Optimal Design For Injection Molding

Kabanemi, Kalonji K.; Derdouri, Abdessalem; Hétu, Jean-Francois

Publisher's version / la version de l'éditeur:

Injection Molding. Technology and Fundamentals, 11, pp. 511-552, 2009

IMI2005-109447-G
CNRC 47913

Optimal Design For Injection Molding

Kalonji K. Kabanemi, Abdessalem Derdouri and Jean-François Hétu

Numerical Modelling of Processes, Industrial Materials Institute (IMI)
National Research Council of Canada (NRC)
75 de Mortagne Blvd., Boucherville, Québec, Canada

Table of Contents

(21 figures, 83 references)

1 Introduction

Over the last twenty years, much progress has been made in the development of models and algorithms capable of predicting different phases involved in the injection molding process. Extensive research work in computer modeling for injection molding has been conducted by many research groups, including the Cornell Injection Molding Program (CIMP) [1-11], Université Catholique de Louvain (UCL) [12-22], IKV [23-27], Ecole des Mines de Paris (CEMEF) [28-31], Industrial Materials Institute (IMI) [32-37], McGill University [38-42], University of Illinois [43-44], Western Electric Company [45-46], among others. Basically, two approaches have emerged. The first uses finite difference method with the grid generation technique and the second uses finite element method with either the volume of fluid method or an automatic remeshing technique for the flow-front tracking. Most of the above numerical methods are based on the lubrication approximation to formulate the mold filling problems, although a number of recent papers use a fully three-dimensional approach [23, 24, 29, 30, 33, 34] to address specific problems that need to solve the full Navier-Stokes equations. As far as final properties and dimensional stabilities of parts are concerned, either elastic or viscoelastic formulation is used to model the polymer behavior during cooling and ejection phases. As a result of the above research, a clear understanding of the flow behavior, thermal exchanges, dimensional stability and mechanisms of stress generation is well established.

In spite of many assumptions and approximations used in the mathematical description of the physics of the underlying problem, nowadays, commercial injection molding simulation software, e.g., Moldflow [47] and Moldex [48], are, nevertheless, routinely used as design optimization tool, to analyze the entire injection molding process, from the filling stage to the ejection. Information collected from simulation, i.e., the temperature distribution during filling, the position of weld-lines, possible short shot problems, warpage and residual stresses, are used to manually modify, by trial-and-error, the initial design, e.g., processing conditions, gate sizes and locations, runner systems and the layout of the cooling systems, to obtain the best design from the available alternatives.

While this 'intuitive' approach is attractive for simple geometries and small number of design variables, it becomes quickly time consuming for complex geometries and results in multiple manual modifications of the design before the best or optimal one is finalized. In the present chapter, we shall be concerned with the prediction of the optimal design, where manual modification by trial-and-error becomes tedious and the need for numerical optimization techniques is essential.

The spectacular progress made in numerical modeling of the injection molding process [1-46], in conjunction with available efficient optimization techniques used in many engineering applications [49-56], have led researchers to develop design optimization tools for injection molding [57-69]. Some of them combine existing injection molding simulation codes with existing numerical optimization program.

These optimization techniques are either, gradient-based methods that use the gradients of the objective and constraint functions, or direct search methods that use only function values. Such an integrated optimization tool, that couples the injection molding program C-Mold/Moldflow [47] with a commercial design optimization tool, were successfully used by Turng and Peic [67] to optimize the cycle time and the volumetric shrinkage. Lame et al. [68], Smith et al. [62-64], and Kabanemi et al. [65-66] also used such an integrated optimization system to determine the optimal design for injection molding. The integrated approach enables the selection and employment of the most suitable optimization algorithm depending on the user's specific needs while using well-developed CAE technologies for injection molding. Therefore, for the sake of clarity in this chapter, we will find it useful to recall the basic equations for the mold filling problems along with various optimization techniques.

The aim of the present chapter is to show how gradient-based methods and the design sensitivity analysis (DSA) can be applied to determine optimal design for injection molding. The chapter is organized as follows: first, we briefly present the basic equations for the injection molding process based on the Hele-Shaw approximation. Next, we introduce the optimization concept and the optimization solution methods. In section 4, we present the design sensitivity analysis along with the direct differentiation method and the adjoint method, to evaluate the sensitivities of the flow equations and the objective functional. Finally, we discuss numerical results involving complex non-planar mold geometries.

2 Basic Equations for the Mold Filling Problem

2.1 Mathematical Model: Hele-Shaw and Energy Equations

In order to describe the flow behavior of a polymer fluid during filling of a mold cavity, an expression for the stress tensor has to be specified. It is outside the scope of this chapter to treat the derivation of constitutive equations in detail. We refer the reader to specialized literature reviews by Bird et al. [70] and Crochet et al. [71], among others.

For an incompressible fluid, the Cauchy stress tensor, is decomposed as follows

$$\sigma = -p\mathbf{I} + 2\eta\mathbf{D} \qquad (1)$$

where p is the pressure, η is the viscosity and $\mathbf{D}$ is the rate-of-strain tensor, from which the shear rate is defined as being the invariant

$$\dot{\gamma} = 2(\mathbf{DD})^{1/2} \qquad (2)$$

In what follows, we study the creeping flow of a generalized Newtonian fluid, with a viscosity dependence on the temperature T and the shear rate $\dot{\gamma}$. In addition, we consider processes with fluid flow through thin cavities of slowly varying cross section and arbitrary in-plane dimensions. Then we derive the continuity and the momentum equations using the lubrication approximation and we average the continuity equation over the thickness of the mold cavity. For a full description of the numerical modeling of the injection molding process, we refer the interested reader to the literature review by Hieber and Shen [3], Dupret et al. [12] and Tucker [43-44], among others. We shall here only briefly recall the basic results of the model.

Taking into consideration the above assumption, the resulting boundary value problem in the mid-surface of a mold cavity is given by the Hele-Shaw equation for the pressure

$$\nabla \cdot (-S(p,T)\nabla p) = 0 \text{ in } \Omega \qquad (3)$$

where p is the pressure field in the flow domain Ω. According to Hele-Shaw theory and the lubrication approximation, the pressure variation in the gape-wise direction is negligible.

The fluidity parameter S, in Eq. 3, is given by

$$S(p,T) = \int_0^h \frac{\zeta^2}{\eta(\dot{\gamma},T)} d\zeta \qquad (4)$$

where h is the local half-thickness of the cavity. The viscosity, η, is given in the present work, by the Carreau-Yasuda model of the form

$$\eta(\dot{\gamma},T) = \eta_0 A_T [1 + (\tau A_T \dot{\gamma})^{\alpha}]^{(n-1)/\alpha} \qquad (5)$$

In Eq. 5, α is a model parameter, n the power index, τ a characteristic relaxation time and η_0 the zero shear rate viscosity. The temperature dependence of the viscosity and relaxation time is taken into account by a shift function A_T, which expresses time-temperature equivalence and is defined by a WLF-type relationship

$$\log A_T = -\frac{c_1(T - T_r)}{c_2 + T - T_r} \qquad (6)$$

where T_r is a reference temperature, and c_1 and c_2 are material constants.

We further need an equation that describes the temperature field during the filling stage. This is written in the following simplified form

$$\rho c \frac{DT}{Dt} = \nabla \cdot (k\nabla T) + 2\eta \mathbf{D} : \mathbf{D} \qquad (7)$$

where ρ, c and k are the density, heat capacity and thermal conductivity, respectively.

Since we consider processes with fluid flow through thin cavities, the heat conduction is dominated by the gape-wise direction, while the in-plane flow dominates the convection. While the flow equation for the pressure, Eq. 3, is locally a two dimensional equation that vary only in the mid-surface of the cavity, the energy equation, Eq. 7, remains a purely three-dimensional equation.

2.2 Boundary Conditions

In order to complete the mathematical description of the filling stage for the injection molding problem, we must specify appropriate boundary conditions.

The Hele-Shaw equation, Eq. 3, involves boundary conditions of three kinds:

1. At the flow front, a vanishing value of the pressure is specified;
2. On the lateral sides of the mold, a vanishing pressure normal derivative is imposed;
3. At any gate, a flow rate profile or a pressure profile is imposed;

In general, the energy equation is a balance between heat advection in the mid-surface direction, heat diffusion in the gap-wise direction and viscous heat. The thermal wall boundary condition takes into account the transient heat transfer inside the walls, and requires as input the coolant temperature and the heat transfer coefficient. At the flow front, an equivalent instantaneous heat advection in the gap-wise direction is applied to account for the fountain flow effect. At any gate, an injection melt temperature is imposed.

2.3 Numerical Discretization

Finally, one needs to solve numerically Eqs. 3 and 7 for the pressure and temperature fields. The Hele-Shaw equation, Eq. 3, is then discretized by means of the Galerkin method, which yields

$$\left.\begin{aligned}
&A_{ij}(p,T)p_j = b_i \\
&A_{ij}(p,T) = \int_{\Omega} S(p,T)(\varphi_{i,x}\varphi_{j,x} + \varphi_{i,y}\varphi_{j,y} + \varphi_{i,z}\varphi_{j,z})d\Omega \\
&b_i = -\int_{\partial\Omega} \varphi_i h \bar{v}_n d\Gamma
\end{aligned}\right\} \qquad (8)$$

where p_j are nodal values of the pressure, $\bar{v}_n$ is the normal average velocity on the boundary Γ of the flow domain Ω and the φ_i's are shape functions. The Newton-Raphson method is used to solve the above non-linear set of equations, Eq. 8.

Spatial and time discretization of the energy equation are performed in two steps. The full systems are, therefore, of the generic matrix form:

$$\left.\begin{aligned} M_{ij}T_j^{n+1/2} &= M_{ij}T_j^n - \Delta t N_{ij}T_j^n + \Delta t L_i \\ K_{ij}T_j^{n+1} &= B_{ij}T_j^{n+1/2} \end{aligned}\right\} \qquad (9)$$

In Eq. 9_1, **M** is the mass matrix and all the terms arising from heat advection and viscous heating are included in the matrix **N** and vector **L**, respectively. The matrix **K**, in Eq. 9_2, is associated with the thermal diffusion and the mass matrix **B**. Equation 9_1 corresponds to the advection problem while Eq. 9_2 is a diffusive problem.

Having formulated the mold filling problem, the optimization techniques and algorithms will now be described to demonstrate how these techniques can be applied to injection molding problems.

3 Optimization Techniques

3.1 Optimization Concept

In injection molding process, we wish to reduce the cycle time as much as possible, to produce products with accurate geometrical shape and to achieve high performance quality. In this regard, the rise of the number of design variables needed to be coordinated with each other in an optimal way, have led to the necessity of using mathematical modeling and numerical optimization techniques. The techniques of numerical optimization provide methods for automatically searching the space of admissible designs to determine the best possible design. They may be seen as procedures, which determine a combination of parameters, i.e., design variables, which minimize or maximize an objective function under given conditions. The combination of such parameters is then termed optimum. The optimum must by definition lie in the feasible space, i.e., satisfy all the constraints. Each design variable can be considered as one dimension in a design space and any combination of variables as a point in this space.

3.2 Optimization Problems

In the unconstrained optimization problem [51], we seek a local minimizer of a real-valued function, F(**x**), where **x** is a vector of real variables. In other words, we seek a vector, **x***, such that

$F(\mathbf{x}^*) \le F(\mathbf{x})$ for all **x** close to **x***.

Global optimization algorithms try to find an **x*** that minimizes F over all possible vectors **x** as illustrated in Fig. 1. This is a much harder problem to solve. No algorithm for optimizing general nonlinear functions exists that will always find the global optimum for a general nonlinear minimization problem in a reasonable amount of time.

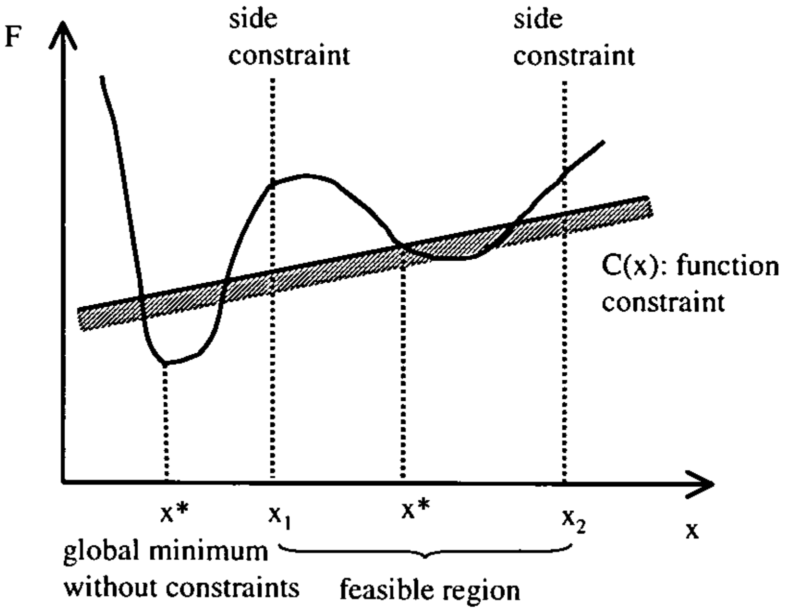


Fig. 1. Unconstrained and constrained minimization problem.

On the other hand, the basic constrained optimization problem can be stated as:

Given a scalar objective function,

$$F = F(\mathbf{x})$$

Find the value of *n* decision variables **x**, that minimizes F and simultaneously satisfy *m* constraint equations. The side constraints represent bounds on the design variables that define the feasible region (Fig. 1). In injection molding, we are only concerned with constrained optimization problems.

3.3 Numerical Solution of Optimization Problems

Most optimization algorithms require that an initial set of design variables, $\mathbf{x}^0$, be specified. Beginning from this starting point, the design is updated iteratively. The most common form of the iterative optimization procedure is given by

$$\mathbf{x}^q = \mathbf{x}^{q-1} + \theta \mathbf{G}^q \qquad (10)$$

where q is the iteration number and **G** a vector search direction in the design space. The scalar quantity θ defines the distance that we wish to move in direction **G**.

Equation 10 involves the determination of the search direction $\mathbf{G}^q$ and the solution of the one dimensional search problem for the scalar θ, which will minimize $F(\mathbf{x})$ as much as possible in direction $\mathbf{G}^q$, without violating any constraints.

For a thorough description of the numerical solution of the optimization problems, we refer the reader to specialized literature reviews [49-52], among others. Here, it suffices to recall the following alternatives for the computation of the parameter θ:

The two most popular methods for finding the minimum of a function of one variable, associated to the scalar θ, are the polynomial interpolation and the golden section method. The procedure for the polynomial interpolation method is to evaluate the function at several points and then fit a polynomial to those known points. The minimum of this polynomial is then found and is considered to be a good estimate to the minimum of true function. The golden section method, on the other hand, consists in evaluating functions on a smaller and smaller interval until a desired interval of uncertainty is achieved. Alternatively, θ can be selected randomly between 0 and 1, for random search methods.

On the other hand, one can use the following alternatives for the numerical solution of the search direction $\mathbf{G}^q$, involved in Eq. 10:

3.3.1 Zero-order Methods

Zero-order methods or direct search schemes, which include random search, genetic search, simulated annealing and Powell's method, do not require gradient information. These methods are usually reliable and easy to program but require thousands of function evaluations to achieve optimum. Because genetic algorithms are essentially random search techniques (stochastic), they offer an improved prospect of finding the global or near global optimum. While the emphasis in this chapter is on gradient-based methods, genetic search methods are only briefly outlined here.

For sake of clarity, we repeat here the genetic search method developed by Young [60] to search the optimal gate location in liquid composite molding, and by Lam et al. [69] for the optimal cooling system design in injection molding. A thorough description of general genetic algorithms is given in [72].

Genetic algorithms are inspired by Darwin's theory of evolution. The basic approach of genetic algorithms is based on the mechanism of biological evolution, natural selection, and genetic recombination.

Consider, for example, a two variable design

$$\mathbf{x} = \begin{Bmatrix} 1011101001 \\ 0101100110 \end{Bmatrix}$$

where each x_i is represented as a binary string. We can think of this design as one long string of zeroes and ones

$$\mathbf{x} = 10111010010101100110$$

Several bit strings generated by different combinations of design parameters constitute the population. Subsequently, the biological reproduction and evolution processes will change the members of the population. Through the natural selection and reproduction, the population is improved and only those who fit in the environment will survive. This approach is facilitated by defining a fitness or objective function indicating the 'goodness' of a member in a generation during the evolution process. The individual with higher fitness means that it has a better chance to survive the evolution. In a biological system, the genetic memory is preserved and transferred from generation to generation in the form of chromosomes (population). In order to use genetic algorithm, the design variables must be represented in a chromosome-like form. For a *n*-dimensional design, the *n* binary strings representing *n* design variables can be connected in a head-to tail way to form a single binary chromosome. Several operations of genetic evolution and selections can be performed on the artificial binary chromosome. The basic operations of genetic search include selection, reproduction, crossover and mutation. The basic genetic search algorithm consists of the following steps:

1. Generate random population of *n* chromosomes.
2. Evaluate the fitness, F(**x**), of each chromosome x in the population.
3. Create a new population by repeating the following steps until the new population is complete:
    a. Selection and reproduction: Select two parent chromosomes from a population according to their fitness. The better fitness, the bigger chance to be selected, the more chances they have to reproduce. Consequently, the string with higher fitness will have more chances to be selected to crossover in generating the succeeding generation.
    b. Crossover: With a crossover probability, cross over the parents to form new offspring (children). If no crossover was performed, offspring is the exact copy of parents.
    c. Mutation: With a mutation probability, mutate new offspring at each locus (position in chromosome or population).
    d. Accepting: Place new offspring in the new population.
4. Use new generated population for a further run of the algorithm.
5. Repeat the process until convergence is achieved.

In a finite element model, for example, the node numbers define the candidates for gate location, and each gate node is coded as a binary string.

Although the genetic search method was successfully used by Lam et al. [69] for a cooling system optimization problem (non-convex optimization problem), the required computational time is still a limit, due to the large number of function evaluation. Young [60] also pointed out the large amount of computational time needed to obtain converged solution, for the gate location optimization problem.

3.3.2 First- and Second-Order Method

Conversely, first-order methods, such as the steepest descent method and the conjugate direction method, those that utilize gradient information for the search direction, G^q, are usually more efficient than random or zero-order methods. The price paid for this efficiency is that gradient information must be supplied. Also, to determine the scalar quantity, θ, an implementation of a line-search algorithm is required. On the other hand, second-order methods (e.g., Newton's method) give rise to a wide and important class of algorithms that require computation of the gradient vector and the Hessian matrix (second derivative) [51]. Although the computation or approximation of the Hessian can be a time-consuming operation, there are many problems for which this computation is justified. Gradient-based methods are in essence local optimization algorithms. A full description of these gradient-based algorithms may be found in [49-52]. The algorithm used in section 7 is sequential quadratic programming (SQP). In the SQP algorithm, the objective function is augmented using Lagrange multipliers and an exterior penalty so that the resulting one-dimensional search is unconstrained. The SQP algorithm then replaces the augmented objective function with the quadratic approximation and replaces the constraint functions by linear approximations. The search direction is, therefore, found by solving a sub-problem with a quadratic objective and linear constraints. The solution procedure involves three main stages: updating of the Hessian matrix of the Lagrangian function, quadratic programming problem solution and line search calculation.

3.3.3 Combination of Zero-Order and Gradient-Based Methods

Finally, we can think of algorithms that combine a mixture of the above methods, as outlined hereafter. The search method introduced by Pandelis and Zou [57] combines a deterministic hill climbing search with a stochastic simulated annealing search method (SANHIL) for gate location optimization. We recall here the basic concept of the algorithms. In the simulated annealing process, a system of particles seeks low energy configurations. If the temperature of the system is gradually lowered and enough time is allowed at each temperature to reach thermal equilibrium, the system will attain a very low energy configuration. If the system is in thermal equilibrium at a given temperature T, the probability $P_T(b)$ of the system in a given configuration, b, is a function of the energy, E(b), of the configuration and satisfies the Boltzman distribution

$$P_T(b) = \frac{e^{-E(b)/k_B T}}{\sum_B e^{-E(B)/k_B T}}$$

Where k_B is Boltzman's constant and B is the set of all possible configurations. To apply the simulated annealing method to optimization problems, the objective function F(x) is used as an energy function E. Instead of finding a low energy configuration, the problem becomes one of seeking a near global optimal solution. A random number generator is used as a way of generating new values for the design variables.

In the case of gate location optimization, the methodology developed by Pandelis and Zou is as follows. Let's assume that the probability distribution of the objective function F(x) for gate g has a Boltzman distribution. Let's suppose that at time t at iteration k, gate location g_1 is chosen. A candidate gate location g_2 is chosen randomly at time t+1 at iteration k. The criterion for whether or not the move from g_1 to g_2 is successful, depends on the difference between the objective function value $F(g_1)$ and $F(g_2)$, and iteration k. The ratio R between the probability of being at location g_1, and the probability of being at location g_2 can be decided by the following equation

$$R = \frac{P_T(g_1)}{P_T(g_2)} = e^{-\frac{F(g_1)-F(g_2)}{k_B T}}$$

When R>1, the objective function value at location g_2 is smaller than that at location g_1. This move is considered successful, and location g_2 is automatically chosen as an optimal gate location at time t+1. When $R \leq 1$, the objective function value at location g_2 is greater than or equal to that at location g_1. This move is then considered successful with probability R, but gate g_2 will not be accepted as optimum gate at time t+1 at iteration k. An iteration limitation M is set for a search process, and a time limitation N is also chosen for each iteration. If there is no successful move after N moves or M iterations are reached, the process is stopped and the last calculated optimum value is taken as an answer.

In the hill climbing search method, after a starting point is chosen, the objective function will go along a path of steepest descent direction until there are no more downhill moves possible. For example, when the initial gate is chosen in the gate location optimization, the objective function will evaluate this starting gate and each of the neighboring nodes. The node associated with the minimum objective function value is then selected as the next starting node. The process continues until no improvement is possible, and the last selected node is considered as the local optimum. It can be seen from the above description that the hill climbing search is good for finding the optimum in a convex region. If there are many local optima in the search space, it is possible to obtain an unsatisfactory solution because of an improper initial gate. On the other hand, the simulated annealing search can find a near global optimum if enough time is spent on function evaluation at each iteration.

The SANHIL search developed by Pandelis and Zou combines the simulated annealing search with the hill climbing search. In SANHIL search, the hill climbing can be considered as a sub-search in the simulated annealing search. To find the local optimum, the hill climbing search is applied to each initial gate randomly selected by the simulated annealing search. The results are then sent back to the simulated annealing search.

3.3.4 Choice of a Method

The choice of a particular algorithm for any situation depends on the problem formulation and application. Direct search methods are not considered to be particularly competitive for solution of continuous optimization tasks. Instead, gradient-based methods are generally considered superior to zero-order methods in their efficiency. Since the emphasis in this chapter is on gradient-based methods, one is required to supply gradient information of the objective functional and function constraints. Information on gradients can be supplied either by finite-difference computations, or by design sensitivity analysis (DSA). Whatever the gradient-based method we choose, an efficient computation of accurate gradients is a major requirement of any design tool. To obtain good estimates of gradients with the finite-difference method, it is important to properly choose the finite difference step size, $\Delta\mathbf{x}$, for each design variable, which depends highly on the objective functional F($\mathbf{x}$) and on the current position of the vector of design variables, $\mathbf{x}^q$, in the design space. When $\Delta\mathbf{x}$ is too large, truncation errors affect the results whereas round-off errors affect the results when $\Delta\mathbf{x}$ is too small. So that several perturbation sizes, $\Delta\mathbf{x}$, may be necessary to verify that reliable results are obtained. For these reasons, the finite difference method is not used here for computing sensitivities. Instead, DSA-based methods are used to evaluate gradients of both the objective functional and function constraints, along with the gradients of the flow equations. Hereafter, we focus our attention on DSA-based methods.

4 Sensitivity Analysis

The use of direct sensitivities and adjoint methods in optimal design is certainly not new and is fairly well developed in structural design [73-81]. However, in material processing problems, much remains to be done on this topic, especially in injection molding process.

The state sensitivities (derivative of the state with respect to the design variables) play two key roles: they provide gradient information for the optimization algorithm and sensitivity information for analysis of a particular design.

4.1 Direct Sensitivity Equation Method

Let F be the objective functional,

$$F = F(\phi, \mathbf{a}, t) = \int_0^{t_s} g(\phi, \mathbf{a}, t)dt \qquad (11)$$

where ϕ is the vector of flow field variables, t_s the terminal time and **a** the vector of design variables. The time-dependent state equation or flow equation may be written symbolically as

$$\left.\begin{array}{l} \dfrac{\partial\phi}{\partial t}(t)+B(\phi(t),\mathbf{a})=0 \text{ in } \Omega \text{ and } \forall t\in\,]0,t_s] \\ \phi(t=0)=\phi_0 \text{ in } \Omega \\ \phi(t)=\phi_b \text{ on } \Gamma \end{array}\right\} \qquad (12)$$

where B is a partial differential operator and Γ denotes the boundary of the flow domain Ω.

Direct sensitivity equations can be obtained by direct differentiation of the above equation with respect to **a** as follows

$$\frac{dF}{d\mathbf{a}}(\phi,\mathbf{a},t)=\frac{\partial F}{\partial\phi}(\phi,\mathbf{a},t)\frac{d\phi}{d\mathbf{a}}+\frac{\partial F}{\partial\mathbf{a}}(\phi,\mathbf{a},t)=\int_0^{t_s}\left[\frac{\partial g}{\partial\phi}(\phi,\mathbf{a},t)\frac{d\phi}{d\mathbf{a}}+\frac{\partial g}{\partial\mathbf{a}}(\phi,\mathbf{a},t)\right]dt \quad (13)$$

We still need to specify how the vector of flow variable sensitivity $d\phi/d\mathbf{a}$ is determined. One may compute the total derivative of the flow equation (Eq. 12), to obtain the linear system for the sensitivities

$$\frac{\partial}{\partial t}\left(\frac{d\phi}{d\mathbf{a}}\right)+\frac{\partial B}{\partial\phi}(\phi,\mathbf{a})\frac{d\phi}{d\mathbf{a}}+\frac{\partial B}{\partial\mathbf{a}}(\phi,\mathbf{a})=0 \qquad (14)$$

The nonlinear flow system, Eq. 12, is solved first using the Newton iteration, then the sensitivities of the flow variables at the new step are found by directly solving Eq. 14 for each design variables. Both the state and the sensitivity equations are integrated forward in time. The forward sensitivity analysis is best suited to the situation of finding the sensitivities of a large number of solution variables with respect to a small number of design parameters.

4.2 Adjoint Equation Method

Alternately, one may use adjoint equation method to determine sensitivities of the objective functional. Introducing a Lagrange multiplier, λ, and using the flow field equation, Eq. 12, as constraint, we get the augmented objective functional

$$G(\phi,\mathbf{a},t)=F(\phi,\mathbf{a},t)-\int_0^{t_s}\lambda^T(t)(\frac{\partial\phi}{\partial t}(t)+B(\phi,\mathbf{a}))dt \qquad (15)$$

The first variation of the objective functional yields

$$\frac{dG}{da}=\frac{dF}{da}=\int_0^{t_s}\left[\frac{\partial g}{\partial \phi}\frac{d\phi}{da}+\frac{\partial g}{\partial a}\right]dt-\int_0^{t_s}\lambda^T(t)\left[\frac{\partial}{\partial t}\left(\frac{d\phi}{da}\right)+\frac{\partial B}{\partial \phi}\frac{d\phi}{da}+\frac{\partial B}{\partial a}\right]dt \qquad (16)$$

By denoting the sensitivity of ϕ with respect to a design variable by

$$\phi_a=\frac{d\phi}{da} \qquad (17)$$

we get

$$\frac{dF}{da}(\phi,a,t)=\int_0^{t_s}\left[\frac{\partial g}{\partial \phi}\phi_a+\frac{\partial g}{\partial a}\right]dt-\int_0^{t_s}\lambda^T(t)\left[\frac{\partial \phi_a}{\partial t}+\frac{\partial B}{\partial \phi}\phi_a+\frac{\partial B}{\partial a}\right]dt \qquad (18)$$

Applying integration by parts to term $\lambda^T\partial\phi_a/\partial t$, yields

$$\frac{dF}{da}=\int_0^{t_s}\left[\frac{\partial g}{\partial \phi}\phi_a+\frac{\partial g}{\partial a}\right]dt-\left[\int_0^{t_s}(-\phi_a\frac{\partial \lambda}{\partial t}+\lambda^T\frac{\partial B}{\partial \phi}\phi_a+\lambda^T\frac{\partial B}{\partial a})dt+(\lambda^T\phi_a)\big|_0^{t_s}\right]$$

or

$$\frac{dF}{da}=\int_0^{t_s}\left[(\frac{\partial \lambda}{\partial t}-\lambda^T\frac{\partial B}{\partial \phi}+\frac{\partial g}{\partial \phi})\phi_a+\frac{\partial g}{\partial a}-\lambda^T\frac{\partial B}{\partial a}\right]dt-(\lambda^T\phi_a)\big|_0^{t_s} \qquad (19)$$

By choosing λ such that it satisfies the adjoint equation

$$\left.\begin{array}{l}\dfrac{\partial \lambda}{\partial t}-\lambda^T\dfrac{\partial B}{\partial \phi}+\dfrac{\partial g}{\partial \phi}=0 \text{ in } \Omega \text{ and } \forall t\in[0,t_s[\\ \lambda(t_s)=0 \text{ in } \Omega\end{array}\right\} \qquad (20)$$

we obtain a simple expression of the sensitivity of the objective functional F as

$$\frac{dF}{da}(\phi,a,t)=\int_0^{t_s}\left[\frac{\partial g}{\partial a}-\lambda^T\frac{\partial B}{\partial a}(\phi,a)\right]dt+(\lambda^T\phi_a)\big|_{t=0} \qquad (21)$$

We note that in Eq. 21, $\phi_a|_{t=0}$, is the sensitivity of the state variables at the initial time. The adjoint equation, Eq. 20, is a terminal value problem, which depend on the solution $\phi(t_s)$ of the original problem, Eq. 12. It must be integrated backward in time for its solution at time t=0. Therefore, a procedure is needed for providing the states ϕ obtained during the forward integration phase of the state equation to Eq. 20, during the backward integration phase of the adjoint equation. The sensitivity, dF/da, can be obtained without any information about the flow variable sensitivity vector, dϕ/da. This makes the computational cost for the sensitivity analysis independent of the number of design variables. Once the parameter, λ, is computed from Eq. 20, the sensitivity formula, Eq. 21, can be used to compute the sensitivity of the objective functional with respect to any design variables. It follows from this analysis that, the adjoint sensitivity method is best suited to the situation of finding the sensitivity of a scalar functional of the solution with respect to a large number of design variables.

4.3 Comparison of Solution Methods

In practice, both direct and adjoint sensitivity methods can be used. The equations related to the direct sensitivity method must be solved for each of the independent design variables, whereas those derived for the adjoint sensitivity method, need to be solved for each of the functions whose sensitivities are sought. Hence, for steady state problems, the best choice will be based on the number of function evaluations needed to obtain the sensitivity sought. While, the direct sensitivity method is more efficient for determining the sensitivity of many outputs (objective functional) to one or a few input parameters (design variables), the adjoint method is better suited for sensitivities of one or a few outputs with respect to many input parameters. However, for transient systems, the direct differentiation method appears to be more suitable. Indeed, the adjoint method requires values of all time steps before the adjoint analysis may begin. This increases both the computation and storage costs. The direct differentiation method allows to evaluate the response sensitivities simultaneously with the transient solution of the problem (see e.g., McDavid and Dantzig [78], Haug et al. [79], Pandelis and Zou [57] and Smith et al. [62-64]).

5 Optimal Design for Injection Molding

Having introduced the design sensitivity method for a general time dependent flow equation, we now specialize to injection molding process. In what follows, we limit ourselves, for sake of clarity, to a scalar objective functional, although the methodology remains valid for a vector of objective functional and various design variables. Additionally, we describe, in more detail, the different steps involved in the definition and solution of a typical mold filling optimization problem, using DSA-based methods. This approach was used by Smith et al. [62-64] and Kabanemi et al. [65, 66] to optimize gate location and processing conditions. It can be easily extended to various optimization problems in injection molding, including the optimization of warpage, differential shrinkage and mold cooling system.

5.1 Problem Parameters

In order to facilitate our subsequent discussion, we begin by defining parameters involved in the specific optimization problem addressed in the following sections:

- Vector of design variables: $\mathbf{a}=\{a_1, a_2,\ldots, a_r, \ldots., a_R\}$. The gate locations in the cavity and the injection pressure profile during the filling stage are design variables considered in this analysis. These variables can be modified during the optimization process to meet our objective;

- Vector of state variables: $\phi=\{p(a_r), T(a_r), f(a_r), a_r\}$. The pressure, temperature and filling fraction fields are the state variables of the optimization problem. These state variables are obtained from the flow equations (see section 2).

- Objective function: $F=t_f(p(a_r), T(a_r), f(a_r), a_r)$. The objective function to be minimized is the total filling time.

- Vector of constraint functions: $C=\{C_1(p(a_r), T(a_r), f(a_r), a_r), C_2(p(a_r), T(a_r), f(a_r), a_r)\}$. The optimization is performed under two constraints, i.e., the maximum clamp force and the maximum flow rate.

5.2 Problem Definition

Having defined the parameters involved in the problem, we can now write the non-linear constrained optimization problem for the filling stage of the injection molding process as follows:

minimize the objective function (fill time):

$$F= t_f(p(a_r), T(a_r), f(a_r), a_r);$$

subject to (maximum clamp force and maximum flow rate):

$$\frac{C_i\,(p(a_r),T(a_r),f(a_r),a_r))}{C_{i,\max}}-1\le 0\,,\; i = 1, 2 \text{ (constraint functions);}$$

$$a_{r,\min}\le a_r \le a_{r,\max}\,,\; r=1,\ldots,R \text{ (side constraints);}$$

where the design variables are the gate locations in the cavity and the injection pressure profile during the filling stage.

5.3 Direct Sensitivity of the State Equations

Having computed the pressure field from Eq. 8, the design sensitivity analysis of the Hele-Shaw and energy equations can now be considered. In practice, one can derive the sensitivity equation at the partial differential equation level, and then discretize it, to obtain a set of discrete sensitivity equations. This continuum approach is known as the material derivative method. Alternately, one can discretize the flow equations first and then differentiate them to obtain another set of discrete sensitivity equations. This approach is known as the discrete approach. The two approaches, i.e., differentiate-then-discretize and discretize-then-differentiate lead to different sensitivity equations and therefore to different approximate sensitivities. The relative merits of the two approaches are discussed in Haug et al. [79] and Lund [80]. The conclusive argument for the selection of a method for design sensitivity analysis has been that the continuum approach takes a lot of analytical work in order to develop the expressions for design sensitivities. For that reason the discrete approach to design sensitivity analysis has been chosen due to its ease of implementation.

Differentiating Eq. 8 with respect to a design variable a_r and rearranging different terms, yields the following discrete version for the flow sensitivities

$$J_{ij}(p_k^m)\frac{Dp_j^m}{Da_r} = -\frac{\partial A_{is}(P_k^m(a_r),a_r,T)}{\partial a_r} p_s^m(a_r) \qquad (22)$$

$$-\frac{\partial A_{is}(p_k^m(a_r),a_r,T)}{\partial T_k}\frac{DT_k}{Da_r} p_s^m + \frac{\partial b_i}{\partial a_r}$$

where

$$J_{ij} = \frac{\partial G_i}{\partial p_j^m} = \frac{\partial A_{ik}}{\partial p_j^m} p_k^m + A_{ij} - \frac{\partial b_i}{\partial p_j^m} \qquad (23)$$

The geometric interpretation of the pressure sensitivity with respect to the gate location, for a 1-D Newtonian filling problem, of a rectangular cavity is shown in Fig. 2. The pressure profile is plotted along the cross line.

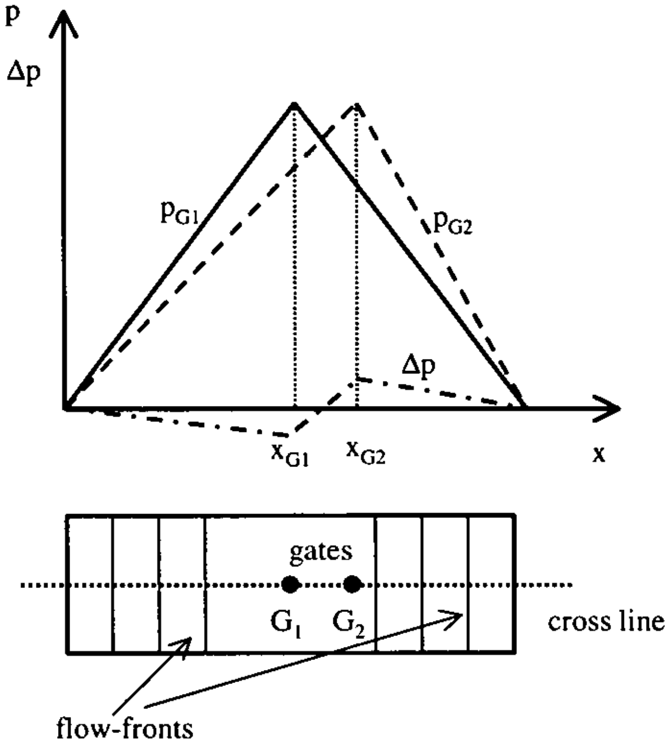


Fig. 2. Pressure profile and pressure gradient across the cavity for two different gate locations.

In Eq. 22, a_r is the rth component of the design variable belonging to the design vector **a**.

Similarly, the sensitivity equations for the advection and diffusion problems, Eq. 9, are given by:

$$\begin{aligned} M_{ij}\frac{DT_j^{n+1/2}}{Da_r} &= M_{ij}\frac{DT_j^n}{Da_r} + (T_j^n - T_j^{n+1/2})\frac{\partial M_{ij}}{\partial a_r} \\ &- \Delta t(\frac{\partial N_{ij}}{\partial P_k}\frac{DP_k}{Da_r} + \frac{\partial N_{ij}}{\partial T_k^n}\frac{DT_k^n}{Da_r} + \frac{\partial N_{ij}}{\partial a_r})T_j^n - \Delta t N_{ij}\frac{DT_j^n}{Da_r} \qquad (24) \\ &+ \Delta t(\frac{\partial L_i}{\partial P_k}\frac{DP_k}{Da_r} + \frac{\partial L_i}{\partial T_k^n}\frac{DT_k^n}{Da_r} + \frac{\partial L_i}{\partial a_r}) \end{aligned}$$

and

$$K_{ij}\frac{DT_j^{n+1}}{Da_r}=\frac{\partial B_{ij}}{\partial a_r}T_j^{n+1/2}+B_{ij}\frac{DT_j^{n+1/2}}{Da_r}-\frac{\partial K_{ij}}{\partial a_r}T_j^{n+1} \qquad (25)$$

The solution of sensitivity equations (Eqs. 22-25) requires the parameterization of the injection pressure profile and the flow domain.

5.4 Sensitivity Formulation of the Objective Function

The optimization problem requires the minimization of the objective functional $F(p(a_r), T(a_r), f(a_r), a_r)$ subject to certain constraints. Gradient-based methods are available to efficiently traverse the design space to locate extrema of F. To use these methods, we need the design sensitivity, i.e., the gradient of F with respect to a design variable a_r, which can be computed by application of the chain rule

$$\frac{DF}{Da_r}=\frac{\partial F}{\partial p}\frac{Dp}{Da_r}+\frac{\partial F}{\partial T}\frac{DT}{Da_r}+\frac{\partial F}{\partial f}\frac{Df}{Da_r}+\frac{\partial F}{\partial a_r} \qquad (26)$$

Since F is an explicit function of p, T, f and a_r, explicit derivatives $\partial F/\partial p$, $\partial F/\partial T$, $\partial F/\partial f$ and $\partial F/\partial a_r$ can be evaluated directly. The solution sensitivity Dp/Da_r is obtained by solving the sensitivity of the Hele-Shaw equation, Eq. 22, while the sensitivity DT/Da_r is obtained by solving the sensitivity of the energy equation, Eqs. 24 and 25. The sensitivity of the flow front is described in a forthcoming section.

The objective function to be minimized represents the total fill time, i.e.,

$$F=t_f(p(a_r),T(a_r),f(a_r),a_r)=\sum_{k=1}^{n_{tot}}\Delta t_k \qquad (27)$$

where n_{tot} is the total number of time-steps needed to fill the mold cavity and Δt_k is the filling time-step size at time t_k. This time step is restricted by the Courant-Friedrichs-Lewy (CFL) condition for numerical stability. Selecting the time-step size such that the advancing front does not completely traverse more than one element anywhere in the flow field during a single time-step satisfies the CFL condition. The maximum filling time-step is thus given by

$$\Delta t_k \le \frac{\Delta x_{min}}{\overline{v}_{max}(p,a_r)} \qquad (28)$$

where Δx_{min} is the characteristic length of the mesh and $\overline{v}_{max}$ is the maximum velocity in the flow field.

5.5 Parameterization of the Injection Pressure and Sensitivities

We use the same injection pressure parameterization as the one used by Smith et al. [62, 63], that we outline bellow. The prescribed injection pressure, $p^P(\psi(a_r), a_r)$, is expressed as a piecewise linear function of the filling fraction ψ. The values ψ=0 and ψ=1 designate an empty and full mold cavity, respectively and, fractional values denote partially filled mold cavity. The gradient of $p^P(\psi(a_r), a_r)$ with respect to a design variable a_r is given by

$$\frac{DP^P}{Da_r} = \frac{\partial P^P}{\partial \psi}\frac{D\psi}{Da_r} + \frac{\partial P^P}{\partial a_r} \tag{29}$$

The filling fraction is defined by

$$\psi = \sum_{k=1}^{n_{tot}} \psi_k = \sum_{k=1}^{n_{tot}} \frac{V^k}{V_{tot}} \tag{30}$$

where V^k and V_{tot} are the volume to be filled during the time step Δt_k and the total volume of the mold cavity, respectively. The volume V^k is given by

$$V^{k+1} = \sum_e V_e(a_r)(1 - f_e^k(a_r)) \tag{31}$$

In Eq. 31, V_e is the volume of element e and f_e^k is the filling fraction of that element at time t_k. The values $f_e^k = 0$ and $f_e^k = 1$ designate an empty and full element, respectively, and, fractional values denote partially filled element. By differentiating Eq. 31, one obtains

$$\frac{DV^{k+1}}{Da_r} = \sum_e (-V_e \frac{Df_e^k}{Da_r} + \frac{\partial V_e}{\partial a_r}(1 - f_e^k)) \tag{32}$$

Equation 29 requires the sensitivity of the filling fraction, which can be computed by

$$\frac{D\psi}{Da_r} = \frac{1}{V_{tot}} \sum_{k=1}^{n} \frac{DV^k}{Da_r} \tag{33}$$

The parameterization of the injection pressure for the filling range, $\psi_i \le \psi(a_r) \le \psi_{i+1}$, is given by

$$p^{P}(\psi(a_r),a_r) = p_i(a_r) + \frac{p_{i+1}(a_r) - p_i(a_r)}{\psi_{i+1} - \psi_i}(\psi(a_r) - \psi_i) \qquad (34)$$

where the abscissa of the control points ψ_i, $i = 1,\ldots, r_p$ are fixed during the optimization process and the ordinates of the control points are such that, $p_i = a_i$, $i = 1,\ldots, r_p$; r_p being the number of parameters that define the transient injection pressure profile.

At each time step, one obtains

$$\Delta t_{k+1} = \frac{V^{k+1}}{Q^k} \qquad (35)$$

where Q^k is the total flow rate during the time step Δt_k. The sensitivity of the time step Δt_k is then given by

$$\frac{D\Delta t_{k+1}}{Da_r} = \frac{1}{Q^k}\frac{DV^{k+1}}{Da_r} - \frac{V^{k+1}}{{Q^k}^2}\frac{DQ^k}{Da_r} \qquad (36)$$

Equation 36 requires sensitivities of both the flow rate and the volume, which are computed with the help of Eqs. 22 and 32, respectively.

5.6 Sensitivities of the Function Constraints

The constraints on the clamp force and flow rate are given respectively by

$$C_1 = \int_{\Omega} p(T,a_r)d\Omega = \int_{\omega} p(T,a_r)|\mathbf{J}(a_r)|d\omega < C_{1,\max} \qquad (37)$$

and

$$C_2 = \sum_{i=1}^{ngate} - \int_{\partial\Omega} \varphi_i \hbar \bar{v}_n d\Gamma = \sum_{i=1}^{ngate} A_{ij}(p,T,a_r)p_j(T,a_r) < C_{2,\max} \qquad (38)$$

where the φ_i's are shape functions, **J** the Jacobian matrix, ω the parent element and ngate the total number of gates.

Since gradient-based methods are used in the resolution of the optimization problem, we need gradients of both constraints. These are given by

$$\left.\begin{aligned}\frac{DC_1}{Da_r} &= \int_{\omega}\left[\frac{Dp}{Da_r}\left|J(a_r)\right| + p\frac{\partial|J|}{\partial a_r}\right]d\omega \\ \frac{DC_2}{Da_r} &= \sum_{i=1}^{ngate}\left[\frac{DA_{ij}}{Da_r}p_j(T,a_r) + A_{ij}(p,T,a_r)\frac{Dp_j}{Da_r}\right]\end{aligned}\right\} \tag{39}$$

5.7 Flow-Front Tracking and Sensitivities

In modeling of the filling phase of the injection process one of the difficult task is the successful tracking of the flow fronts. Performing a mass balance on an element, e, produces an explicit evolution equation for the element fill state

$$f_e^{k+1} = f_e^k + \frac{q_e^k \Delta t_{k+1}}{V_e} \tag{40}$$

where q_e^k is the total flow rate of fluid into element, e, at the kth time step.

In order to compute the filling fraction sensitivity, we differentiate Eq. 40 with respect to a design variable a_r

$$\frac{Df_e^{k+1}}{Da_r} = \frac{Df_e^k}{Da_r} + \frac{\Delta t_{k+1}}{V_e}\frac{Dq_e^k}{Da_r} + \frac{q_e^k}{V_e}\frac{D\Delta t_{k+1}}{Da_r} - \frac{q_e^k \Delta t_{k+1}}{V_e^2}\frac{\partial V_e}{\partial a_r} \tag{41}$$

Equation 41 implies computation of several sensitivities, i.e., the flow rate sensitivity $\frac{Dq_e^k}{Da_r}$, the element volume sensitivity $\frac{\partial V_e}{\partial a_r}$ and the sensitivity of the time step, $\frac{D\Delta t_{k+1}}{Da_r}$.

5.8 Parameterization of the Flow Domain and Sensitivities

Injection molding process involves, most of time, flow in a non-planar cavity of witch the mesh domain is defined through its nodal rectangular coordinates **X**(x,y,z). Using the isoparametric description, these components may be written as follows:

$$\left.\begin{aligned} x(a_r) &= x_i(a_r)\varphi_i \\ y(a_r) &= y_i(a_r)\varphi_i \\ z(a_r) &= z_i(a_r)\varphi_i \end{aligned}\right\} \tag{42}$$

where the φ_i's are shape functions.

The mesh derivatives needed to solve sensitivity equations described in previous sections are obtained from

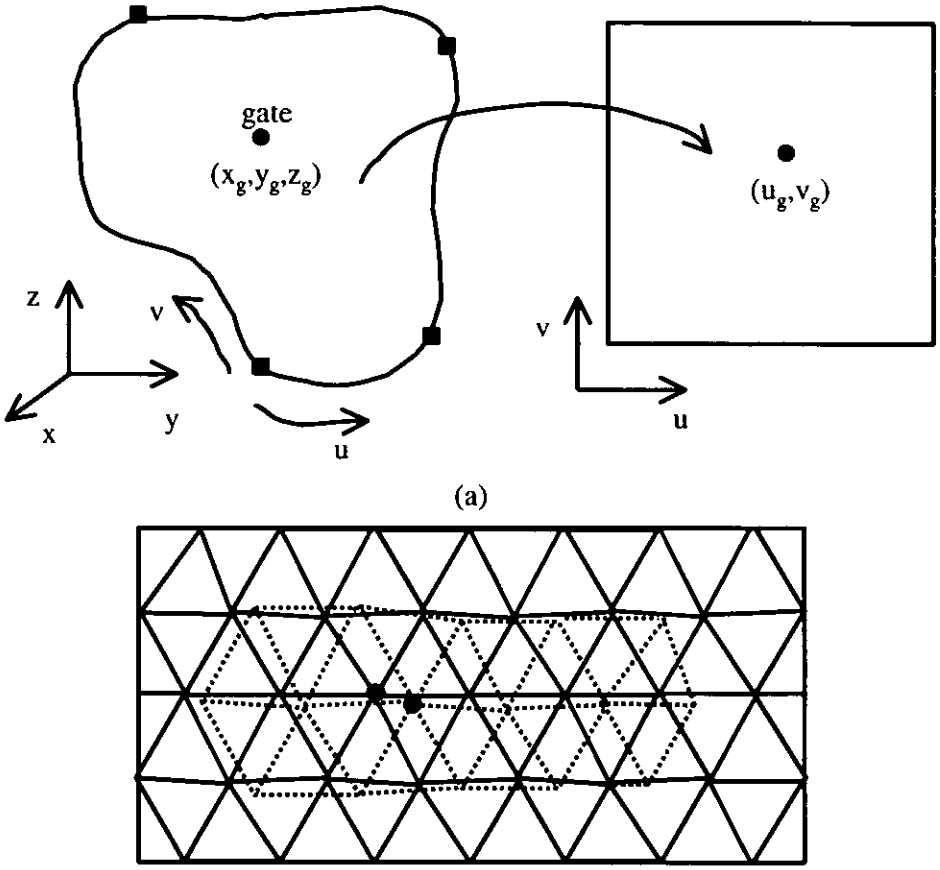
$$\left.\begin{aligned}\frac{dx}{da_r} &= \frac{dx_i}{da_r}\phi_i \\ \frac{dy}{da_r} &= \frac{dy_i}{da_r}\phi_i \\ \frac{dz}{da_r} &= \frac{dz_i}{da_r}\phi_i\end{aligned}\right\} \qquad (43)$$

We recall that side constraints for the gate movement define the region of search for the optimum in the mesh domain. While it is straightforward to define these side constraints for planar mesh domain, it becomes impossible to define for non-planar geometry without adding additional inequality function constraints. For that reason, we used a mapping technique in which a physical non-planar mesh domain, representing the region of search, is mapped into a rectangular geometry as shown in Fig. 3(a). The mapping process generates a curvilinear coordinate system and makes defining side constraints much easier.

Thus, the nodal coordinates are defined as complex function in the design, based on parametric spline surfaces. More explicitly, one may write for each component (see, e.g., Dierkx [83] and Tortorelli et al. [81])

$$\left.\begin{aligned}x(u(a_r),v(a_r)) &= \sum_{i=-k}^{g}\sum_{j=-l}^{h} c_{i,j}^{x} N_{i,k+1}(u) M_{j,l+1}(v) \\ y(u(a_r),v(a_r)) &= \sum_{i=-k}^{g}\sum_{j=-l}^{h} c_{i,j}^{y} N_{i,k+1}(u) M_{j,l+1}(v) \\ z(u(a_r),v(a_r)) &= \sum_{i=-k}^{g}\sum_{j=-l}^{h} c_{i,j}^{z} N_{i,k+1}(u) M_{j,l+1}(v)\end{aligned}\right\} \qquad (44)$$

where u and v are the parametric coordinates of the region of search $(0 \leq u, v \leq 1)$, $N_{i,k+1}$ and $M_{j,l+1}$ are the normalized B-splines of degree k and l, respectively, and $c_{i,j}^{x}$, $c_{i,j}^{y}$ and $c_{i,j}^{z}$ are called B-spline coefficients. Thus, each gate is characterized by its parametric coordinates, (u_g, v_g), in the regular geometry as shown in Fig. 3 (a).



(a)

(b)

Fig. 3. (a) Mapping of an irregularly shaped physical flow domain into a rectangular computational domain and (b) Schematic illustration of the mesh sensitivity with respect to a small gate movement.

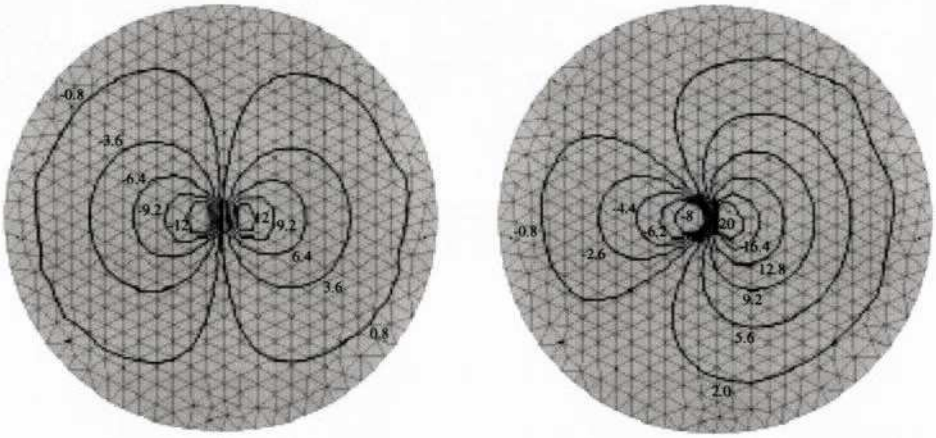
One can then derive the sensitivities for the x-component with respect to a given gate location as follows

$$\left.\begin{aligned} \frac{dx}{du_g} &= \frac{\partial x}{\partial u}\frac{du}{du_g} + \frac{\partial x}{\partial v}\frac{dv}{du_g} \\ \frac{dx}{dv_g} &= \frac{\partial x}{\partial u}\frac{du}{dv_g} + \frac{\partial x}{\partial v}\frac{dv}{dv_g} \end{aligned}\right\} \qquad (45)$$

A geometrical interpretation of such sensitivities is highlighted in Fig. 3(b), where we see how the mesh may deform due to a small perturbation of the gate position.

Similarly, the sensitivities for the y and z-components can be obtained by replacing in Eq. 45 the x-component by the y and z-components, respectively.

The sensitivity of the state equations greatly depends on the topology of the mesh connected to the gate (aspect ratio of elements connected to the gate). These results are illustrated in Fig. 4, where we show pressure sensitivity fields with respect to the gate movement during the filling stage of a center-gated disk, with the following data: 150 mm diameter, uniform thickness of 3 mm and the viscosity approximated by a power law model with n=0.4 and m=9360 Pa.s$^{0.4}$. Also, Fig. 4 highlights the effect of mesh distortion around the gate on the sensitivity of the pressure, which greatly influences the sensitivity of the objective function. While in Fig. 4(a) we observe a symmetrical distribution of the pressure sensitivity field about the center gate, Fig. 4(b) shows a completely asymmetrical sensitivity, due to the distortion of elements around the gate. Therefore, the rate-of-convergence of the global optimization algorithm will greatly depend on the mesh configuration of the region of search for the optimum.



(a) (b)

Fig. 4. Pressure sensitivity with respect to the gate movement at the end of filling of the center-gated disk. Effect of distortion of elements connected to the gate. Uniform (a) and deformed (b) elements around the center gate.

6 Algorithm

The optimization methodology is completely automated. The objective functional (fill time) and its sensitivities, the constraints and their sensitivities are computed and transferred to DOT [82], a commercially available optimization code. The optimization algorithm proceeds as follows:

Start from an initial guess a_r^q for the design variables (gate locations and injection pressure profile) and for each iteration:

1. Solve the complete filling problem:
   - At each time step during the filling simulation:
     - o Solve the Hele-Shaw equation for p, the energy equation for T and the flow front for f.
     - o Solve the sensitivity equations for Dp/Da_r , DT/Da_r and Df/Da_r.
     - o Evaluate the objective function F, the sensitivities DF/Da_r, the constraints C_i and the sensitivities DC_i/Da_r.
2. At the end of each complete filling:
   - Send different functions and corresponding sensitivities to DOT program for δa_r^q.
   - Compute a new design $a_r^{q+1} = a_r^q + \delta a_r^q$.
3. Go to 1 with the new design until satisfactory convergence is achieved.

7 Illustrative Applications

We now demonstrate with two examples the use of DSA-based methods in optimal design for mold filling problems. The main object here in studying complex geometries is to exhibit the type of convergence that one may expect from the optimization algorithm.

7.1 Automotive Part

As a first application of the DSA-based approach in injection molding, we analyze the filling of a complex automotive non-planar geometry, which has a uniform thickness of 3 mm. The geometry is shown in Fig. 5. The finite element mesh consists of 7887 linear triangular elements and 3965 nodes in the middle plane.

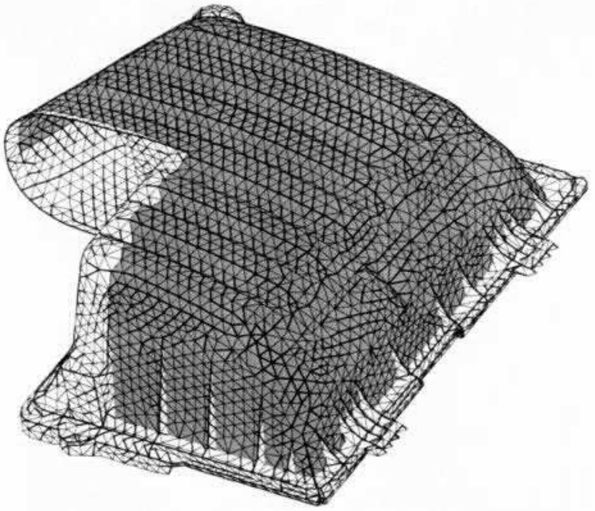


Fig. 5. Mesh and domain search for the optimum in gray.

The material used is a polypropylene, whose viscosity is approximated by the Carreau-Yasuda-WLF model. The material properties and processing conditions are summarized in Table 1.

Table 1 Polypropylene material properties and processing conditions (PP Ma3U from Mitsubishi).

Material Properties	Data
Zero shear rate η_0	13982.6986 Pa.s
Relaxation time λ	0.0598 s
Power index n	0.3423
Yasuda coefficient a	0.7613
WLF coefficient C_1	16.4989
WLF coefficient C_2	496.3101 °C
Reference temperature T_r	203.6024 °C
Density ρ	700 Kg/m^3
Specific heat c	2000 J/Kg °C
Thermal conductivity k	0.19 W/m °C
Processing Conditions	
Injection temperature Ti	220 °C
Interface heat transfer coefficient h	20 W/m^2 °C
Mold Temperature T_w	40 °C

There are four pressure profile parameters and one gate. Therefore the total number of design variables is six. Side constraints of the design variables are set as follows:

$1 \text{ Mpa} \le P_i \le 70 \text{ MPa} (i=1,4)$

$0 \le u, v \le 1$ (Gray region in Fig. 5)

The region of search in the mesh domain that was parameterized to track the gate movement during the optimization process is also shown in Fig. 5 (in gray). The injection temperature was imposed on nodes connected to the gate and on all nodes lying within three element layers around the gate.

The clamp force and flow rate limits are 7E03 kN and 2E04 cm^3/s, respectively. The clamp force is evaluated at 100% fill.

Optimization results are summarized in Table 2. The optimization algorithm requires 16 gradient and 45 function evaluations. Each function or gradient evaluation corresponds to a complete filling simulation. The fill time was reduced in the optimization process from 11.1 s to 3.3 s. The gate location moves from the initial location (-261.1; 651.8; 621.4) mm to the final location (-344.73; 510.73; 659.21) mm, as shown in Fig. 6.

Table 2 Optimization results for the filling of an automotive part.

Optimization Data	Initial Design	Optimal Design
Objective function (s)	11.18	3.31
Constraint C_1 (clamp force)	-6.725E-02	-2.687E-03
Constraint C_2 (flow rate)	0.2727	-0.118
Design Variables		
Injection Pressure (MPa)		
P_1	55	60.03
P_2	56.65	70
P_3	58.3	70
P_4	60	56.92
Gate Coordinates (mm)		
x	-261.1	-344.73
y	651.8	510.73
z	621.4	659.21

Successive gate locations during the optimization process are depicted in Fig. 6. The trajectory shown is only for illustration purposes. The constraints on clamp force (C_1) and gate flow rate (C_2) remain within allowable tolerances, i.e., –2.687E-02 and –0.118, respectively.

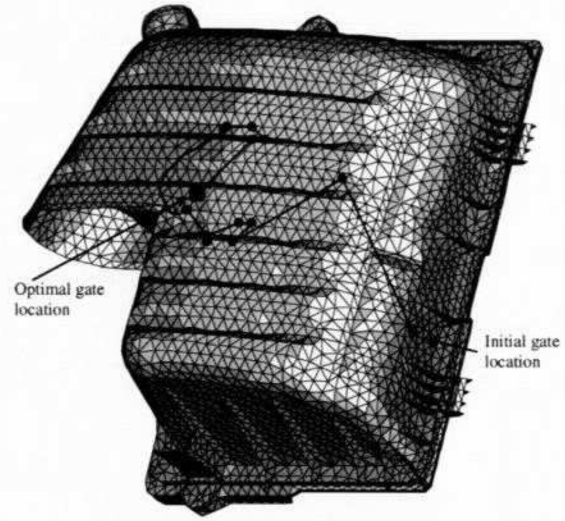
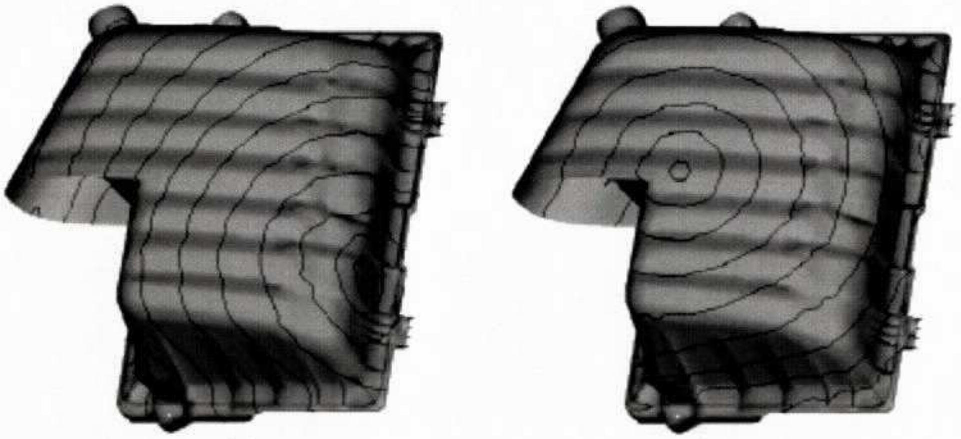


Fig. 6. Successive gate locations during the optimization process. The trajectory is only for illustration purposes.

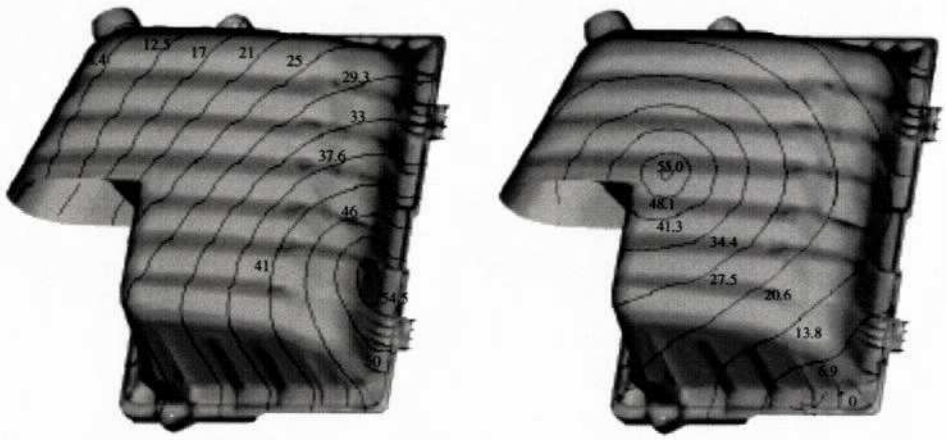
The Pressure, flow-front patterns and average temperature results, for both initial and optimal design, are shown in Figs. 7-9. As can be seen from these figures, optimal gate location considerably affects the fill time, the pressure distribution, the temperature field and the direction of the polymer flow. The optimization produces the desired effect, i.e. the melt reaches the opposite boundaries of the mold almost simultaneously. We also observed that the flow is much better balanced in the optimal design. This will have a beneficial effect on the quality of the part, such as the differential shrinkage and warpage.



(a) (b)

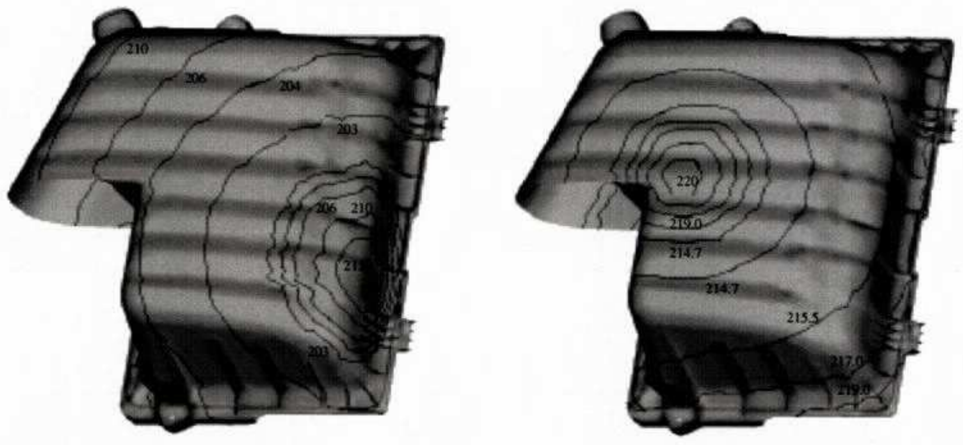
Fig. 7. Initial (a) and optimal (b) designs: successive flow fronts.

The effect of the optimization on the temperature at the end of flow is highlighted in Fig. 9(b). A reasonably uniform average temperature distribution throughout the cavity is achieved. The average temperature difference between the highest and the lowest temperature, in the optimal design, is about 6 °C, while in the initial design, this difference is about 24 °C as shown in Fig. 9(a). This temperature distribution will affect the final deformation of the part, i.e., the shrinkage and warpage as reported in Kabanemi and Crochet [18] and Kabanemi and Dupret [19].



(a) (b)

Fig. 8. Initial (a) and optimal (b) designs: pressure distribution at the end of filling (in MPa).



(a) (b)

Fig. 9. Initial (a) and optimal (b) designs: average temperature distribution (in °C).

A comparison between the initial injection pressure profile and the optimal pressure profile is shown in Fig. 10. At the end of filling, the pressure decreases to satisfy the constraint on the clamp force. The optimal pressure profile is such that side and function constraints are not violated as shown in Table 2. We also observe that the effect of the design variable, pressure P_1, on the objective function (the total fill time) is very small. Therefore, its value does not change much during the optimization process, consistent with the gradient-based method used. Yet, the constraint on the maximum flow rate greatly influences the design variable P_1, while design variables P_3 and P_4, are limited (in their change) by the constraint on the maximum clamp force, in addition to be bounded by side constraints.

In Fig. 11, we show the history of the objective function during the optimization process. After 12 iterations, there is no change in the objective function.

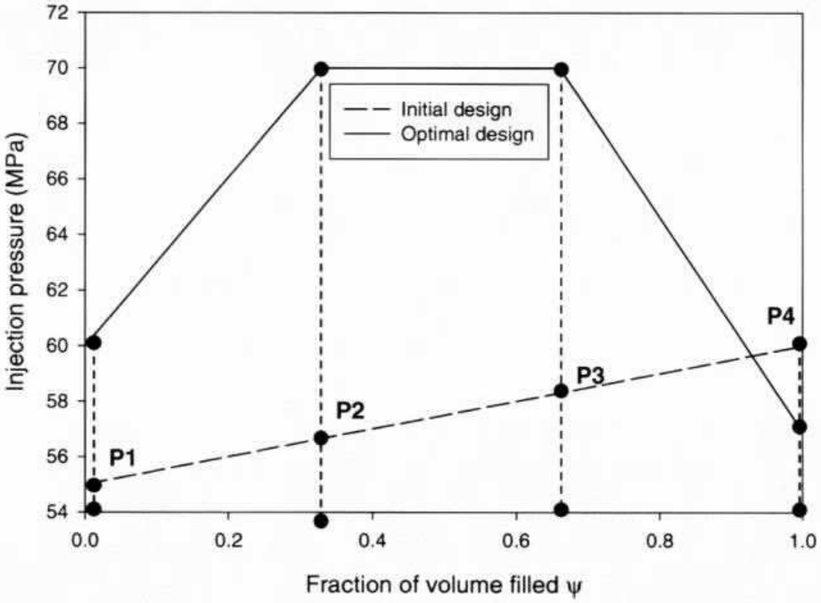


Fig. 10. Injection pressure profiles before and after optimization.

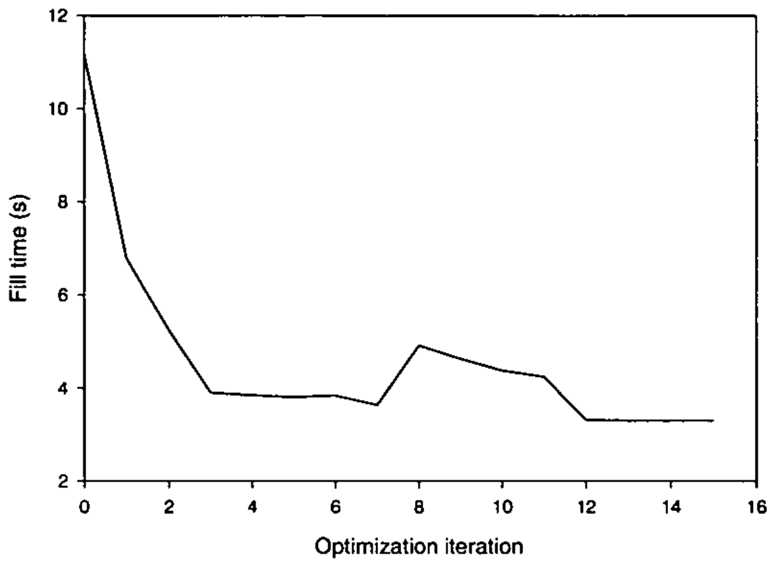


Fig. 11. Fill time history for the optimization of an automotive part.

Table 3. Optimization results for the filling of an automotive part.

Optimization Data	Initial Design	Optimal Design
Objective function (s) Constraint C_1 (clamp force) Constraint C_2 (flow rate)	12.52 -0.117 -0.298	3.60 -2.951E-02 -0.3166
Design Variables		
Injection Pressure (MPa)		
P_1	55	53.41
P_2	56.65	70
P_3	58.3	70
P_4	60	54.87
Gate Coordinates (mm)		
x	-201.6	-343.0
y	561.3	502.6
z	660.5	660.9

Let us now analyze the effect of changing the initial gate location (Fig. 12) on the optimal design. The material data, side and function constraints remain the same. The optimization algorithm requires 7 gradient and 21 function evaluations, much lower than the number of function evaluations needed to achieve optimal design in the preceding initial configuration. Results are summarized in Table 3.

The gate location moves from the initial location (-201.62; 561.39; 660.59) mm to the final location (-343.02; 502.65; 660.94) mm, as shown in Fig. 12. In the same figure, successive gate locations during the optimization process are also depicted. Constraints on clamp force (C_1) and gate flow rate (C_2) remain within allowable tolerances, i.e., -2.951E-02 and –0.3166, respectively.

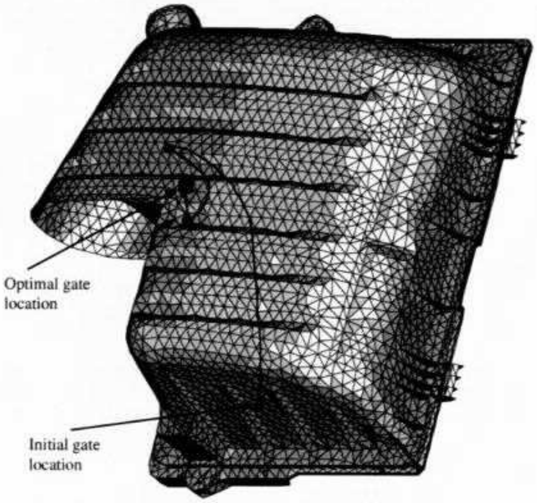


Fig. 12. Successive gate locations during the optimization. The trajectory is only for illustration purposes.

A comparison between the initial pressure profile and the optimal pressure profile is shown in Fig. 13. Once again, at the end of filling the pressure decreases to satisfy the constraint on the clamp force. Figure 14 shows the history of the objective function during the optimization process. After 4 iterations, there is no change in the objective function. These results show that the initial gate location has a small effect on the optimal design (of the order of percentage), though the objective is about the same. Therefore, the optimal design obtained, i.e., the gate location and the injection pressure profile, can thus be considered to be the global optimum of this problem.

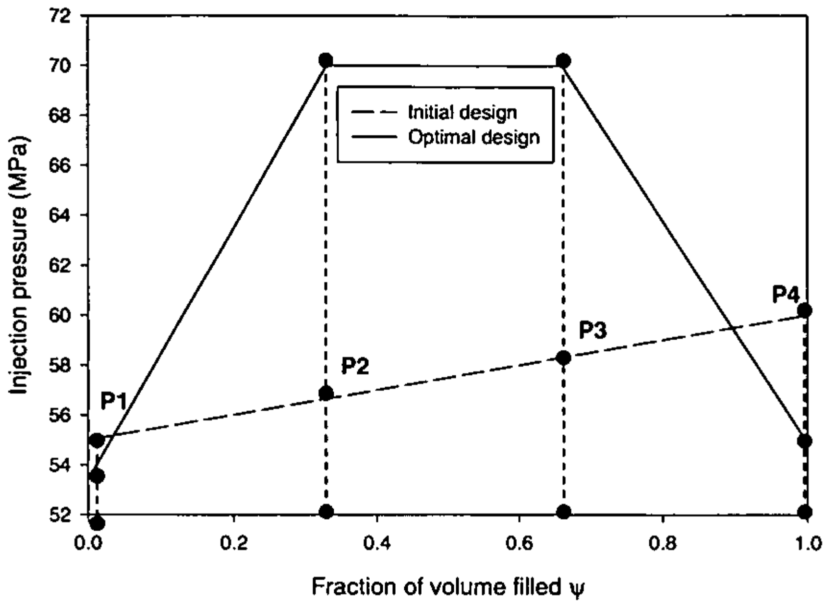


Fig. 13. Injection pressure profiles before and after optimization.

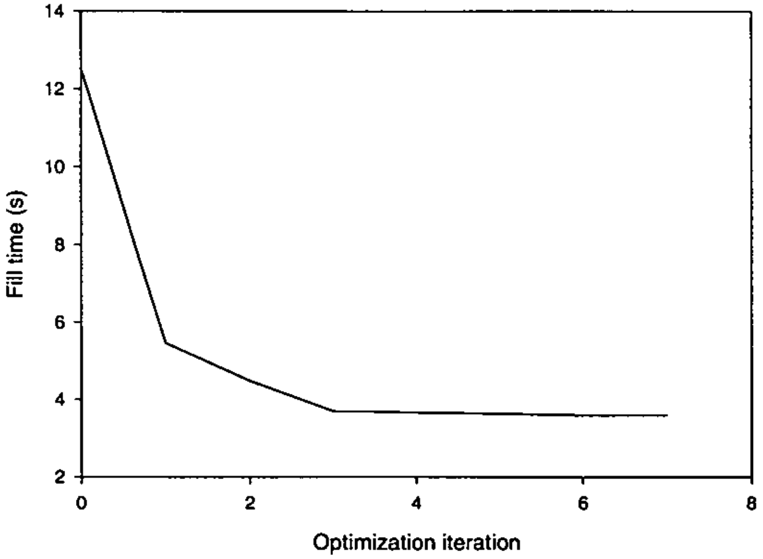


Fig. 14. Fill time history for the optimization of an automotive part.

7.2 Automotive Lens

As a second application, we analyze the non-isothermal filling of an automotive lens mold cavity shown in Fig. 15, which has a uniform thickness of 3 mm. This case is analyzed to demonstrate the applicability of the method to multiple gate locations. We also choose to perform edge gate optimization. The finite element mesh consists of 3393 linear triangular elements and 1840 nodes in the middle plane. The material used is a PMMA, whose viscosity is approximated by the Carreau-Yasuda model. The material properties and processing conditions are summarized in Table 4.

Table 4. PMMA material properties and processing conditions (LUCRYL G88, PMMA, BASF AG).

Material Properties	Data
Zero shear rate η_0	10071.1678 Pa.s
Relaxation time λ	0.0642 s
Power index n	0.4
Yasuda coefficient a	1.0
WLF coefficient C_1	14.3245
WLF coefficient C_2	144.6981 °C
Reference temperature T_r	220.0981 °C
Density ρ	1010 Kg/m^3
Specific heat c	2554 J/Kg °C
Thermal conductivity k	0.16 W/m °C
Processing Conditions	
Injection temperature Ti	230 °C
Interface heat transfer coefficient h	200 W/m^2 °C
Mold Temperature T_w	90 °C

There are four pressure profile parameters and three edge gates (Fig. 15). Side constraints for the design variables are set as follows:

$1 \text{ Mpa} \leq P_i \leq 60 \text{ MPa} \, (i=1,4)$

$0 \leq u \leq 1$ and $v = 0$

Therefore the total number of design variable is seven. The edge segments parameterized to track the gates movement during the optimization process are shown in Fig. 15. The clamp force and flow rate limits are 1500 kN and 2E04 cm^3/s, respectively. The clamp force is evaluated at 100% fill.

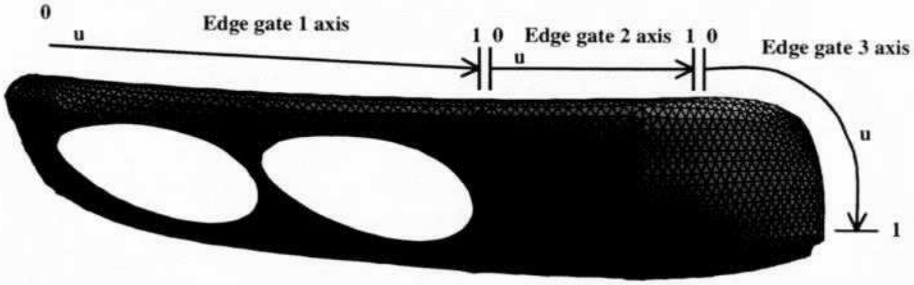


Fig. 15. Gate axis definition and finite element of an automotive lens mold cavity with three edge gates.

The optimization algorithm requires 4 gradient and 8 function evaluations. Optimization results are summarized in Table 5. The fill time was reduced in the optimization process from 2.33 s to 0.59 s. The gate locations move from their initial edge locations, i.e., u=0.566, u=0.943 and u=0.743, for the first, second and third gate, respectively, to their final locations, i.e., u=0.049, u=0.116 and u=0.940, as shown in Fig. 16. The constraints on clamp force (C_1) and gate flow rate (C_2) remain within allowable tolerances, i.e., –1.487E-03 and –0.759, respectively.

Table 5. Optimization results for the filling of an automotive lens mold cavity.

Optimization Data	Initial Design	Optimal Design
Objective function (s)	2.33	0.59
Constraint C_1 (clamp force)	0.137	-1.487 E-03
Constraint C_2 (flow rate)	-0.221	-0.759
Design Variables		
Injection Pressure (MPa)		
P_1	35	21.23
P_2	36.65	46.85
P_3	38.3	60
P_4	50	46.15
Parametric Gate Coordinates		
u for gate 1	0.566	0.0497
u for gate 2	0.943	0.116
u for gate 3	0.743	0.940

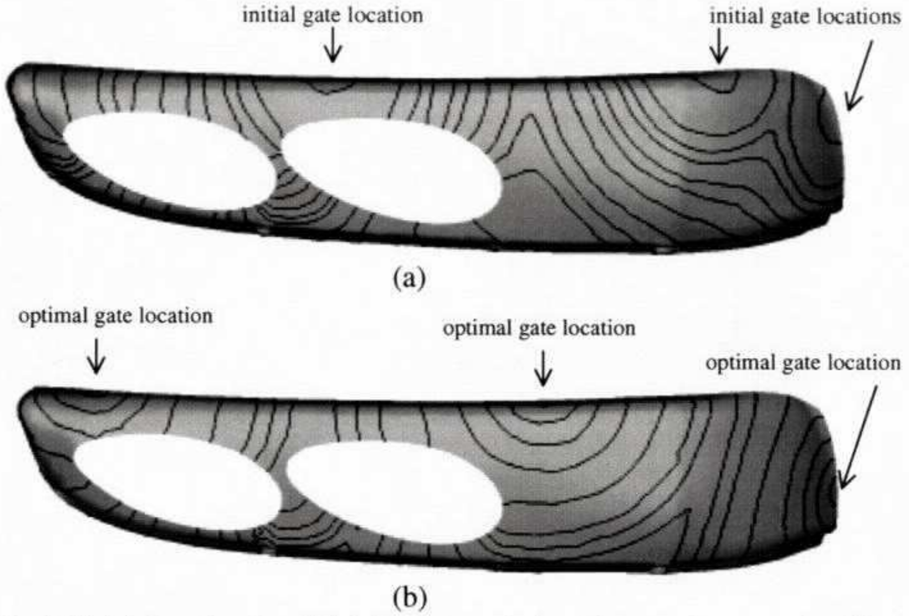


Fig. 16. Initial (a) and optimal (b) designs: gate locations and successive flow fronts. The time step between successive flow fronts plotted is not constant.

Filling fronts and pressure results, for both initial and optimal design, are shown in Figs. 16 and 17. As can be seen from these figures, optimal gate location considerably affects the fill time, the pressure distribution, and the position of weldlines.

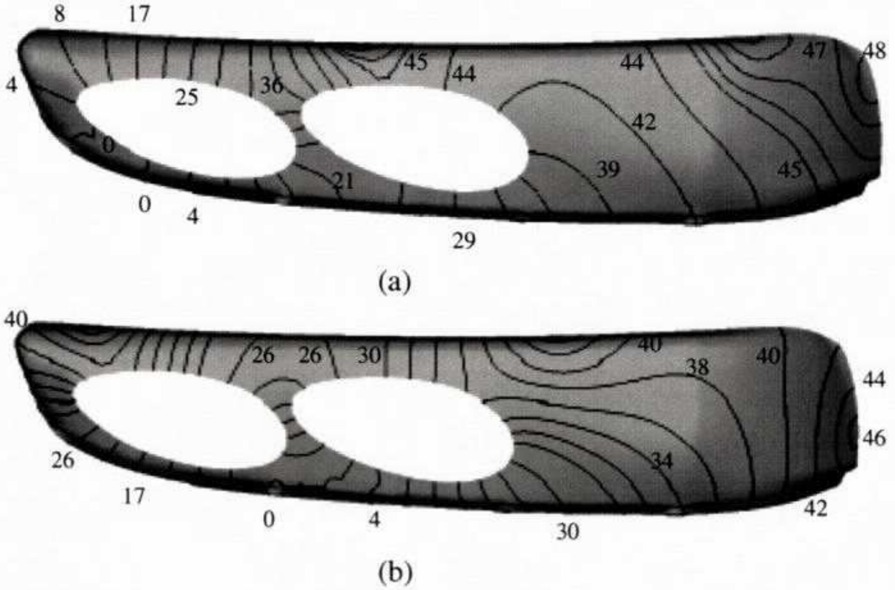


Fig. 17. Initial (a) and optimal (b) designs: predicted pressure distribution at the end of filling (in MPa).

A comparison between the initial and the optimal injection pressure profiles is shown in Fig. 18. As in the preceding application, we observe a decrease of the pressure at the end of filling, to satisfy the constraint on the clamp force.

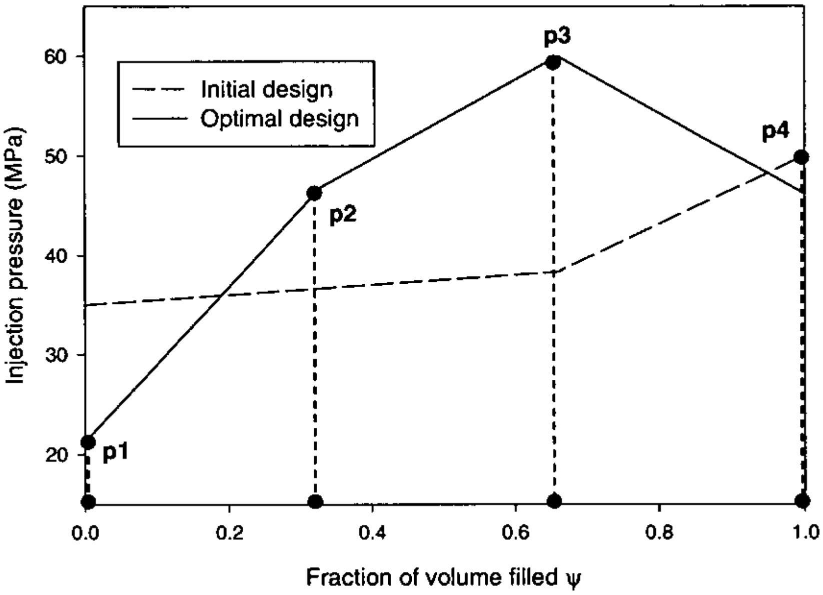


Fig. 18. Injection pressure profiles before and after optimization.

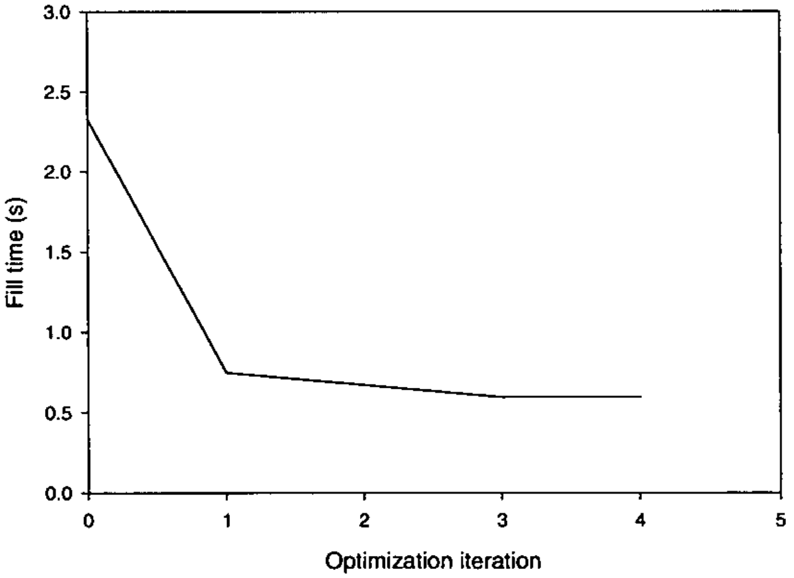


Fig. 19. Fill time history for the optimization of an automotive lens mold cavity with three edge gates.

Figure 19 shows the history of the objective function during the optimization process. Only 3 global iterations were necessary to achieve the optimal design.

The mathematical optimization model developed here has revealed that the objective functional is a highly nonlinear functional with respect to the design variables. For this reason, multiple optimal gate locations may be found from our optimization solution that depend on: the initial gate locations; the number of gates; the geometry of the mold cavity; and the symmetry of the part. Such results are shown in Fig. 20, where we can see the effect of the part symmetry on the optimal design.

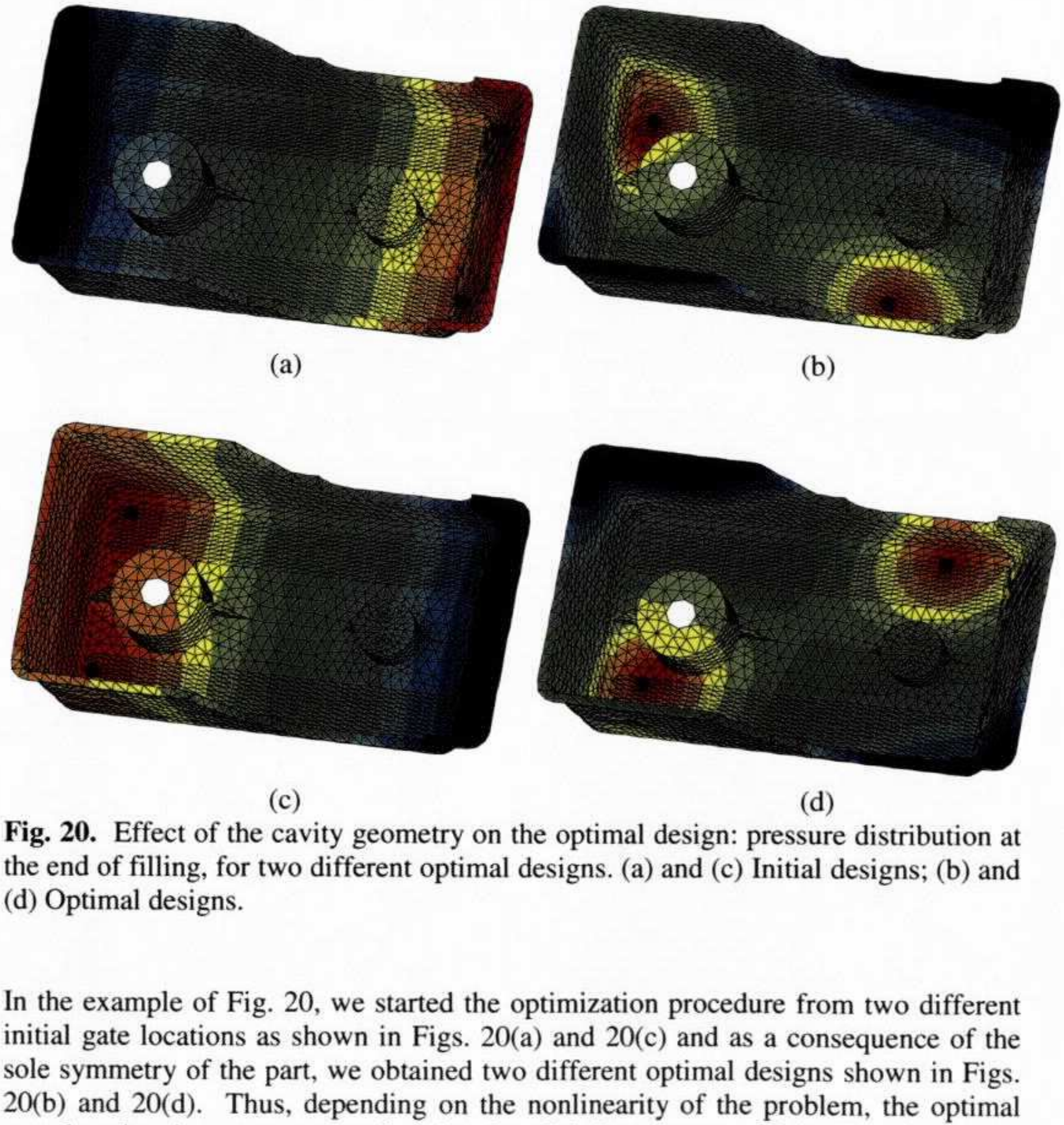


Fig. 20. Effect of the cavity geometry on the optimal design: pressure distribution at the end of filling, for two different optimal designs. (a) and (c) Initial designs; (b) and (d) Optimal designs.

In the example of Fig. 20, we started the optimization procedure from two different initial gate locations as shown in Figs. 20(a) and 20(c) and as a consequence of the sole symmetry of the part, we obtained two different optimal designs shown in Figs. 20(b) and 20(d). Thus, depending on the nonlinearity of the problem, the optimal gate location is not guaranteed to be the global optimum. Therefore, it may be necessary to restart the optimization process from several different gate locations to provide reasonable assurance of obtaining the global optimum. Yet, since the design space is often ''flat'', starting from different initial points will usually produce a slightly different optimum, though the objective function will be about the same. In case of multiple optimum gate locations, the final decision of the placement of gates in a mold cavity will be dictated by factors such as, weld-line position in the cavity, pressure distribution, temperature field, shrinkage and warpage, among others.

Finally we wish to point out that in the course of the optimization procedure, the function evaluation involves a complete filling simulation. Such a result, for the mold cavity of Fig. 20, is shown in Fig. 21, which gives the history of the objective functional during the optimization process. This figure clearly shows the increasing number of function evaluation before the optimal design is reached. While this may seem to be time consuming for highly nonlinear problems and complex mold cavity geometries, these optimization techniques nevertheless provide to engineers an efficient tool for automatically finding the best possible design from admissible ones, instead of using 'intuitive' design methods based on trial-and-error.

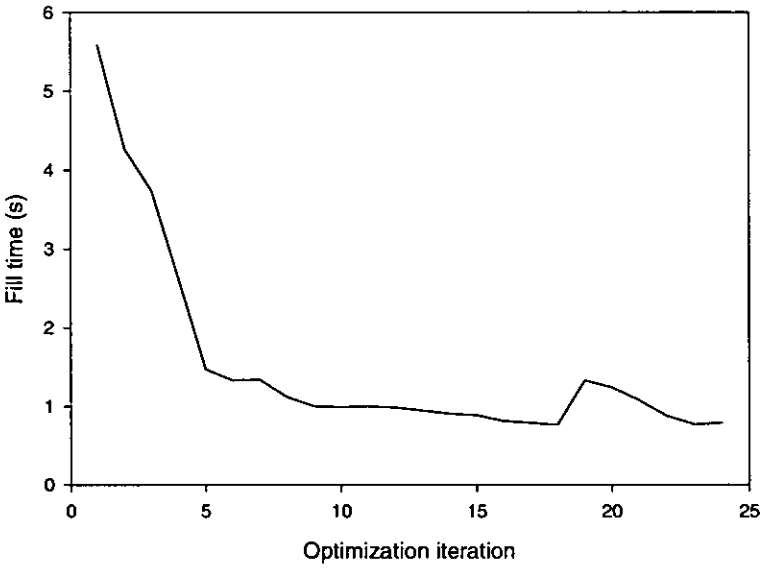


Fig. 21. Optimization history.

8 Conclusions

We have discussed in this chapter the applicability of the DSA-based approach to optimal design for mold filling problems. This technique enables efficient analyses and optimal design for computationally expensive injection molding systems. Its effectiveness as a tool for the optimization of the injection molding process has been demonstrated from two different industrial applications. The approach developed here combines an existing injection molding simulation code with an existing numerical optimization program (DOT), which enables the selection and employment of the most suitable optimization algorithm. The emphasis was put on gradient-based optimization methods. Those methods are local optimization methods and best suited in situations where the design space is convex and the objective functional continuous. We have also briefly discussed the use of stochastic-based methods as they provide a tool for finding a global or a nearly global optimum and can deal with non-convex and discontinuous functions. The drawback of those stochastic techniques remains the computational time that is required to achieve the global optimum.

References

1 Himasekhar, K., Lottey, L. and Wang, K.K.: CAE of mold cooling in injection molding using a three-dimensional numerical simulation, Trans. ASME, J. Eng. For Industry, vol. 114, pp. 213-221, (1992).

2 Wang, V.W., Hieber, C.A. and Wang, K.K.: Dynamic simulation and graphics for the injection molding of three-dimensional thin parts, J. Polym. Eng., vol. 7, pp. 21-45, (1986).

3 Hieber, C.A. and Shen, S.F.: A finite element/finite-difference simulation of the injection molding filling process, J. Non-Newt. Fl. Mech., 7, pp. 1-32, (1980).

4 Hieber, C.A.: Injection and compression molding fundamentals, Chapter 1, Isayev, A.I., ed., Marcel Dekker, New York, (1987).

5 Chiang, H.H., Hieber, C.A. and Wang, K.K.: A unified simulation of the filling and postfilling stages in injection molding. Part I: Formulation, Polym. Eng. Sci., 31(2), pp. 116-124, (1991).

6 Chiang, H.H., Hieber, C.A. and Wang, K.K.: A unified simulation of the filling and postfilling stages in injection molding. Part II: Experimental verification, Polym. Eng. Sci., 31(2), pp. 125-139, (1991).

7 Turng, L.S., Wang, V.W. and Wang, K.K.: Numerical simulation of the coinjection molding process, J. Eng. Mat. Techn., vol. 115, pp. 48-53, (1993).

8 Chiang, H.H., Himasekhar, K., Santhanam, N. and Wang, K.K.: Integrated simulation of fluid flow and heat transfer in injection molding for the prediction of shrinkage and warpage, J. Eng. Mat. Techn., vol. 115, pp. 37-47, (1993).

9 Himasekhar, K., Turng, L.S., Wang, V.W., Chiang, H.H. and Wang, K.K.: Current trends in CAE: simulations of lates innovations in injection molding, Adv. Polym. Technol., 12(3), pp. 233-241, (1993).

10 Turng, L.S.: Development and application of CAE technology for the gas-assisted injection molding process, Adv. Polym. Technol., 14(1), pp. 1-13, (1995).

11 Wang, K.K.: CAE for injection molding: what's next, In Proc.: CAE and Related Innovation for Polymer Processing ASME 2000, MD-vol. 90, pp. 299-305, (2000).

12 Dupret, F., Couniot, A., Mal, O., Vanderschuren, L. and Verhoyen, O.: Modelling and Simulation of Injection Molding, In Advances in the Flow and Rheology of Non-Newtonian Fluids, D.A. Siginer, D. De Kee and R.P. Chhabra editors, Rheology Series, Elsevier, Amsterdam, (1999).

13 Dupret, F. and Verleye, V.: Modelling the flow of fiber suspensions in narrow gaps, In Advances in the Flow and Rheology of Non-Newtonian Fluids, D.A. Siginer, D. De Kee and R.P. Chhabra editors, Rheology Series, Elsevier, Amsterdam, (1999).

14 Dupret, F. and Vanderschuren, L. : Calculation of the temperature field in injection molding, AIChE Journal, 4 (12), pp. 1959-1972, (1988).

15 Crochet, M.J, Dupret, F. and Verleye, V.: Injection Molding, In Flow and Rheology in Polymer Composites Manufacturing, pp. 415-463, edited by S.G. Advani, Elsevier, (1994).

16 Couniot, A. and Crochet, M.J.: Finite elements for the numerical simulation of injection molding, In Proc.: Numiform 86, Gothenburg, Sweden, pp. 165-170, (1986).

17 Couniot, A., Dheur, L., Hansen O. and Dupret, F.: A finite element method for simulating injection molding of thermoplastics, In Proc. Numiform 89, Thomson et al. (eds), Balkema, Rotterdam, pp. 235-241, (1989)

18 Kabanemi, K.K. and Crochet, M.J.: Thermoviscoelastic calculation of residual stresses and residual shapes of injection molded parts, Int. Polymer Processing, VII(1), pp. 60-70, (1992).

19 Kabanemi, K.K. and Dupret, F.: Analysis of the influence of the packing stage on residual stresses and shrinkage of injection molded parts, In Proc. Numiform 92, Chenot, Wood & Zienckiewicz (eds), Balkema, Rotterdam, pp. 357-363, (1992).

20 Mal, O., Dheur, L., Pirotte, P., Van Rutten, N., Couniot, A. and Dupret, F.: A realistic wall thermal boundary condition for simulating the injection molding of thermoplastics, In Proc. Numiform 95, Shen & Dawson (eds), Balkema, Rotterdam, pp. 1165-1171, (1995)

21 Verhoyen, O., Dupret, F. and Legras, R.: Polym. Eng. Sci., (1998)

22 Verhoyen, O., Dupret, F. and Legras, R.: Computer simulation of crystallization of PET in injection molding, In Proc. Numiform 95, Shen & Dawson (eds), Balkema, Rotterdam, pp. 1209-1212, (1995)

23 Michaeli, W., Hoffmann, S., Kratz, M. and Webelhaus, K.: Simulation opportunities by a three-dimensional calculation of injection moulding based on the finite element method, Intern. Polymer Processing, vol. 16(4), pp. 398-403, (2001).

24 Michaeli, W., Kratz, M. and Hoffmann, S.: 3D-Flow analysis by means of streamline calculation, Macromolecular Materials and Engineering, vol. 286, pp. 774-779, (2001).

25 Niggemeier, P. and Michaeli, W.: Improvement of the simulation of shrinkage and warpage by characterizing the material behaviour more exactly, In Proc. ANTEC '99, New York 57, pp. 737-743, (1999).

26 Michaeli, W. and Hoffmann, S.: Calculation of crystallization with injection moulding simulation software based on three-dimensional approaches, Macromolecular Materials and Engineering, vol. 286, pp. 43-47, (2001).

27 Hoffmann, S. and Michaeli, W.: Analysis of internal structure of injection molded parts based on a three-dimensional simulation software, Journal of Injection Molding Technology, vol. 5(2), pp. 98-104, (2001).

28 Agassant, J.F., Alles, H., Philipon, S. and Vincent, M.: Experimental and theoretical study of the injection molding of thermoplastic materials, Polym. Eng. Sci., vol. 28(7), pp. 460-468, (1988).

29 Pichelin, E. and Coupez, T.: Finite element solution of the 3D mold filling problem for viscous incompressible fluid, Comput. Methods Appl. Mech. Eng., vol. 163, pp. 359-371, (1998).

30 Pichelin, E. and Coupez, T.: A Taylor discontinuous Galerkin method for the thermal solution in 3D mold filling, Comput. Methods Appl. Mech. Eng., vol. 178, pp. 153-169, (1999).

31 Denizart, O., Vincent, M. and Agassant, J.F.: Thermal stresses and strains in injection molding: experiments and computations, J. Mat. Sci., vol. 30, pp. 552-560, (1995).

32 Hetu, J.F., Gao, D.M., Garcia-Rejon, A. and Salloum, G.: 3D finite element method for the simulation of the filling stage in injection molding, Polym. Eng. Sci., vol. 38(2), pp. 223-236, (1998).

33 Ilinca, F. and Hétu, J.F.: Three-dimensional filling and post-filling simulation of polymer injection molding, Int. Polymer Processing, vol. 16(3), pp. 291-301, (2001).

34 Ilinca, F. and Hétu, J.F.: Three-dimensional numerical modeling of co-injection molding, Intern. Polymer Processing, vol. 17(3), pp. 265-270, (2002).

35 Kabanemi, K.K., Hétu, K.K. and Garcia-Rejon, A.: Numerical simulation of the flow and fiber orientation in reinforced thermoplastic injection molded products, Int. Polymer Processing, vol. 12, pp. 182-191, (1997)

36 Kabanemi, K.K., Vaillancourt, H., Wang, H. and Salloum, G.: Residual stresses, shrinkage, and warpage of complex injection molded products: Numerical simulation and experimental validation, Polym. Eng. Sci., vol. 38(1), pp. 21-37, (1998).

37 Wang, H., Kabanemi, K.K. and Salloum, G.: Numerical and experimental studies on the ejection of injection-molded plastic products, Polym. Eng. Sci., vol. 40(3), pp. 826-840, (2000).

38 Kamal, M.R. and Kenig, S.: The injection molding of thermoplastics. Part I: Theoretical model, Polym. Eng. Sci., vol. 12(4), pp. 294-308, (1972).

39 Kuo, Y. and Kamal, M.R.: The fluid mechanics and heat transfer of injection mold filling of thermoplastic materials, AIChE J., vol. 22(4), pp. 661-669, (1976)

40 Kamal, M.R. and Lafleur P.G.: Polym. Eng. Sci., vol. 26, pp. 103-110, (1986).

41 Kamal, M.R. and Ryan, M.R., in: A.I. Isayev (ed.), Injection and compression molding fundamentals, Marcel Dekker, New York, (1987).

42 Kamal, M.R., in: L.T. Manzione (ed.), Applications of CAE in injection molding, Hanser, Munich, (1987).

43 Tucker III, C.L.: Computer Modeling for Polymer Processing, Hanser, (1989).

44 Tucker III, C.L., in: Isayev, A.I. (ed.), Injection and compression molding fundamentals, Marcel Dekker, New York, (1987).

45 Williams, G. and Lord, H.A.: Mold-filling studies for injection molding of thermoplastic materials: Part I: The flow of plastic materials in hot- and cold-walled circular channels, Polym. Eng. Sci., vol. 15(8), pp. 553-568, (1975).

46 Lord, H.A. and Williams, G.: Mold-filling studies for injection molding of thermoplastic materials: Part II: The transient flow of plastic materials in the cavities of injection-molding dies, Polym. Eng. Sci., vol. 15(8), pp. 569-582, (1975).

47 Moldflow, www/moldflow.com.

48 Moldex, www/moldex.com.

49 Kirsch, U.: Structural optimization: fundamental and applications, Springer-Verlag, (1993).

50 Vanderplaats, G.N.: Numerical optimization techniques for engineering design, McGraw-Hill, New York, (1984).

51 Vanderplaats, G.N.: Numerical optimization techniques for engineering design, 3^{rd} ed., Second printing, VR&D, Inc., Colorado Spring, (1999).

52 Kirsch, U.: Optimum structural design, MacGraw-Hill, New York, (1981).

53 Kuang-Hua, C.: Shape design sensitivity analysis and what-if design workstation for elastic structural components, Ph.D. thesis, The university of Iowa, (1990).

54 Adeli, H.: Advances in design optimization, Chapman & Hall, (1994).

55 Haftka, R.T. and Gürdal, Z.: Elements of structural optimization, 3rd ed., Kluwer Academic Publishers, (1992).

56 Fox, R.L.: Optimization methods for engineering design, Addison-Wesley, Reading, Mass., (1971).

57 Pandelidis I. And Zou, Q.: Optimization of injection molding design. Part I: Gate location optimization, 30(15), pp. 873-882, (1990).

58 Pandelidis I. And Zou, Q.: Optimization of injection molding design. Part II: Molding conditions optimization, 30(15), pp. 883-892, (1990).

59 Lee, B.H. and Kim, B.H.: Optimization of part wall thickness to reduce warpage of injection-molded parts based on the modified complex method, Polym.-Plast. Technol. Eng., vol. 34(5), pp. 793-811, (1995).

60 Young, W.B.: Gate location optimization in liquid composite molding using genetic algorithms, J. Composite Mat., vol. 28(12), pp. 1098-1113, (1994).

61 Kang, S., Hieber, C.A. and Wang, K.K.: Optimum design of process conditions to minimize residual stresses in injection-molded parts, in Proc.: Antec'95, pp. 991-996, (1995).

62 Smith, D.E., Tortorelli, D.A. and Tucker III, D.L.: Analysis and sensitivity analysis for polymer injection and compression molding, Comput. Methods Appl. Mech. Eng., 167, pp. 325-344, (1998).

63 Smith, D.E, Chen, C.J., Usman, M. and Koskey, J.: Multi-gate injection molding process optimization using design sensitivity analysis, ASME, vol. 227, pp. 259-272, (1997).

64 Smith, D.E.: Optimal design and analysis for polymer extrusion and molding, Ph. D. Thesis, University of Illinois at Urbana-Champaign, (1996).

65 Kabanemi, K.K., Hétu, J.F. and Derdouri, A.: Design sensitivity analysis applied to injection molding for optimization of gate location and injection pressure, Intern. Polymer Processing, vol. 17(3), pp. 254-264, (2002).

66 Kabanemi, K.K., Derdouri, A. and Hétu, J.F.: Optimization of the injection molding process using design sensitivity analysis, National Research Council Canada, Report IMI-2000-84100-C, (2000).

67 Turng, L.S. and Peic, M.: Computer aided process and design optimization for injection molding, Proc. Instn. Mech. Engrs., vol. 216, Part B: J. Engineering Manufacture, pp. 1523-1532, (2002).

68 Tan, K.H. and Yuen, M.F.: A fuzzy multiobjective Approach for minimization of injection molding defects, Polym. Eng. Sci., vol. 40(4), pp. 956-971, (2000).

69 Lam, Y.C., Zhai, L.Y., Tai, K. and Fok, S.C.: An evolutionary approach for cooling system optimization in plastic injection moulding, Int. J. Prod. Res., vol. 42(10), pp. 2047-2061, (2004).

70 Bird, R.B., Armstrong, R.C. and Hassager, O.: Dynamics of Polymeric Liquids. John Wiley & Sons, New York, Chichester, Brisbane, Toronto, Singapore (1987).

71 Crochet, M.J., Davies, A.R. and Waletrs, K.: Numerical simulation of non-Newtonian flow, Rheology Series 1, Elsevier, Amsterdam, Oxford, New York, Tokyo, (1984).

72 Goldberg, D.E.: Genetic algorithms, Addison Wesley, (1989).

73 Totorelli, D.A.: Sensitivity analysis for non-linear constrained elastostatic systems, Intern. J. Num. Meth. Eng., vol. 33, pp. 1643-1660, (1992).

74 Totorelli, D.A. and Michaleris, P.: Design sensitivity analysis: Overview and review, Inverse Problems in Engineering, vol. 1, pp. 71-105, (1994).

75 Totorelli, D.A. and Haber, R.B.: First-order design sensitivities for transient conduction problems by an adjoint method, Intern. J. Num. Meth. Eng., vol. 28, pp. 733-752, (1989).

76 Vidal, C.A. and Haber, R.B.: Design sensitivity analysis for rate-independent elastoplasticity, Computer Meth. Appl. Mech. Eng., vol. 107, pp. 393-431, (1993).

77 Chang, K.H. and Choi, K.K.: A geometry-based parameterization method for shape design of elastic solids, Mech. Struct. & Mach., vol. 20(2), pp. 215-252, (1992).

78 McDavid, M.R. and Dantzig, J.A.: Design sensitivity and finite element analysis of free surface flows with application to optimal design of casting rigging systems, International J. for Numer. Methods in Fl., 28, pp. 419-442, (1998).

79 Haug, E.J., Choi, K.K. and Komkov, V.: Design analysis of structural systems, Mathematics in Science and Engineering, Vol. 177, Academic Press, (1986).

80 Lund, E.: Finite element based design sensitivity analysis and optimization, Ph. D. Thesis, Institute of Mechanical Engineering, Aalborg University, (1994).

81 Tortorelli, A.D., Tomasko, J.A. and Morthland, T.E.: Optimal design of nonlinear parabolic systems. Part II: Variable spatial domain with application to casting optimization, Comput. Meth. Appl. Mech. Eng., 113, pp. 157-172, (1994).

82 VR&D, Inc., DOT Design Optimization Tools, Users Manual, Version 5.0, (1999).

83 Dierckx, P.: Curve and surface fitting with splines, Oxford Science Publication, Clarendon Press, Oxford, (1993).