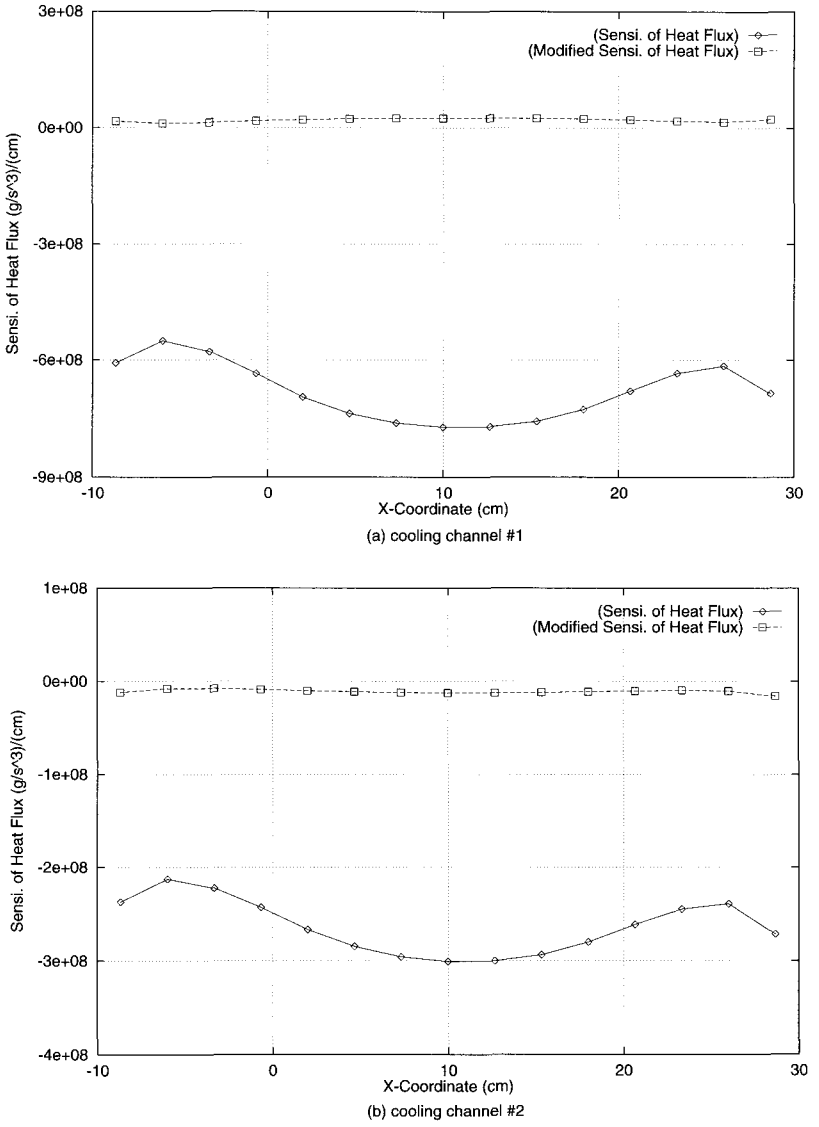


Fig. 24. Results of the sensitivity and the modified sensitivity of the heat flux by DDA: (a) at cooling channel #1 with respect to the radius of the cooling channel #1 and (b) at cooling channel #2 with respect to the radius of the cooling channel #2.

the other portion. As expected, the line of no change of the part surface temperature, that is, zero sensitivity coincides exactly with the location of cooling channel #1. The sensitivity of cooling time in this case is 0.0705 s/cm, and thus the cooling time decreases when the cooling channel #1 moves towards the center of the part surface. It may be mentioned that in the case of cooling channel #2, similar results to the case of cooling channel #1 were obtained.

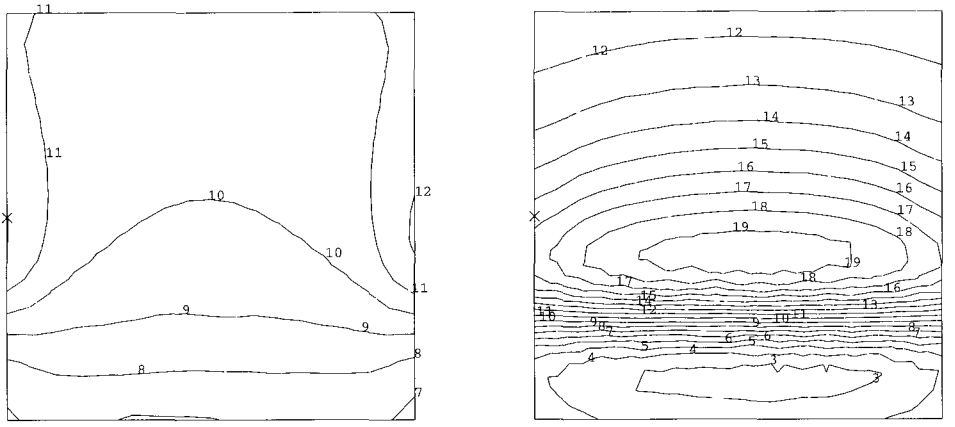


Fig. 25. Sensitivity results of the temperature on part surface with respect to the z-coordinate of cooling channel #1 by the DDA [level value = (level number − 11) × 3/10].

3.7. *Concluding remarks*

An efficient and accurate approach for the design sensitivity analysis of the injection mold cooling system was presented. Sensitivity analysis formulae have been derived based on the implicit differentiation of the modified boundary integral equations (described in Sec. 2) with respect to each design variable to obtain the sensitivity boundary integral equations. Attention was focused on the sensitivities of boundary conditions on part surface, cooling channel surface and mold exterior surface. Finally, the appropriate numerical algorithms and solution procedure are presented for obtaining the sensitivities of cooling time (related to productivity) and the part surface temperature distribution (related to uniformity). In this sensitivity analysis program, various design variables are considered as follows: (i) (design variables related to processing conditions) the inlet coolant bulk temperature and inlet coolant volumetric flow rate of each cooling channel, and (ii) (design variables related to mold cooling system design) the radius and location of each cooling channel. The major advantage of the present approach lies in the fact that the accurate sensitivity analysis results are obtained simultaneously for all design variables thus saving computational time.

Using an illustrative example problem, the numerical results of the sensitivity analysis developed here is discussed in terms of accuracy and computational efficiency. The accuracy and efficiency of the present method is demonstrated through excellent agreements between the results from the present direct differentiation approach and those from the forward finite difference method. It was concluded

that the former is much faster and more accurate than the latter. Furthermore, our understanding of the characteristics of each design variable has been enhanced from this sensitivity analysis results.

This sensitivity analysis program, when incorporated with an optimization method, would be very useful for injection mold designers to obtain an optimal configuration of a injection mold cooling system in terms of the radii and locations of the cooling channels and to determine the optimal processing conditions of the cooling stage (inlet coolant bulk temperature and inlet coolant volumetric flow rate of each cooling channel) by minimizing certain objective function related to uniformity (part quality) or productivity in injection molding processes. This will be the very subject of our next section.

4. Optimization

4.1. *Introduction*

A typical optimization process starts with a preliminary design (or initial) and searches for a better design with the help of the numerical analysis and/or the corresponding design sensitivity analysis of the current design. This optimization process essentially requires the choice of an objective function to be minimized, the constraints for the design reality, the thermal analysis and the design sensitivity analysis for any first order methods. Among them, the thermal analysis and the corresponding design sensitivity analysis have been discussed in Secs. 2 and 3, respectively. In any first order methods, a new design is proposed based on the design sensitivity analysis during an iterative process by using a nonlinear programming algorithm, such as the steepest descent algorithm and the CONMIN algorithm, etc. If no better designs can be found, the iterative optimization process stops. Otherwise, the iterative process is continued until an optimal design is obtained. There are some research works on the subject of optimization problems in the manufacturing process of plastic materials: Barone and Caulk[17] presented the two-dimensional thermal design of injection molds, Forcucci and Kwon[18] presented the three-dimensional thermal design of compression molds, Matsumoto *et al.*[19] presented the three-dimensional cooling/heating design of compression molds, and Park and Kwon[20] persented the three-dimensional thermal design of the steady conduction in special geometries, etc.

In this section, the objective function is to minimize a weighted combination of the temperature nonuniformity over the part surface and the cooling time related to the productivity. But first of all, the constraints for design realistics are proposed. The present study has developed efficient optimization procedures using the CONMIN algorithm which has been adopted to obtain the optimal configuration of the design variables in conjunction with the special boundary integral formulation in Sec. 2 and the corresponding design sensitivity analysis formulation in Sec. 3. Based on the categories of design variables, two different optimization strategies

are suggested to optimize this kind of problem. Two sample problems are solved to demonstrate the efficiency and the usefulness of the present optimization procedures.

4.2. *Optimization*

The choice of an objective function to be minimized, the constraints for the design reality, the thermal analysis and the design sensitivity analysis for any first order methods are essentially required for an optimal design. Among them, the thermal analysis and the corresponding design sensitivity analysis are achieved in Secs. 2 and 3, respectively. In this section, the choice of an objective function to be minimized, the constraints for the design reality, the optimization algorithm, the optimization strategies, and the overall structure are discussed.

Objective function

To achieve the design goal of minimizing the combination of the nonuniformity in the part surface temperature distribution and the cooling time, the normalized objective function is chosen as:

$$F(\vec{X}) = \alpha \frac{F_1(\vec{X})}{\bar{F}_1} + (1-\alpha)\frac{F_2(\vec{X})}{\bar{F}_2}, \tag{55}$$

$$\text{where } F_1(\vec{X}) = \frac{\int_{\text{part}} (T-\bar{T})^2 dA}{\bar{T}^2 \int_{\text{part}} dA} \quad \text{and} \quad F_2(\vec{X}) = t_c.$$

In Eq. 55, $\vec{X}$ is a design variable vector (All the design variables considered in this study have been discussed in Sec. 1.), $\bar{T}$ is the average temperature over the part surface, $\bar{F}_1$ is the reference value for normalization, and $\bar{F}_2$ is the reference (desired) cooling time. The user can adjust the weighting parameter, α, according to his interest. The effect of the weighting parameter will be discussed in following section.

Constraints

Proper constraints have to be imposed upon all the design variables to keep the design realistic for this optimization problem. In this particular problem, the upper and lower bounds must be placed on the radius and the position of each cooling channel respectively to keep the design realistic from the manufacturing point of view. In addition, the upper and lower bounds must also be placed on the inlet coolant bulk temperature and the inlet coolant volumetric flow rate of each cooling channel for the realistic processing conditions. Inequality constraint associated with

a design variable X_i having the upper and lower inequality bounds expressed by

$$A_i \leq X_i \leq B_i \qquad i = 1, \ldots, n \tag{56}$$

can be modified to an equality constraint by introducing a new slack design variable Y_i as follows:

$$G_i(X_i, Y_i) = X_i - A_i - (B_i - A_i) \sin^2 Y_i = 0 \qquad i = 1, \ldots, n, \tag{57}$$

where n is the number of design variables.[21]

Optimization algorithm

In order to solve the above constrained minimization problem, the present study has employed the CONMIN algorithm developed by Haaroff and Buys[22] since it has been successfully used in an optimal design of heating systems in compression molds, which is quite similar to the present model problem, by Barone and Caulk[17] and Forcucci and Kwon.[18] The CONMIN algorithm employs the augmented Lagrangian multiplier (ALM) method to deal with the equality constraints, as well as the Davidon–Fletcher–Powell method[23] for the unconstrained minimization during the successive unconstrained minimization procedure.[24]

Overall structure

The overall structure of the proposed optimal design system is schematically shown in Fig. 26. The proposed optimal design system consists of the thermal analysis module, the sensitivity analysis module, and the optimization module.

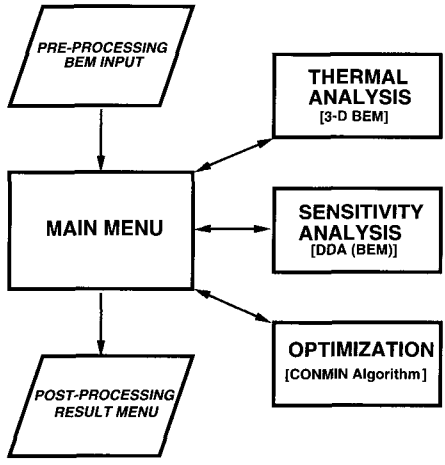


Fig. 26. Diagram for overall structure.

4.3. *Results and discussions*

Three representative examples are solved by the optimization method proposed in this chapter. The following are the relative norm convergent criteria used for stopping the iteration:

$$\left| \frac{F^{\text{new}} - F^{\text{old}}}{F^{\text{old}}} \right| \leq 10^{-2},$$

where F is the objective function.

Flat plate with two cooling channels

The example introduced for the thermal analysis in Sec. 2 and the sensitivity analysis in Sec. 3 was also applied to the proposed optimization method. This simple problem is chosen to investigate the role and effect of the weighting parameter, α, in the objective function as indicated in Eq. 55. The radii, the locations, and the processing conditions of cooling channel #1 and cooling channel #2 are made intentionally unsymmetrical to see if the optimized result leads to a symmetric configuration as a way of validating the proposed method. In this example, the design variables are the inlet coolant bulk temperature, the inlet coolant volumetric flow rate, the radius, the y-coordinate and the z-coordinate of each cooling channel, as described in Sec. 2. It is again noted that the effect of the x-coordinate of each cooling channel is negligible. Strategy A is applied to this example. In the optimization procedure for this example, the value of the objective function F_1 and the cooling time in the initial design are referred to as $\bar{F}_1$ and $\bar{F}_2$, respectively. It is noted that the cooling time at the initial design is 6.47 seconds. The optimization procedure required five iterations of unconstrained minimizations to yield the optimal configuration and the elapsed CPU time was about 40 hours. Table 1 shows the obtained optimal values of design variables for several weighting parameters. (In the table, j, $T_{i,j}$, $Q_{i,j}$, a_j, x_j, y_j, and z_j denote the cooling channel identification number, the inlet coolant bulk temperature, the inlet coolant volumetric flow rate, the radius, the x-coordinate, the y-coordinate and the z-coordinate of the jth cooling channel, respectively.) From the above results, the following characteristics of each design variable have been found as expected:

(a) As α increases, the inlet coolant bulk temperature increases.
(b) As α increases, the inlet coolant volumetric flow rate decreases.
(c) As α increases, the radius decreases.
(d) As α increases, the distance between the cooling channel and the part surface increases.
(e) The optimal value of the z-coordinate coincides with the symmetrical location, that is, the center of the part for all weighting parameters.

Figure 27 shows the corresponding values of normalized objective functions F_1, F_2 and F for several values of weighting parameters. Figure 28 shows the path of design

changes towards a symmetric optimal configuration (Fig. 28a) and the changes of corresponding normalized objective functions in the iterative optimization at $\alpha = 0.5$ (Fig. 28b). It is interesting to note that the configuration changes towards a symmetric configuration for all weighting parameters as it should. Figure 29 shows the distribution of $(T/\bar{T}) - 1$ over the part surface (referred to as the uniformity distribution, hereafter) in the initial design (Fig. 29a) and in the optimal design (Fig. 29b) at $\alpha = 1.0$, which clearly indicates that the optimal design has more uniform temperature distribution than the initial one. The satisfactory results of this

Table 1. Optimal values of design variables for several weighting parameters.

α		0.00	0.25	0.50	0.75	1.00
$T_{i,j}$	$j=1$	23.7	22.4	19.7	14.5	13.3
	$j=2$	24.9	23.5	19.8	15.3	14.0
$Q_{i,j}$	$j=1$	172	184	204	230	246
	$j=2$	160	172	201	227	237
a_j	$j=1$	0.483	0.489	0.498	0.510	0.517
	$j=2$	0.486	0.491	0.499	0.511	0.518
y_j	$j=1$	5.10	4.67	4.30	3.66	3.09
	$j=2$	−5.15	−4.71	−4.25	−3.72	−3.12
z_j	$j=1$	0.0137	0.0398	0.0287	0.0399	0.0678
	$j=2$	−0.0501	−0.0973	−0.0371	−0.0349	−0.0200

(unit: °C, cc/s, cm, cm, cm)

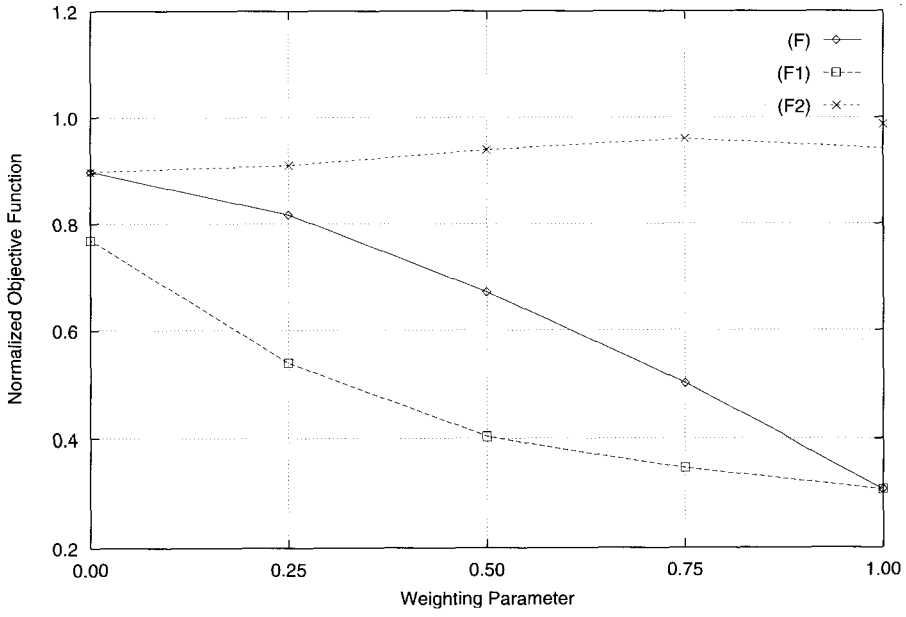


Fig. 27. Normalized objective functions versus weighting parameter at the optimal design.

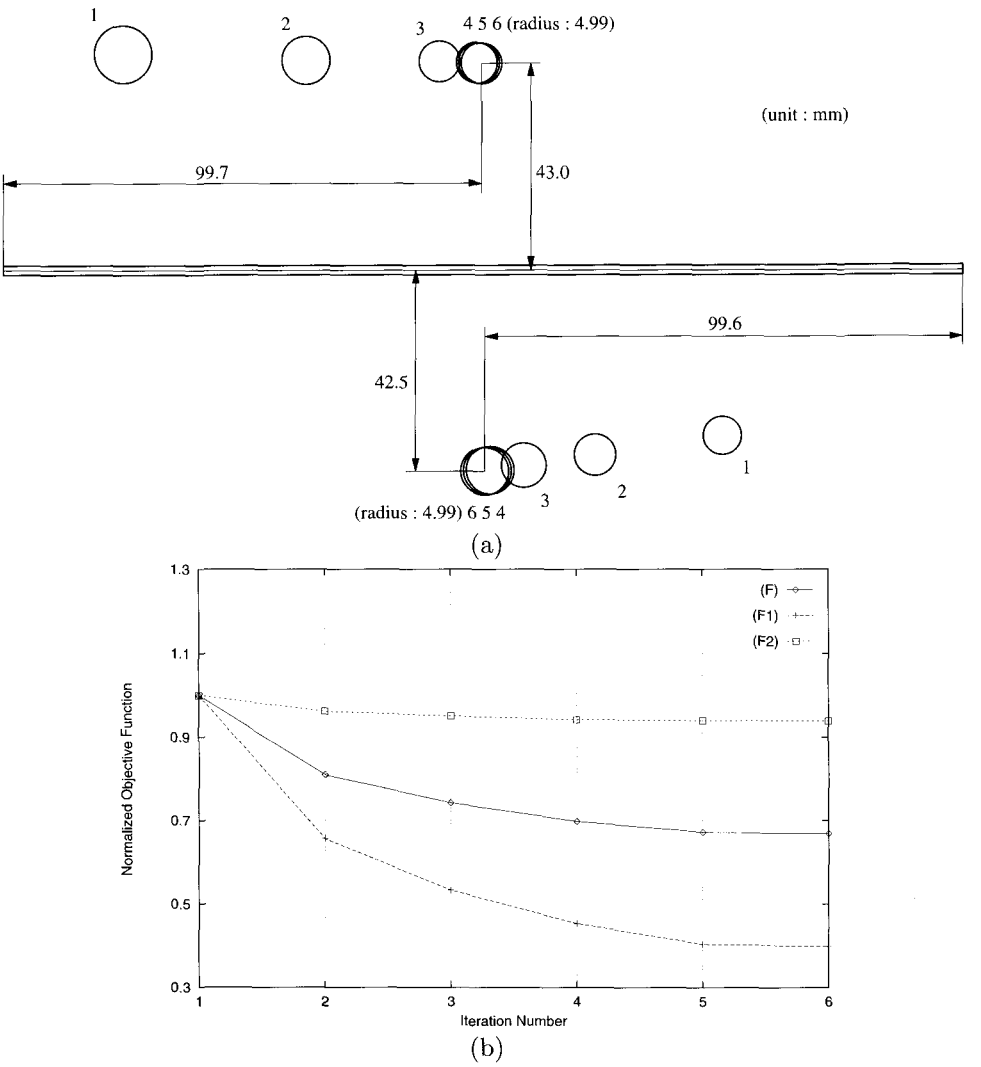
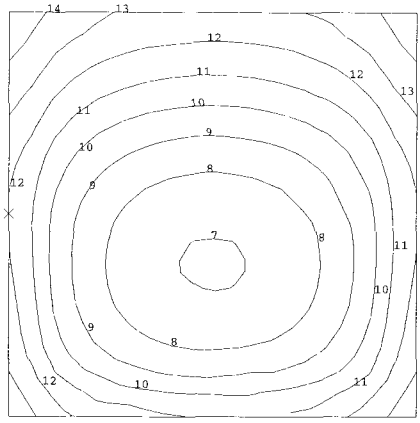


Fig. 28. Results of optimization at $\alpha = 0.5$: (a) path from initial design to optimal design and (b) normalized objective functions versus iteration.

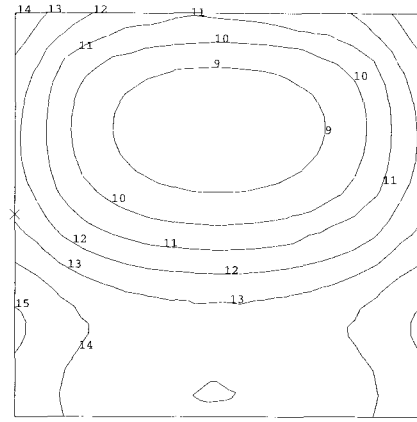
example may imply that the proposed optimization method works quite successfully. Furthermore, it have been examined that almost the same optimal configuration had been obtained by the proposed optimization method from several different initial designs at $\alpha = 1.0$.

Boxed shape with six cooling channels

The boxed-shape part geometry with six cooling channels, as shown in Fig. 30, was considered as a more practical example than the previous examples. The number

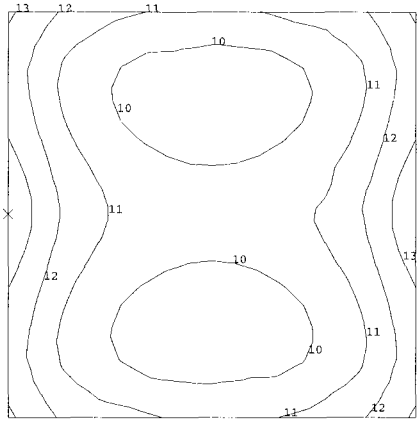


min:−0.19 max: 0.16
plus plane

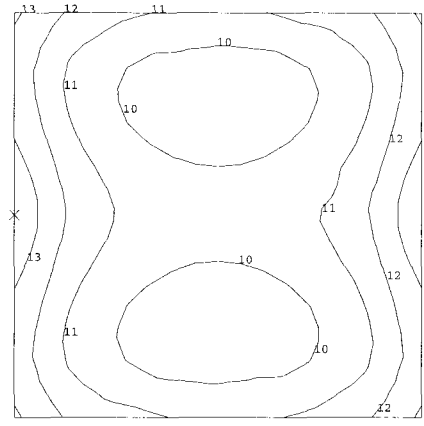


min:−0.19 max: 0.12
minus plane

(a) initial design



min:−0.14 max: 0.07
plus plane



min:−0.14 max: 0.07
minus plane

(b) optimal design

Fig. 29. Uniformity distribution on the part surface at $\alpha = 1.0$: (a) initial design ($\bar{T} = 74.97$) and (b) optimal design ($\bar{T} = 73.00$) [level value = (11 − level number)/20].

of boundary elements in the part and the six cooling channels are 1148 and 78, respectively. For this example, the following processing conditions are used:

(i) the injection and ejection temperatures of the part material are 250°C and 110°C, respectively;

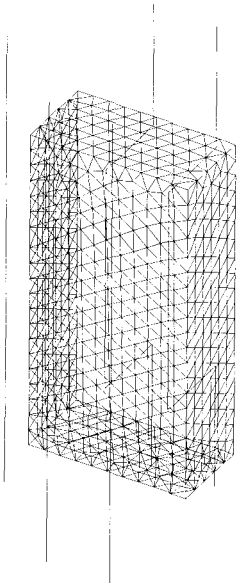


Fig. 30. Boxed shape with six cooling channels: initial design.

(ii) the inlet coolant temperature and the volumetric flow rate are 20°C and 200 cc/s respectively for all cooling channels.

The design variables of this example are the inlet coolant bulk temperature, the inlet coolant volumetric flow rate, the radius, the x-coordinate and the z-coordinate of each cooling channel. It is noted that the effect of the y-coordinate of each cooling channel is negligible. At $\alpha = 0.5$, nine iterations are required to obtain the optimal configuration and the elapsed CPU time was 255:46:24. Figure 31 shows the cross-sectional view of the initial design (dotted lines) and the final optimal one (solid lines). It is noted that the radii of all cooling channels in the initial design are 0.5 cm. The overall objective function F at the optimal design is 0.768 ($F_1 = 0.613$ and $F_2 = 0.923$). The cooling times at the initial and optimal designs are 5.54 seconds and 5.11 seconds respectively. Figure 32 shows the uniformity distribution over the part surface both in the initial design and in the optimal one. This figure indicates that this optimization method improves both the temperature uniformity and the cooling time at the given weighting parameter. One can improve the uniformity or the productivity by adjusting the weighting parameter, α, in Eq. 55. The proposed optimization procedure applied to this practical example again seems to work successfully.

4.4. *Concluding remarks*

This section deals with the overall numerical procedure for the optimal cooling system design for injection molds with a thin cavity. The CONMIN algorithm is

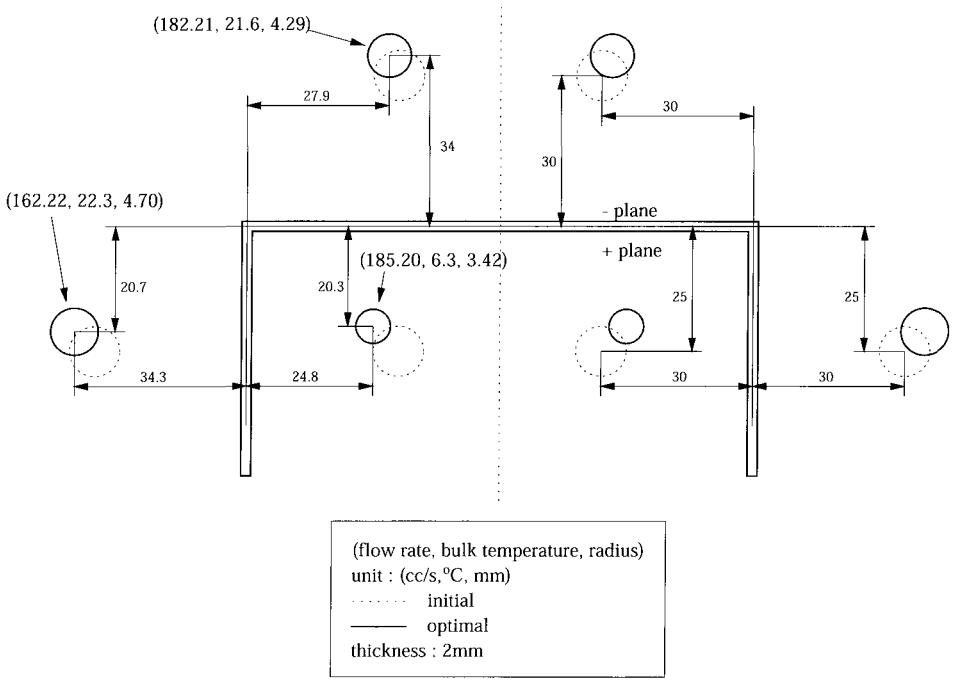


Fig. 31. Initial design and optimal design.

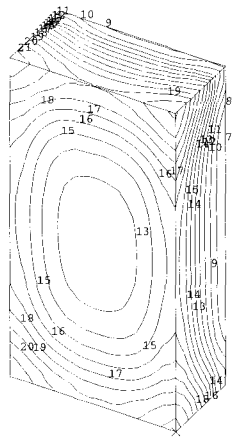
successfully applied to this optimal design problem with the help of the thermal analysis and the corresponding design sensitivity analysis via the boundary element method.

For this optimal design, an objective function is proposed to minimize a weighted combination of the cooling time and the temperature nonuniformity over the part surface. The former has to do with the warpage in the final part, while the latter is directly related to the overall productivity of the injection molding process. Side (interval) constraints for all design variables are introduced for design reality. In the proposed objective function, the weighting parameter between the temperature nonuniformity and the cooling time can be adjusted according to user's interest. The direction of the design variable vector which decreases the objective function and related to the pure part nonuniformity is usually opposite to that which decreases another objective function and related to pure productivity. The effect of this weighting parameter on an optimal design was examined using an example.

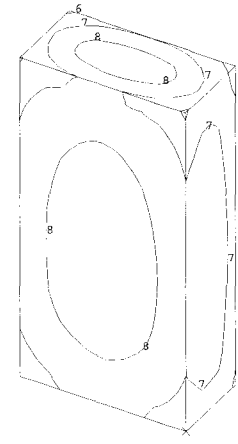
5. Summary and Future Works

5.1. *Summary*

In recent years, increased attention has been paid to the design of cooling systems in injection molding, as it became clear that cooling affects both productivity and

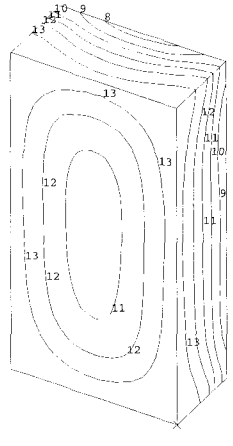


min:−0.20 max: 0.54
plus plane

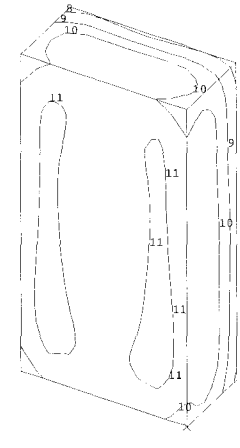


min: −0.27 max: −0.11
minus plane

(a) initial design



min:−0.18 max: 0.14
plus plane



min:−0.17 max: 0.01
minus plane

(b) optimal design

Fig. 32. Uniformity distribution on the part surface: (a) initial design ($\bar{T} = 62.93$) and (b) optimal design ($\bar{T} = 63.53$) [level value = (11 − level number)/20].

part quality. In order to systematically improve the performance of a cooling system to obtain rapid, uniform cooling, the designer may need an optimal design system for the mold cooling system design as well as the process conditions of the cooling stage. In this study, an efficient optimization procedure for this kind of problem is proposed utilizing (i) the special boundary element analysis, (ii) the corresponding design sensitivity analysis using the direct differentiation approach and (iii) the

optimization algorithm. For this optimal design, an objective function is proposed to minimize a weighted combination of the cooling time and the temperature nonuniformity over the part surface. The latter has to do with the warpage in the final part, while the former is directly related to the overall productivity of the injection molding process. In this optimization program, various design variables are considered as follows: (i) (design variables related to processing conditions) the inlet coolant bulk temperature and the inlet coolant volumetric flow rate of each cooling channel, and (ii) (design variables related to mold cooling system design) the radius and location of each cooling channel.

Each step of the proposed optimization procedure will be briefly explained below. First, for the optimal design system, the designer needs a thermal analysis tool for a three-dimensional mold heat transfer during the cooling stage of an injection molding process as the first step of optimization. This thermal analysis tool should be able to predict the cooling time (and thus the cycle time), the temperature and the temperature gradients on the mold surface, and so on. Several thermal analysis tools have been developed by many researchers using the modified boundary element method under the cycle-averaged concept. In these simulation packages, the mold heat transfer is considered as a cyclic-steady, three-dimensional conduction; the heat transfer within the melt region is treated as a transient, one-dimensional conduction; the heat exchange between the cooling channel surfaces and coolant is considered steady, and so is the heat exchange between the ambient air and the mold exterior surfaces. Such numerical simulation packages were validated and are being used by many relevant industries with a reasonable satisfaction. However, it was found that seemingly negligible inaccuracy in the thermal analysis result sometimes gives rise to meaningless sensitivity analysis results. Thus quite an accurate thermal analysis is critically essential to obtain a precise sensitivity analysis result. Developing a successful sensitivity analysis tool is our ultimate goal of this study. With that the accuracy of the thermal analysis system has been improved based on the modified boundary element method. In addition the rigorous treatments of boundary conditions appropriate for the sensitivity analysis have been developed by considering the following issues: (i) the numerical convergency for obtaining a temperature distribution accurate enough to warrant a meaningful design sensitivity analysis; (ii) the series solution in part analysis for obtaining appropriate equations for determining the cooling time and cycle-averaged heat flux on cavity surface; (iii) the treatment of the tip surface of line elements to satisfy the basic assumptions of the boundary integral equation; (iv) the treatment of coolant in consideration of the cooling process condition; and (v) the treatment of the mold exterior surface associated with an algorithm to determine the temperature on mold exterior surface. All these issues contribute greatly towards overcoming the difficulties in obtaining successfully the corresponding sensitivity analysis. It may be mentioned that most of them are closely related to the rigorous treatment of boundary conditions and that all design variables considered in the sensitivity analysis are defined only on boundary surfaces. Next, an efficient and accurate approach for the design

sensitivity analysis of the injection mold cooling system was presented. We have derived the sensitivity analysis formulae based on the implicit differentiation of the modified boundary integral equations with respect to each design variable to obtain the sensitivity boundary integral equations. In addition, emphasis was placed on the sensitivities of boundary conditions on part surface, cooling channel surface and mold exterior surface. Finally, appropriate numerical algorithms and solution procedures are presented for obtaining the sensitivities of cooling time (related to productivity) and part surface temperature distribution (related to uniformity). The major advantage of the present approach lies in the fact that the accurate sensitivity analysis results are obtained simultaneously for all design variables thus saving computational time. It was concluded that the present direct differentiation approach is much faster and more accurate than the forward finite difference method through excellent agreements between the results from both methods. Furthermore, our understanding of the characteristics of each design variable has been enhanced from this sensitivity analysis results. Finally, the CONMIN algorithm is applied for the optimization program with the help of the above thermal analysis and the corresponding design sensitivity analysis. The CONMIN algorithm employs the augmented Lagrangian multiplier method to deal with the equality constraints, as well as the Davidon–Fletcher–Powell method for the unconstrained minimization during the successive unconstrained minimization procedure. In this optimization program, the proper objective function of the combination of both the uniform part surface temperature distribution and the productivity and constraints imposed upon the design variables to keep the design realistic are proposed. The user can adjust the combination in the proposed objective function using the weighting parameter, according to his interest. The effects of this weighting parameter were examined by solving sample problems. Three different optimization strategies based upon the natures (categories) of design variables are proposed: Strategy A, Strategy B, and Strategy C. It can be concluded that Strategy A is best considering optimal results and the elapsed CPU time as well.

The developed computer aided optimal design system would be very useful for injection mold designers to obtain an optimal configuration of an injection mold cooling system in terms of radii and locations of cooling channels and to determine the optimal processing conditions of the cooling stage (inlet coolant bulk temperature and inlet coolant volumetric flow rate of each cooling channel) by minimizing certain objective functions related to uniformity (part quality) and/or productivity in the injection molding processes.

5.2. *Future works*

One may consider the following studies in the future under the developed computer aided design system featured in this study:

(a) experimental verification of the developed computer aided design system;

(b) boundary element system introducing part thickness effect and circumferential effect of cooling channels;
(c) unified system of filling, packing and cooling analysis; and
(d) periodic or transient cooling analysis.

Appendix

A. Coordinate Transformation Rule

The following orthogonal matrix represents the coordinate transformation rule used for the sensitivity analysis with respect to the location of the cooling channel. The basis $\{\hat{\imath}, \hat{\jmath}, \hat{k}\}$ represents the local coordinate system in each line element of the cooling channel and the basis $\{\hat{I}, \hat{J}, \hat{K}\}$ represents the global coordinate system as shown in Fig. 4. All symbols used in the following orthogonal matix are shown in Fig. 4.

$$\begin{Bmatrix} \hat{\imath} \\ \hat{\jmath} \\ \hat{k} \end{Bmatrix} = \begin{bmatrix} \dfrac{P_1 - Z_P l_x}{R_P} & \dfrac{P_2 - Z_P l_y}{R_P} & \dfrac{P_3 - Z_P l_z}{R_P} \\ \dfrac{P_3 l_y - P_2 l_z}{R_P} & \dfrac{P_1 l_z - P_3 l_x}{R_P} & \dfrac{P_2 l_x - P_1 l_y}{R_P} \\ l_x & l_y & l_z \end{bmatrix} \begin{Bmatrix} \hat{I} \\ \hat{J} \\ \hat{K} \end{Bmatrix},$$

where $P_1 = P_x - x$, $P_2 = P_y - y$, and $P_3 = P_z - z$.

In the special case that P is on the axis of the same element as Q, the above coordinate transformation rule can be simplified to permit the following form (in representing $\vec{\nu}$, the unprimed system $\{\nu_x, \nu_y, \nu_z\}$ for the global coordinate system and the primed system $\{\nu'_x, \nu'_y, \nu'_z\}$ for the local coordinate system):

$$\begin{aligned} \nu'_x \hat{\imath} + \nu'_y \hat{\jmath} &= (\nu_x - \nu'_z l_x)\hat{I} + (\nu_y - \nu'_z l_y)\hat{J} + (\nu_z - \nu'_z l_z)\hat{K}, \\ \nu'_z \hat{k} &= \nu'_z (l_x \hat{I} + l_y \hat{J} + l_z \hat{K}) \\ \text{where} \quad \hat{k} &= \hat{l} = l_x \hat{I} + l_y \hat{J} + l_z \hat{K}, \\ \nu'_z &= \hat{l} \cdot \hat{\nu} = l_x \nu_x + l_y \nu_y + l_z \nu_z. \end{aligned}$$

B. Integral Formulae for Thermal Analysis

B.1. *Singular integrals in triangular element*

The singular integrals over a flat area A in a triangular element can be evaluated analytically. The symbols used in the following closed forms are shown in Fig. 33.

$$\begin{aligned} \int_A \frac{1}{r}\, dS = \frac{2\Delta}{3} \Bigg[& \frac{1}{r_{23}} \ln \left| \frac{\tan\{(\theta_1 + \alpha_2)/2\}}{\tan(\alpha_2/2)} \right| \\ & + \frac{1}{r_{31}} \ln \left| \frac{\tan\{(\theta_2 + \alpha_3)/2\}}{\tan(\alpha_3/2)} \right| \\ & + \frac{1}{r_{12}} \ln \left| \frac{\tan\{(\theta_3 + \alpha_1)/2\}}{\tan(\alpha_1/2)} \right| \Bigg], \end{aligned}$$

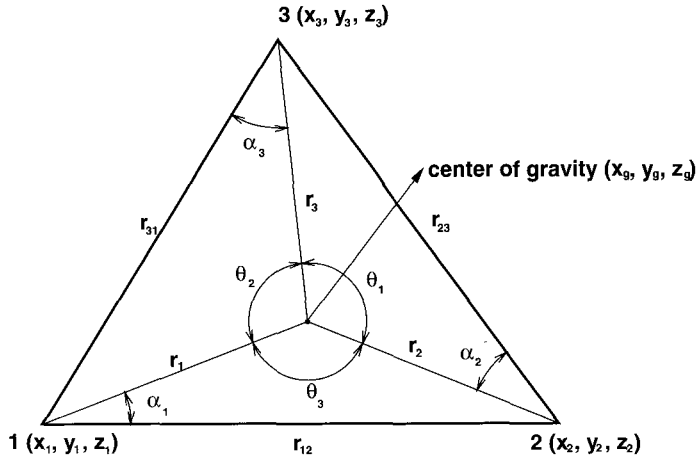


Fig. 33. Notations for geometric information of a triangular element.

$$\int_A \frac{\partial}{\partial n}\left(\frac{1}{r}\right) dS = 0,$$

$$\int_A \frac{\partial}{\partial \nu}\left(\frac{1}{r}\right) dS = 0,$$

$$\int_A \frac{\partial}{\partial \nu}\left(\frac{\partial}{\partial n}\left(\frac{1}{r}\right)\right) dS = \frac{3}{2\Delta}\left[r_{23}\left\{\cos(\theta_1+\alpha_2)-\cos\alpha_2\right\}\right.$$
$$+r_{31}\left\{\cos(\theta_2+\alpha_3)-\cos\alpha_3\right\}$$
$$\left.+\, r_{12}\left\{\cos(\theta_3+\alpha_1)-\cos\alpha_1\right\}\right].$$

In above equations, Δ denotes the area of the triangular element.

B.2. *Closed form of integrals in line element*

The integrals over θ can be evaluated in closed forms. In the special case that P is on the axis of the same element as Q, the integrals over θ and l can also be evaluated in closed forms. The symbols used in the following definitions and closed forms are shown in Fig. 4.

One may first define some functions used in evaluating integrals over θ. To avoid conflicts with other symbols in the study, one shall use non-standard, calligraphic symbols for the following functions. $\mathcal{K}$ and $\mathcal{E}$ denote the complete elliptic integrals of the first and second kind, respectively, defined as:

$$\mathcal{K}(m) = \int_0^{\pi/2} \frac{d\theta}{\sqrt{1-m\sin^2\theta}},$$

$$\mathcal{E}(m) = \int_0^{\pi/2} \sqrt{1-m\sin^2\theta}\, d\theta.$$

$\mathcal{P}$, $\mathcal{Q}$, $\mathcal{R}$ and $\mathcal{S}$ denote the simple modifications of $\mathcal{K}$ and $\mathcal{E}$ and can be evaluated using the derivatives of $\mathcal{K}$ and $\mathcal{E}$:

$$\mathcal{P}(m) = \int_0^{\pi/2} \frac{d\theta}{(A^2 + B^2 \sin^2\theta)^{1/2}} = \frac{1}{\sqrt{A^2 + B^2}} \mathcal{K}(m),$$

$$\mathcal{Q}(m) = \int_0^{\pi/2} \frac{d\theta}{(A^2 + B^2 \sin^2\theta)^{3/2}} = \frac{1}{A^2\sqrt{A^2 + B^2}} \mathcal{E}(m),$$

$$\mathcal{R}(m) = \int_0^{\pi/2} \frac{d\theta}{(A^2 + B^2 \sin^2\theta)^{5/2}},$$

$$= \frac{1}{3A^2(A^2 + B^2)^{3/2}} \left[\left(4 + 2\frac{B^2}{A^2} \right) \mathcal{E}(m) - \mathcal{K}(m) \right],$$

$$\mathcal{S}(m) = \int_0^{\pi/2} \frac{d\theta}{(A^2 + B^2 \sin^2\theta)^{7/2}}$$

$$= \frac{1}{15A^2(A^2 + B^2)^{5/2}} \left[\left(23 + 23\frac{B^2}{A^2} + 8\frac{B^4}{A^4} \right) \mathcal{E}(m) - \left(8 + 4\frac{B^2}{A^2} \right) \mathcal{K}(m) \right],$$

where $m = B^2/(A^2 + B^2)$.

$\mathcal{W}$, $\mathcal{X}$, $\mathcal{Y}$ and $\mathcal{Z}$ are more general forms of $\mathcal{P}$, $\mathcal{Q}$, $\mathcal{R}$ and $\mathcal{S}$, respectively:

$$\mathcal{W}(m; I_0) = \int_0^{\pi/2} \frac{I_0}{(A^2 + B^2 \sin^2\theta)^{1/2}} d\theta = \{I_0\} \mathcal{P}(m),$$

$$\mathcal{X}(m; I_0, I_2) = \int_0^{\pi/2} \frac{I_0 + I_2 \sin^2\theta}{(A^2 + B^2 \sin^2\theta)^{3/2}} d\theta$$

$$= \left\{ \frac{1}{B^2} I_2 \right\} \mathcal{P}(m) + \left\{ I_0 - \frac{A^2}{B^2} I_2 \right\} \mathcal{Q}(m),$$

$$\mathcal{Y}(m; I_0, I_2, I_4) = \int_0^{\pi/2} \frac{I_0 + I_2 \sin^2\theta + I_4 \sin^4\theta}{(A^2 + B^2 \sin^2\theta)^{5/2}} d\theta$$

$$= \left\{ \frac{1}{B^4} I_4 \right\} \mathcal{P}(m) + \left\{ \frac{1}{B^2} I_2 - 2\frac{A^2}{B^4} I_4 \right\} \mathcal{Q}(m)$$

$$+ \left\{ I_0 - \frac{A^2}{B^2} I_2 + \frac{A^4}{B^4} I_4 \right\} \mathcal{R}(m),$$

$$\mathcal{Z}(m; I_0, I_2, I_4, I_6) = \int_0^{\pi/2} \frac{I_0 + I_2 \sin^2\theta + I_4 \sin^4\theta + I_6 \sin^6\theta}{(A^2 + B^2 \sin^2\theta)^{7/2}} d\theta$$

$$= \left\{ \frac{1}{B^6} I_6 \right\} \mathcal{P}(m) + \left\{ \frac{1}{B^4} I_4 - 3\frac{A^2}{B^6} I_6 \right\} \mathcal{Q}(m)$$

$$+\left\{\frac{1}{B^2}I_2 - 2\frac{A^2}{B^4}I_4 + 3\frac{A^4}{B^6}I_6\right\}\mathcal{R}(m)$$
$$+\left\{I_0 - \frac{A^2}{B^2}I_2 + \frac{A^4}{B^4}I_4 - \frac{A^6}{B^6}I_6\right\}\mathcal{S}(m).$$

Now, one may introduce some intermediate definitions to shorten the integral formulae:

$$\begin{aligned}
&s_1 = a - R_P && s_2 = l - Z_P\\
&s_3 = s_1\nu'_x + s_2\nu'_z && s_4 = s_1^2\nu'_x - s_2\nu'_z R_P\\
&s_5 = -s_1(s_1 - R_P)\nu'_x + 2s_2\nu'_z R_P && s_6 = -s_1(s_1 + a)\nu'_x + s_2\nu'_z R_P\\
&s_7 = s_1(s_1 - R_P)\nu'_x - s_2\nu'_z R_P && s_8 = 2(s_1 + a)\nu'_x + s_2\nu'_z\\
&t_1 = L - Z_P && t_2 = \sqrt{a^2 + Z_P^2}\\
&t_3 = \sqrt{a^2 + t_1^2}\\
&A^2 = s_1^2 + s_2^2 && B^2 = 4aR_P\\
&m = B^2/(A^2 + B^2).
\end{aligned}$$

In the analysis formulation, the following integral formulae are needed:

$$\begin{aligned}
I_1 &= \int_0^{2\pi} \frac{1}{r}\, a\, d\theta = 4a\,\mathcal{W}(m;1),\\
I_2 &= \int_0^{2\pi} \frac{\partial}{\partial n}\left(\frac{1}{r}\right) a\, d\theta = 4a\,\mathcal{X}(m;s_1,2R_P),\\
I_3 &= \int_0^{2\pi} \frac{\partial}{\partial \nu}\left(\frac{1}{r}\right) a\, d\theta = 4a\,\mathcal{X}(m;s_3,-2a\nu'_x),\\
I_4 &= \int_0^{2\pi} \frac{\partial}{\partial \nu}\left(\frac{\partial}{\partial n}\left(\frac{1}{r}\right)\right) a\, d\theta\\
&= 12a\,\mathcal{Y}(m;s_1s_3,-2s_4,-B^2\nu'_x) + 4a\nu'_x\,\mathcal{X}(m;-1,2).
\end{aligned}$$

In the special case that P is on the axis of the same element as Q, the above formulae can be simplified to obtain closed forms of integrations over l. In such a case, the following integral formulae are needed:

$$\begin{aligned}
J_1 &= \int_0^L \int_0^{2\pi} \frac{1}{r}\, a\, d\theta\, dl = 2\pi a \ln\left|\frac{t_3 - t_1}{t_2 - Z_P}\right|,\\
J_2 &= \int_0^L \int_0^{2\pi} \frac{\partial}{\partial n}\left(\frac{1}{r}\right) a\, d\theta\, dl = 2\pi\left(\frac{Z_P}{t_2} + \frac{t_1}{t_3}\right),\\
J_3 &= \int_0^L \int_0^{2\pi} \frac{\partial}{\partial \nu}\left(\frac{1}{r}\right) a\, d\theta\, dl = 2\pi a\nu'_z\left(\frac{1}{t_2} - \frac{1}{t_3}\right),\\
J_4 &= \int_0^L \int_0^{2\pi} \frac{\partial}{\partial \nu}\left(\frac{\partial}{\partial n}\left(\frac{1}{r}\right)\right) a\, d\theta\, dl = 2\pi a^2\nu'_z\left(\frac{1}{t_2^3} - \frac{1}{t_3^3}\right).
\end{aligned}$$

C. Integral Formulae for Design Sensitivity Analysis

In the sensitivity analysis formulation with respect to the radius of the cooling channel, the following integral formulae are needed:

$$\begin{aligned}
\frac{\partial I_1}{\partial a} &= \frac{I_1}{a} + 4a\,\mathcal{X}(m;-s_1,-2R_P),\\
\frac{\partial I_2}{\partial a} &= \frac{I_2}{a} - 12a\,\mathcal{Y}(m;s_1^2,4R_Ps_1,4R_P^2) + 4a\,\mathcal{X}(m;1,0),\\
\frac{\partial I_3}{\partial a} &= \frac{I_3}{a} - 12a\,\mathcal{Y}(m;s_1s_3,-2s_4,-B^2\nu_x') + 4a\,\nu_x'\mathcal{X}(m;1,-2),\\
\frac{\partial I_4}{\partial a} &= \frac{I_4}{a} - 60a\,\mathcal{Z}(m;s_1^2s_3,2s_1s_5,4R_Ps_6,-2B^2R_P\nu_x')\\
&\quad + 12a\,\mathcal{Y}(m;2s_1\nu_x'+s_3,-2(3s_1-R_P)\nu_x',-8R_P\nu_x').
\end{aligned}$$

In the sensitivity analysis formulation with respect to the location of the cooling channel, the following integral formulae are needed:

$$\begin{aligned}
\nabla_\xi(I_1) &= -\nabla_x(I_1) = 4a\,\mathcal{X}(m;-s_1,2a)\,\hat{\imath} - 4as_2\,\mathcal{X}(m;1,0)\hat{k},\\
\nabla_\xi(I_2) &= -\nabla_x(I_2) = \left\{12a\,\mathcal{Y}(m;-s_1^2,2s_1^2,B^2) + 4a\,\mathcal{X}(m;1,-2)\right\}\hat{\imath}\\
&\quad - 12s_2\,\mathcal{Y}(m;s_1,2R_P,0)\,\hat{k},\\
\nabla_\xi(I_3) &= -\nabla_x(I_3) = -12a\,\mathcal{Y}(m;s_1s_3,-2a(s_1\nu_x'+s_3),4a^2\nu_x')\hat{\imath}\\
&\quad + 48a^3\nu_y'\,\mathcal{Y}(m;0,-1,1)\hat{\jmath} - 12as_2\,\mathcal{Y}(m;s_3,-2a\nu_x',0)\hat{k}\\
&\quad + 4a\hat{\nu}\,\mathcal{X}(m;1,0),\\
\nabla_\xi(I_4) &= -\nabla_x(I_4)\\
&= \left\{-60a\,\mathcal{Z}(m;s_1^2s_3,-2s_1^2(s_3+a\nu_x'),4as_7,2aB^2\nu_x')\right.\\
&\quad \left. + 12a\,\mathcal{Y}(m;s_1\nu_x'+s_3,-2s_8,8a\nu_x')\right\}\hat{\imath}\\
&\quad + \left\{-240a^3\nu_y'\,\mathcal{Z}(m;0,s_1,3R_P-a,-2R_P)\right.\\
&\quad \left. + 96a^2\nu_y'\,\mathcal{Y}(m;0,1,-1)\right\}\hat{\jmath}\\
&\quad + \left\{-60as_2\,\mathcal{Z}(m;s_1s_3,-2s_4,-B^2\nu_x',0)\right.\\
&\quad \left. + 12as_2\nu_x'\,\mathcal{Y}(m;1,-2,0)\right\}\hat{k}\\
&\quad + 12a\hat{\nu}\,\mathcal{Y}(m;s_1,2R_P,0).
\end{aligned}$$

Next, in the sensitivity analysis formulation with respect to the radius of the cooling channel for the special case that P is on the axis of the same element as Q, the following integral formulae are needed:

$$\begin{aligned}
\frac{\partial J_1}{\partial a} &= \frac{J_1}{a} - 2\pi\left(\frac{Z_P}{t_2}+\frac{t_1}{t_3}\right),\\
\frac{\partial J_2}{\partial a} &= \frac{J_2}{a} + \frac{2\pi}{a}\left\{\frac{Z_P}{t_2}+\frac{t_1}{t_3} - \frac{Z_P(2t_2^2+a^2)}{t_2^3} - \frac{t_1(2t_3^2+a^2)}{t_3^3}\right\},
\end{aligned}$$

$$\frac{\partial J_3}{\partial a} = \frac{J_3}{a} - 2\pi a^2 \nu_z' \left(\frac{1}{t_2^3} - \frac{1}{t_3^3} \right),$$
$$\frac{\partial J_4}{\partial a} = \frac{J_4}{a} - 6\pi a^3 \nu_z' \left(\frac{1}{t_2^5} - \frac{1}{t_3^5} \right) + 2\pi a \nu_z' \left(\frac{1}{t_2^3} - \frac{1}{t_3^3} \right).$$

Finally, in the sensitivity analysis formulation with respect to the location of the cooling channel for the same special case as above, the following integral formulae are also needed:

$$\nabla_\xi(J_1) = -\nabla_x(J_1) = -2\pi a \hat{k} \left(\frac{1}{t_2} - \frac{1}{t_3} \right),$$
$$\nabla_\xi(J_2) = -\nabla_x(J_2) = -2\pi a^2 \hat{k} \left(\frac{1}{t_2^3} - \frac{1}{t_3^3} \right),$$
$$\begin{aligned}\nabla_\xi(J_3) &= -\nabla_x(J_3) \\ &= \frac{2\pi\hat{\nu}}{a} \left(\frac{Z_P}{t_2} + \frac{t_1}{t_3} \right) - \frac{\pi(\nu_x'\hat{\imath} + \nu_y'\hat{\jmath})}{a} \left\{ \frac{Z_P(2t_2^2 + a^2)}{t_2^3} + \frac{t_1(2t_3^2 + a^2)}{t_3^3} \right\} \\ &\quad - \frac{2\pi\nu_z'\hat{k}}{a} \left(\frac{t_1^3}{t_3^3} + \frac{Z_P^3}{t_2^3} \right),\end{aligned}$$
$$\begin{aligned}\nabla_\xi(J_4) = -\nabla_x(J_4) &= \frac{2\pi(\hat{\nu} + \nu_x'\hat{\imath} + \nu_y'\hat{\jmath})}{a^2} \left\{ \frac{Z_P(2t_2^2 + a^2)}{t_2^3} + \frac{t_1(2t_3^2 + a^2)}{t_3^3} \right\} \\ &\quad - \frac{\pi(\nu_x'\hat{\imath} + \nu_y'\hat{\jmath})}{a^2} \left\{ \frac{Z_P(8t_2^4 + 4a^2 Z_P^2 + 7a^4)}{t_2^5} + \frac{t_1(8t_3^4 + 4a^2 t_1^2 + 7a^4)}{t_3^5} \right\} \\ &\quad - \frac{2\pi\nu_z'\hat{k}}{a^2} \left\{ \frac{Z_P^3(2t_2^2 + 3a^2)}{t_2^5} + \frac{t_1^3(2t_3^2 + 3a^2)}{t_3^5} \right\}.\end{aligned}$$

References

1. Z. Tadmor and C. G. Gogos, *Principles of Polymer Processing* (John Wiley & Sons, Inc., New York, 1979).
2. K. J. Singh, Design of mold cooling system, ed. A. I. Isayev, *Injection and Compression Molding Fundamentals* (Marcel Dekker, New York, 1987) 567–605.
3. T. H. Kwon, Mold cooling system design using boundary element method, *ASME Journal of Engineering for Industry* **110** (1988) 384–394.
4. K. Himasekhar, J. Lottey and K. K. Wang, CAD of mold cooling in injection molding using a three-dimensional numerical simulation, *ASME Journal of Engineering for Industry* **144** (1992) 213–221.
5. M. Rezayat and T. Burton, A boundary-integral formulation for complex three-dimensional geometries, *International Journal of Numerical Methods Engineering* **29** (1990) 263–273.
6. F. D. Incropera and D. P. DeWitt, *Introduction to Heat Transfer* (John Wiley & Sons, Inc., New York, 1985).
7. F. M. White, *Fluid Mechanics* (McGraw-Hill Book Company, New York, 1986).
8. C. A. Brebbia, J. C. F. Telles and L. C. Wrobel, *Boundary Element Techniques — Theory and Application in Engineering* (Springer-Verlag, Berlin, 1984).

9. S. J. Park and T. H. Kwon, Thermal and sensitivity analysis for cooling system of injection mold: Part 1. Accurate thermal analysis, *ASME Journal of Engineering for Industry* **120** (1998) 287–295.
10. C. A. J. Fletcher, *Computational Techniques for Fluid Mechanics 2* (Springer-Verlag, Berlin, 1988).
11. S. J. Park and T. H. Kwon, Sensitivity analysis formulation for three-dimensional conduction heat transfer with complex geometries using a boundary element method, *International Journal of Numerical Methods Engineering* **39** (1996) 2837–2862.
12. M. Barone and R. J. Yang, Boundary integral equations for recovery of design sensitivities in shape optimization, *AIAA Journal* **26** (1988) 589–594.
13. S. Saigal, J. T. Borggaard and J. H. Kane, Boundary element implicit differentiation equations for design sensitivities of axisymmetric structures, *International Journal of Solids Structure* **25** (1989) 527–538.
14. S. Saigal and A. Chandra, Shape sensitivities and optimal configurations for heat diffusion problems: a BEM approach, *Journal of Heat Transfer* **113** (1991) 287–295.
15. K. G. Prasad and J. H. Kane, Three-dimensional boundary element thermal shape sensitivity analysis, *International Journal of Heat & Mass Transfer* **35** (1992) 1427–1439.
16. S. J. Park and T. H. Kwon, Thermal and sensitivity analysis for cooling system of injection mold: Part II. Sensitivity analysis, *ASME Journal of Engineering for Industry* **120** (1998) 296–305.
17. M. R. Barone and D. A. Caulk, Optimal thermal design of injection molds for filled thermosets, *Polymer Engineering Science* **25** (1985) 608–617.
18. S. J. Forcucci and T. H. Kwon, A computer aided design system for three-dimensional compression mold heating, *ASME Journal of Engineering for Industry* **111** (1989) 361–368.
19. T. Matsumoto, M. Tanaka and M. Miyagawa, Boundary element system for mold cooling/heating design, eds. C. A. Brebbia and J. J. Rencis, *Boundary Element XV — Vol. 2: Stress Analysis* (Computational Mechanics Publications, Boston, 1993) 461–475.
20. S. J. Park and T. H. Kwon, Optimization method for steady conduction in special geometry using a boundary element method, *International Journal of Numerical Methods Engineering* **43** (1998) 1109–1126.
21. R. L. Fox, *Optimization Methods for Engineering Design* (Addison-Wesley, Menlo Park, 1971).
22. P. C. Haarhoff and J. D. Buys, A new method for the optimization of a nonlinear function subject to nonlinear constraint, *The Computer Journal* **13** (1970) 178–184.
23. R. Fletcher and M. J. D. Powell, A rapidly convergent descent method for minimization, *The Computer Journal* **6** (1963) 163–168.
24. J. L. Kuester and J. H. Mize, *Optimization Techniques with Fortran* (McGraw-Hill Book Company, New York, 1973).

CHAPTER 4

COMPUTER CONTROL SYSTEMS TECHNIQUES AND APPLICATIONS IN MANUFACTURING SYSTEMS

ZAHRA IDELMERFAA and JACQUES RICHARD
Centre de Recherche en Automatique de Nancy (CRAN),
University HP, Nancy I, B.P. 239,
54506 Vandoeuvre-lès-Nancy Cedex, France
E-mail: zahra.idelmerfaa@cran.uhp-nancy.fr,
jacques.richard@cran.uhp-nancy.fr

In the last few decades, the increasing development in manufacturing systems has emerged to simultaneously fulfill the requirements of quality, autonomy, flexibility and modularity. To support the extremely distributed, complex, heterogeneous and dynamic nature of this kind of manufacturing systems, a methodological approach for developing, implementing and maintaining generic, hence reusable, manufacturing control system is necessary. The major part of current researches which focus on reusable models and systems propose the application of a general architecture for enterprise modeling as an efficient support to provide software reusability for manufacturing systems. A general architecture is a structured plan, a framework on the basis of which a product, a system or an organization of an enterprise can be constructed. The central aspect of the approach relies on conceptual, organizational and operational properties. It allows on one hand to specify the design cycle of a manufacturing control system and on the other hand to take into account new requirements for manufacturing systems such as the reconfigurability and the ability to adapt to changes of target manufacturing systems configuration.

Keywords: Control system; manufacturing system; Flexible Manufacturing Cell (FMC); Computer Integrated Manufacturing (CIM); flexibility; genericity; modeling framework.

1. Introduction

Reliable operations, quality assurance, reactivity and modularity are essential to ensure economic interest in the manufacturing system facilities. These aims can be achieved by integrating control loop functions into each manufacturing level, including the design of the product, the maintenance of the means of production and, in particular, the manufacturing process ensure immediate remedy once a fault is detected. The integration of these functions allows us to search for a global optimization of the production quality. This integration needs a suitable structure which must be distributed since the intelligence of the various functions is geographically

distributed. This requires methods and efficient tools for the use and maintenance of:

(i) integration in a computer integrated manufacturing (CIM) structure;
(ii) databases which store this information and achieve their coherence and consistency;
(iii) real-time, distributed and reusable computer control systems which have to work and cooperate permanently;
(iv) communication networks which transmit all types of information.

The presentation of these control functions is limited to the level of a flexible manufacturing cell. The main concepts are:

(i) quality management in the manufacturing system;
(ii) maximum flexibility of the system related to the shop floor level, products and control;
(iii) quality information system using entity-relationship modeling; and
(iv) distribution and modularity of the control system by means of the definition of a conceptual model and its implementation with the ISO standard manufacturing message specification.

The following sections are dedicated to these concepts and their application in a flexible manufacturing cell, paying attention to quality management and reusability features.

2. Manufacturing Control System Requirements

2.1. *Quality*

Many firms are being confronted with increasing market pressures induced by competitiveness, reduced prices, better qualities, minimal response times and a rise in product diversity. In the last few decades, the increasing development in flexible manufacturing systems and cells has emerged to simultaneously fulfill the requirements of efficiency, quality and flexibility. Manufacturing systems are large, as well as complex systems that are made up of a variety of numerically controlled machine tools, coordinate measuring machines (CMM), machining centers, storage-loading-unloading-clamping-unclamping areas and pallet transport systems. Manufacturing cells are reduced-scale manufacturing systems controlled as automation islands dedicated to the manufacturing of small sets of product classes, and acting as computer controlled stand alone entities of an overall manufacturing system. Flexible manufacturing systems and cells are considered as the shop floor start-up of a CIM architecture involving contributions and effort from all the departments of a company.

As one of the major goals in manufacturing systems, quality concerns the whole life-cycle of both product and process, thus covering all quality management activities, including quality planning, control and monitoring with appropriate feedback actions.

The quality objective stands on horizontal and vertical integration flows in a CIM structure. For the horizontal viewpoint, in-process quality assurance and process certification methods require the significant use of in-process quality sensors and deterministic metrology methods supported by a reactive architecture. The QIA (Quality in automation) project has made big efforts on this subject.[1] Vertical integration of quality assurance operates from the manufacturing system automation stage to the computer aided design (CAD) stage by means of an information system with quality management features.

Quality control has to be integrated into the whole manufacturing process, as close to the production operations as possible to induce feedback actions on the operational and/or informational stages of the system.

Optimal management of manufacturing operations requires the setting up of feedback control-loops within the system architecture. To achieve this, the system must be equipped with loop-controlled functions. The difficulty here is to survey manufacturing control loop solutions using existing and heterogeneous equipment, to specify necessary modifications for their integration, and to propose adaptable and reconfigurable solutions for various types of equipment.

Thus, a solution based on a centralized production control organization, as well as coordinating NC machines without initiative would be implemented easily but would prove much too sensitive to perturbations: the smallest equipment failure can lead the manufacturing system towards a degraded and sometimes incoherent functioning mode. For these reasons, a distributed computer control solution that attempts to grant more intelligence and also more autonomy to the machines is preferred. Many studies have demonstrated the profitability of such an approach.[2]

2.2. *Autonomy*

To increase the autonomy of manufacturing system, the production management of the shop floor and the cell must be coupled as closely as possible. The manufacturing orders can be created differently, either from:

(i) information received from a next cell (Kanban),
(ii) information determined by the shop floor (MRP), or
(iii) information decided by the cell (OPT).

These differences alter the data flow consequences.

Once the cell has received the manufacturing orders, it can sequence them, but only in a very short-term way. The integration of defects and manufacturing exceptions into the planning of computing time is quite impossible. It does not take into account the due dates which are computed by the MRP (material requirement planning) as priorities, and thus the FIFO criterion is used. As for the Kanban and the OPT, the estimated processing time of the manufacturing operations may serve to compute the due dates.

The cell may also use the Kanban or the OPT criteria, or any other criterion, in maintaining both its autonomy and the decisional framework defined by the entire manufacturing orders. The cell may decide to perform more quality control tasks especially if it is not overloaded or if it is not a bottleneck.

The cell also fulfills the reporting activity, which concerns both the executed tasks and the obtained qualities. Nevertheless, the reports must be suited to the production management method because this method does not consider the same indicators.

2.3. *Flexibility*

The manufacturing abilities and control must be able to manage varying manufactured articles (for example, using a group technology configuration), or different products conceived to match the customer's requests.

The equipment flexibility in the manufacturing system is managed by the product system design and the control flexibility by the chosen control architecture. The former is related to the integrated management of the quality, and the latter needs modular software and hardware. In the long term, control system reconfiguration depends on the facility to substitute the software in the computers, the numerical controllers, and the programmable logic controllers. In the manufacturing process, the new incoming product should be able to:

(i) explain the manufacturing specification using the design stage information (the CAD/CAM product data exchange standards enable the consistency of data format from the design stage to the manufacturing and inspection stage);
(ii) perform the report of the adapted manufacturing, at least to satisfy statistical process control (SPC) methodology.

The manufacturing system adjusts itself on the control variation following the quantity and the due date in JIT (just in time) and OPT ways. The manufacturing capabilities must change with the control estimates. That is not within the cell's power, but it affects the manufacturing management method applied in the factory or the shop floor. On the other hand, the control criterion relative to this order can progress along with the methods and the manufacturing systems. Thus the manufacturing control system must be adaptive: its programming is parameterized by the criterion.

2.4. *Modularity*

A manufacturing system is composed of a computer-controlled collection of communicating and generally distributed groups of modular, automated material handling systems and interchangeable numerically controlled machine tools.[3] These various and heterogeneous components are all connected by communication links and integrated by a hierarchical network of computers. They simultaneously contribute to the efficient manufacture of a variety of parts at low to medium-sized volumes.[4]

Three essential components of a manufacturing system must consequently be taken into account. They are:

(i) the CNC machine tools to process the parts;
(ii) the material handling systems to move the parts and tools; and
(iii) an overall control system to manage the manufacturing components.

The overall control system manages the various manufacturing components and coordinates their activities to provide a cohesive structure that can react deterministically to the events occurring on the shop floor. It therefore carries out several activities such as detailed planning, direct control as well as the monitoring of all manufacturing components and establishes a link between them and the superior functions found in the shop floor.

The important criteria to structure the manufacturing control activities and their relations are *abstraction*, *decisional autonomy* and *modularity*. With regards to these principles, a distributed control solution is a natural way to grant more intelligence and therefore more autonomy and flexibility to the manufacturing components.

In a distributed control solution, a manufacturing system acts as an adaptive, dynamic system in which a wide variety of jobs are continuously and randomly introduced.[3] These jobs are broken down into operations which then have to be scheduled on various manufacturing components. The computers in a manufacturing system carry out different levels of planning and control using heterogeneous, intelligent, autonomous, and spatially distributed processors that share a common goal:

(i) at the highest level, the facility level deals with manufacturing engineering and production management;
(ii) the shop level manages, coordinates and monitors the cell in the shop floor;
(iii) the cell level manages, coordinates and monitors the stations in the cell;
(iv) the station level deals with local planning, coordination and monitoring of the equipment within the station; and
(v) the equipment level directly controls and monitors manufacturing resources such as robots, machine tools and devices.

Each level handles a set of manufacturing components. To model a manufacturing system in an abstract manner, these manufacturing components can be described by generic elements which are called reception posts.[5] Each reception post is interfaced to a physical location in a manufacturing system and is represented by a machine state graph. It thus informs the manufacturing control system when the component is available, occupied or unavailable (Fig. 1). Moreover, several reception posts can be regrouped in a reception zone if they are handled by a same level (for example, in the case of several pallets on a conveyor).

These concepts of reception post and reception zone provide an external and abstract image of the various manufacturing components of a manufacturing system. Indeed, a manufacturing control system captures a view of each manufacturing

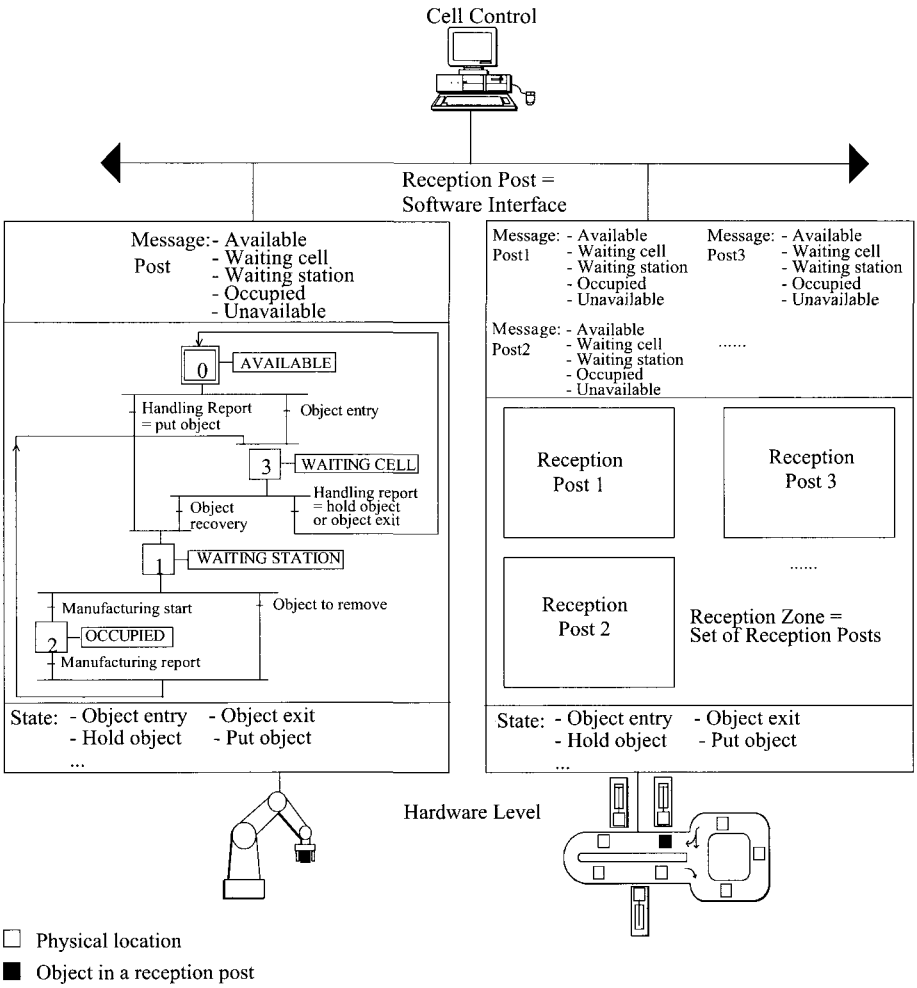


Fig. 1. Concepts of reception posts and reception zones.

component by observing its associated reception zone without knowing its internal functioning.

3. Quality in the Manufacturing Control System

One of the main difficulties in implementing quality in manufacturing control systems is the structuring of information flows. The quality circles are useful if the quality measurements and the improvements are correctly performed, but this condition is not always satisfied for the following tasks:

(i) gathering of the quality data;
(ii) their analysis;

(iii) the communication of proposed corrections; and
(iv) their applications in the manufacturing process.

Quality management methods are clearly defined but are difficult to apply in a coherent way when the manufacturing system is not structured. If the product life-cycle is not totally controlled, the quality function cannot be handled. The following parameters must be safeguarded:

(i) the product design specifications on the process design;
(ii) the design adjustments;
(iii) the qualities relative to the products and processes; and
(iv) the difficulties which occurred during the manufacturing stage.

The computer aided design and manufacturing (CAD/CAM) level and the shop floor level transmit dated objectives and quality to the manufacturing system. The manufacturing system objectives are:

(i) to obtain the required qualities; and
(ii) to report non-quality.

Thus, the control loop architecture joins the PDCA cycle (plan-do-check-action).[6] According to this cycle, the quality control function ensures that every manufacturing task is well performed, and that the required quality of each operation is obtained. The relationships with both CAD/CAM and the manufacturing management include the feedback loops, which reconsider the design specifications, and the planning of work in progress. The corrective actions of the manufacturing cell depend on the CAD/CAM and the manufacturing management capabilities. Inside the cell, the control loop architecture has effects on the manufacturing control system. This must be able to take into account the manufacturing exceptions, and to verify the effects of its orders. Then the non-quality relative to the parts or to the equipment induces reactions. The main problem is to organize these reactions into a coherent architecture, which includes informational and decisional abilities.

Relative to product and process design, the cell receives the design specifications, and then executes them according to the shop floor manufacturing orders. The specifications transmission may use the following CAD/CAM data exchange standards: IGES, CAD*I, SET, PDES, STEP, $\cdots$.[7–11] These standards generally include the quality data in the product and process models, but they do not always consider corrective actions. In keeping with the Action stage of the PDCA cycle, it should indeed be possible to consider corrective actions depending on the manufacturing exceptions and drifts. In other words, the manufacturing process must contain alternative solutions, especially those generated by the process planning but not recorded because they are not optimal. The control loop from the manufacturing cell to the design level involves redesigning the manufacturing process, as well as the product. In this regard, the cell reports any quality data for statistical analysis or for exhaustive analysis. This feedback enables TQM (total quality

management).[12] If the report activity deals with the whole data relative to both parts (dimensions, roughness, etc.) and equipment (tool wear, machine axis accuracy, etc.), the first versions of the Ishikawa diagrams may be improved. Reports like the one by Akman,[13] contain qualitative parameters that only knowledge-based systems or fuzzy logic can process, and these parameters enable redesign to be automated in an interactive manner.

Inside the manufacturing cell, the control system initially proposes a set of functions which process work in progress, equipment, actions and their sequencing. The control function is therefore distributed throughout the whole manufacturing system, and is as close as possible to the equipment level. The control of the parts and equipment must review their design and process plans as often as is necessary. The control of the actions (handling, machining, for instance) must adjust their progress in real-time, if possible. The control of the action monitoring modifies the real-time control of the handling inside the cell, depending on the problems encountered. In the case of a default and in order to avoid the intervention of an operator, the manufacturing control system must be able to interpret any risk, to know which operation has to be corrected, and then determine the corrective action.

4. Computer Control System

4.1. *Manufacturing control system modeling techniques*

Adequate manufacturing control systems must support the extremely distributed, complex, heterogeneous and dynamic nature of manufacturing systems.[14] Therefore, control systems are expected to rise in numbers and in complexity as new advanced technology is introduced in manufacturing systems.[15] In this context, the current practices for manufacturing control systems design are no longer sufficient. In these practices, manufacturing control systems are developed individually after the manufacturing system has been designed. This often results in an extended development period, high cost, inflexible and consequently non-reusable control systems. Such a control system does not support evolution nor adaptation to the manufacturing requirements. To solve these problems, a methodological approach for developing, implementing and maintaining generic, hence reusable, manufacturing control systems is necessary. This approach should serve as the guideline for efficient realization of manufacturing controllers in companies.

Although researchers have employed modeling techniques to analyze and specify manufacturing control systems, the modeling methods that are generally used do not naturally support the specification and reuse of the control systems models.[16] These modeling efforts are generally developed for a particular manufacturing system. The models generated by these methods tend to be application specific and are very difficult to generalize. Hence they cannot sufficiently cope with new requirements for manufacturing systems such as the reconfigurability and the ability to adapt to changes of target manufacturing systems configuration.

Facing these trends, many researches aim at producing consistent models within a framework, which can be used to systemize design and build reusable systems. We can mention about the Esprit project VOICE[17] which in particular deals with model-based operation control, the researches carried out by the MSI Research Institute at Loughborough University entitled "*Model-Driven CIM*",[18,19] and researches carried out by the AR Research Institute at South Fort Worth Texas which develops model-based control techniques capable of driving reconfigurable manufacturing systems.[16]

Most of the current researches which focus on reusable models and systems propose the application of a general architecture for enterprise modeling as an efficient support to provide software reusability for manufacturing systems. A general architecture is a structured plan, a framework on the basis of which a product, a system or an organization of an enterprise can be constructed.[20] Therefore it allows us to identify and to represent, at a high abstraction level, the main components, processes, activities, constraints, information and decision providers, which are necessary for the functioning of systems or organizations.

In this framework, different approaches exist such as the CIM-OSA (open system architecture) architectural framework,[21] the CAM-I DPPM (discrete part manufacturing model), and the NBS AMRF,[22] cover the whole life-cycle of CIM, where flexibility and modularity are the key elements. CIMOSA (computer integrated manufacturing—open system architecture) and the associated methodology,[20,23] as proposed by the ESPRIT Consortium AMICE (European computer integrated manufacturing architecture)[21] are probably one of the most advanced open architecture. CIMOSA is based on a modeling framework which specifies three principles, namely instanciation, derivation and generation. Furthermore, the CIMOSA integrating infrastructure[24] proposes the means to implement its own solution with existing tools or technology. Indeed, it allows, according to the instanciation principle, the execution of particular implementation models, using a set of services (common, information, presentation, management and business) generally used in CIM systems.[25]

Therefore the application of CIMOSA modeling framework in designing manufacturing control systems should allow the development of each new implementation of a control system using the same rules, the same set of information, and the above all should contribute in a coherent way to the existing organizations and applications. The definition of an integrating infrastructure should then allow for its operational implementation and use.

4.2. *Generic modeling of the manufacturing control system*

4.2.1. *Modeling levels*

The ESPRIT Consortium AMICE[21] has already accomplished its mission in the development of the CIMOSA concepts. Thus, the CIMOSA modeling framework provides a set of methods and tools for each view and each modeling level. The

function view adopts a process-based, event-driven approach to describe the functionality and the behavior of the enterprise operation.[23] The information view makes use of three modeling paradigms, namely a semantic object-oriented model, an extended entity-relationship approach based on the M* methodology and finally conventional data modeling techniques like SQL. Regarding the resource view, MMS (manufacturing message specification) has been selected to represent and gain access to the communicating manufacturing resources.[26,34]

Today, projects like VOICE 5510 aim at demonstrating the validity of CIMOSA for the integration and the optimization of manufacturing systems.[33] However, the application of this modeling framework still needs some efforts in the details of specification and in the implementation of technology.[27] In particular, the different proposed methods and tools were not defined to be integrated in a same design approach. Application of the CIMOSA modeling framework and the integrating infrastructure in performing the modeling and implementation of a manufacturing control system is nevertheless very interesting in the purpose of control system reusability.

To achieve the goal of a reusable manufacturing control system, a high abstraction level must be adopted. This requires the definition of a unified conceptual model which covers the functional, informational and organizational aspects. This conceptual model can then serve to derive a particular control model and finally an executable control solution. The conceptual model is independent of any implementation specific characteristics. The material considerations only occur at its operational implementation. A particular model is duplicated from the previous one, either personalized or developed to respond to a particular requirement. Figure 2 summarizes the different modeling levels and the respective models and tools.

Firstly, the design specification level aims at defining the conceptual control model. At this level, the description of the function view can be used for the functional specification phase, the structured analysis and design technique (SADT)[28] or the data processing models as defined in the extended entity-relationship-based MERISE method.[29] The information view deals with the definition of the manufacturing information system by means of the extended entity-relationship-based MERISE method which ensures a unified data representation, and provides communication and data processing models based on Petri nets. This method is founded on three design cycles in order to facilitate the enterprise organizational survey (i.e. the system life, the decisional and the abstraction cycles). These three cycles cover the CIMOSA information and organization views. In particular, the abstraction cycle consists of gradually defining the organization to be studied. It evolves in three analysis stages: conceptual, organizational and operational, which respectively model the organizational performance, the integration of the organizational constraints and the physical implementation of the system.

Finally, the implementation description level aims at generating an executable control model by means of the integrating infrastructure services applied to the MMS services.

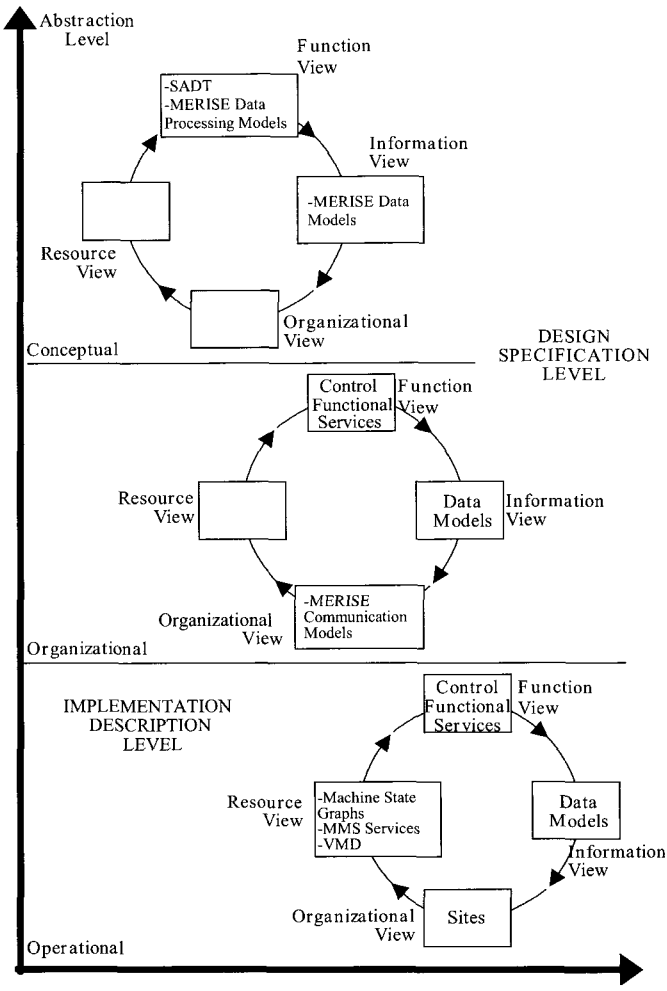


Fig. 2. Modeling levels.

4.2.2. *Information system for the manufacturing control*

The first purpose of the information system is to handle the communications inside the manufacturing system. Models have to define information flows according to both management rules and decisional activities. Thus, these models gather information which is representative of the considered system to ensure that the decisional activities are well informed. Its second purpose is to define the following means:

(i) to be representative of the organization, on how to get information,
(ii) to provide information access facilities to the decisional activities.

In the case of a manufacturing system, information must be available to any function. In other words, the information system integrates the manufacturing system

from an informational point of view. In the case of a control architecture of a manufacturing cell, the information system must deal with the quality information.[30] Therefore, information classes can be classified as follows:

(i) cell and workstation configurations;
(ii) acceptable objects (products, pallets, tools, gaugers, etc.);
(iii) physical and manufacturing capabilities;
(iv) products and manufacturing process models;
(v) feasible fixturing and handling tasks;
(vi) work in progress and available resources;
(vii) performed tasks;
(viii) obtained qualities; and
(ix) corrective actions performed.

Finally, the information system must provide a uniform access to these information classes. The different functions inside the manufacturing system must not be developed taking into account whether some information is distributed or not, or whether more than one network has to be used, etc. In this way, the distributed computer control system is completely independent of the operative system, meaning that the flexibility is further increased. It must have access to any information relative to the qualities, to the manufacturing reports and to the system configuration. This entails the following kinds of resources:

(i) communications;
(ii) information storage; and
(iii) data processing.

These are the three facets of an information system study. For this study, the extended entity-relationship-based MERISE method is perfectly adapted since[29]:

(i) it uses the extended entity-relationship formalism; and
(ii) it provides communications and data processing models based on Petri nets.

4.2.2.1. *Entity-relationship formalism*
The entity-relationship formalism gathers a set of objects, which are:

(i) entity represents concrete or abstract things or concepts of the real-world which can be named; they usually denote a person, a place, a thing or an event of informational interest;
(ii) relationship is an association (or link) among two or more entities;
(iii) attribute is a property or a descriptive characteristic of an entity or a relationship. It is defined by its name and its data type; an attribute can either be an identifier which uniquely identifies information items (entity or relationship), or a single item of information defined on basic data types;

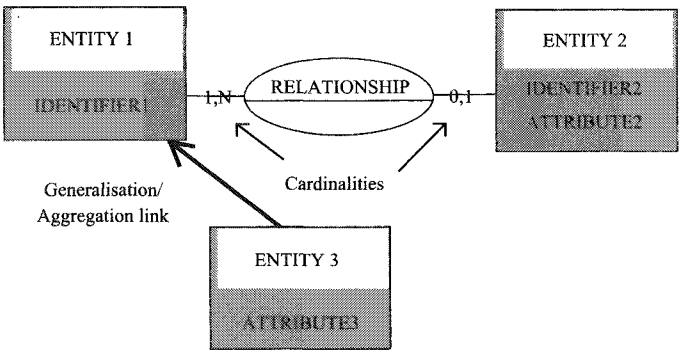


Fig. 3. MERISE graphic representation of entity-relationship objects.

(iv) generalization link allows a class E, named a super-class to be defined as a generalization of a set of class El, E2, $\cdots$, En (called the sub-classes or the specialization of E) if each occurrence of the subclass is also an occurrence of the super-class;
(v) aggregation link allows a class, named as a composite class, to be specified as a composition of existing classes, named as the component classes.

The graphic representation used by the MERISE method is given in Fig. 3.

Communication model

A distributed control solution involves the communications inside the manufacturing system as well as its environment. The communication between two entities always results in a data exchange. The information study by means of the entity-relationship formalism aims at defining the information flows between the different manufacturing functions. The communication study must allow the structuring of the interactions between the different manufacturing functions. The MERISE communication model is used for this purpose.

These communication model gathers a set of objects which are (Fig. 4):

(i) the site corresponds to a physical location where a part of the manufacturing activities is performed and where information is recorded; they can send and receive messages;
(ii) the data model represents the result of a modeling work; they can send and receive messages;
(iii) the message represents information flow between the sites and data model; each message supports the information which are transmitted; and
(iv) the timer gives the frequency of the message sent.

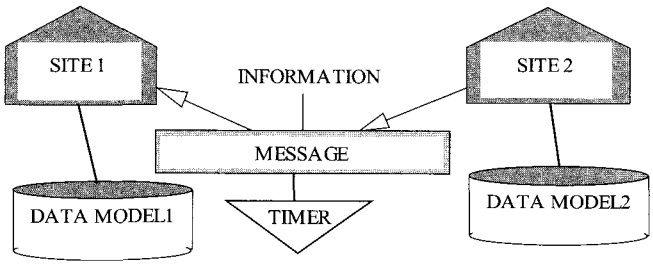


Fig. 4. MERISE graphic representation of communication model objects.

4.2.3. *Distributed control with MMS*

The communication between the different distributed functions of the control system must be performed through a network adapted to industrial applications, i.e. able to operate with real-time constraints. A network based on standard protocols guarantees independence with respect to vendors, and protects against rapid changes. At the application layer ISO (International Standard Organization) has defined the OSI (open system interconnection) model, which allows the heterogeneous equipment interconnection. In this model, the user interacts with the communication system through the application layer, where he can find a set of specialized service groups. One of them, the MMS (manufacturing message specification)[31] has been defined in order to meet the communication requirements in the manufacturing industry. MMS offers services for data communication between manufacturing devices, and provides a common set of commands. This guarantees effective communication between individual components in open systems. At first the MMS was specified and developed for the MAP (manufacturing automation protocol) network. The MAP is in conformity with the OSI model, and it only integrates ISO protocols at each layer.

A MMS model includes for instance, MMS objects, VMD (virtual manufacturing device), MMS services groups, etc.

MMS objects

This standard makes use of the abstract object modeling technique in order to fully describe the MMS device model and the MMS services procedures. The objects (Fig. 5), their attributes, as well as their respective operations are described.

This object modeling technique is a formal tool which helps to understand the intent and the effects of MMS services better than any kind of verbal description.

Semaphore	Program Invocation	Variable and Type	Action and Condition Event	Journal	Domain

Fig. 5. MMS objects.

The relationship between the services and the objects and vice versa is formally described in the standard. On implementing the MMS, a real system maps the concepts described in the model to the real device.

An object is then represented by a data structure. MMS defines a number of object classes. Each object is a class instance and it constitutes an abstract entity which exhibits some specific features and may be affected by some MMS services and operations. For each class a name is given by which it may be referenced.

Each class is characterized by a number of attribute types, which describe some externally visible feature(s) of all objects of this class. Each instance of a class (object) has the same set of attribute types, but has its own set of attribute values. The values of these attributes are either defined by this standard or are established by the MMS services, and hence their effect on the device may be modeled by a change in one or more attribute values of an object (or objects).

Each object must be uniquely identified among all instances of the same class. For this purpose, one or more of the object's attribute values, as a combination, must be unique. In MMS, each attribute, which is a part of this combination of attributes and which makes the object unique, is identified as a key attribute.

Finally, some objects contain attributes which are conditional, in the sense that they are relevant to the object if certain conditions hold. MMS expresses such attributes through the use of a constraint, which specifies a condition. Attributes that are subjected to a constraint are considered to be object attributes for an object if the corresponding constraint is satisfied for that object.

In MMS, classes are syntactically defined as a set of objects as follows:

Object: (name of class)
  Key attribute (name of attribute type (values))
  ...
  Key attribute (name of attribute type (values))
  Attribute: (name of attribute type (values))
  ...
  Attribute: (name of attribute type (values))
  Constraint: (constraint expression)
    Attribute: (name of attribute type (values))
    ...
    Attributes: (name of attribute type (values))

It should be noted that some objects contain attributes which make reference to other objects. Such attributes, called the reference attributes, provide a mechanism to create a linkage from one object to another.

Virtual manufacturing device object

The virtual manufacturing device (VMD) serves as a model to represent the behavior of a real device. A VMD is an abstract representation of the common

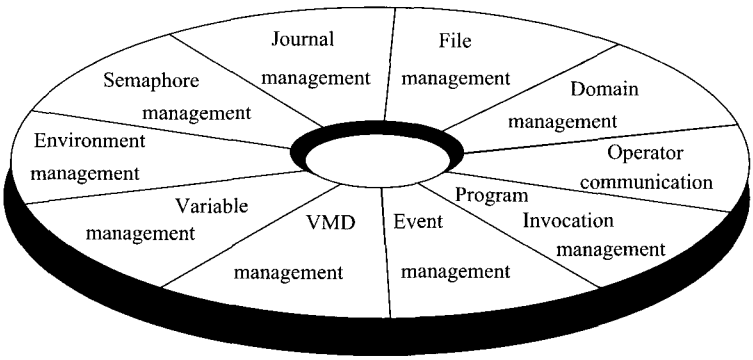


Fig. 6. MMS services.

characteristics of all real manufacturing devices (generalization of manufacturing devices) and it represents the externally (from the network) visible behavior relevant to the MMS. All standardized services refer to this virtual device, which by the implementation has to be mapped to real functions. MMS only describes the effects of the services on the VMD and does not prescribe a specific implementation or transformation to real functions.

MMS services groups
A service group comprises of those services described in the standard which all refer to as the same object or object group. The name of each service is fixed by the standard. A reference to a service therefore consists of the name and the corresponding parameters. In all, a reference to a service makes up the functionality of MMS. Ten service groups exist in the MMS (Fig. 6).

5. Conceptual Stage

The conceptual stage concurs to the modeling of the information and decision processes in a manufacturing control system. This modeling is independent of any application specific characteristics so as to define a control model which can then be used as a basis to derive a particular control model.

To be independent of any implementation, one must distinguish the scope of an application from its utilization. This means that the external description of the manufacturing system components is separated from their internal functioning (the way they are used). From an external point of view, a scope brings out generally two entities: a *client* (the manufacturing system) and a *server* (the manufacturing control system) which provides services for the realization of a global manufacturing goal.

Within a manufacturing system, the reference entities are the manufacturing components. Therefore, the manufacturing control system conceptual model shows how each manufacturing component contributes to the realization of a set of goals

and then, how the control system can reply to these goals. Thus, in a first step, the conceptual model defines the functions and then the data necessary to perform the control and the decision tasks within the manufacturing system. That means that it has to characterize the services offered by the control system in response to the manufacturing goals.

The basic functions of a manufacturing control system are determined by the analysis of the interactions between the different manufacturing components. Thus, each control function handles a set of information flows which are produced by other control functions and generates in turn information flows addressed to other control functions to achieve the global manufacturing goal. They are taking place in the form of messages passed from one control function to another. These messages contain information, arrive at a certain time and are responsive to a specific need or request: so they are events.

These interactions are based on a client/server model. This means that a client-control-function requests for a service by sending a message to a server-control-function. For these purposes, the different messages are called *functional services.* The functional services (FS) correspond to the "primitives" that the control system dispatches to the different manufacturing components and leaves them to these components to completely manage them locally.

Each functional service has to deal with a discontinuous flow of information. Their specification allows us to gather the whole data and functions necessary to perform the control and decision tasks within the CIM system. By this way, the CIM system is characterized by a set of communicating objects which receive, process and send functional services. These functional services are themselves communicating objects with their own protocol. Indeed, functional services always result in a request to be sent to another functional service or in a response to be re-sent to another control functional service. Once a functional service has received a request or a response data, it takes full responsibility of this data. The generic structure of a functional service is illustrated in Fig. 7. This kind of dialogue perfectly conforms to the type of interactions between two CIMOSA functional entities.[32]

The principal function of a manufacturing control system is to manage and to coordinate the overall operations of the different manufacturing components. The functional services are therefore used to determine the appropriate commands to be executed by each manufacturing component. For example, in the context of a flexible manufacturing cell, a manufacturing order to produce a batch of pieces is released from the shop floor level (shop floor manufacturing request FS) where checks are made to ensure that the necessary process plans, NC files, toolings, probes, fixturing elements and raw materials are available within the cell (cell configuration FS). Upon receipt of the order, the cell control generates a production schedule based on a simple priority rule. With the corresponding process plans, the cell control decomposes the schedule into tasks like machining and inspection tasks (scheduling FS). The cell level then dispatches the tasks to the station level (cell manufacturing request FS and station configuration FS). The station level schedules and dispatches

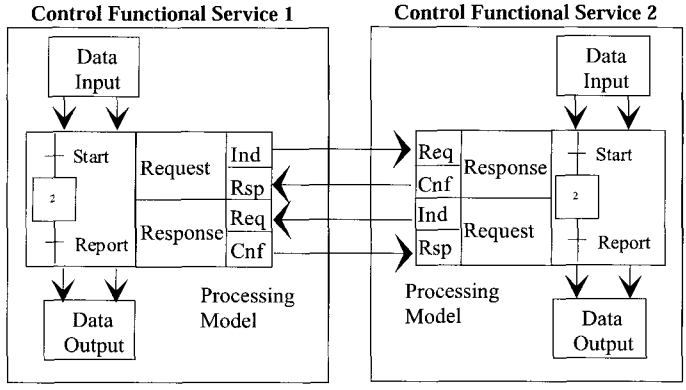


Fig. 7. Structure of a control functional service.

in its turn the tasks to the equipment level (equipment configuration FS and transformation request FS for a machining task execution, qualification request FS for an inspection task execution and handling request FS for a handling operation execution). The manufacturing control system also makes dynamic routing decisions, monitors the progress of the tasks, reports order and manufacturing resource status from one level to another, as well as manages alarms and the various data files required by the different levels. These functions are internal to the controllers (software programs).

6. Organizational Stage

The organizational study allows us to represent and to organize the information produced and used by the control functional services. Thus, each manufacturing component is described by its own particular data structure, independently of its material support. That means that each data structure of a manufacturing component is a general representation of the component: it does not reveal either the type of device used or the details associated with implementing the control functional services.

Each data structure is defined by means of a MERISE data model. The set of defined data models represents the data gathered by the different manufacturing components. The modeling of the informational flow ensures data consistency and integrity within the manufacturing system, independently of the means of supporting these data.

Then, the modeling of organizational communications by means of the MERISE communication models allows us to identify interactions between the different manufacturing components.

Thanks to that, a manufacturing system is constructed as an assembly of a set of sites (computers, numerical controllers, robots, $\cdots$). Each site is described by one or more data models. The messages between the different sites represent the various

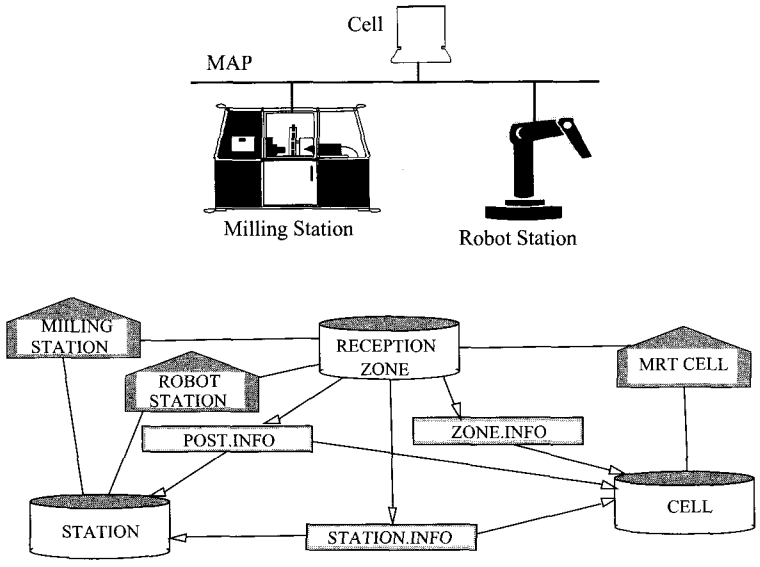


Fig. 8. Example of an application and its communication model.

functional services of the manufacturing control system and they specify how the different manufacturing sites cooperate to reach the realization of the manufacturing operation sequences as defined by process planning. Thus, the communication models gather the information elements which are significant for the manufacturing system configuration: they constitute the conceptual control model of a manufacturing system.[33] A control model for a particular computer integrated manufacturing (CIM) application can then be derived from it. For example, a conceptual control model applied to a flexible manufacturing cell is depicted in Fig. 8. It brings out three sites:

(i) the MRT cell site which is described by the cell data model;
(ii) the milling station site which is described by the station data model; and
(iii) the robot site which is described by the station data model.

7. Operational Stage

The previous modeling approaches (conceptual and organizational stages) aim at facilitating the implementation of a manufacturing system, whatever the manufacturing system configuration or the kind of products it may be. From a conceptual control model, the manufacturing components description must indeed allow the automatic generation of an executable control solution. To ensure a sufficient abstraction level, the previously defined conceptual model does not take into account material specifications and constraints. A manufacturing component is essentially identified through the services it realizes: its internal structure does not intervene.

The description of the real physical component, which is required and/or implemented in a manufacturing system, only occurs at this last stage which corresponds to the CIMOSA implementation description level. To interpret and execute the physical implementation model, the CIMOSA integrating infrastructure provides a common set of services.[32] These services make use of the client-server model.

In this framework, MMS (manufacturing message specification) is used to represent and access the different manufacturing real physical components. This means that each component is modeled as an MMS, virtual manufacturing device (VMD) and the MMS services are used to manage and control the different manufacturing components. Thus, the data structure of each manufacturing component is specified into MMS objects. These MMS objects are directly derived from the previously defined data models. In fact, each data model is translated into a VMD (virtual manufacturing device) which serves as a model to represent the behavior of a real manufacturing device. In the same manner, each control functional service (i.e. a message defined in a communication model) is directly implemented into one or several MMS services. The required MMS services identification is specified from the communication models. These MMS services can then access MMS objects which model the physical manufacturing components and facilitate coordination and communication of the whole connected components. Figure 9 illustrates the implementation of a control functional service into MMS services.

8. Application to a Flexible Manufacturing Cell

8.1. *Description of the FMC*

The approach for developing a manufacturing control system has been applied to a flexible manufacturing cell constituted of one cell controller and three stations (Fig. 10):

(i) a milling station which drives a CNC machine-tool (Charly-Robot) for milling parts;
(ii) a vision station which controls the rough parts; and
(iii) a robot station which ensures the handling operations inside the cell.

The equipment is connected by a MAP network, and we use the MMS-EASE software from SISCO to implement the set of application processes. The interface between the equipment and the network is performed by PCs which run under the multitasking operating system OS/2, except for the robot which supports a MAP card in its controller. The cell has five reception posts which are part fixture assigned to the Charly-Robot machine tool, robot prehensile, rough part vision post, input post of the cell and the output post of the cell. An SQL Server is the relational DBMS, which in particular records any cell state information.

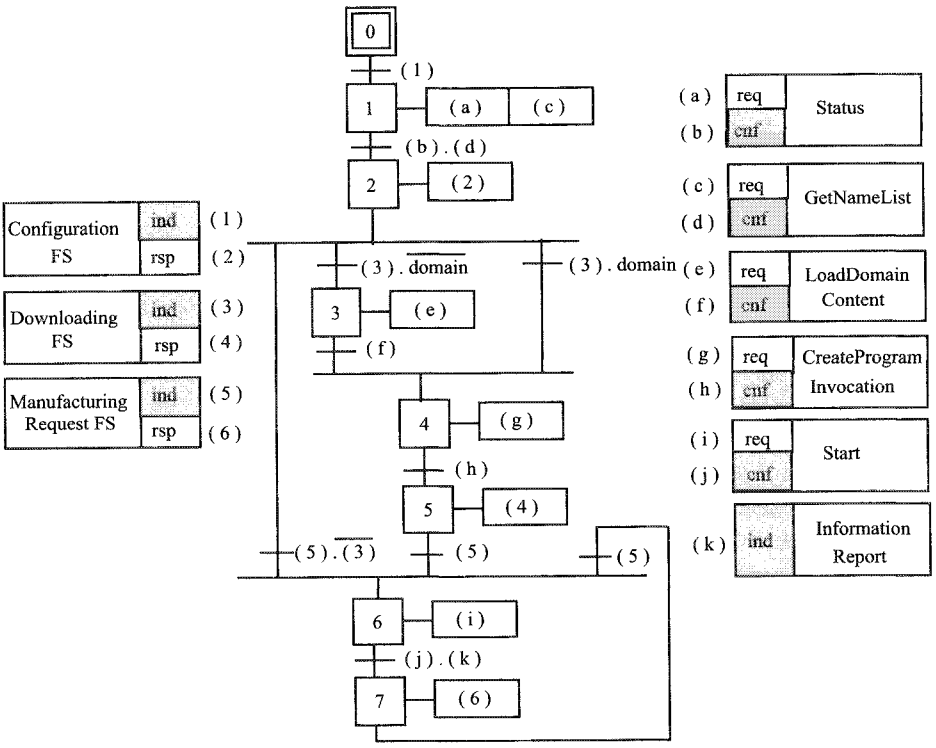


Fig. 9. From control functional services to MMS services.

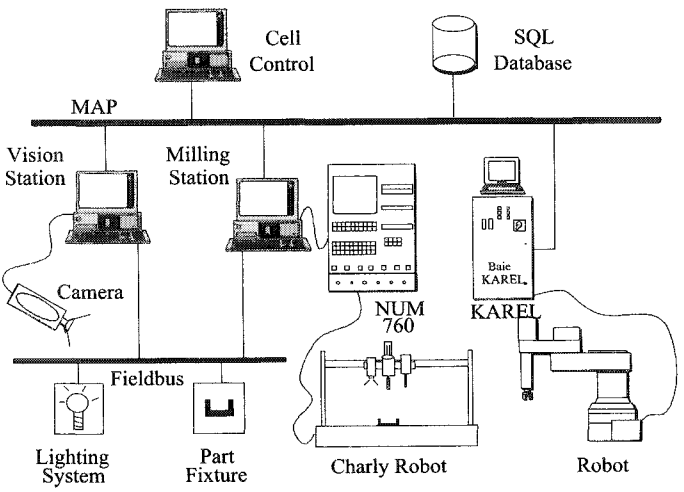


Fig. 10. Description of the flexible manufacturing cell.

8.2. *Design specification of the FMC control system*

The management information handled by the database and the vision VMD models have been individually developed. As regards to the robot and the CNC machine tool VMD, we have generated and implemented parts of them so as to validate the proposed approach. The communication and control of the different components are then realized in implementing the adapted MMS services. For this, we have applied the conceptual control model to describe the cell components and their functions by means of a set of sites and functional services.

The organizational communication model of the cell as depicted by Fig. 11 has been directly derived from the general conceptual control model. Four sites are present, each corresponding to a network node. They are:

(i) the STATION PC site which models the milling station computer. It contains two organizational data models (a station model and a reception zone model which corresponds to the part-fixture).
(ii) the CELL PC site which models both the cell computer, the handling station computer and the vision station computer. So, it is described by a cell data model, a station data model and a reception zone model which corresponds to the rough part vision post.
(iii) the A600 ROBOT equipment site which models the robot. It is described by a Robot VMD.
(iv) the NUM 760 equipment site which models the numerical controller. It is described by a NC VMD.

Arrows which join boxes indicate communications (functional services) between physical entities, and thus the ones which exchange messages can be identified.

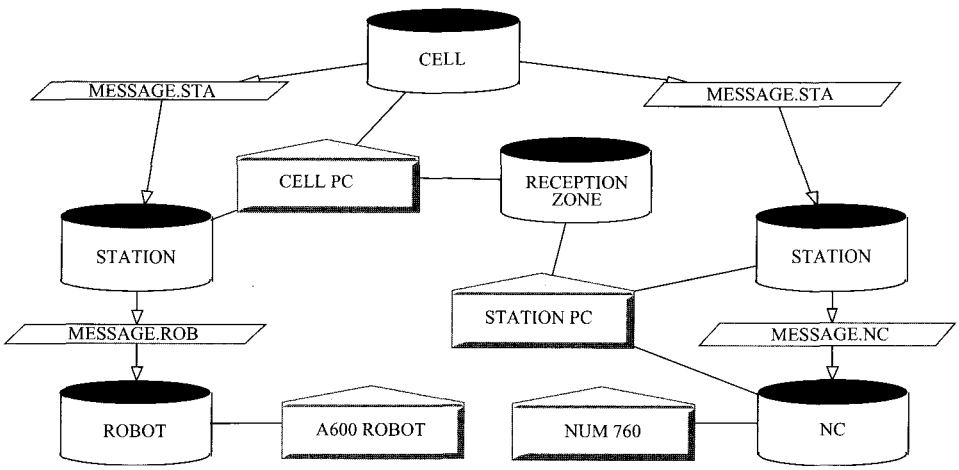


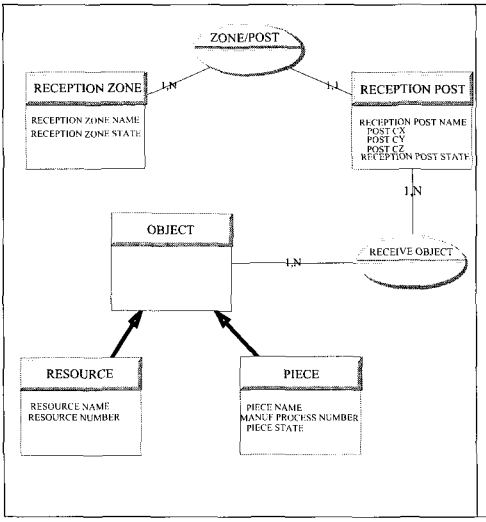
Fig. 11. Organizational communication model of the FMC.

The informational study has previously specified the real data supported by these messages and the induced constraints.

8.3. *Implementation of the FMC control system*

The executable control model allows us to build all the MMS objects handled by the control system. The different cell activities are described by a set of MMS services. The control of these activities only consists of simple services dispatching between the different cell components. Each manufacturing component uses a set of MMS services to communicate with the other virtual manufacturing devices. For example, the cell computer communicates with the milling station computer through the medium of the reception posts. The information needed for modeling the reception post have been previously encapsulated by a reception zone data model. This data model is then translated into a VMD directly implemented on the corresponding handling equipment.

Figure 12 shows a translation example of a reception zone data model into a reception zone MMS object. It defines a reception post domain which corresponds to the presence of a reception post. This domain can be reproduced up to as many as the number of reception posts. For example, a robot with two prehensiles is described by a reception zone VMD which contains two domains. Thus, a prehensile substitution does not change the VMD structure, and the insertion of a new prehensile only corresponds to the addition of a new reception post domain. This action can be simply made for example with the "DOMAIN DOWNLOAD SEQUENCE" service included in MMS.



Object: *VMD*
Attribute: *Reception Zone Name*
Attribute: *Reception Zone State*
Attribute: *Domains List*

Object: *Domain*
Key Attribute: *Domain Name*
Attribute: *Reference*
Attribute: *Post CX*
Attribute: *Post CY*
Attribute: *Post CZ*
Attribute: *Reception post State*
Attribute: *Object Type= (Piece, Resource)*
Constraint: *Object Type= Piece*
Attribute: *Piece Name*
Attribute: *Manuf Process Number*
Attribute: *Piece State*
Constraint: *Object Type= Resource*
Attribute: *Resource Name*
Attribute: *Resource Number*

Fig. 12. Translation of a data model to a MMS object.

Each manufacturing component responds to a set of well-defined services and can in turn request for services from the other cell components to reach the global manufacturing goal. This structure supports its own evolution to adapt to both the evolution of the cell configuration (addition or removal of a component and its control FS), and the situation within the cell (a component out of order or disconnected). No software modification is necessary.

9. Conclusion

In order to support the evolution and adaptation to the manufacturing requirements, generic and reusable manufacturing control system are required. Many researches have established that the application of a general architecture is an efficient support to provide reusability for manufacturing control systems. An approach based on the definition of a generic conceptual model and its implementation for a particular manufacturing system, contributes to systemize design-and-build reusable systems. The central aspect of this approach relies both on its organizational and generic properties which allow us on one hand to perfectly specify the design cycle of a manufacturing control system and on the other hand to make an abstraction of the manufacturing system configuration or reconfiguration. Indeed, it results in a modeling which ensures control system genericity and the harmonization of the presentation and access of the manufacturing components information. It thus facilitates the integration and the addition or removal of new functionalities or new manufacturing components, without reconsidering the existing control system, and in this fact, the reusability of the manufacturing control system. Finally, the interests of this approach are not only a great flexibility in the control system implementation but also in the portability of the implemented control system.

References

1. NISTIR, *Progress Report in Quality In Automation Project FY88*, National Institute of Standards and Technology , Gaithersburg, MD, 1989, 89–4045.
2. N. A. Duffie and R. S. Piper, Non-hierarchical control of manufacturing systems, *Journal of Manufacturing System* **5**, 2 (1991) 137–149.
3. A. Chaudhury and S. Rathnam, Informational and decision processes for flexible manufacturing systems, *IEEE Expert* **6** (1992) 53–62.
4. L. K. Goh, Y. G. Lim and B. S. Lim, Design and implementation of a FMS control and communication system, *International Conference on Computer Integrated Manufacturing, ICCIM'91*, Singapore, October 2–4 (1991) 319–322.
5. Z. Idelmerfaa, J. Richard and E. Bajic, Integrated approach for manufacturing control system design, *International Conference on Industrial Engineering and Production Management, IEPM'95*, Marrakech, Morocco, April 4–7 (1995) 183–192.
6. W. E. Deming, *Out of Crisis* (MIT Press, Cambridge, MA, 1986).
7. NIST, *Initial Graphics Exchange Specification V4.0*, National Institute of Standards and Technology, U.S. Department of Commerce, Gaithersburg, MD, 1988.
8. I. Bey and U. Gengenbach, The CAD*I interface for solid model exchange, *Computer & Graphics* **12**, 2 (1998) 181–190.

9. G. Stil, S.E.T. Standard for exchange and data transfer, *3rd International Conference on CAD-CAM-CAE Integration Technologies in the Automative Industry*, Torino, Italy, 1988.
10. NTIS, *Product Data Exchange Specification, First working draft*, National Institute of Standards and Technology, U.S. department of Commerce, Gaithersburg, MD, 1988.
11. ISO TC 184/SC/N83, Standard for the exchange of product data model, draft proposal, *International Organization for Standardization* (1989).
12. K. Ishikawa, *Le TQC ou la qualitè à la Japonaise*, Collection AFNOR Gestion, Edition Eyrolles, 1984.
13. V. Akman, P. T. Hagen and T. Tomiyama, A fundamental and theoretical framework for an intelligent CAD system, *Computer Aided Design* **22**, 6 (1990) 352–367.
14. W. B. Hadj-alouane, J. K. Chaar and A. W. Naylor, Developing control and integration software for flexible manufacturing systems, *Journal of Systems Integration* **1** (1991) 7–34.
15. H. J. Lynggaard, K. Siggaard and L. Alting, A generic software design for cell control systems, *Third International Conference on Flexible Automation and Integrated Manufacturing*, Limerick, Ireland, (1993) 326–335.
16. B. L. Huff, V. K. Varner and D. H. Liles, Model-based control of a dynamically reconfigurable assembly system, *Third International Conference on Flexible Automation and Integrated Manufacturing*, Limerick, Ireland, June 1993, 602–612.
17. VOICE Consortium, *Voice Architecture of the IIS*, Voice Open Workshop, Berlin IPK, September 14, 1993.
18. M. W. Aguiar, I. Coutss and R. H. Weston, Model enactment as a basis for rapid prototyping of manufacturing systems, *European Workshop on Integrated Manufacturing Systems Engineering, IMSE'94*, Grenoble, France, 1994, 86–96.
19. R. H. Weston and J. M. Edwards, A model-driven toolset for flexibly integrating manufacturing systems, *European Workshop on Integrated Manufacturing Systems Engineering, IMSE'94*, Grenoble, France, 1994, 441–450.
20. K. Kosanke, CIM-OSA: its role in manufacturing control, *Proceeding of the 11th triennial world congress of the International Federation of Automatic Control*, USSR, August 13–17 (1990) 309–313.
21. Amice Consortium, *CIMOSA: Open System Architecture for CIM*, 2nd revised and extended version (Springer-Verlag, Berlin, 1993).
22. G. J. Olling, *CIM Status and Direction in the USA CAPE'91*, Bordeaux, France (1991) 33–40.
23. F. B. Vernadat, CIMOSA: Enterprise modeling and enterprise integration using a process-based approach, *Information Infrastructure Systems Manufacturing*, eds. H. Yoshikawa and J. Goosenaerts, Amsterdam, North-Holland (1993) 65–84.
24. B. Querenet, The CIM-OSA integrating infrastructure, *Computing and Control Engineering Journal* (1991) 118–125.
25. J. Vlietstra, CIMOSA: Integrating the production, *International Federation of Information Processing, Towards World Class Manufacturing 1993*, Phoenix, USA (1993) 1–20.
26. M. Didic, Rapid prototyping for MAP/MMS based CIM-OSA environments, *Third International Workshop on Rapid System Prototyping*, North Carolina, USA, June 23–25 (1992).
27. W. N. Hou and H. Trauboth, An approach to the development of the machine front-end services in a CIM-OSA environment. *Third International Conference on Flexible Automation and Integrated Manufacturing*, Limerick, Ireland, June (1993) 115–124.
28. D. T. Ross, Application and extensions of SADT. *IEEE Computer* (1985) 25–34.

29. Y. Tabourier, *De L'autre Côté de Merise, Systèmes D'information et Modèles D'entreprise* (Les Editions d'organisation, Paris, 1986).
30. J. Richard, Z. Idelmerfaa, F. Lepage and V. Veron, Quality control in a flexible manufacturing cell, *Annals of the CIRP* **41**, 1 (1992) 561–564.
31. ISO 9506/1, Manufacturing message specification (MMS) — service definition, *International Organization for Standardization* (1988).
32. M. Klittich, CIM-OSA part 3: CIM-OSA integrating infrastructure — the operational basis for integrated manufacturing systems, *International Journal of Computer Integrated Manufacturing* **3–4** (1990) 168–180.
33. Z. Idelmerfaa and J. Richard, CIM systems modeling for control system re-usability, *International Journal of Computer Integrated Manufacturing* **11**, 3 (1998).
34. M. Didic, CIMOSA model creation and execution for a casting process and a manufacturing cell, *Computers in Industry, Special Issue on CIM Architectures* **24**, 2–3 (1994) 237–247.
35. VOICE Consortium, *Computers in Industry, Special Issue on Application and Validation of CIMOSA* (Spring-Verlag, Berlin, 1995).

CHAPTER 5

RAPID PROTOTYPING TECHNOLOGIES AND LIMITATIONS

CHEE KAI CHUA and SIAW MENG CHOU
School of Mechanical and Production Engineering,
Nanyang Technological University,
Nanyang Avenue,
Singapore
E-mail: mckchua@ntu.edu.sg and *msmchou@ntu.edu.sg*

Each rapid prototyping (RP) process has its special and unique advantages and disadvantages. The chapter presents a state-of-the-art study of RP technologies and classifies broadly all the different types of rapid prototyping methods. Subsequently, the fundamental principles and technological limitations of different methods of RP will be closely examined. Comparison of the present and ultimate performance of the rapid prototyping processes will be made so as to highlight the possibility of future improvements for a new generation of RP system.

Keywords: Limitations; rapid prototyping; resolution; speed.

1. Introduction

The first rapid prototyping (RP) system, the 3D systems' Stereolithography Apparatus (SLA), made its commercial debut in 1987. Since then, many commercial systems using various technologies are now available. These include Selective Laser Sintering (SLS), Solid Ground Curing (SGC), Laminated Object Manufacturing (LOM), 3-Dimensional Printing (3DP), Fused Deposition Modeling (FDM), Solid Creation System (SCS), Solid Object Ultraviolet-laser Plotter (SOUP), Selective Adhesive and Hot Press (SAHP), Multi-Jet Modeling system (MJM), Direct Shell Production Casting (DSPC), Multiphase Jet Solidification (MJS) and Ballistic Particle Manufacturing (BPM), etc.

RP can be defined as a layer by layer fabrication process, with the exception of holographic techniques. It is used to build a three dimensional (3D) object from a 3D CAD data. The CAD system converts solid or surface model to a .STL file.[1,2] The .STL file is a list of triangular facets representing the surfaces of an object to be built together with a unit normal vector associated with the outer surface of each triangle. Facets are created by a process called "tessellation" which generates triangles that approximate the object surface described in the CAD solid model. This faceted file is then loaded into a RP system to build the model.

For most RP systems, the building is fully automated. Thus, the operator can leave the machine on to build the model overnight. The building process may take several hours depending on the size and the number of parts required. The RP system's computer will analyse the .STL file, slice the model into cross-sections and, depending on the system used, create the support for the building process. The cross-sections are recreated through the solidification of either liquids or powders, or the fusing of solids, layer by layer to form the 3D model. Finally, after the model is built, depending on the system, post-processing will be required for cleaning, removal of supports, sanding, painting, post-curing, etc.

Over the past few years, there have been many articles and works published in books, conferences and journals in the area of rapid prototyping. Besides a book written by Johnson,[3] a few works have actually looked into a unified description of all RP technologies based on their fundamental principles. Ippolito *et al.*[4] made use of SLA, SGC, SCS, FDM and LOM to investigate and compare their dimensional accuracy and surface finish. Dickens[5] gave a brief qualitative and historical overview of existing developments in RP technology in terms of their techniques and applications.

The purpose here is to address the above shortfall by reviewing the limitations of these RP systems according to each of its fundamental principle. A qualitative and quantitative assessment is also provided.

2. Description and Classification of Different Rapid Prototyping Processes

Fundamentally, the development of RP can be seen in four primary areas, mainly input, methods, materials and applications as stated by Chua *et al.*[6] This is depicted in Fig. 1. There are many different ways of classifying RP systems according to the different methods used.

Burns[7] classified RP processes into additive and hybrid processes. Under the additive process, it is further categorized under different techniques used by the RP systems such as laser curing, masked-lamp curing, laser sintering and droplet deposition. Burns used the additive process to describe the RP process since the object is built by successively adding raw material in particles or layers to create a solid volume of the desired shape. Adhesion of cut sheet such as LOM is a hybrid subtractive/additive process because the contour of the cross-section and the unwanted parts are cut by laser after being bonded to the previous layer.

Jerome[3] classified the RP processes according to the method of controlling layer fabrication. The interaction of raw material mass m and energy W produces the physical layer in a total variation occurring as:

$$\delta(mW) = m\delta W + W\delta m. \tag{1}$$

The first term represents a process where a uniform mass m is selectively activated, removed, or bonded by a variable energy δW controlled by the layer description.

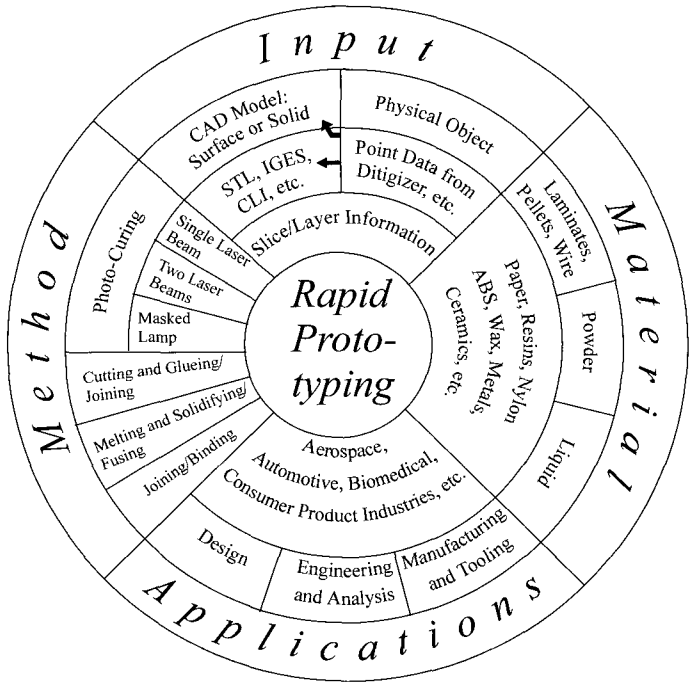


Fig. 1. Rapid prototyping wheel depicting the four major aspects of RP.

Molecular bonding, particle bonding and sheet lamination can be classified under this variable energy process. The second term represents a process where layer information controls a variable mass δm acted on by a control energy W. Droplet deposition, particle deposition and melt deposition are classified under this variable mass process.

Kochan and Chua[8] categorized the RP process by their initial material state and method. The RP processes are grouped under solid, liquid and powder. Under each group, the RP processes are further classified into the different method adopted which includes single or dual laser beams, lamps, holography, masked lamp, cutting and gluing/joining, melting and solidifying/fusing and joining/binding.

RP systems can be classified using similar method adopted by Kochan and Chua[8] as shown in Table 1. However, only commercialized RP systems are considered.

3. Principles of Different Systems

3.1. *3D system's SLA*

The SLA process is based fundamentally on the following principles[9]:

(i) Parts are built from a photo-curable liquid resin that solidifies when sufficiently exposed to a laser beam (basically undergoing the photopolymerization process) which scans across the surface of the resin.

Table 1. Classification of RP process by method.

Variable Energy					Variable Mass		
Molecular Bonding (Photopolymerization)		Particle Bonding					
Laser Curing	Masked Lamp	Sinter Bonding	Adhesive Bonding	Sheet Lamination	Droplet Deposition	Particle Deposition	Melt Deposition
3D System's SLA	Cubital's SGC	DTM's SLS	MIT's 3D Printing	Helisys' LOM	3D Systems' MJM	No commercial system	Fraunhoner's MJS
Teijin Seiki's Soliform			Soligen's DSPC	KIRA's SAHP	BPM Technology's BPM		Stratasys' FDM
Mitsubishi's SOUP				Kinergy's Zippy System			
EOS's Stereos System							
Meiko's RP System							

(ii) The building is done layer by layer, each layer being scanned by the optical scanning system and controlled by an elevation mechanism, which lowers at the completion of each layer.

The first principle deals mostly with photocurable liquid resins, which are essentially photopolymers and the photopolymerization process. The second principle deals mainly with the CAD data, the laser, and the control of the optical scanning system as well as the elevation mechanism.

3.2. *Teijin Seiki's Soliform*

The Soliform creates models from photo-curable resins based essentially on the principles described in SLA.[6] The resin developed by Teijin is an acrylic-urethane resin with a viscosity of 40 000 centipoise and a flexural modulus of 52.3 MPa, compared to 9.6 MPa for a grade used to produce conventional prototype models.

Parameters that influence performance and function are generally similar to the SLA, but for Soliform, the properties of the resin and the accuracy of the laser beam are considered more significant.

3.3. *Mitsubishi's SOUP (solid object ultraviolet-laser plotter)*

The SOUP system is based on the laser lithography technology, which is similar to the SLA.[6] The main trade-off is in the scanning speed and consequently, the building speed. Parameters that influence performance and functionality are the galvanometer mirror precision for the machine, the laser spot diameter, the slicing thickness and the resin properties.

3.4. *EOS's STEREOS system*

The fundamental fact of the STEREOS system is that a photo-curable liquid resin is cured in layers by a computer-controlled laser to create 3-dimensional plastic models directly from the CAD data without tooling.[6] This is based very much on the principles similar to the SLA.

Parameters that affect the performance and function are also similar to the SLA. In particular, they are more dependent on the accuracy of the laser, the properties of resins and the complexity of the part.

3.5. *Meiko's RP system*

The fundamental principle behind the method is the laser solidification process of photo-curable resins.[6] As with other liquid-based systems, its principles are similar to the SLA. The main difference is in the controller of the scanning system. Meiko uses an XY (plotter) system with NC controller instead of the galvanometer mirror scanning system.

Parameters that influence the performance and functionality of the system are the properties of resin, the diameter of the beam spot, and the XY resolution of the machine.

3.6. *Cubital's SGC (solid ground curing)*

Cubital's RP technology creates highly physical models directly from computerized 3D data files.[6] Parts of any geometric complexity can be produced without tools, dies or molds.

The process is based on the following principles:

(i) Parts are built, layer by layer, from a liquid photopolymer resin that solidifies when exposed to UV light. The photopolymerization process is similar to the SLA, except that the irradiation source is a high power collimated UV lamp and the image of the layer is generated by masked illumination instead of optical scanning of a laser beam. The mask is created from the CAD data input and then 'printed' on a transparent substrate (the mask plate) by a non-impact ionographic printing process, a process similar to the Xerography process used in photocopiers and laser printers (Ref. 10, Chapter 2). The image is formed by depositing black powder, a toner which adheres to the substrate electrostatically. This is used to mask the uniform illumination of the UV lamp. After exposure, the electrostatic toner is removed from the substrate for reuse and the pattern for the next layer is similarly 'printed' on the substrate.
(ii) Multiple parts may be processed and built in parallel by grouping them into batches (runs) using Cubital's proprietary software.
(iii) Each layer of a multiple layer run contains cross-sectional slices of one or many parts. Therefore, all slices in one layer are created simultaneously. Layers are created thicker than desired. This is to allow the layer to be milled precisely to its exact thickness, thus giving overall control of the vertical accuracy. This step also produces a roughened surface of cured photopolymer, assisting the adhesion of the layer next to it. The next layer is then built immediately on the top of the created layer.
(iv) The process is self-supporting and does not require the addition of external support structures to emerging parts since continuous structural support for the parts is provided by the use of wax, acting as a solid support material.

3.7. *DTM's SLS (selective laser sintering)*

The SLS process is based on the following two principles[6]:

(i) Parts are built by sintering when a CO_2 laser beam hit a thin layer of powdered material. The interaction of the laser beam with the powder raises the temperature to the point of melting, resulting in particle bonding, fusing the particles to themselves and the previous layer to form a solid.

(ii) The building of the part is done layer by layer. Each layer of the building process contains the cross-sections of one or many parts. The next layer is then built directly on top of the sintered layer after an additional layer of powder is deposited via a roller mechanism on top of the previously formed layer.

The density of the packing of particles during sintering will have a profound effect on the results of bonding and consequently on the mechanical properties of the model. In the studies of particle packing with uniform sized particles[3] and particles used in commercial sinter bonding,[10] the packing densities are found to range typically from 50% to 62%. Generally, the higher the packing density, the better would be the expected mechanical properties. However, it must be noted that scan pattern and exposure parameters are also major factors in determining the mechanical properties of the part.

3.8. *3D systems' MJM (multi-jet modeling)*

The principle underlying the Actua^{TM} 2100 is the layering principle used in other RP systems and the new MJM process.[6] MJM builds models using a technique akin to inkjet or phase-change printing, applied in three dimensions. A 'print' head comprising of 96 jets oriented in a linear array builds models in successive layers, with each jet applying a special thermopolymer material only where required. The MJM heads shuttle back and forth like a line printer (X-axis), building a single layer of what will soon be a 3-dimensional concept model. If the part is wider than the MJM head, the platform (Y-axis) will continue building that layer. When the layer is completed, the platform is distanced from the head (Z-axis) and the head begins building the next layer. This process is repeated until the entire concept model is completed.

The main factors that influence the performance and functions of the Actua^{TM} 2100 are the thermopolymer materials, the MJM head, and the X, Y and Z controls.

3.9. *Soligen's DSPC (direct shell production casting)*

The principle of Soligen's DSPC is based on three-dimensional printing (3DP), a technology invented, developed and patented by the Massachusetts Institute of Technology (MIT).[6] 3DP is licensed exclusively to Soligen on a worldwide basis for the field of metal casting.

In the process, the parameters that influence performance and function are the layer thickness, the powder's properties, the binders and the pressure of the rollers.

3.10. *MIT's 3DP (3-dimensional printing)*

3DP creates parts by a layered printing process and adhesive bonding, based on sliced cross-sectional data.[6] A layer is created by adding another layer of powder. The powder layer is selectively joined, where the part is to be formed, by 'inkjet'

printing of a binder material. The process is repeated layer by layer until the part is completed.

As described in the SLS process, the packing density of the powder particle has a profound impact on the results of the adhesive bonding, which in turn affects the mechanical properties of the model. Like powders used on the SLS, packing densities ranges from 50% to 62%.[10] When the ink droplet impinges on the powder layer, it forms a spherical aggregate of binder and powder particles. Capillary forces will cause adjacent aggregates, including that of the previous layer, to merge. This will form the solid network which will result in the solid model. The binding energy for forming the solid comes from the liquid adhesive droplets. This energy is composed of two components, one its surface energy and the other its kinetic energy. As this binding energy is low, it is about 10^4 times more efficient than sinter binding in converting powder to a solid object.[3]

Parameters that influence the performance and functions of the process are the properties of the powder, the binder material, and the accuracy of the XY table and Z-axis controls.

3.11. *Helisys' LOM (laminated object manufacturing)*

The LOM process is based on the following principles[6]:

(i) Parts are built, layer by layer, by laminating each layer with paper or other sheet-form materials and a CO_2 laser cuts the contour of the part on that layer.
(ii) Each layer of the building process contains the cross-sections of one or many parts. The next layer is then built directly on top of the laser-cut layer.
(iii) The Z-control is activated by an elevation platform, which lowers when each layer is completed and the next layer is then laminated and is ready for cutting. The Z-height is then measured for the exact height so that the corresponding cross-sectional data can be calculated for that layer.
(iv) No additional support structure is necessary as the excess material, which are crosshatched for later removal acts as the support.
(v) When the part is completed, it is removed from the platform and undergoes a brief postprocessing to extract the part.

The parameters which influence performance and functionality of the parts are the consistency and the thickness of the sheet material, the laser used, the laser scanning speed, and the lamination pressure, temperature and speed.

3.12. *KIRA's SAHP (selective adhesive hot press)*

The SAHP process is based on photocopy principles, conventional mechanical layering and cutting techniques.[6] A typical laser stream printer is used for printing and resin powder instead of print toner is used as toner which is applied to the paper in the exact position indicated by the section data to adhere the two adjacent layers of paper.

Cutter plotter, temperature and humidity are the three factors that affect the accuracy of the model being built. The accuracy of the cutter plotter affects the accuracy of the model in the X and Y direction. The shrinkage of the model occurs when the model is cooled down in the hot press unit. The expansion of the model occurs when the model is exposed to varying humidity conditions in the hot press unit.

3.13. *Kinergy's Zippy system*

The Zippy system can automatically build the model by layering and laser cutting principles, similar to LOM.[6] After the special chemically treated high temperature-resistant paper is supplied, the heated roller moves reciprocally across the paper to bond the paper to the top of the model stack. The laser movement is controlled by the Cartesian robot with mirrors which reflect a beam from a CO_2 laser and lens which focuses the beam on the upper surface of the laminated stack. Scrap pieces remain on the platform as the part is being built.

The temperature and pressure of the heated roller and the accuracy of the Cartesian robot are factors that influence the performance and functions of the system. The control of temperature is critical in fabrication. High temperatures would cause the distortion of the model whereas low temperatures would result in poor paper adhesion. Roller pressure is also another important factor. If the pressure is not high enough, it will allow the formation of air bubbles. On the other hand, if the pressure is too high, distortion will occur. The accuracy of the Cartesian robot affects the cutting accuracy of the machine.

3.14. *BPM Technology's BPM (ballistic particle manufacturing)*

The BPM Technology's ballistic particle manufacturing uses the droplet deposition principle, by directing the tiny droplets (76 microns in diameter) of the thermo-plastic material ejected from a robotically controlled ejector head to the desired position where they are needed.[6,11,12] The molten droplets soften the material of the previous layer and cause them to bind together as they solidify.

3.15. *Fraunhoner's MJS (multiphase jet solidification)*

The basic concept of the MJS process is comparable to the fused deposition modeling process (FDM) with regards to the deposition of low viscosity molten material layer by layer with a nozzle system.[6] The main differences between the two processes are in the raw material used to build the model and the feeding system. For the MJS process, the material is supplied in different phases using powder-binder-mixture or liquefied alloys instead of using material in the wire-form. As the form of the material is different, the feed and nozzle system is also different.

In the MJS process, parameters that influence its performance and function are the layer thickness and the feed material. The variables include liquefied alloys (usually low melting-point metals) or powder-binder-mixture (usually materials with high melting point), chamber pressure, machining speed (build speed), jet specification, material flow and operating temperature.

3.16. *Stratasys' FDM (fused deposition modeling)*

The principle of the FDM is based on the surface chemistry, thermal energy and layer manufacturing technology.[6] The material in filament (spool) form is melted in a specially designed head, which extrudes it on the model. As it is extruded, it is cooled and thus solidifies to form the model. The model is built layer by layer, like the other RP systems.

The parameters which affect the performance and functions of the system are material column strength, material flexural modulus, material viscosity, positioning accuracy, road widths, deposition speed, volumetric flow rate, tip diameter, envelope temperature, and part geometry.

4. Fundamental Limitations of Different Technologies

4.1. *Molecular bonding (photopolymerization)*

There are two basic types of liquid-based commercial machines that use photopolymerization processes, namely: laser curing and masked lamp. The first uses a deflected laser beam to irradiate a thin polymer layer at the surface of a vat filled with liquid photopolymer resin. The irradiated areas of the photopolymer react chemically, turning into solid, and the performance of which is dependent on the energy and wavelength of the incident radiation. For example, the 3D systems' SLA uses an ultraviolet laser to solidify the epoxy resin. The second type of machines uses a masked illumination, instead of a point-by-point method, to irradiate a complete layer of polymer at a time. An example of this masked lamp technique is the Cubital's SGC.

Photopolymer resins are mixtures of simple low molecular weight monomers capable of chain reactions, forming solid long-chain polymers when activated by radiant energy of specific wavelength range. Photoinitiators are added to act as efficient energy absorbers, releasing catalysts that initiate the polymerization of the monomers to form high molecular weight polymer. The photopolymerization process is schematically represented by Fig. 2.

Photopolymerization is polymerization initiated by a photochemical process whereby the starting point is usually the induction of energy from the radiation source.[9] At least five different photopolymerization processes are known, characterized by the catalyst used in the chain reaction. They are: free radical polymerization, cationic polymerization, anionic polymerization, condensation polymerization, and addition polymerization.

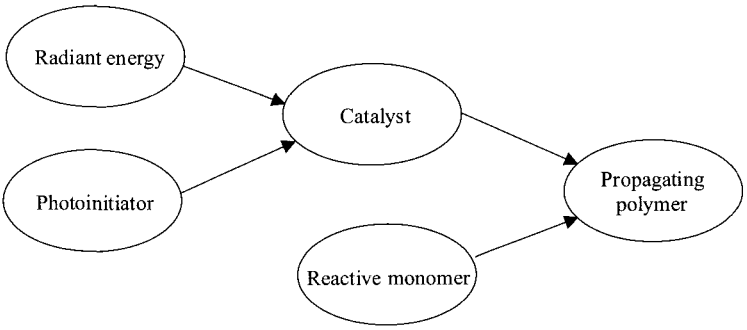


Fig. 2. Schematic diagram of photopolymerization.

Free radical polymerization has been the dominating mechanism used. Incident photons pass through the liquid polymer layer. They are then absorbed, and free radicals are created in a few picoseconds. The free radicals react with the monomer molecules but most are quenched by the oxygen molecules present in the resin through diffusion from the air in the chamber. Hence, significant polymerization requires overcoming the oxygen by the generation of excess free radicals through incident energy exposure.

Cationic polymerization monomers are used based on epoxy[13] or vinylether resin families. Epoxies are attractive because of their higher mechanical strength and lower volumetric shrinkage compared to the acrylate systems. Water and other hydroxyl compounds are inhibitors. The lower catalyst formation rate of cationic polymerization requires a higher radiation energy to achieve the same polymerization rate as the acrylate-based polymer.

Specific energy is used to described photopolymer photosensitivity: it is the amount of radiant energy exposure to solidify a unit of photopolymer resin. Photopolymers with high values of specific energy require more energy to solidify. Specific energy for stereolithography is expressed by[3]:

$$W^* = W_c'/l_c \exp(l_c/l_p), \tag{2}$$

where W_c' is the threshold curing exposure for the transition of photopolymer resin from the liquid to the solid phase, l_c is depth of polymer curing and l_p is the penetration depth, which is the depth of resin that will reduce the exposure to $1/e$ of the incident exposure.

Object fabrication speed[3] is expressed as a linear dimension built per unit time as:

$$\text{Object fabrication speed} = l_l/(t_{\text{image}} + t_{\text{reset}}) \tag{3}$$

where l_l is the layer thickness, t_{image} is the image time per layer, and t_{reset} is the sum of the total layer recoating time, the computer slicing time and the total support material placement time, divided by the total number of layers.

Imaging time per layer t_{image} is determined by the resin photosensitivity, the illumination power P, the layer area A_l, and certain imaging method parameters W_i'. For the masked lamp technology,[3]

$$t_{\text{image}} = W_i' A_l / P, \tag{4}$$

$$\text{Objection fabrication speed for masked lamp} = l_l / (W_i' A_l / P + t_{\text{reset}}). \tag{5}$$

For the laser technology that uses raster scan,[3]

$$t_{\text{image}} = W_i' A l A_{\text{spot}} s^2 / P, \tag{6}$$

where A_{spot} is the smallest dot area accomplished by the laser to solidify the resin and s is the addressibility or the reciprocal of the distance between the centre of two adjacent dots. Hence,

$$\text{Object fabrication speed for laser} = l_l / (W_i' A l A_{\text{spot}} s^2 / P + t_{\text{reset}}). \tag{7}$$

The minimum resolution volume Vr of laser curing and masked lamp processes is based on the minimum feature area and the minimum layer thickness. For laser curing process,[3]

$$\begin{aligned} Vr &= \pi/4 \cdot (\text{energy beam diameter})^2 \cdot (\text{layer thickness}) \\ &= \pi d r^2 l_l / 4. \end{aligned} \tag{8}$$

The Mark 1000 Laser Modeling System[14] has a beam diameter of 89 μm and a layer thickness of 51 μm, producing a resolution ($R = 1/Vr$) of 3152 elements per mm^3. For masked imaging process such as the Cubital's SGC,[15] there exists a minimum feature size of 0.38 mm and a minimum layer thickness of 0.1 mm, producing a resolution of 69 elements per mm^3.

Ikuta *et al.*[16] managed to optimize the UV beam and other optical factors in order to obtain a minimum size of $5 \times 5 \times 3$ μm of harden polymer (so called the "5 μm rule in three dimensional space"). Objects with a resolution of 1.33×10^7 element per mm^3 is possible.

For the mask lamp technology, Kochan and Hovtun[17] stated that the limiting factor of the smallest feature to be drawn by the SGC process depends on the fragile nature of the thin section. The measured average horizontal surface roughness is 4.22 μm (Ra). Bad surface finish can be caused by damages of the in-process milling cutters and an imperfect balance of the milling plate. Thus, these parameters have to be controlled periodically.

4.2. *Sinter bonding*

For laser sintering, parts are built by sintering when an infrared laser beam hits a thin layer of powdered material such as wax, polycarbonate, nylon, fine nylon and metal.[18,19] The interaction of the laser beam with the powder raises the temperature to the glass transition temperature which is below the point of melting, resulting in

particles bonding, fusing the particles to themselves and the previous layer to form a solid. Infrared laser is preferred because of its higher power density.

The sintering energy per unit area Ws'^3 needed to raise a mass of powder particles above the glass transition temperature is given by:

$$Ws' = \rho c l_l \delta T, \tag{9}$$

where l_l is the total thickness of unsintered toner layer and δT is the temperature rise from ambient to polymer sintering temperature. The rough approximation of the sintering energy for a 100 μm thick polymer powder layer using the constant polymer properties from Ref. 18 is $Ws' = 1.68\,\mathrm{J/cm^2}$. The particle sintering requires 300 to 500 times more energy than photopolymerization.

From Ref. 3,

$$t_{\text{imaging}} = Ws' A_l A_{\text{spot}} s^2 / p. \tag{10}$$

Hence,

$$\text{the object fabrication speed} = l_l / (Ws' A_l A_{\text{spot}} s^2 / p + t_{\text{reset}}). \tag{11}$$

The resolution of laser sintering is the same as Eq. 8. The powder particles are typically in the range of 80 to 120 μm. The minimum energy beam diameter of 16 μm[20] is able to produce an estimated resolution of 244000 elements/mm^3.

4.3. *Adhesive bonding*

Three-dimensional parts are created by a layered printing process as well as adhesive bonding, based on sliced cross-sectional data. The powder layer is selectively joined where the part is to be formed by "ink-jet" printing of a binder material. The process is repeated layer by layer until the part is completed. Examples of such technology are 3DP and DSPC.

The droplet energy is composed of its surface energy and kinetic energy as shown by Eq. 3:

$$\text{Droplet energy } W = \sigma \pi d_d^2 + 1/2 \rho V_d v^2 \tag{12}$$

Adhesive binding energy W^* is the droplet energy per aggregate volume associated with the droplet. Aggregate volume $Va = K^3 V_d = \pi/6(K d_d)^3$, where K is a constant.

The specific energy is

$$W^* = (\sigma \pi d_d^2 + 1/2 \rho V_d v^2) / \{\pi/6 (K d_d)^3\}. \tag{13}$$

Using the parameters from Ref. 21, $W^* = 0.026\,\mathrm{J/cm^3}$. Thus, the adhesive binding is 10^4 times more efficient than sinter binding in converting powder to a solid object.

For the adhesive bonded process, the minimum feature is the volume of particle and the adhesive associated with one droplet after bonding to form a somewhat spherical powder primitive.

From Ref. 3,

$$Vr = \pi/6(\text{primitive diameter})^3. \tag{14}$$

A representative experimental machine[22] has a powder primitive diameter of 100 μm for 20 μm powder and 200 μm for 75 μm powder, producing a minimum resolution of 1900 elements/mm^3. Particle size is an ultimate limit on the resolution for the adhesive bonded process. For the strength of the minimum feature to be useful, a cube of two particle on a side is considered useful.

4.4. *Sheet lamination*

Sheet lamination fabrication such as the patented laminated object manufacturing process[23–25] builds 3-dimensional cross sections out of a roll of thermoplastic adhesive lined sheets with CO_2 laser and then binds each layer to the previous one by applying heat and pressure. The area outside the layer outline and inside any internal closed areas are cut into small sections called tiles. These tiles are removed during post-processing after the completion of all the layers.

The specific energy is,[3]

$$W^* = \rho cT, \tag{15}$$

where c is the specific heat capacity, ρ is the density of paper used and T is the disintegration temperature of the sheet lamination. The laser cutting of paper requires 336 J/cm^3[26] which is approximately about 300 times more than photopolymerization.

Imaging time per layer is,[3]

$$t_{\text{imaging}} = W^* A c l_l / P. \tag{16}$$

$$\text{Object fabrication speed} = l_l/(W^* A c l_l/P + t_{\text{reset}}). \tag{17}$$

Layer thickness influences all three possible parameter groups. The imaging time is affected by the infrared power required to cut the full layer depth. The reset time is affected by the number of layers in the object.

The resolution limits are based on the accuracy of energy beam position control regardless of the beam size. The beam must be controlled to move in a closed path leaving a minimum feature area. A representative commercial machine[27] has a rated position accuracy of ±51 μm and a minimum layer of thickness of 51 μm, producing a resolution of 1907 elements/mm^3. The ultimate resolution will be limited by the thickness of the paper used and the spot size of the laser beam. Using the same spot size as described by Ikuta *et al.*,[16] the ultimate resolution of sheet lamination is 200000 element/mm^3.

Pak and Nisnevich[28] show that both the model and experimental results indicated that for increased lamination speed, high deformation, temperature and increased contact area between the paper and the roller are required.

4.5. *Droplet deposition*

For droplet deposition, the molten droplets deposited on the working area will soften the material of the previous layer and solidify as one piece. Unfilled areas may be filled to the same thickness with a soluble molten wax as the support layer material. When all layers have been deposited, the object is removed from the solid block of support material by dissolving the support material. An example of a machine using this method is the BPM Technology's ballistic particle manufacturing.[6]

The imaging time per layer is determined by the droplet volume V_d, the droplet rate f, the layer area A_l, and the layer thickness l_l.

Therefore,[3]

$$t_{\text{imaging}} = A_l l_l / f V_d, \text{ and} \tag{18}$$

$$\text{object fabrication speed} = l_l / (A_l l_l / f V_d + t_{\text{reset}}). \tag{19}$$

The minimum resolution volume of a droplet deposition fabrication processes is the volume of one droplet. This is given as:

$$Vr = \pi / 6 d_d^3, \tag{20}$$

where d_d is the diameter of the droplet.

A representative machine[29] has a droplet diameter of 50 μm, producing a resolution of over 15200 elements/mm^3. The ultimate resolution will be determined by machine limitations in the production and control of liquid droplets. A similar application in the production and control on a similar size scale is a page size ink jet printer where 56 dots/mm have been demonstrated. The 12 μm diameter orifice of that machine produces 23 μm diameter droplets, giving an estimated ultimate resolution of 157000 elements/mm^3.

4.6. *Melt deposition*

For melt deposition, each layer is built by extruding molten material through an orifice and depositing it across the 2-dimensional plane. When the deposited material cools, it solidifies and adheres to the neighboring material, as well as the previous layer. Examples of machines using this method include the Fraunhofer's MJS and the Stratasys's FDM.

The imaging time per layer is determined by the melt material volume deposition rate U, the layer area A_l, and the layer thickness l_l.

$$t_{\text{imaging}} = A_l l_l / U, \text{ and} \tag{21}$$

$$\text{object fabrication speed} = l_l / (A_l l_l / U + t_{\text{reset}}). \tag{22}$$

The minimum resolution volume of the melt deposition fabrication process is dependent on the minimum feature area and the minimum layer thickness. A representative machine, the Stratasys's FDM, has an orifice diameter of 0.25 mm and a minimum layer thickness of 25 μm, producing a resolution volume of 640 elements/mm^3.[30]

5. Comparison of Present and Ultimate Performance

The maximum performance for each of the freeform fabrication technologies is shown in Table 2. The ultimate resolution is either inferred or given in the respective referenced papers (see Table 2, Column 4).

Figure 3 shows the present and the ultimate resolution of different methods of the RP process. It is obvious that laser curing of photolithography has the most significant ultimate resolution as compared to the other methods of fabrication.

Table 2. Rapid prototyping process performance. (*[ ] quotes the reference number)

Rapid Prototyping Processes	Fabrication Speed (cm/h)	Resolution (elements/mm^3)	
		Present	Ultimate
Laser Curing	1.5 [31]*	3152 [14]	13 300000 [16]
Masked Lamp	1.5 [15]	69 [21]	97500 [3]
Sinter Bonding	2.5 [18]	211 [18]	244000 [20]
Adhesive Bonding	1.27–1.9 [22]	1900 [22]	244000 [20]
Sheet Lamination	0.45 [23]	1907 [27]	200000 [16]
Droplet Deposition	0.4 [29]	15,200 [29]	157000 [29]
Melt Deposition	0.4 [30]	640 [30]	Not available

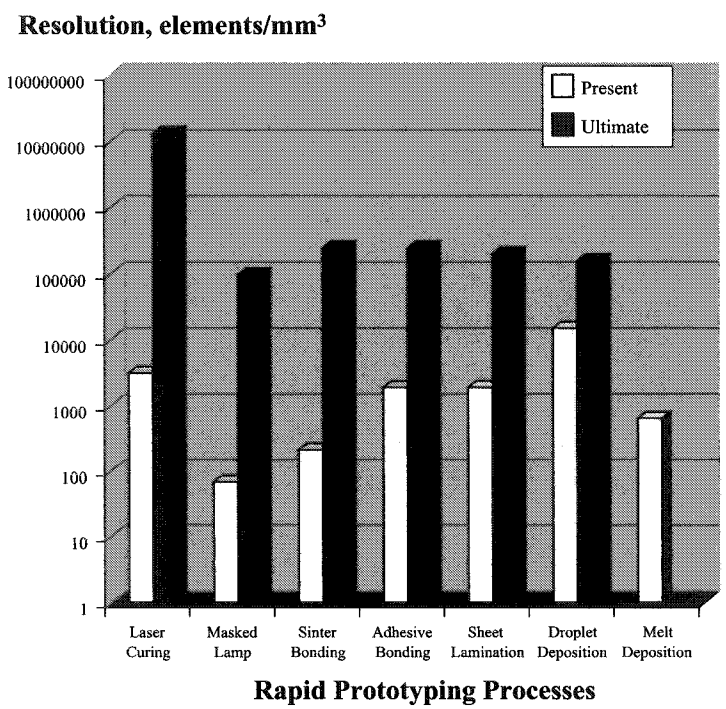


Fig. 3. Ultimate and present resolution of rapid prototyping process.

In terms of fabrication speed, laser sintering is faster due to the fact that the technology does not require us to build support, it requires little post-processing, but no post-curing and high laser scan speed.

Several additional figures of merits have been used to compare the performance, cost, and reliability of the rapid prototyping processes. The present technology values of those figures of merit are shown in Table 3.

A minimum process step is usually associated with a simple process whose cost would be generally lower and reliability higher than those technologies with more process steps. A low imaging specific energy is similarly associated with lower cost and higher reliability. Therefore, the ideal technology that has the smallest number of process steps and minimum imaging specific energy must closely approach the origins of Fig. 4.

Besides the comparison of those parameters, the merits of the different methods of RP adopted during the fabrication process can be compared. Hence, a list of the different advantages and disadvantages is shown in Table 4.

Table 3. Rapid prototyping process figures of merit.

Rapid Prototyping Processes	Imaging Specific Energy J/cm^3	Surface Roughness μm rms	Process Steps
Laser Curing	0.94 [32]	0.10 [17]	4
Masked Lamp	0.94 [32]	4.22 [18]	11
Sinter Bonding	300 [33]	1.3–3.0 [19]	4
Adhesive Bonding	0.026 [23]	6.6–15 [35]	4
Sheet lamination	336 [34]	Not available	3
Droplet Deposition	0.06 [29]	1.6–2 [36]	3
Melt Deposition	0.0005 [30]	Not available	1–3

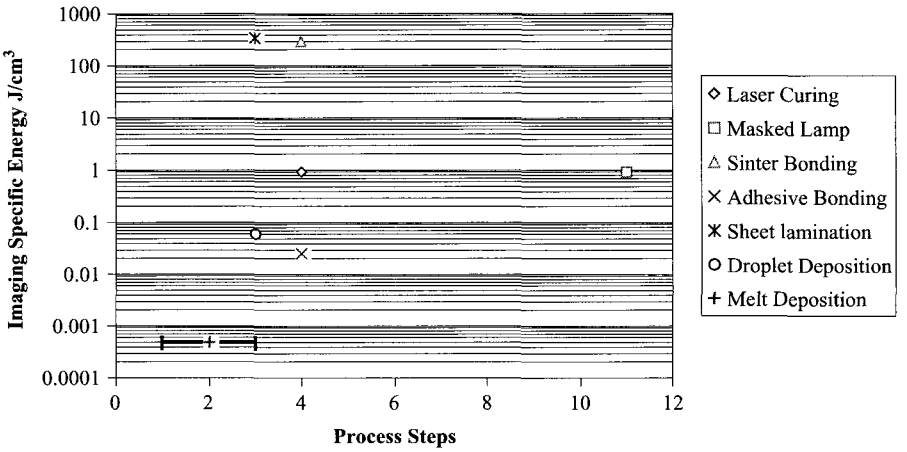


Fig. 4. Rapid prototyping process complexity.

Table 4. Rapid prototyping processes' advantages and disadvantages.

RP Processes	Advantages	Disadvantages
Stereolithography	• an established, proven technology • high layer fabrication speed capacity • high resolution capacity • low imaging specific energy	• a complex process • requires support • limited types of material (mainly Ciba-Geigy resins and Allied Signal Exactomer) • material creep • expensive material
Sinter Bonding	• no support is required • very little waste in material • large number of potential materials with better mechanical properties than photopolymer • cheaper and no smelly material	• high imaging specific energy • rough surface texture • porosity of the completed object • requires a separate curing step for some materials
Adhesive Bonding	• large number of potential materials • able to create ceramic molds for metal casting • support structures are included automatically in layer fabrication • low imaging specific energy	• rough or grainy appearance • poor resolution • post-processing required to remove moisture or preheat to appropriate temperature
Sheet Lamination	• rapid fabrication speed for objects with a high ratio of volume to surface area • support structures are included automatically in layer fabrication • low internal stress and distortion • a variety of organic and inorganic building materials are possible • multiple building materials or colours are possible in the same object on a layer basis	• specific cleaning is required especially for support material trapped in internal cavities. • significant expense associated with the large material waste • thermal cutting produces noxious fumes • possible warpage of lamination as a result of induced by heat of laser • high imaging specific energy
Droplet Deposition	• low material cost • very little material waste • material waste is limited to the support material • low imaging specific energy • multiple materials or colours are possible in the same object and even in the same layer	• poor surface finish and grainy appearance • limited to material of relatively low melting point
Melt Deposition	• very low imaging specific energy • few number of process steps	• poor surface roughness • low fabrication speed • average resolution

6. New Generation Rapid Prototyping Processes

In the years to come, micro-machines will have profound impact on the lives of ordinary people. The height of micro-electronic circuits and micro-electron-mechanical

system is typically between 2 and 10 micrometres.[37] Hence, stereolithography can be another feasible way of fabricating these micro-mechanism. However, the ideal RP process should be as fast as possible and requires no post-processing.

The advantages of different methods could be combined to improve the present RP process. Masked lamp technology and laser curing can be combined together to produce a multi-laser system that is able to irradiate the resin in a raster scan format. This system will allow the resin to be irradiated layer by layer and thus, this will significantly reduce the time for fabrication. Vertical straight parts of different height can be produced by increasing the power of the individual laser. The speed of producing vertical straight parts can be increased by solidifying thicker layer of resin. For the sloping parts, a smaller layer is recommended in order to reduce the ledges and produce better resolution and surface finish. However, there must also be a balance in the combination of different RP processes so that they would not result in drastic increases in cost and design/operation complexity problems.

7. Conclusion

The different merits of rapid prototyping processes are investigated. Laser curing is believed to have a greater potential in improving its resolution. A hybrid of different methods can be used to optimize the performance of rapid prototyping by producing the part with lesser time.

Acknowledgment

The authors express their appreciation to the Agency for Science, Technology and Research (A*STAR) of Singapore for sponsoring this research work.

References

1. R. J. Famieson, Direct slicing of CAD models for rapid prototyping, *Rapid Prototyping Journal* **1–2** (1995) 4–12.
2. R. J. Donahue, CAD model and alternation methods of information transfer for rapid prototyping systems, *Proceeding of the Second International Conference on Rapid Prototyping* (1991) 217–235.
3. J. L. Johnson, *Principles of Complete Automated Fabrication* (Paletino Press, 1994).
4. R. Ippolito, L. Luliano and A. Gatto, Benchmarking of rapid prototyping techniques in terms of dimensional accuracy and surface finish, *Annals of the CIRP* **44** (1995) 157–160.
5. P. M. Dickens, Research developments in rapid prototyping, *Proceedings of the Institution of Mechanical Engineerings, Part B; Journal of Engineering Manufacture* **209** (1995) 261–266.
6. C. K. Chua, K. F. Leong and C. S. Lim, *Rapid Prototyping: Principles and Applications* 2nd Edition (World Scientific, 2003).
7. M. Burns, *Automated Fabrication. Improving Productivity in Manufacturing* (PTR Prentice Hall, 1993).

8. D. Kochan and C. K. Chua, State-of-the-art and future trends in advanced prototyping and manufacturing, *International Journal of Information Technology* **1** (1995) 173–184.
9. P. F. Jacobs, *Rapid Prototyping and Manufacturing: Fundamentals of Stereolithography* (Society of Manufacturing Engineers, 1992), Chapter 1, pp. 11–18, Chapter 2, pp. 25–32, Chapter 3, p. 89.
10. M. S. M. Sun, J. C. Nelson, J. J. Beaman and J. J. Barlow, A model for partial viscous sintering, *Proceedings of the Solid Freeform Fabrication Symposium* (1991) 46–55.
11. Rapid prototyping report, *BPM Technology Introduces Low-priced Personal Modeler* **5** (1995) 3–5.
12. K. E. Richardson, The production of wax models by the ballistic particle manufacturing process, *Proceedings of the Second International Conference on Rapid Prototyping* (1991) 15–22.
13. M. A. Huniziker, Schulthess, and M. Hofmann, Developments in stereolithography resins for investment casting, *Proceedings of the Fourth International Conference on Rapid Prototyping* (1993) 225–237.
14. Quadrax Corporation, *Mark 1000 Laser Modeling System* (1990).
15. Cubital Ltd, *Solider* (1992).
16. K. Ikuta, K. Hirowatari and T. Ogata, Ultra high resolution stereo lithography for three dimensional micro fabrication, *Proceedings of the Fourth International Conference on Rapid Prototyping* (1994) 37–46.
17. D. Kochan and R. Hovtun, Precise and optimized process realization for solid ground curing, *Proceedings of the Fourth International Conference on Rapid Prototyping* (1994) 77–89.
18. DTM Corporation, *Sinterstation 2000* (1995).
19. M. Durham, T. Grimm and J. Rollins, *SLS and SLA: Different Technologies for Different Applications* (Accelerated Technologies, Inc., 1996).
20. G. K. Starkweather, A high resolution laser printing system, *Second International Congress on Advances in Non-Impact Printing Technologies, Advance Papers Summary, SPSE* (1984) 198–199.
21. E. Sachs, M. Cima and J. Cornie, Three dimensional printing: ceramic shells and cores for casting and other applications, *Proceedings of the Second International Conference on Rapid Prototyping* (1991) 39–53.
22. S. Michaels, E. Sachs and M. Cima, Metal parts generation by three dimensional printing, *Proceedings of the Fourth International Conference on Rapid Prototyping* (1993) 25–31.
23. Helisys Inc., *LOM 1015 and LOM 2030* (1992).
24. M. Feygin, *Apparatus and Method for Forming an Integrated Object from Laminations* (1988) U.S. Patent No. 4, 752, 352.
25. M. Feygin, *Apparatus and Method for Forming an Integrated Object from Laminations* (1994) European Patent No. 0, 272, 305.
26. M. Feygin, *Apparatus and Method for Forming an Integrated Object from Laminations* (1994) U.S. Patent No. 5, 354, 414.
27. N. Rykalin, A. Uglov and A. Kokora, *Laser Machining and Welding* (Pergamon Press, 1992) 230.
28. S. S. Pak and G. Nisnevich, Interlaminate strength and processing efficiency improvements in laminated object processing efficiency improvements in laminated object manufacturing, *Proceedings of the Fifth International Conference on Rapid Prototyping* (1994) 171–180.

29. W. Crooks, E. W. Luttman and A. B. Jaffe, High quality color printing with continuous ink jet, *First International Congress on Advances in Non-impact Printing Technologies for Computer and Office Applications* (1982) 1007–1031.
30. Stratasys Inc., *3D Modeler* (1992).
31. J. L. Hirsch, DuPont SOMOS solid imaging system, *Proceedings of the National Conference on Rapid Prototyping* (1990) 11–18.
32. W. F. Hug and P. F. Jacobs, Laser technology assessment for stereolithography systems, *Proceedings of the Second International Conference on Rapid Prototyping* (1991) 29–38.
33. J. W. Barlow, M. S. M. Sun and J. J. Beaman, Analysis of selective laser sintering, *Proceedings of the Second International Conference on Rapid Prototyping* (1991) 1–14.
34. N. Rykalin, A. Uglov and A. Kokora, *Laser Machine and Welding* (Pergamon Press, 1978).
35. P. K. Subramanian, G. Zong and H. L. Marcus, Selective laser sintering and reaction sintering of ceramic composites, *Proceedings of the Solid Freeform Fabrication Symposium* (1992) 63–71.
36. M. E. Orme, A novel technique of rapid solidification net-form materials synthesis, *Journal for Materials Engineering and Performance* (1993) 399–407.
37. H. Hogan, Invasion of the Microm, *New Scientist* (June 1996) 28–33.

This page is intentionally left blank

CHAPTER 6

VISUAL ASSESSMENT OF FREE-FORM SURFACES IN CADCAM

ROBERT J. CRIPPS and ALAN A. BALL
Geometric Modeling Group,
Manufacturing and Mechanical Engineering,
The University of Birmingham,
Birmingham, United Kingdom
E-mail: r.cripps@bham.ac.uk

We review the current graphical tools available for checking the quality of CAD/CAM (computer-aided design/manufacture) surface models and highlight the difficulties of their use. A new range of geometrically based tools is introduced, especially designed for the task of enabling design engineers to visually assess the quality of free-formed surfaces at a workstation screen to within an accuracy comparable with working to full-scale drawings. The tools should be easy to use without recourse to understanding the underlying mathematical theory of surface differential geometry on which the techniques are ultimately based. We propose a hierarchical approach to surface quality assessment that addresses the problem of gross geometric features swamping more subtle features of a surface definition.

Keywords: Shape quality; NURBS; geometric parameterization; FANGA analysis; CAD/CAM; free-form surfaces.

1. Introduction

Computer aided engineering was once the tool of the large wealthy high-tech industries like aerospace and automotive manufacturers. Indeed it was in the aerospace industry where the seeds of modern CADCAM systems first germinated. These large companies had the resource to develop their own software and train their own experts in its use. Today, the cost of computing hardware and software has placed this technology within the reach of even the smallest manufacturing businesses. Most large companies now insist on CADCAM as the tool for producing engineering solutions and its communication between companies. Computer aided engineering has come of age. The best evidence of this is in the wide range of products that are now designed, modeled, analyzed and manufactured using CADCAM systems. Such products include: aircraft, cars, ships, computers, electrical appliances, toys, containers, medical equipment, dentures, footware, etc. Examples of

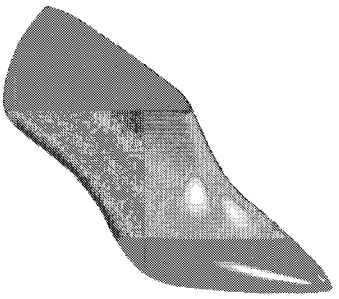


Fig. 1. A shoe last representation.

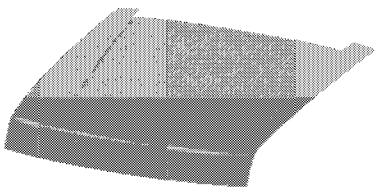


Fig. 2. A car bonnet.

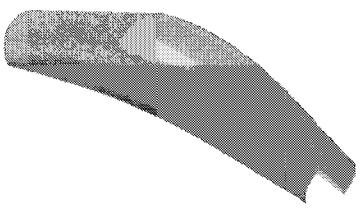


Fig. 3. An aircraft windscreen and canopy.

such computer-based models are a shoe last (Fig. 1), a car bonnet (Fig. 2), and an aircraft windscreen and canopy (Fig. 3).

Even though its use is growing, there are still fundamental problems associated with the way objects are described on a digital computer. Of primary concern is the assessment of the geometric and manufacturing qualities of computer-based definitions, for example a car roof. Having identified a problem either in the geometry or in the manufacturability of an object, how does one correct it? Not least of the problems is in viewing an object on a screen whose size is a fraction of the actual object. We aim to review how CADCAM is used to describe objects on a computer. This requires some knowledge of the mathematical representation used by the CADCAM system. Given this mathematical description, we can then develop methods of assessing the quality of the computer-based representations of the real objects. We give detailed descriptions of a wide range of current quality assessment techniques and detail a novel approach based on geometric parametrization. The next stage is

to consider how to use these quality assessment tools and how to identify potential problems to help determine whether the finished computer model is acceptable in terms of its geometry.

The remainder of this chapter is organised in self contained sections. We begin with a brief overview of how CADCAM is used to describe three-dimensional objects and consider the criteria for assessing the quality of computer-based representations. The next section gives the mathematical basis of surface representations and mathematical preliminaries for the assessment techniques discussed later. This section can be omitted for readers who are not interested in implementation of these techniques. Section 4 looks at the practical aspects of the quality assessment techniques. Each technique is applied to the car roof definition with a discussion of the results and their implications. Section 5 considers those techniques that are aimed at simplifying the task of assessing the quality of a definition. The penultimate section aims to specify a structured approach to assessing the quality of computer-based definitions. The conjecture being that a systematic approach is more likely to successfully determine the quality of a definition and to give more insight to the geometric cause of any imperfections detected along the way.

The question of how to correct for a geometric defect has received very little attention from the CADCAM vendors or academics alike and yet it is this area that offers the greatest potential to increase the impact of CADCAM.

2. CADCAM: Object Description and Quality Assessment

Free form surfaces in CADCAM are typically defined by a vector valued parametric equation of the form:

$$\mathbf{r}(u,v) = [x(u,v), y(u,v), z(u,v)] \qquad 0 \leq u,\ v \leq 1,$$

which is a mathematical description of a surface that can be interrogated at every point, not just at the data used to define the surface.

The most popular representations are the Bézier and B-spline forms.[9] For example, the rational Bézier form can be represented as:

$$\frac{\sum_{i=0}^{n}\sum_{j=0}^{m}\omega_{i,j}b_{i,n}(u)b_{j,m}(v)\mathbf{P}_{i,j}}{\sum_{i=0}^{n}\sum_{j=0}^{m}\omega_{i,j}b_{i,n}(u)b_{j,m}(v)} \qquad 0 \leq u,\ v \leq 1;\ \omega_{i,j} > 0,$$

where $b_{i,n}(u)$ and $b_{j,m}(v)$ are the Bézier basis functions, $w_{i,j}$ are the scalar weights that are normally restricted to being positive, and $\mathbf{P}_{i,j}$ are the vertices of the control polyhedron. We note that when the weights are constant we have the non-rational Bézier form. Each surface is then a rectangular grid of (rational) polynomial patches of the form $\mathbf{r}(u,v)$ and a typical definition is created as a collection of one or more surfaces. However, the surface representation should not affect the way in which we are able to visualise its form. For example a shoe (Fig. 1) should have the same shape, to within some tolerance, whatever its representation. It is the geometry of the object that is important to the designer, not its computer-based representation.

The definition can then be used to generate a sequence of points and normals that are used to define numerically controlled (nc) tool cutter paths to drive an nc milling machine, producing the dies and moulds for the components.[9] Thus with CADCAM we have the capability of matching the computer-based definition and the production tool to machining tolerances, i.e. what we define is what we get. This has resulted in the need to create high quality surface definitions. This quality is not just for mathematical rigor or computational speed and robustness, nor just for the aesthetics imposed by the stylists or designers. Small surface imperfections can have large consequences in down-stream activities such as nc tool manufacturing and robotic assembly.[6]

Before the introduction of CADCAM, the quality of surfaces was assessed by skilled craftsmen looking at large scale drawings of section curves and physical models or patterns. With a CADCAM generated definition, it is now possible to consider its quality on a workstation screen. To aid in this assessment, CADCAM systems offer a range of interrogation tools. Some are computationally equivalent to the familiar manual techniques, for example surface/plane intersects (intersects). Others are new techniques with no manual equivalents, for example curvature based analysis.

We review the standard techniques available for quality assessment; their effectiveness and the techniques that have been developed to ease their use. These techniques give results that are easy to interpret with high quality definitions, but when used on definitions with geometric imperfections, interpretation becomes very difficult.

When assessing any technique for visualising surface quality the following criteria are useful:

(i) *Intuitive:* Techniques should be easy to use and interpret by design engineers. The rationale here is that these tools are not for mathematicians who may have a good understanding of surface geometry. The techniques are for users of CADCAM systems and therefore they need to be presented in terms that are intuitive to the design engineer. So to expect engineers to be comfortable with moving control points about on the screen is one thing. To require a deep understanding of surface differential geometry is expecting too much.

(ii) *Applicability:* Detection of a range of geometric imperfections.
Here, it is expected that techniques should be available to detect geometric flaws in an engineering context, i.e. creases, inflections, non-harmonious variations, etc. Thus we should not give cryptic messages about C^2 discontinuities, rather messages relating to the manufacturability, etc.

(iii) *Robust:* Detection independent of the choice of interrogation, i.e. location, spacing, light source, curvature colour maps, etc.
Ideally, any technique should be able to detect all flaws without special knowledge of the definition it is applied to. For example, the sampling of the definition

to evaluate surface invariants should depend on the geometry and not be biased to the underlying parametrization. Also one should not have to rely on inside information of a technique in order for it to detect flaws, i.e. best choice of a light source for optimum reflection lines, etc.

(iv) *Visibility:* Features must be visible on a workstation screen.
This is crucial if we wish to have techniques that can be used by engineers at a workstation screen. Scale should not affect the technique's ability to detect flaws. Working at a high level of 'zoom' should not lose the integrity of the overall definition.

(v) *Local:* The ability to assess patches both independently and as a surface.
This clearly follows as a consequence of the *Visibility* criteria. Thus changes made 'locally' at the patch level should maintain overall integrity with the total definition.

(vi) *Geometric:* Interrogations should be related to the geometry.
This enables *Robust* techniques to be developed and more importantly, if the technique relates directly to the geometry then it should be possible to relate the cause of the flaw back to the geometry and hence automate corrective measures. Such automatic modification must always have an override in the cases of intended features, i.e. crease lines, etc.

Applying the above criteria to existing quality assessment tools highlights the need for a new approach. Indeed, most state-of-the-art CAD software packages, for example EDS,[7] have many of the standard tools for surface visualization including basic isoparameter lines, planar and curvature contours and curvature analysis including isophotes, reflection and silhouette lines — see Secs. 2, 3 and 4. It is recognized that these tools help identify very quickly what should not be there, rather than quantifying what is there.

Having identified an offending flaw in a definition, users need to have significant automation to correct the geometry to reduce if not totally eliminate such flaws. Most CADCAM systems have all the underlying visualization and manipulation algorithms, the problem is that they just need to be more accessible. There is growing recognition that the fundamental support for assessing the quality of CADCAM generated surface definitions has still not been addressed. Current techniques, when expertly used, will detect geometric flaws in surfaces. They do not address why they exist nor how to remove them.

The important message of this chapter is the proposal of a structured approach to free form surface quality assessment of reasonable surfaces at a workstation screen. The proposal is based on parameter lines that are constructed geometrically. The aim is not to replace existing methods, rather to develop geometric equivalent, more accessible forms that can be used in a hierarchical approach. Since the assessment will then be in terms of geometrically constructed surface lines, corrective procedures may be developed.

3. Implementation Details

We shall keep this section to a minimum but give sufficient details in order that all the methods discussed in the text can be implemented. We shall also try to keep the majority of the mathematics in this section so that when we come to discuss the techniques, we can concentrate on their use. Readers who are not interested in implementation details may omit this section.

All the visualization techniques require point and at most second order partial derivative evaluation of the surface. Some techniques require that the surface be subdivided or split. We therefore simply state the results needed for the techniques. We shall also give hints and guidance on possible strategies for the implementation of techniques when we discuss them in more detail in the following sections. Since most polynomial-based representation forms can be written as a NURBS,[9] for simplicity we shall assume our surface $\mathbf{R}(u, v)$ has a NURBS form.

3.1. *Standard representations*

The most general parametric representation form is the Non-Uniform Rational B-Splines (NURBS) which can be represented as:

$$\mathbf{R}(u,v) = \frac{\sum_{i=0}^{n}\sum_{j=0}^{m}\omega_{i,j}N_{i,n}(u)N_{j,m}(v)\mathbf{P}_{i,j}}{\sum_{i=0}^{n}\sum_{j=0}^{m}\omega_{i,j}b_{i,n}(u)b_{j,m}(v)} \quad 0 \le u,\ v \le 1;\ \omega_{i,j} \ge 0,$$

where $N_{i,n}(u)$ and $N_{j,m}(v)$ are the normalized B-Spline basis functions defined by the non-uniform knot vectors $[\mathbf{U}]$ and $[\mathbf{V}]$, where:

$$[\mathbf{U}] = [u_0, u_1, \ldots, u_{n-p+1}] \text{ and } [\mathbf{V}] = [v_0, v_1, \ldots, v_{m-q+1}],$$

where $u_i \le u_{i+1}$ and $v_j \le v_{j+1}$. The $w_{i,j}$ are the scalar weights that are normally restricted to being positive, and $\mathbf{P}_{i,j}$ are the vertices of the control polyhedron. There are several reasons why this form has been widely adopted by the CADCAM vendors. We list a few:

(i) A NURBS representation is able to represent both rational and non-rational curves and surfaces. Further, NURBS contains both rational and non-rational Bézier forms as follows:

- Rational Bézier:

$$[\mathbf{U}] = [\underbrace{0,0,\ldots,0}_{p+1},\underbrace{1,\ldots,1}_{p+1}], \quad [\mathbf{V}] = [\underbrace{0,0,\ldots,0}_{q+1},\underbrace{1,\ldots,1}_{q+1}].$$

- Non-Rational Bézier $[\mathbf{U}]$ & $[\mathbf{V}]$ as above; constant weights.

(ii) Any NURBS definition can be represented in Bézier form by simply ensuring that all knots in the $[\mathbf{U}]$-knot vector have multiplicity p and all knots in the $[\mathbf{V}]$-knot vector have multiplicity q.

(iii) Generally the control polyhedron $\mathbf{P}_{i,j}$ will extend beyond the surface edges unless we force the polyhedron to start and end at the surfaces' corners. This is achieved by letting the end u-knots u_0, u_{n-p+1} have multiplicity p and similarly, the end v-knots, v_0, v_{m-q+1} have multiplicity q.

(iv) A multi-patch NURBS surface gives guaranteed continuity between the surface patches. This can be used to good effect when devising interrogation and manipulation algorithms, for example the knot insertion algorithm of DeBoor.[2] However it places a large onus on end-users of NURBS systems to ensure mathematical coherency between such multi-patch surfaces. This usually results in a restriction on the number and location of surface defining knots of adjoining surfaces and is a common cause of surface irregularities.

(v) The high level of continuity between NURBS surfaces is not always desirable as this is very restrictive in terms of the geometry.

(vi) The rational form can represent all natural quadric surfaces (i.e. the plane, cylinder, cone and sphere) exactly.[19]

3.2. *Point evaluation*

This is best achieved using knot insertion that is a generalization of the de Casteljau Algorithm.[2] In its simplest form it evaluates a point on the NURBS surface by repeatedly inserting new knots into the knot vectors. To evaluate a point, knots v_j are inserted q times into $[\mathbf{V}]$ which results in a new 'knot-line', knots u_i are then inserted p times into $[\mathbf{U}]$ to give $\mathbf{P}(u_i, v_j)$. This process is effectively repeated linear interpolation defined (in the curve case) by:

$$\mathbf{P}^m(u) = \alpha^m_{i,\cdot}\mathbf{P}^{m-1}_{i,\cdot} + (1-\alpha^m_{i,\cdot})\mathbf{P}^{m-1}_{i-1,\cdot} \quad m = 1, 2, \ldots, p,$$

$$\text{where } \alpha^m_{i,\cdot} = \frac{u - u_i}{u_{i+p-m+1} - u_i} \quad i = k - p + m, \ldots, k,$$

$$\text{and } \mathbf{P}^0_{i,\cdot} = \mathbf{P}_{i,\cdot}$$

and

$$\omega^m_{i,\cdot} = \alpha^m_i \omega^{m-1}_{i,\cdot} + (1-\alpha^m_{i,\cdot})\omega^{m-1}_{i-1,\cdot},$$

and where u is the point to be evaluated and the subscript '.' has been used to remind the reader that we treat rows and columns of the control polyhedron as NURBS curves.

3.3. *Derivative evaluation*

All the derivatives of a NURBS surface can be obtained from the knot insertion algorithm, see, for example Farin[9] for more details. This is completely analogous to the way de Casteljau's algorithm generates points and derivatives by repeated linear interpolation.

3.4. *Surface sub-division*

Again we can use the knot insertion algorithm to sub-divide our NURBS surface. Since inserting a knot $\langle u, v\rangle \langle p, q\rangle$ times results in a collection of auxiliary vertices, we find that we have the entire polyhedron for all the four sub-divided surfaces.

3.5. *Surface differential geometry*

Again, for completeness, we give the necessary formulae for deriving surface invariants that will be used in the following sections. Further details can be found in Ref. 18. All the surface quantities that we shall need can be obtained from the knot insertion algorithm of Sec. 3.2.

Unit surface normals

The *unit surface normal* for a NURBS surface $\mathbf{R}(u, v)$ is defined by:

$$\mathbf{n}(u, v) = \frac{\mathbf{R}_u(u, v) \times \mathbf{R}_v(u, v)}{\|\mathbf{R}_u(u, v) \times \mathbf{R}_v(u, v)\|}$$

where $\mathbf{R}_u(u, v) = (\delta\mathbf{R}(u, v))/\delta u$ and $\mathbf{R}_v(u, v) = (\delta\mathbf{R}(u, v))/\delta v$ are the partial parametric derivatives with respect to u and v respectively of the surface $\mathbf{R}(u, v)$. Geometrically, the unit surface normal is a vector of unit length that is normal to the surface at the given point.

In general, surface normals are well defined but is not the case if either of the partial derivatives are zero or if they are parallel, i.e. the cross product is zero. In such cases we must determine whether the normal is 'well defined' at such points and in terms of evaluation, we must take special care when these cases arise.

Surface twists

The *twist* of a surface is its mixed partial derivative:

$$\frac{\delta\mathbf{R}(u, v)}{\delta u \delta v} = \frac{\delta\mathbf{R}(u, v)}{\delta v \delta u}.$$

The twist vector at a surface corner point gives a measure of the deviation of the internal vertex from the tangent plane at that corner. For example at $\mathbf{R}(0, 0) = \mathbf{P}_{0,0}$ the twist vector is a measure for the deviation of $\mathbf{P}_{1,1}$ from the tangent plane at the corner, $\mathbf{P}_{0,0}$.[9]

First and second fundamental form

The first fundamental form, I and second fundamental form, II of a surface are given by[21]:

$$\begin{aligned} \mathrm{I} \; &= \mathbf{R}_u(u, v) \cdot \mathbf{R}_u(u, v) \left(\frac{du}{ds}\right)^2 + 2\mathbf{R}_u(u, v) \cdot \mathbf{R}_v(u, v) \left(\frac{du}{ds}\right)\left(\frac{dv}{ds}\right) \\ &\quad + \mathbf{R}_v(u, v) \cdot \mathbf{R}_v(u, v) \left(\frac{dv}{ds}\right)^2 \end{aligned}$$

$$= E\left(\frac{du}{ds}\right)^2 + 2F\left(\frac{du}{ds}\right)\left(\frac{dv}{ds}\right) + G\left(\frac{dv}{ds}\right)^2.$$

$$\begin{aligned}\text{II} &= \mathbf{n}(u,v)\cdot\mathbf{R}_{uu}(u,v)\left(\frac{du}{ds}\right)^2 + 2\mathbf{n}(u,v)\cdot\mathbf{R}_{uv}(u,v)\left(\frac{du}{ds}\right)\left(\frac{dv}{ds}\right) \\ &\quad + \mathbf{n}(u,v)\cdot\mathbf{R}_{vv}(u,v)\left(\frac{dv}{ds}\right)^2 \\ &= L\left(\frac{du}{ds}\right)^2 + 2M\left(\frac{du}{ds}\right)\left(\frac{dv}{ds}\right) + N\left(\frac{dv}{ds}\right)^2.\end{aligned}$$

Both forms are *surface invariants* as they are independent of the surface parametrization. These forms enable us to calculate the curvature, κ, of a curve $\mathbf{S}(t)$ lying on the surface $\mathbf{R}(u,v)$.

Curvature of surface curve

A curve on the parametric surface $\mathbf{R}(u,v)$ can be represented parametrically by the equations:

$$u = u(t) \qquad v = v(t),$$

i.e. $\mathbf{S} = \mathbf{S}(t)$ where $\mathbf{S}(t) = [u(t),\, v(t)]^T$ and $\mathbf{S}(t)$ is a point on the curve that is also a point of the surface. The tangent to this curve is given by:

$$\mathbf{S}_t(t) = \frac{\delta\mathbf{S}(t)}{\delta u}u_t + \frac{\delta\mathbf{S}(t)}{\delta v}v_t,$$

and thus the curvature κ of the curve $\mathbf{S}(t)$ with the tangent direction $\mathbf{S}_t(t)$ is given by:

$$\kappa = \frac{1}{\cos(\theta)}\frac{Ldu^2 + 2Mdudv + Ndv^2}{Edu^2 + 2Fdudv + Gdv^2},$$

where θ is the angle between the normal $\mathbf{m}(t)$ of the curve and the unit surface normal $\mathbf{n}(u,v)$ at a given point.[3]

Normal curvature of surface curve

When $\theta = 0$, i.e. $\cos(\theta) = 1$, we have $\mathbf{m}(t) = \mathbf{n}(u,v)$ and the osculating plane of the curve is perpendicular to the surface tangent plane at the point. The curvature of such a curve is called the *normal* curvature of the surface in the direction of the tangent du/dv and is given by:

$$\kappa_N = \frac{Ldu^2 + 2Mdudv + Ndv^2}{Edu^2 + 2Fdudv + Gdv^2}.$$

Our surface curves can be planar, since $\mathbf{m}(t) = \mathbf{n}(u,v)$, and we call such curves *normal section* curves.

Lines of curvature

We now consider planes containing the normal vector at some point, i.e. the tangent $\lambda = dv/du$ is no longer fixed. Thus the normal curvature becomes a function of the tangent, i.e.

$$\kappa(\lambda) = \frac{L + 2M\lambda + N\lambda^2}{E + 2F\lambda + G\lambda^2},$$

where $\kappa(\lambda)$ is, in general, a rational quadratic in λ. The two extrema, κ_1 and κ_2 of $\kappa(\lambda)$ are the roots of:

$$\begin{vmatrix} \kappa E - L & \kappa F - M \\ \kappa F - M & \kappa G - N \end{vmatrix} = 0,$$

where κ_1 and κ_2 define directions in the (u, v) plane and the corresponding directions in the tangent plane are called the *principal directions*. The net of lines that have these directions at all their points is called the net of *lines of curvature*.

This net may be constructed by analytic integration. An alternative formulation for the lines of curvature[10] is given by the following coupled differential equations:

$$\frac{du}{ds} = \beta(M + \kappa F) \quad \text{and} \quad \frac{dv}{ds} = -\beta(L + \kappa E),$$

where s is the arc-length and:

$$\beta = \frac{\pm 1}{\sqrt{E(M + \kappa F)^2 - 2F(M + \kappa F)(L + \kappa E) + G(L + \kappa E)^2}},$$

and κ represents one of the extreme curvature values. These can be solved using Runge-Kutta 4th Order: writing the two differential equations as:

$$\frac{d\mathbf{y}}{ds} = f(s, \mathbf{y}) \quad \text{where } \mathbf{y} = [u, v]^T,$$

we can determine the increment $\Delta\mathbf{y}$ corresponding to a step Δs. The initial conditions are the coordinates (u, v) of a start point and interrogations are typically made at equally spaced intervals along the surface boundaries.

Gaussian curvature

The product of the principal curvatures is called the *Gaussian* curvature, i.e.

$$K = \kappa_1\kappa_2 = \frac{LN - M^2}{EG - F^2}.$$

Some of the properties of K[21] are:

(i) If the normal, $\mathbf{n}(u, v)$ is reversed then κ_1 and κ_2 change sign but K does not.
(ii) For a given point, $\mathbf{R}(u, v)$, if:
   - $K > 0$: the point is called *Elliptic*, i.e. the surface in the neighbourhood of $\mathbf{R}(u, v)$ is an ellipsoid.

- $K = 0$: the point is called *Parabolic*, i.e. the surface in the neighbourhood of $\mathbf{R}(u, v)$ is a paraboloid.
- $K < 0$: the point is called *Hyperbolic*, i.e. the surface in the neighbourhood of $\mathbf{R}(u, v)$ is a hyperboloid.

(iii) If the surface is deformed using a transformation that preserves length, K is unaltered.

Mean curvature

The arithmetic mean of the principal curvatures is called the *Mean* curvature, i.e.

$$H = \frac{\kappa_1 + \kappa_2}{2} = \frac{NE - 2MF + LG}{2(EG - F^2)}.$$

At a point $\mathbf{R}(u, v)$ where both K and H are zero, the point is said to be *flat.*

Reflection lines

A light line is defined by: $\mathbf{l}(t) = \mathbf{l}_0 + t\mathbf{l}_d$ where $\mathbf{l}_0$ is a point on the line and $\mathbf{l}_d$ is the direction of the line. To find a reflection line one has to specify a viewpoint $\mathbf{v}$ and the light line, $\mathbf{l}(t)$. Then for each point $\mathbf{l}$ on the line, we calculate point(s) that reflect(s) $\mathbf{l}$ onto the line $\mathbf{vl}$, thus we solve for[13]: (Fig. 4)

$$\lambda \cdot \mathbf{v}_d = 2\mathbf{n}(\mathbf{n} \cdot \mathbf{b}) - \mathbf{b},$$

where

$$\mathbf{v}_d = \mathbf{v} - \mathbf{l}, \ \mathbf{b} = \mathbf{l} - \mathbf{v} \quad \text{and} \quad \lambda = \frac{\|\mathbf{b}\|}{\|\mathbf{v}_d\|}.$$

We can eliminate λ from the three equations yielding two equations in u and v that can be solved numerically.

If the surface is C^r continuous, then the reflection lines will be C^{r-1} continuous.

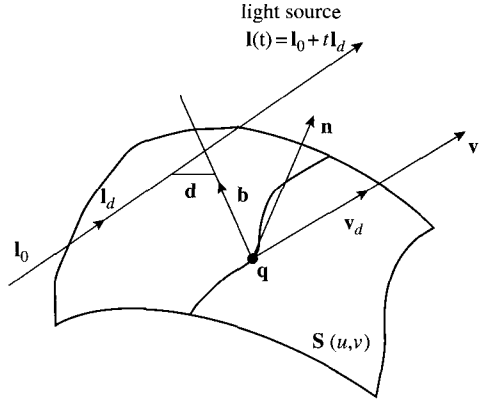


Fig. 4. Construction of reflection lines, isophotes and silhouettes.

Isophotes and silhouettes

We again have a light direction $\mathbf{l}_d$ of a light source and a viewpoint $\mathbf{v}$. Then the function $g(u, v)$ is defined across the surface $\mathbf{R}(u, v)$ as:

$$g(u, v) = \cos(\Theta)\cos(\phi),$$

where Θ is the angle between the light direction vector and the surface normal, i.e. $\Theta = \angle(\mathbf{l}, \mathbf{n})$ and ϕ is the angle between the viewpoint and the surface normal, i.e. $\phi = \angle(\mathbf{v}, \mathbf{n})$. An **lv** curve[11] is made up of all points on the surface $\mathbf{R}(u, v)$ where $g(u, v)$ is a constant. In the case where $\phi = 0$ and $\mathbf{l}$ = constant then we have an *isophote*. As with reflection lines, if the surface is C^r continuous, then the reflection lines will be C^{r-1} continuous. Isophotes become *silhouette* lines when $\Theta = 90^o$. The **lv**-curve can be further characterized as follows:

$$\begin{cases} g(u, v) = \textit{isophote} \text{ when } \phi = 0^o,\ \mathbf{l} = \text{constant} \\ \qquad\quad \textit{silhouette} \text{ when } \Theta = 90^o \\ \qquad\quad \textit{pseudo-central isophote} \text{ when } \phi = 0^o,\ \mathbf{l} \neq \text{constant} \\ \qquad\quad \textit{iso-pheng} \text{ when } \mathbf{v} = \text{constant},\ \mathbf{l} = \text{constant} \\ \qquad\quad \textit{pseudo-central isopheng} \text{ when } \mathbf{v} = \text{constant},\ \mathbf{l} \neq \text{constant}. \end{cases}$$

Highlight lines

Here we dispense with the need to define a viewpoint. As before, we have $\mathbf{l}(t) = \mathbf{l}_0 + t\mathbf{l}_d$ (see Fig. 5). For a given point $\mathbf{q}$ on a surface $\mathbf{S}(u, v)$, let $\mathbf{n}$ be the surface normal vector at $\mathbf{q}$. The extended surface normal $\mathbf{E}(s)$ is a line passing through $\mathbf{q}$ with the direction given by $\mathbf{n}$ and defined by

$$\mathbf{E}(s) = \mathbf{q} + \mathbf{n}(s).$$

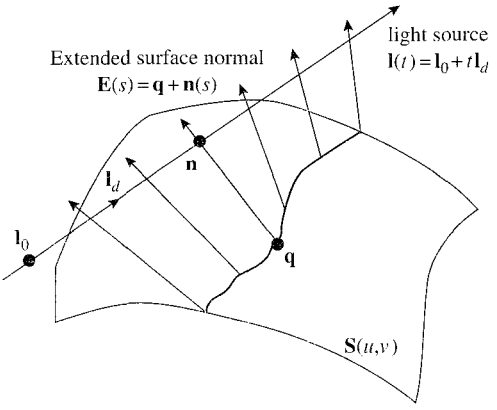


Fig. 5. Highlight lines and bands.

The point $\mathbf{q}$ belongs to the *highlight line*[1] if: $\mathbf{l}(t)$ and $\mathbf{E}(s)$ intersect or if the perpendicular distance d between both lines is zero, i.e.

$$d = \frac{|\,(\mathbf{l}_d \times \mathbf{n}) \cdot (\mathbf{l}_0 - \mathbf{q})\,|}{\|\mathbf{l}_d \times \mathbf{n}\|} = 0$$

to some tolerance.

Replacing $\mathbf{l}(t)$ by a cylinder of radius r we obtain a *highlight band.* $\mathbf{q}$ belongs to the highlight band if $d \leq r$ where d is the perpendicular distance of the extended normal from the centre of the light source and r is the radius of the light source cylinder. The highlight band provides extra information through its width. Highlight bands in a plane normal to the light source $\mathbf{l}(t)$ reflect the shape of the surface. The bandwidth is directly related to the local shape and the local curvature of the surface in the intersecting plane normal to the light cylinder. A convex shape has a narrow band, a flat shape has a wider band and a concave shape has the widest band.

Since every point on a highlight band carries a distance d within the range $0 \leq d \leq r$ we can colour-code or grey-scale shade the point. For example a dark intensity indicates where a point passes through the centre of the cylinder light source ($d = 0$) and light intensity indicates that the normal extension is tangential to the light source cylinder ($d = r$), with white indicating the point is not a member of the highlight band. The colour-code or grey-scale indicates the path of the highlight band and provides information regarding the normal changes within the band.

3.6. *FANGA analysis*

FANGA analysis[14] is a method of assessing the quality of a surface by comparing the quality of a set of parallel planar intersects along a surface. On each intersect we determine the points $\mathbf{A}_1, \mathbf{A}_2$ and $\mathbf{A}_3$ with associated tangent angles $\alpha = \angle(\mathbf{C}_1, \mathbf{C}_3)$, $\beta = \angle(\mathbf{C}_2, \mathbf{C}_3)$ and $\gamma = \angle(\mathbf{C}_1, \mathbf{C}_3)$ where $\mathbf{C}_1$ is the chord $\mathbf{A}_3 - \mathbf{A}_1$, $\mathbf{C}_2$ is the chord $\mathbf{A}_2 - \mathbf{A}_3$ and $\mathbf{C}_3$ is the chord $\mathbf{A}_2 - \mathbf{A}_1$ (see Fig. 6)

We then plot the angle functions α, β, γ and the chord length $\mathbf{C}_3$ against the distance along the surface (perpendicular to the planes of intersect). The resulting curves are known as the FANGA curves. In practice it is sufficient to examine the

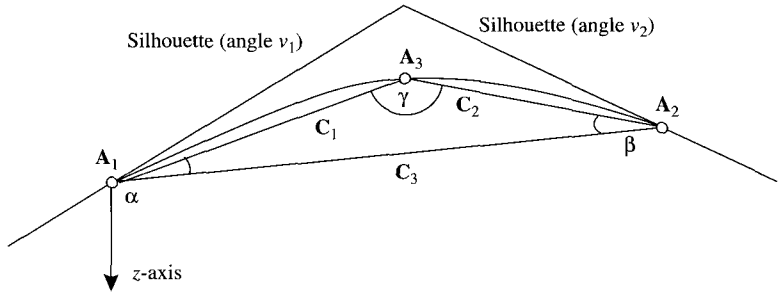


Fig. 6. The lengths and angles of FANGA analysis.

variation in $\mathbf{C}_3$. This can be achieved by finding the silhouette (Sec. 3.5) lines with viewing angles v_1 and v_2. These are then intersected with the intersect planes. Finally, we plot the distance between the silhouette lines intersection point against the distance along the surface (perpendicular to the intersect planes).

3.7. *Geometric parameter lines*

Geometric parameter lines were developed by extending methods for the geometric reparametrization of a curve $\mathbf{r}(u)$ with respect to either arc-length spanned or turned tangent-angle across the curve. Both of these schemes used the general reparametrization function

$$g(u) = \frac{\int_0^u G(u)du}{\int_0^1 G(u)du},$$

where a suitable function $G(u)$ for reparametrizing with respect to arc-length is

$$G(u) = \|\mathbf{r}_u(u)\| \quad \text{where } \mathbf{r}_u(u) = \frac{d\mathbf{r}(u)}{du}.$$

Extending these techniques for a parametric surface is straightforward, with boundary curves being reparametrized with respect to the chosen geometric invariant to produce a number of geometric points on each of the boundaries of a regular four-sided definition. The function $G(u)$ for reparametrization with respect to tangent-angle is given by

$$G(u) = \frac{\|\mathbf{r}_u(u) \times \mathbf{r}_{uu}(u)\|}{\|\mathbf{r}_u(u)\|^2}.$$

$G(u)$ for normal-angle reparametrization for the surface $r(u, v)$ is given by:

$$G(u) = \frac{\|\mathbf{n} \times \mathbf{n}_u\|}{\|\mathbf{n}_u\|^2} \quad \text{with } \mathbf{n} = \mathbf{r}_u(u, v) \times \mathbf{r}_v(u, v).$$

Equally spaced geometric points $\mathbf{g}_j, j = 0, \cdots, n$, can be generated along each of the four boundaries $\mathbf{r}(u_0, 0)$, $\mathbf{r}(u_1, 1)$, $\mathbf{r}(0, v_0)$ and $\mathbf{r}(1, v_1)$. Then the surface interior geometric points are generated using a linear blend of the underlying u and v parameter values between equivalent geometric points lying on opposite boundaries. For example, the opposite boundaries $\mathbf{r}(u_0, 0)$ and $\mathbf{r}(u_1, 1)$ with equivalent geometric points $\mathbf{g}_{0,j}$ and $\mathbf{g}_{1,j}$, provide geometric parameter lines, i.e. surface curves, given by $\mathbf{r}((1 - v)u_0 + vu_1, v)$, where $0 \leq v \leq 1$.

This is clearly an approximation to the true geometric parametrization and in practice produces geometric parameter lines that are dominated by the features on the boundary curves. The approximation can be improved by subdividing the surface with respect to the geometric parameter and using linear interpolation between the subdivided boundaries. This gives more information regarding the surface interior as illustrated in Fig. 7.

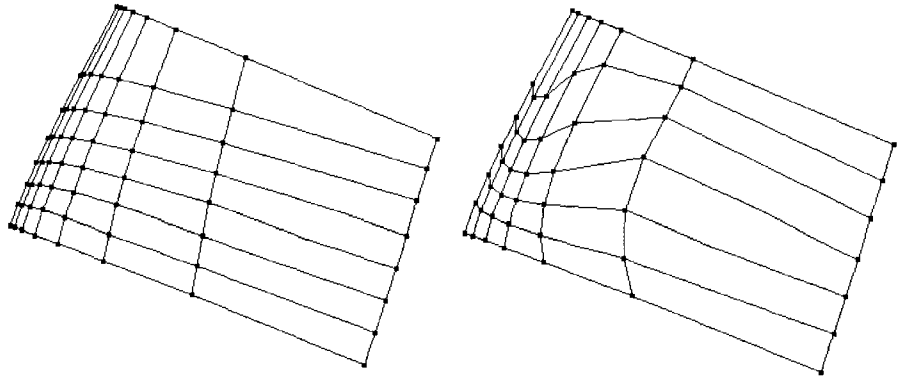


Fig. 7. Boundary-based and improved geometric parameter lines.

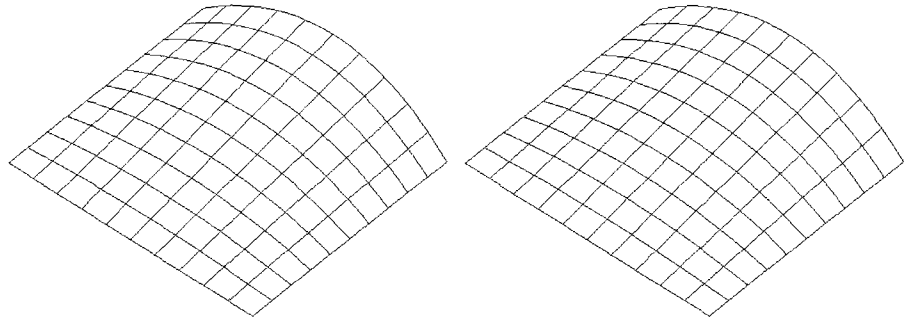


Fig. 8. (a) Isoparametric lines (b) Arc-length geometric parameter lines on a surface.

The geometric parameter lines are similar to isoparameter lines in that surface curves join equally valued u and v parameter points to produce an intersecting grid of u and v parameter lines, where the u, v parameters relate directly to the geometry of the surface rather than to any underlying representation (Figs. 8 and 9).

4. Standard Procedures

The standard procedures available for surface visualization have been well documented.[1,4,10,12,15,16,17,20] In increasing order of complexity these are isoparameter lines, planar intersects and curvature analysis. Other methods do exist, for example examining the nature of offset curves on a surface and geodesic paths,[10] but as yet are little used or understood. In general, the level of complexity of the interrogation relates to the amount of information generated about the surface which in turn relates to the level of difficulty of interpretation.

We shall review each method, with reference to the criteria above, and discuss the limitations of their use. We have chosen to illustrate each method by applying it

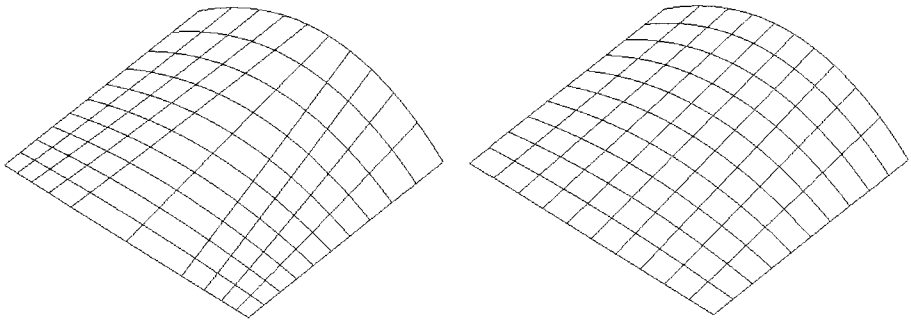


Fig. 9. (a) Tangent-angle geometric parameter lines (b) Normal-angle geometric parameter lines on a surface.

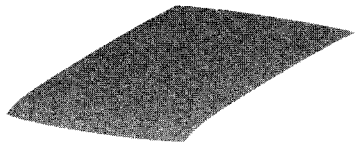


Fig. 10. A car roof.

to the same surface definition, namely the car roof shown in full in Fig. 10. However, due to its symmetry, we shall only illustrate the results of applying each method to half the definition. The approximate dimensions of the half-roof are 2015 mm in x, 450 mm in y and 65 mm in z. All plots are scaled (15 : 1) to give an indication of how a designer might view the information whilst working at a typical graphics screen.

At first glance the roof definition looks very simple and of not much interest. However, this is precisely why it has been used. We did not want the complexity of a definition to get in the way of the illustration of the methods performance. It is also worth bearing in mind two aspects of the NURBS-based approach. Firstly it creates surfaces from blends across opposite boundaries, so if these boundaries are incompatible, the mathematics has to work harder than it should to produce a quality surface. Looking at the roof, the front surface has incompatible boundaries, so it would be wise to pay particular attention to the quality of this definition in this region. Secondly joints between surfaces give the opportunity to introduce geometric flaws in the form of discontinuities that are unacceptable in downstream activities, especially nc machining and robot assembly. Again, particular attention should be given to this aspect of the quality and we shall highlight the performance of those techniques which aim at detecting such flaws.

4.1. *Isoparameter lines*

This is the simplest of the interrogation techniques and is based on surface point data. Isoparameter lines are simply lines of constant parametric values. Thus, given

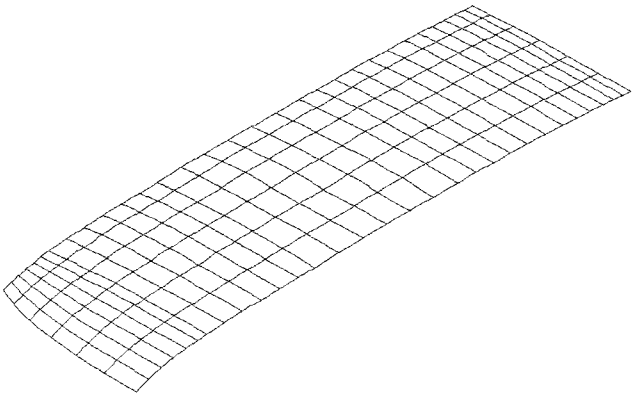


Fig. 11. Isoparameter lines on the car roof.

a surface $\mathbf{r}(u, v)$, an isoparameter line is generated by assigning a value v_c to the parameter v, say, and evaluating the surface curve defined by $\mathbf{r}(u, v_c)$. They are quick to compute and can give a good impression of the underlying surface shape. Figure 11 displays eleven equally spaced isoparameter curves in both the u and v directions. Gross imperfections will be visible as oscillating parameter lines, although more subtle features may be more difficult to detect visually. Figure 11 gives no indication of any geometric irregularities, especially at this scale.

This is especially true with high quality surface definitions that have been geometrically constructed—a good net of parameter lines virtually guarantees a quality surface. This technique is totally dependent on the underlying parametrization and works on a patch basis. Clearly, an isoparameter net is intrinsically a feature of the surface representation not the geometry of the surface itself. Thus a poor net of isoparameter lines may indicate a poor choice of parametrization or a poor surface.

An equally simple and quick alternative to plotting isoparameter lines is to plot the control polyhedron. Figure 12 displays the 6×6 polyhedron for each of the five biquintic NURBS surface patches of the roof definition. Like the isoparameter lines, a good polyhedron net indicates a quality surface. Again this is dependent on the representation and produces nets of vertices that exaggerate the features of the isoparameter lines. Like, Fig. 11, Fig. 12 indicates a good quality surface definition when viewed at this scale.

Thus any imperfections will be more visible as the oscillations in the polyhedron net are more exaggerated than the parameter lines. Since the isoparameter lines are derived from the polyhedron, the polyhedron can be manipulated to resolve problems in the isoparameters. This approach is very direct and still widely used. Figures 11 and 12 give no indication of surface imperfections at this scale.

Whilst this method is intuitive and can be used locally, it is neither geometric nor efficient, being poor at isolating surface imperfections and capable of giving false impressions. A greatly oscillating polyhedron net does not necessarily imply a

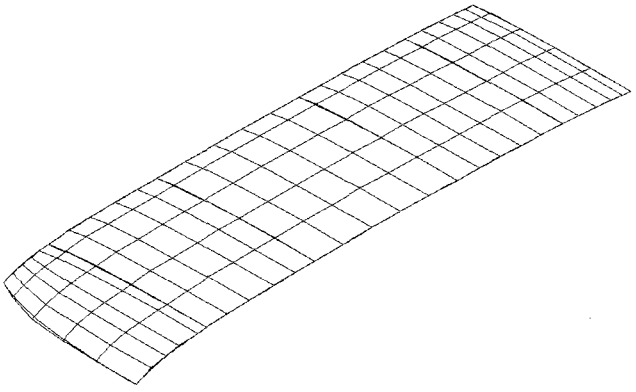


Fig. 12. Control polyhedron and patch boundary curves for the car roof.

poor surface. Robustness is another problem in that the step size can be important especially for small localized imperfections. Plotting only a sparse net of lines can hide imperfections that lie between the net. Like the isoparameter lines, a good polyhedron net indicates a quality surface. However, the exaggerated nature of nets of vertices can be even more difficult to interpret than the nets of isoparameter lines. Finally, the approach of manipulating the polyhedron to get a 'good' net may reduce the variation in shape which may result in 'bland' surfaces.

4.2. *Planar intersects*

Planar intersects[4] are often described by taking the analogy with a geographic relief map. We can indicate the general shape of the terrain by joining points of equal heights to give a series of contours or curves, the points of which are all the same height from sea level (our reference plane). The closer the contours are together the more rapidly the terrain is changing. The geometric interpretation of equally spaced sets of intersects is reasonably straightforward. Mathematically the intersection curve or contour, between a parametric surface $\mathbf{r}(u, v)$ and a plane having unit normal $\mathbf{p}$ satisfies $\mathbf{r}(u, v).\mathbf{p} - d = 0$, where d is the 'height' of the planar intersect.

The criteria used when analyzing intersect curves are that they are individually good and that as a set they are in harmony with their neighboring intersects. The variation from one intersect to the next must be pleasing to the (trained) eye with no unnecessary oscillations. Planar intersects are truly geometric and therefore tell us more about the underlying shape of a parametric surface than the isoparameter lines. A set of such intersects is given in Fig. 13 and is seen to be regular (the spacing between the contours do not oscillate) at this scale, indicating a good surface.

Intersects are geometric in their construction and are intuitive in their interpretation. However, they are computationally more difficult to generate than isoparameters and a large number of sections may be needed to detect all imperfections in a

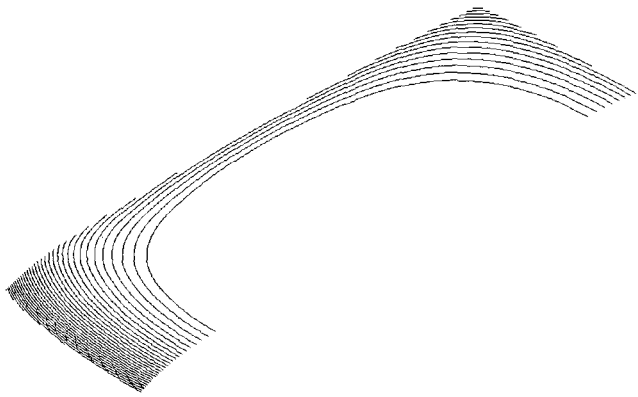


Fig. 13. Planar intersects on the car roof.

definition. The direction and spacing of the intersects cannot be chosen at random, although a natural direction is usually prescribed. For the car roof the natural coordinate for the intersects is the z-component since the roof will have been defined, in general terms, by curves in the x, z plane (sections) with user-defined x values and curves in the y, z plane (longitudinals) with user-defined y values. Ideally the spacing of the contours should be constant to make interpretation easier. This can result in parts of the definition where the intersects are too close and parts of the definition where they are too far apart to give meaningful information.

In Fig. 13 there are 41 z contours in the range 1169.602–1210.2815 mm. The main body of the roof looks acceptable. However, little can be deduced for the front and rear corner of the roof because the contours are too tightly packed due to the scale of the plot. Finally, for a high quality surface, any imperfections would manifest themselves as oscillations in the intersects. Such oscillations are typically very small and may be difficult to detect on a workstation screen.

4.3. *Curvature analysis: contours and lines of curvature*

Curvature contours

The most convenient way of viewing curvatures is via a curvature contour map[10] where colour can be used to enhance the variation in the scalar quantities. The two most common methods are to assign colours to the curvature values directly or to the curvature values on a logarithmic scale. For the following figures the range of curvatures is $4.00568e^{-06}$ (white) to $8.60942e^{-09}$ (black).

Figure 14 illustrates the mean curvature distributions for the car roof (increasing from white to dark grey). There is a regular transition of curvatures from the roof centre to the outboard line giving an impression of a quality surface. However there is a marked irregularity at the front of the roof that indicates a subtle geometric flaw in the definition. Given the location of the irregularity, i.e. in the vicinity of

a patch boundary, this is most likely indicating a discontinuity in curvature across the common boundaries. It is worth noting that neither the isoparameter lines nor the contour plots were able to detect this feature.

Figure 15 illustrates the Gaussian curvature distributions for the car roof (increasing from white to dark grey). This gives the same overall impression as Fig. 14 although the irregularity is slightly less marked.

Figure 16 illustrates the mean curvature distributions on a logarithmic scale for the car roof (increasing from white to dark grey). Again, there is a regular transition of curvatures from the roof centre to the outboard line giving an impression of a quality surface. In contrast to Fig. 14 there is less irregularity in the contours at the front of the roof but there is a distinct line along the join between the first two patches. This gives an even stronger indication of a discontinuity across the common boundaries.

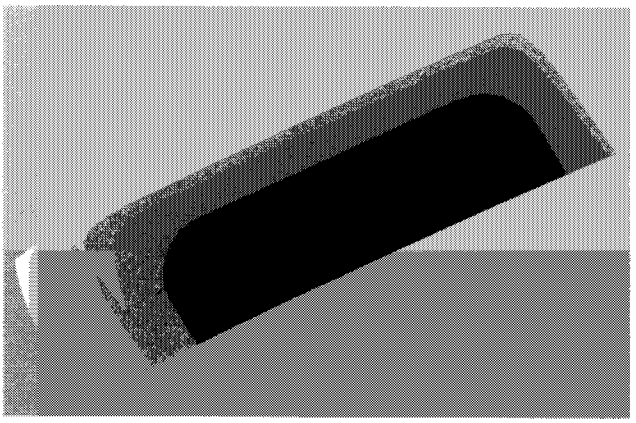


Fig. 14. Mean curvature contours on the car roof.

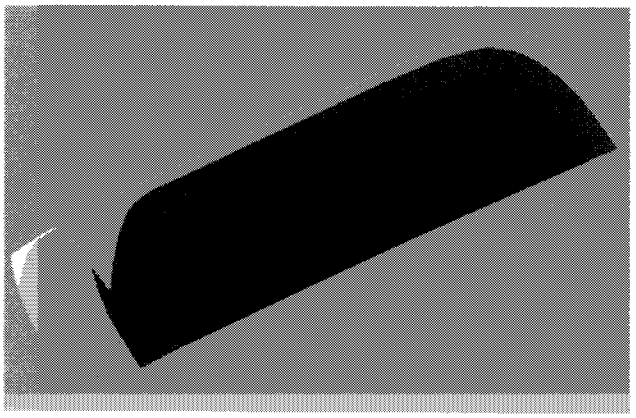


Fig. 15. Gaussian curvature contours on the car roof.

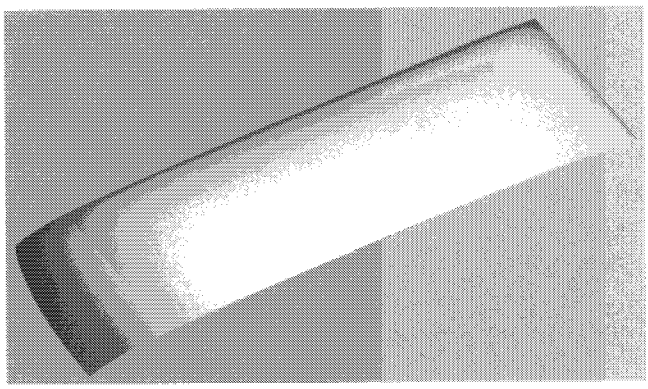


Fig. 16. Mean curvature contours on a logarithmic scale on the car roof.

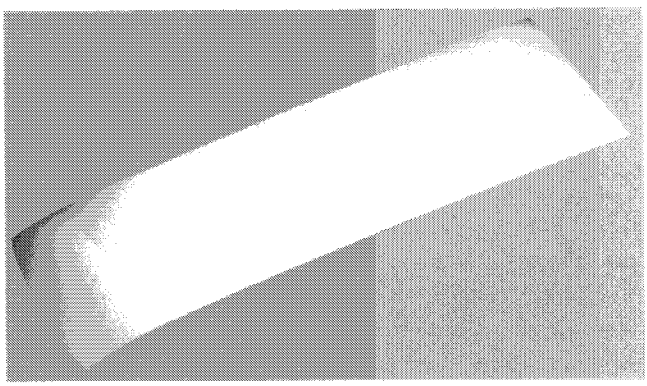


Fig. 17. Gaussian curvature contours on a logarithmic scale on the car roof.

Figure 17 illustrates the Gaussian curvature distributions on a logarithmic scale for the car roof (increasing from white to dark grey). This gives the same overall impression as Fig. 16 although the irregularity is slightly less marked. This is due to a less than optimal choice of range of curvature values.

Although the curvature plots illustrated above have given insight into the quality of the roof definition, it is important to note that the choice of the range of curvature values was of great importance. A poorly chosen range would completely mask the irregularity at the front of the roof and as with all the assessment techniques, the range and the number of interrogations is difficult to pre-determine.

Lines of curvature

Rather than looking at scalar quantities we can construct lines of curvature whose tangent at every point is in a principal direction. These lines indicate a directional flow for $k_1 = \max$ and $k_2 = \min$ over the surface. Unfortunately, the evaluation

of lines of curvature requires the numerical solution to two coupled non-linear differential equations. To avoid the large computational overhead of generating lines of curvature, an alternative is to display principal directions directly onto a (reasonably) sparse grid of surface points.[10] There are also problems associated with spherical points(umbilics) where the direction of the lines of curvature may become undefined. Here, special techniques have to be used which further complicate the interpretation of the lines.

The curvature-based methods do not have a manual equivalent and therefore the design engineers are unlikely to have any practical working knowledge to apply to aid interpretation. The advantage of these methods is that all the information about the surface is given. The resulting pictures are easy to see with imperfections causing irregularities in the curvature contours and lines. However the interpretation of the resulting pictures is non-intuitive and very difficult since the choice of the range of colours and the range of k will affect what is seen. Furthermore the robustness of the method is dependent on the range and the number of curvature lines plotted. As with all the curvature-based methods, the number and location of the lines are crucial in detecting imperfections and hence are of paramount importance to the assessment of quality. Finally, these methods are global, i.e. they are independent of patches, thus local discontinuities may be difficult to detect as they may appear regular when viewed locally.

To summarize, both the curvature-based methods and planar intersects give geometric lines and are therefore useful for confirming a quality definition. They are also potentially very effective for detecting imperfections. Imperfections manifest themselves as irregularities in contours and curvature lines but are difficult to interpret and hence relate back to the original geometry. Isoparameter lines are easy to interpret in terms of the surface representation but their effectiveness for quality assessment is reduced because they do not relate to the geometry. What is needed is a way of generating geometric parameter lines on a surface that are easy to interpret and relate directly to the geometry, i.e. geometric parameter lines.

In the next section we review several techniques that have been proposed to aid interpretation of the contour and curvature techniques and we introduce geometric parameter lines.

5. Improving the Effectiveness of Standard Procedures

5.1. *Curvature analysis: isophotes and reflection lines*

Reflection lines and isophotes mimic the manual approach of surface assessment by shining a number of beams of light at a surface and looking at the patterns generated by the light reflected off the surface. In effect they measure the variation over the surface of the unit surface normal and hence are related to the curvature methods and are generally preferred to curvature plots.

The reflection lines[12] on the surface produced for a fixed light source have a constant angle of reflection off the surface. They can be plotted either as discrete

lines or colour (grey-scales) shaded between lines of constant intensity. The most common implementation uses direct ray-tracing with both ambient and directional light sources, usually of equal intensity. The ambient light gives an overall illumination to the surface and the directional light source gives the reflection line contrasts. Using a dynamic light source location enables the surface characteristics to be viewed under varying lighting conditions. This approach is very sensitive to the quality of the surface. More importantly, it is dependent on the choice of range of interrogations and the range of colours. Imperfections at the limits of the ranges are easily masked by the saturation effects of the colours at the limits.

Figure 18 illustrates a set of eleven reflection lines from the roof with a fixed light source positioned at $\langle 2700.00, -170.00, 1700.00\rangle$ with light direction $\langle -1000.00, 47.00, 1200.00\rangle$. The set of lines appear smooth and in reasonable harmony with each other. As they relate to the curvature of the surface, these lines suggest a quality definition. They give no indication of any continuity problems across the patch boundaries especially in the region suggested by the mean curvature plot (Fig. 14).

Isophotes[20] are similar in concept to reflection lines. They are lines of constant light intensity where the surface is illuminated by parallel beams of light. Geometrically, an isophote is a line on which the angle α between the surface normal and the direction of the parallel light beams is constant. Connecting all such points of the surface results in an isophote line. When $\alpha = 0°$ we obtain silhouette lines. It can be shown that smooth isophotes indicate a good quality surface definition whereas irregularity of the isophotes indicate irregularities in the first and second derivatives of the surfaces. The technique can differentiate between different orders of discontinuities but relating the possible causes to the underlying definition is difficult.

Figure 19 illustrates a set of 16 silhouettes on the car roof. The light vector direction is $\langle 0.0, -0.5, 1.0\rangle$ with an incremental vector of $\langle 0.0, -3.0, 0.0\rangle$. As with

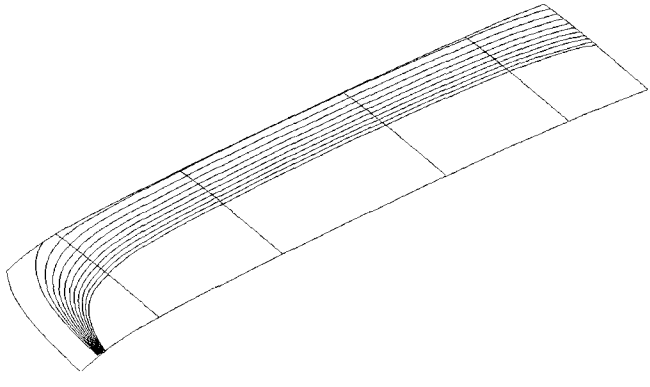


Fig. 18. Reflection lines on the car roof.

the reflection lines, the set of silhouettes are regular and appear to be in harmony with each other. There is no indication of any surface problems relating to the curvature and one would conclude that the quality of the definition was acceptable.

Figure 20 gives a set of isophote lines for the car roof. The light source is in the z-direction and the viewing angle ranges from 166° to 178°. Again these lines are smooth and regular and appear in harmony with each other over the majority of the surface. However across the join of the front two patches the isophotes have a marked slope discontinuity. This gives a strong indication of an undesirable feature between the front two surface patches.

These methods are highly sensitive and can give seemingly poor results because easily detectable imperfections dominate. They are also very sensitive to the choice of direction of the light source. Surfaces that produce regular isophotes and reflection lines, whatever the direction of the light source, are of a very high quality thus these methods are useful for confirming the quality of a good surface definition.

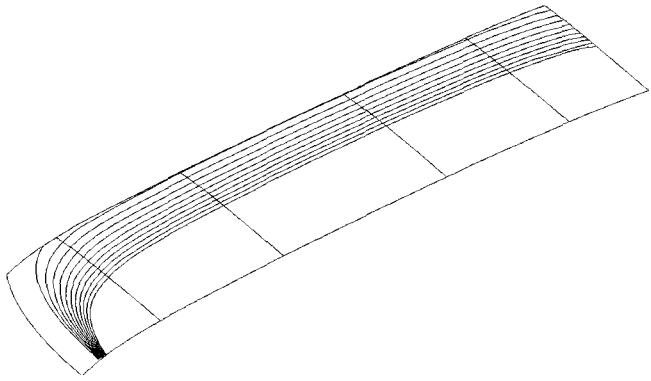


Fig. 19. Silhouettes on the car roof.

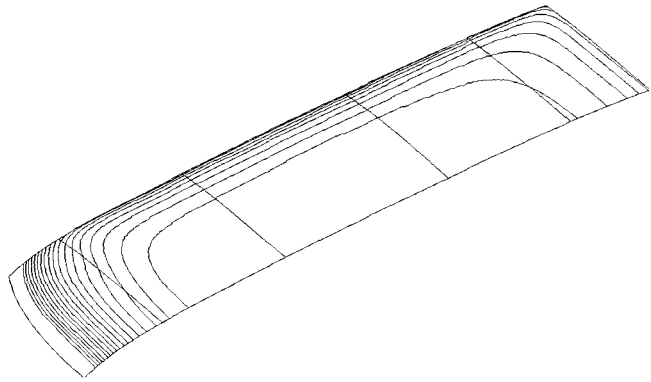


Fig. 20. Isophote lines on the car roof; light direction [0,0,1] angles 166°–178°.

5.2. *Highlight lines and bands*

Figure 21 illustrates a single highlight line on the car roof. This gives the impression of a quality definition along the length of the highlight line. The highlight line gives information related to the curvature along the line and this indicates that the surface curvature is acceptable, at least in the neighbourhood of this line. Contrast this with the curvature plots in Figs. 14–17.

Figure 22 illustrates a highlight band on the car roof. Again, along the length of the highlight band we have the impression of a quality surface. The highlight band gives information related to the curvature in the region of the band. The width of the band suggests a convex shape to the roof surface in the direction of the highlight band, which is correct. However little can be deduced from this plot except that

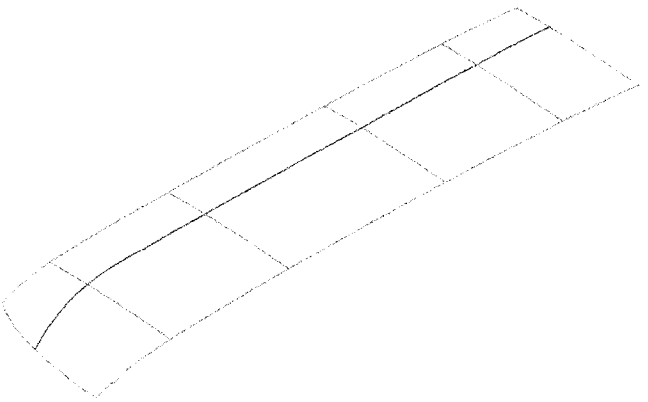


Fig. 21. Highlight lines on the car roof; light source $(2000, 400, 3000)$, direction $(50, 0, 0)$.

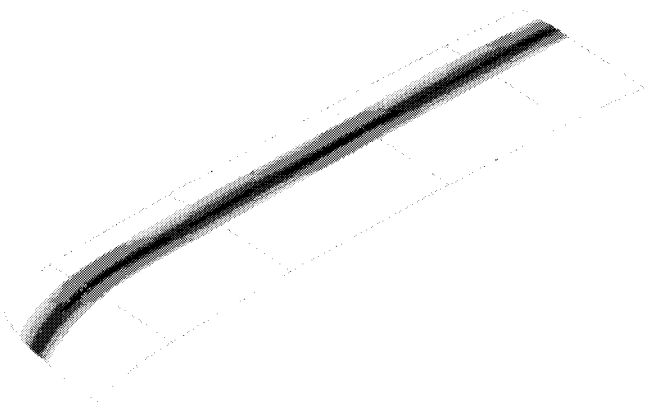


Fig. 22. Highlight bands on the car roof; light source $(2000, 400, 3000)$, direction $(50, 0, 0)$, radius 1000.

the surface curvature is acceptable, at least about this band. Contrast this with the curvature plots in Figs. 14–17.

Whilst these techniques make interpretation of basic geometric imperfections easier, for example position and tangent discontinuities, the general interpretation of curvature-based methods is still extremely difficult and not well understood. These methods are also global and therefore even more difficult to use within the constraints of a workstation screen.

5.3. *Contour interpretation: FANGA analysis*

To help interpret planar intersects SAAB have developed a technique for assessing the quality of a surface by relating points on sets of intersects. Intersects give more insight into the surface geometry as the angle of the intersecting plane and the angle of the surface normal reduces. By connecting points on intersects having the same slope, the resulting curve is a silhouette line of the surface when viewed at that slope angle. The shape variation of a surface can then be assessed by viewing a range of silhouette lines.[14] Since the spacing of the silhouette lines indicates the rate of change of the tangent angle in the intersects, comparing sets of silhouettes gives an insight into the turning of the normals which in turn relates to the curvature of the surface.

FANGA analysis is based essentially on the variation of chord length distances and angles between pairs of silhouette lines. Any surface imperfection will cause an irregularity in the variation of the triangle defined by the two silhouette lines and the shoulder line as shown in Fig. 6. It is therefore a highly sensitive method of detecting imperfections in surface quality. A surface that exhibits regularity of intersects and associated silhouette lines is virtually guaranteed to be a high quality surface.[4] SAAB has developed an optimization procedure for correcting surfaces based on FANGA analysis.[15] Figure 23 shows three silhouette lines, the centreline ($y = 0$) and two viewed at approximately 20° and 70° to the y-axis. Figure 24 shows the

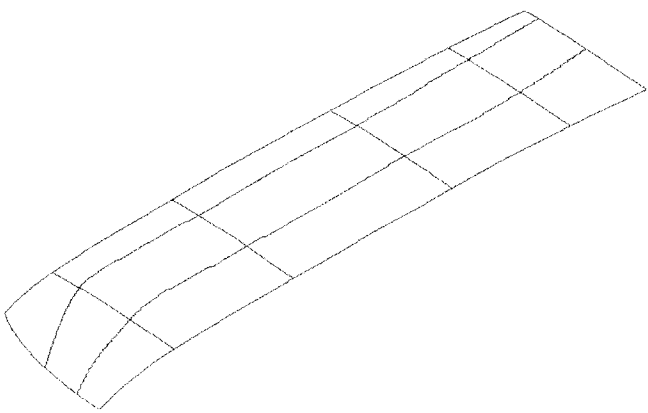


Fig. 23. Two silhouettes on the car roof; light source $(2000, 400, 3000)$, direction $(50, 0, 0)$.

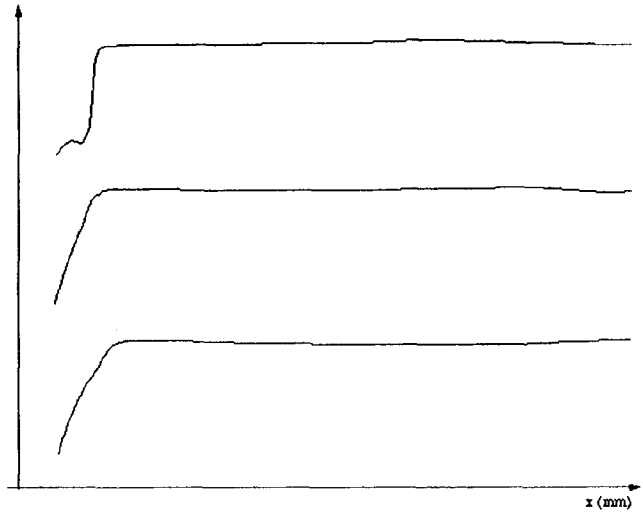


Fig. 24. FANGA curves for the silhouettes of Fig. 23.

resulting FANGA curves. The top FANGA curve resulting from the top silhouette, clearly illustrates a different character at the start of the curve than the other two. This indicates a problem with the geometry of the definition in this region.

It is clear that these techniques aid in the interpretation of both the planar contours and curvature methods. However, they do not address the problem of accessibility to design engineers. To understand the resulting pictures one needs a good understanding of the underlying mathematics. We therefore return to the idea of simple isoparameter lines. Clearly we are not looking to replace the methods based on contours or curvatures, after all these methods do tell us everything about the surface. We wish to develop techniques that meet the criteria outlined at the beginning of this chapter that give the same information but in a more accessible form.

5.4. *Geometric parameter lines*

To make isoparameter lines more effective at detecting imperfections we need a method of relating them to the geometry. Rather than interrogating a surface at regularly spaced parameter values $\mathbf{r}(u, v)$, which is intrinsically a feature of the representation, it is possible to sample the surface patch at regularly spaced values of some geometric metric.[8]

This approach combines the intuitive interpretation of isoparameter lines with geometric information about the definition. This method, like isoparameter lines, is a patch-based interrogation and ideally suited to working at a screen.

We can choose other geometric metrics, for example based on the total turning angle of the tangent along each boundary or the total turning angle of the surface

normal along each boundary. Essentially arc-length is a function of the first parametric derivative information, tangent angle is a function of the first and second parametric derivative information and normal angle is a function of both u and v parametric derivative information.

Thus the sensitivity of these geometric schemes increases from arc-length to tangent angle to normal angle. Hence we should look at these interrogations in this order. We would expect 'regularity' of the geometric parameter lines (arc, tangent and normal) in a quality definition. Figures 25–27 illustrate patch-based arc-length, tangent angle and normal angle parameter lines respectively on the car roof. The arc-length lines are regular whilst the angle-based lines show irregularities in the front region. This is consistent with the curvature interrogation (Figs. 14 and 15) which is highly encouraging since it highlights the previously suggested geometric flaws.

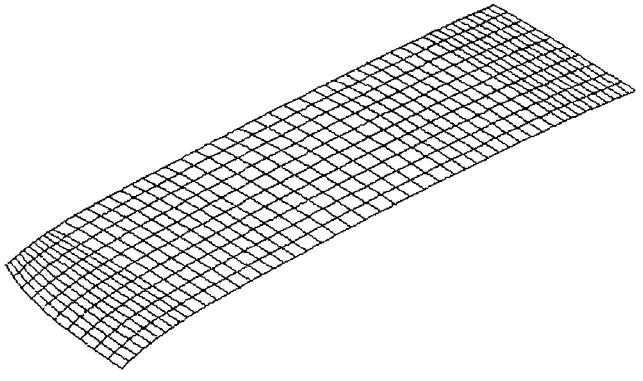


Fig. 25. Geometric arc-length parameter lines on the car roof.

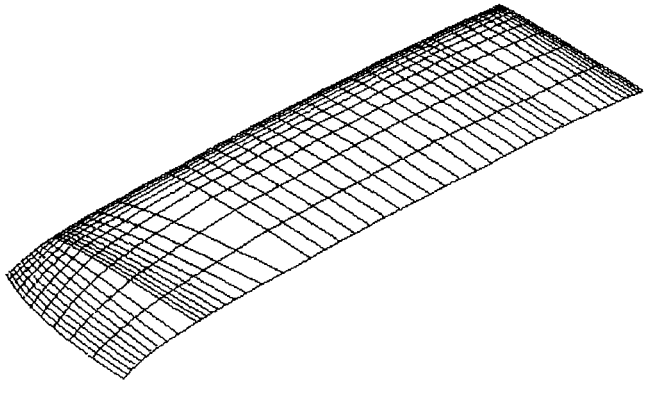


Fig. 26. Geometric tangent angle parameter lines on the car roof.

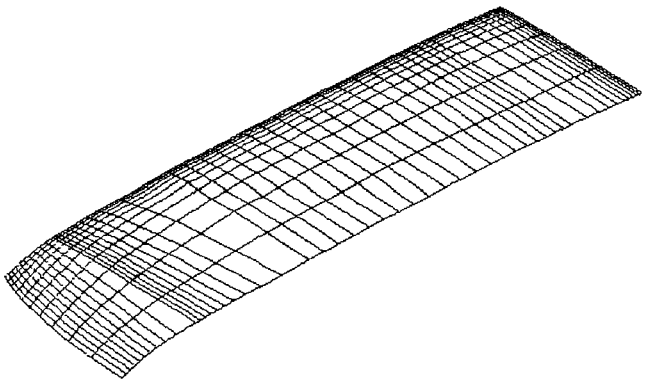


Fig. 27. Geometric normal angle parameter lines on the car roof.

6. Hierarchical Procedure

To make the assessment of quality more intuitive and applicable to working at a screen, we propose a hierarchical procedure first to detect gross imperfections and then to look for more subtle imperfections in an iterative sense in the same way as a sculptor would use different sizes of chisel to smooth the workpiece. The approach addresses the problems of gross geometric features swamping the more subtle features of a surface definition and acknowledges that surface quality is directly linked to functionality. Thus an injection mould tool maker would not wish to produce a computer-based definition to the same exacting quality requirements of, for example, an exterior panel tool for an automobile.

The proposed procedure uses the following methods:

(i) Geometric parameter lines: arc-length; tangent angle; normal angle;
(ii) Isophotes;
(iii) Curvature methods.

By using the geometric parameter lines in the order arc-length; tangent angle; normal angle, we are able to detect finer and finer imperfections. The expectation then is that FANGA analysis would be used to confirm the quality of the definition or to detect extremely fine imperfections. Finally the curvature-based methods could then be used on high quality definitions as an overall visual check. It is not expected that any further problems would be highlighted.

7. Conclusions

As we have seen all the current methods of assessing the quality of computer-based surface definitions at a workstation screen are less than satisfactory. They require a high degree of skill both to use and to interpret the resulting pictures. All methods suffer from the choice of location and number of interrogations. Curvature-based methods are difficult to interpret, but with the aid of isophotes and reflection lines,

they are a good method which are capable of visually confirming the quality of surfaces. Planar intersects, when coupled with FANGA analysis, give a method of ensuring high quality definitions. Recognizing the limitations of isoparameter lines, we propose a series of geometric parametrizations that give a graded method for detecting geometric imperfections that are easy to interpret on a screen. Furthermore, any imperfections can be related back to the geometry and a corrective procedure prescribed. A hierarchical approach has been proposed that uses geometric parametrizations to detect imperfections and then uses the strengths of existing methods for quality confirmation. The research into characterizing geometric parametrizations is in its early stages, but having related 'irregularity' of surface definition to the geometric interrogations; we can then define algorithms for removing or improving surface quality by regularizing the geometric parameter lines or isophotes. For example, the tangent angles may be varying too quickly at one end of a curve in relation to the other. This will be detected by the geometric parametrization and therefore gives guidance to correction.[5]

Acknowledgments

The authors are pleased to acknowledge the support of EPSRC (Grant No GR/K 31268) and Delcam plc. The diagrams of the geometric parameter lines were generated by Robert Howe and the surface definitions were kindly supplied by British Aerospace, Clarks Shoes Ltd and Rover Group.

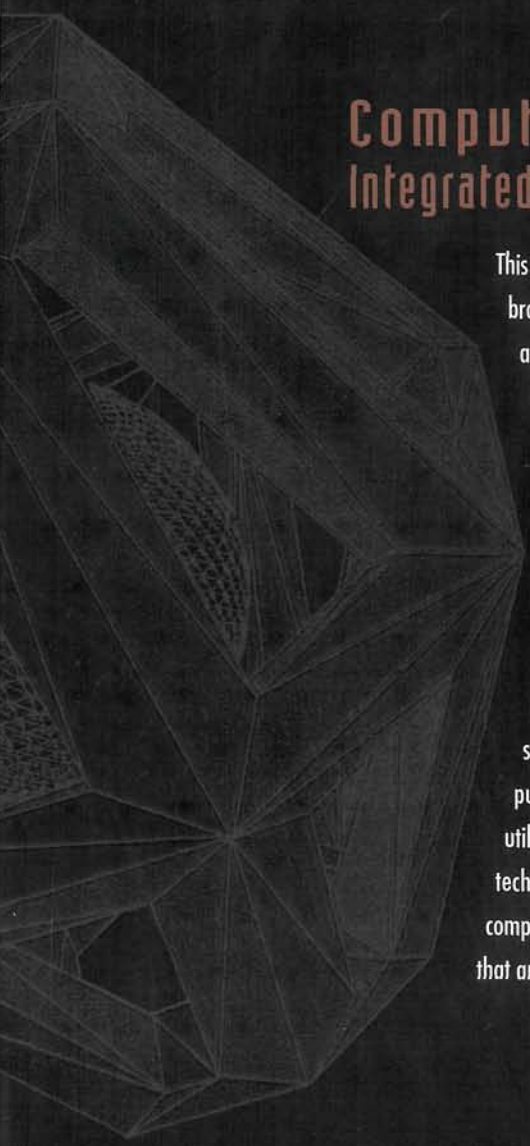
References

1. K. P. Beier and Y. Chen, Highlight-line algorithm for real time surface-quality assessment, *Computer Aided Geometric Design* **26**, 4 (1994) 268–277.
2. W. Böhm, G. Farin and J. Kahmann, A survey of curve and surface methods in CAGD, *Computer Aided Geometric Design* **1**, 1 (1984) 1–60.
3. W. Böhm, Differential geometry II. In G. Farin, *Curves and Surfaces for Computer Aided Geometic Design: A Practical Guide, 2nd Edition* (Academic Press, 1993).
4. R. J. Cripps and A. A. Ball, Visualization and quality assessment of free-form surfaces, *Proceedings of the Institution of Mechanical Engineers, Part B* **212** (1998) 207–214.
5. R. J. Cripps and R. E. Howe, Surface visualization and assessment using geometric parameter curves, *Advances in Manufacturing Technology XII*, Eds. R. W. Baines, A. Taleb-Bendiab and Z. Zhao (Professional Engineering Publishing Ltd, 1998) pp. 335–342.
6. R. J. Cripps and S. A. Barley, A geometric characterization of springback in drawn panels, *Mathematical Engineering in Industry* **3**, 3 (1992) 205–214.
7. C. Clarke, Review of EDS, *Unigraphics 11.0. CADCAM* (1996) 25–28.
8. A. M. Czerkawski, *Fitting Procedures for Curves and Surfaces*, PhD Thesis, University of Birmingham, UK, 1996.
9. G. Farin, *Curves and Surfaces for Computer Aided Geometic Design: A Practical Guide, 2nd Edition* (Academic Press, 1993).
10. R. T. Farouki, Graphical methods for surface differential geometry, *The Mathematics of Surfaces II*, ed. R. R. Martin (Oxford University Press, 1987) 363–385.

11. N. Guid, C. Oblonsek and B. Zalik, Surface interrogation methods, *Computers and Graphics* **19**, 4 (1995) 557–574.
12. H. Hagen, S. Hahmann, T. Schreiber, Y. Nakajima, B. Wördenweber and P. Hollemann-Grundstedt, Surface interrogation algorithms, *IEEE Computer Graphics and Applications* **12**, 9 (1992) 53–60.
13. R. Klass, Correction of surface irregularities using reflection lines, *Computer Aided Design* **12**, 2 (1980) 73–77.
14. G. Liden and A. A. Ball, Intersection techniques for assessing surface quality, *The Mathematics of Surfaces V*, ed. R. B. Fisher (Oxford University Press, 1993) 363–385.
15. G. Liden and S. K. E. Westberg, Fairing of surfaces with optimization techniques using FANGA curves as the quality criterion, *Computer Aided Design* **25**, 7 (1993) 411–420.
16. T. Maekawa, F. E. Wolter and N. M. Patrikalakis, Umbilics and lines of curvature for shape interrogation, *Computer Aided Geometric Design* **13**, 2 (1996) 133–161.
17. H. P. Moreton, Simplified curve and surface interrogation via mathematical packages and graphical libraries and hardware, *Computer Aided Design* **27**, 7 (1995) 523–543.
18. A. W. Nutbourne and R. R. Martin, Differential geometry for curve and surface design, Vol. 1, *Foundations* (Ellis Horwood, 1982).
19. L. Piegl, On NURBS: A survey, *IEEE Computer Graphics and Applications* **11**, 1 (1991) 55–71.
20. T. Poeschl, Detecting surface irregularities using isophotes, *Computer Aided Geometric Design* **1**, 2 (1984) 163–168.
21. D. J. Struik, *Lectures on Classical Differential Geometry, 2nd Edition* (Dover, NY, 1961).

This page is intentionally left blank

INDEX



Computer Aided and Integrated Manufacturing Systems

This is an invaluable five-volume reference on the very broad and highly significant subject of computer aided and integrated manufacturing systems. It is a set of distinctly titled and well-harmonized volumes by leading experts on the international scene.

The techniques and technologies used in computer *aided and integrated manufacturing systems* have produced, and will no doubt continue to produce, major annual improvements in productivity, which is defined as the goods and services produced from each hour of work. This publication deals particularly with more effective utilization of labor and capital, especially information technology systems. Together the five volumes treat comprehensively the major techniques and technologies that are involved.

World Scientific
www.worldscientific.com
5249 hc

ISBN 981-238-339-5(set)
9 789812 383396

ISBN 981-238-981-4
9 789812 389817