# Meshfree Simulation of Deforming Domains*

Vadim Shapiro† Igor Tsukanov‡
Spatial Automation Laboratory
University of Wisconsin-Madison
1513 University Avenue Madison, WI 53706

**PRELIMINARY DRAFT of May 7, 1999**

**Abstract**

Spatial discretization of the domain and/or boundary conditions prevents application of many numerical techniques to physical problems with time-varying geometry and boundary conditions. By contrast, the $R$-functions method (RFM) for solving boundary and initial value problems discretizes not the domain but the underlying functional space, while the prescribed boundary conditions are satisfied exactly. The clean and modular separation of geometric information from the numerical procedures results in a solution technique that is essentially meshfree and allows an almost effortless modification of geometrical shapes, boundary conditions, and the governing equations. We show that these properties of the RFM make it highly suitable for automated modeling and simulation of non-stationary physical problems with time-varying geometries and boundary conditions.

# 1 Introduction

## 1.1 Problems with time-varying geometry

Many practical engineering situations require solving problems with time-varying geometry or boundary conditions. Important examples of such problems include simulation of engine combustion, metal forming and removing operations, and processes involving phase transitions (with moving boundaries). A two-dimensional example of such a problem, simplified from [12], is illustrated in Figure 1. In this case, the geometric domain is the union of a (moving) piston and cylinder; at any given time instant, the time-varying domain is determined by the relative position of the piston and the cylinder. Modeling and simulating the space-time temperature distribution for this problem requires solving a non-stationary heat conduction problem, which is usually approximated by solving a quasi-steady boundary value problem at small time intervals.

Most modern methods for solving boundary and initial condition problems, including finite elements, finite differences, and boundary-integral methods, involve spatial discretization of both the domain and the boundary conditions. Spatial discretizations are an inconvenience and a source of errors for boundary value problems with fixed geometric conditions. However, for boundary value problems with time-varying geometry, spatial discretizations present insurmountable difficulties. Figure 2(a) shows the spatial discretization of the domain in Figure 1. As pointed out in [12], the high temperature transience requires higher discretization density near the inner walls of the cylinder and the piston. However, it should be clear from Figure 2(b), that mismatch between the meshes of the piston and the cylinder does not allow them to be joined together at desired time intervals. In other words, the nonuniform grid (which makes good sense from the physical point of view) makes modeling of uniform motion difficult or impossible. In this particular case, the problem was solved by adapting a much finer grid than was dictated by the problem, and at a significant additional computational cost. This simple example illustrates the fundamental limitation of all spatial discretization methods: spatial discretization greatly restricts the ability of the domain to deform and/or move; it has to be constructed by a tedious and lengthy process that must take into consideration and accommodate all *a priori*

*To appear in Computer Aided Design
†Corresponding author, e-mail: vshapiro@engr.wisc.edu
‡e-mail: igor@sal-cnc.me.wisc.edu

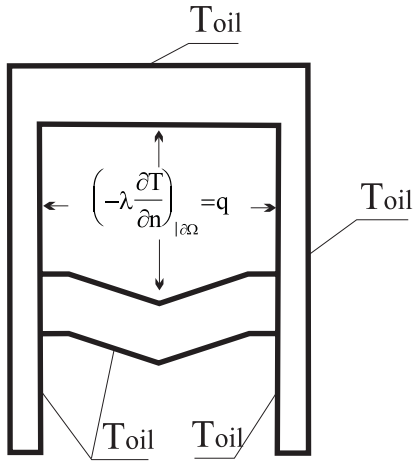



Figure 1: A simplified model of heat transfer in the engine combustion chamber

known motions. This may be particularly difficult or impossible for problems involving more general and sometimes unpredictable motions that may or may not preserve orientation, size, or shape of the geometric domain.

In this paper, we propose a different approach to solving problems with time-varying geometry that does not employ spatial discretization of the domain or the boundary conditions. Our approach is based on the $R$-functions method (RFM) for solving boundary and initial value problems [22, 24, 25]. RFM allows the given boundary conditions to be satisfied *exactly* and discretizes not the domain but the underlying functional space. In this sense, RFM is essentially a *meshfree* method of analysis and simulation. In this paper, the word 'meshfree' does not necessarily imply the absence of a spatial grid. A grid may be convenient or even necessary for integration and/or visualization purposes. For example, many of the pictures in this paper were generated by plotting the functional values over a uniform rectangular grid of points. Rather, the term 'meshfree' means that any spatial decomposition and the resulting grid of points apply to the underlying space and do not conform to the modeled geometric domain. Below we explain the general method, illustrate its application on the example in Figure 1, and discuss the major issues in developing a general-purpose system for automated modeling and simulation of non-stationary physical problems with time-varying geometries and boundary conditions.

## 1.2 Related work

There have been a number of attempts to address the issue of the solution dependence on spatial discretization. Notably, Cox and Anderson [7, 36] proposed approximate solutions using spatial discretizations that are locally restricted to common shape primitives; Zagajac [37] proposed a method to approximate a local solution at a point using a Monte-Carlo method whose numerical complexity is not greatly affected by the geometric complexity of the domain; Sethian [27] proposed a method for simulating a variety of problems with moving interfaces by first reformulating them in terms of the motion of a higher-dimensional hypersurface. Finally, the recent advances in "meshfree" methods seek more flexible spatial discretizations that do not have to conform to the geometric conditions of the boundary value problems [4].

Despite the great recent progress, none of the above methods have matured enough to provide the tools necessary for automated analysis and simulation of problems with time-varying geometric conditions such as discussed in this paper.

## 1.3 Outline

The rest of paper is organized as follows. Section 2 explains the basic idea behind the RFM and shows that it cleanly separates the geometric information from the physical models and numerical procedures. This results in a solution technique that is essentially meshfree and allows an almost effortless modification of geometrical shapes, boundary

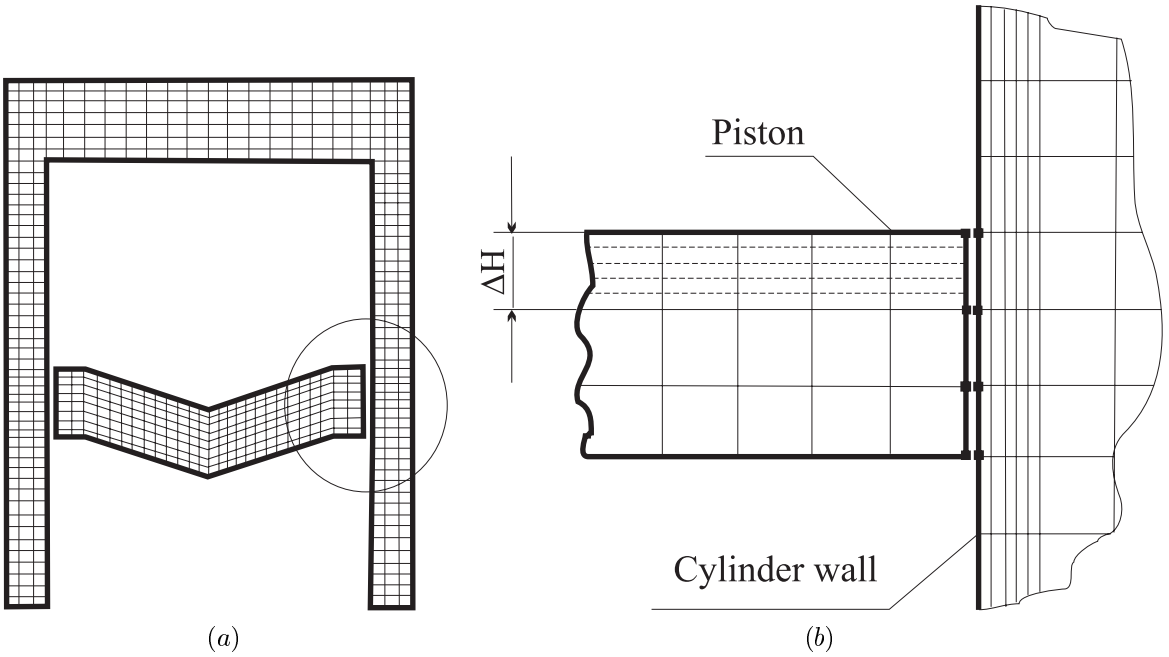



Figure 2: Dependence between the spatial discretization and motion

conditions, and the governing equations. These properties of RFM make it highly suitable for handling time-varying geometry and boundary conditions.

Section 3 describes the application of RFM to the heat conduction problem illustrated in Figure 1. The example is inspired by the investigation of the real engine analysis described in [12], is simplified for illustration purposes, but is validated using the same techniques.

The issues related to automating RFM analysis and simulation for general problems are briefly discussed in section 4. In particular, we explain how RFM can be applied directly to conventional solid models, discuss how to handle time in the solution procedure, and consider other computational issues.

# 2 Problem Formulation with RFM

## 2.1 Boundary value problems with time-varying geometry

Solution of any field problem (which includes most problems of mathematical physics and engineering analysis) depends on the physical law governing the distribution of physical quantities throughout the domain, boundary conditions describing the interaction of the domain and its external environment, and initial conditions which determine the field at some starting point in time.

The physical law is typically formulated using partial differential equations. Its mathematical formulation does not depend on the shape of the domain, but is determined by the physical phenomenon, material properties, and so on. It is common to represent such a physical law using a general operator $A$ as

$$Au = f, \tag{1}$$

where $u(x, y, z, t)$ is a (usually unknown) function describing the distribution of some physical quantity in space and time, and $f$ is some known function. The function $u$ is constrained by boundary conditions which specify the values and/or derivatives of $u$ on boundaries $\partial\Omega$ of a given geometric domain $\Omega$. Different boundary conditions may be prescribed on the different portions of the boundary; the $i$th boundary condition can be written as

$$L_i u_{|\partial\Omega_i} = \varphi_i, \tag{2}$$

where $L_i$ is the corresponding operator and $\varphi_i$ is a given function.

In the most general situation, the physical law $A$, the domain $\Omega$ and its boundaries $\partial\Omega_i$, and the boundary conditions $\varphi_i$ can vary with time, resulting in complex non-stationary problems. The initial conditions for such non-stationary problems may include the initial values of the sought distribution (for parabolic problems) and its time derivative (for hyperbolic problems):

$$u_{|t=0} = u_{start}; \qquad \frac{\partial u}{\partial t}_{|t=0} = \psi_0 \tag{3}$$

Together, the physical law (1), the boundary conditions (2), and the initial conditions (3), constitute the mathematical formulation of a boundary value problem.

Exact solutions are available only for very few and very simple boundary value problems; realistic problems with complex geometric information are usually solved only approximately using one of the many known numerical techniques. This dependence on the approximate numerical methods is even more apparent for most non-stationary and non-linear problems. The difficulty in constructing acceptable approximations to a boundary value problem comes from the requirement that the sought function $u$ has to satisfy both the analytic constraints expressed by the differential or integral equation (1) and the imposed boundary conditions (2).

The latter requirement in particular is partly responsible for the great proliferation of numerical techniques that have been devised to deal with increasingly complex geometries. Classical solution methods, such as Fourier, deal with simple geometries by serendipitous choices of the coordinate systems; conformal mappings extend the solution techniques to transformed simple domains; variational methods require careful choice of the appropriate basis functions. More complex geometric conditions are typically *approximated* by spatially discretizing the geometric domain and its boundary, and then constructing the approximate solution to the boundary value problem by interpolating solutions at discrete spatial locations. This general description applies to finite element, boundary-integral equation, and finite difference methods.

Thus, the spatial grid serves two distinct purposes: to approximate the imposed geometric boundary conditions and to construct an approximate analytic solution. Therefore, the type of finite or boundary elements is determined to a large extent by the shape of the domain, as well as the order of differential equation and type of the boundary conditions. Clearly, the accuracy and the quality of the solution very much depends on the particular spatial discretization used. It is quite understandable that automatic construction of appropriate spatial grids and meshes became a major research topic [1].

## 2.2 RFM for homogeneous Dirichlet problems

The RFM was originally developed in Ukraine by Rvachev and his students[21, 22, 24, 25]. In its early form, RFM was an extension of the approach to solving two-dimensional homogeneous boundary value problems with Dirichlet boundary conditions proposed by Kantorovich [9]. The idea of the method is based on the observation that the solution of a differential equation with boundary conditions

$$u_{|\partial\Omega} = 0 \tag{4}$$

can be represented in the form

$$u = \omega\Phi, \tag{5}$$

where $\omega : R^n \to R$ is a known function that takes on zero values on the boundary of the domain $\partial\Omega$, and is positive in the interior of $\Omega$, and $\Phi$ is some unknown function. For example, Figure 3(a) shows a plot of function $\omega(x, y)$ that is identically zero on the boundary of the shown two-dimensional domain and is positive in the domain's interior. Such a function $\omega$ completely describes all the geometric information for the homogeneous Dirichlet boundary value problem, and in fact any function $u$ of the form (5) will satisfy the boundary conditions (4) *exactly.*

The expression (5) contains no information about the differential equation of the boundary value problem. Rather, it represents the *structure* of any solution to a boundary value problem satisfying the given boundary conditions. The purpose of function $\Phi$ is to satisfy the differential equation of the boundary value problem — exactly or approximately. For any given boundary value problem, determination of the unknown $\Phi$ immediately translates into solution to the boundary value problem. Since we usually cannot expect to determine such $\Phi$ exactly, we can approximate it by a

finite (convergent) linearly-independent series

$$\Phi = \sum_{i=1}^{n} C_i \chi_i, \tag{6}$$

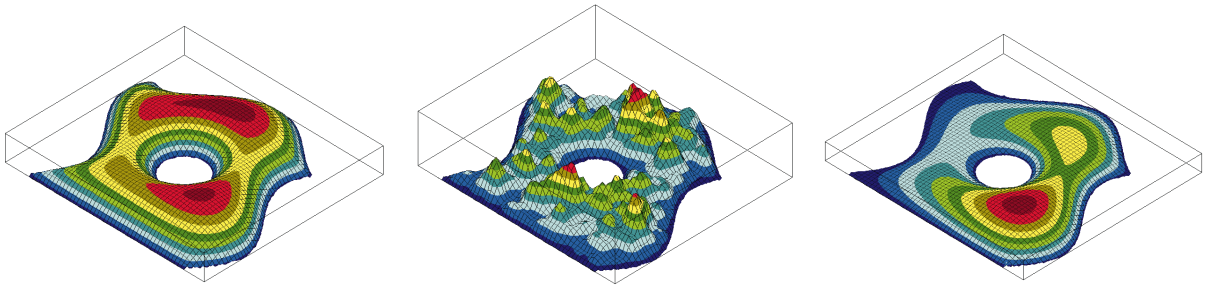
where $C_i$ are scalar coefficients and $\chi_i$ are some basis functions. For example, Figure 3(b) shows combination of the function $\omega$ in Figure 3(a) with a two-dimensional uniform $30 \times 30$ rectangular grid of bicubic B-splines $\chi_i$ and randomly chosen coefficients $C_i$. It is important that the structure (5) does not place any constraints on the choice of the coordinate functions $\{\chi_i\}$ that approximate the function $\Phi$. In particular, the choice of the coordinate functions does not depend on any particular spatial discretization of the domain. The grid of B-splines in our example is aligned with the space and not with the domain.

For any given boundary value problem and a choice of the coordinate basis $\{\chi_i\}$, the approximate solution is obtained as

$$u = \omega \sum_{i=1}^{n} C_i \chi_i, \tag{7}$$

using variational, projection, or a variety of other numerical methods to solve for the numerical values of the coefficients $C_i$. Thus, Figure 3(c) shows the same of combination of B-splines and $\omega$ as in Figure 3(b), but with coefficients chosen to approximate the solution of the differential equation $\nabla^2 u = 1 - sin(y)$ in the least square sense.

From a computational point of view, the intrinsic advantage of the above procedure is in the clean and modular separation of the geometric information represented by function $\omega$ from the differential operator $A$ and the numerical method used to determine the unknown function $\Phi$.



$(a)$ $(b)$ $(c)$

Figure 3: (a) Function $\omega$ satisfying the homogeneous Dirichlet boundary condition; (b) the combination of function $\omega$ with B-splines on a $30 \times 30$ grid with randomly chosen coefficients; (c) the combination of function $\omega$ with B-splines on a $30 \times 30$ grid that approximates the solution of $\nabla^2 u = 1 - sin(y)$

## 2.3 RFM for general boundary value problems

The applicability of RFM to homogeneous boundary value problems stems from the recognition that the solution $u$ can be written in the form of equation (5), and therefore depends on the ability to construct such a function $\omega$ for the specified geometric domain. Fortunately, this construction problem has been solved using the theory of $R$-functions [22, 28, 25, 29]. Briefly, an $R$-function is a real-valued function whose sign is completely determined by the signs of its arguments. For example, the function $xyz$ can be negative only when the number of its negative arguments is odd. Such functions 'encode' Boolean logic functions and are called $R$-**functions**. Every Boolean function is a companion to infinitely many $R$-functions, which form a logical *branch* of the set of $R$-functions. Perhaps the most popular $R$-functions are

$$\begin{aligned} x_1 \wedge_0 x_2 &\equiv \left(x_1 + x_2 - \sqrt{x_1^2 + x_2^2}\right); \\ x_1 \vee_0 x_2 &\equiv \left(x_1 + x_2 + \sqrt{x_1^2 + x_2^2}\right), \end{aligned} \tag{8}$$

corresponding to the logical conjunction $\wedge$ and logical disjunction $\vee$ respectively. These $R$-functions are *analytic* everywhere except when $x_1 = x_2 = 0$. Many other systems of $R$-functions are proposed in [22]. The choice of

an appropriate system of $R$-functions is dictated by many considerations, including simplicity, continuity, differential properties, and computational convenience [31]. Just as Boolean functions, $R$-functions are closed under composition. Using $R$-functions, any object defined by a logical predicate on "primitive" geometric regions (e.g. regions defined by a system of inequalities) can now also be represented by a *single* inequality, or equation.

Equation (5) defines the *structure of the solution* to the boundary value problem with homogeneous Dirichlet boundary conditions and is known to be complete in the sense of convergence to the exact solution [10]. The generalization of RFM to other boundary value problems involves derivation of the corresponding solution structures. Below we briefly summarize the solution structures for common second-order boundary value problems with general Dirichlet and mixed boundary conditions. The solution structures for numerous other boundary value problems with higher order derivatives have been derived, analyzed, and experimentally verified by Rvachev and his pupils [22, 25].

General Dirichlet boundary conditions are commonly written as

$$u_{|\partial\Omega} = \varphi_0, \tag{9}$$

where $\varphi_0$ is the value of the function $u$ prescribed on the boundary $\partial\Omega$. The solution structure is only a slight modification of the homogeneous structure described in section 2.2:

$$u = \omega\Phi + \varphi_0. \tag{10}$$

In practice, the function $\varphi_0$ may be specified in a piecewise fashion, that is, a different value $\varphi_i$ may be prescribed on each portion of the boundary $\partial\Omega_i$. If each such portion is described by an equation $\omega_i = 0$, then the single function $\varphi_0$ can be assembled together using transfinite Lagrangian interpolation:

$$\varphi_0 = \frac{\sum\limits_{i=1}^{m} \varphi_i \prod\limits_{j=1 j\neq i}^{m} \omega_j}{\sum\limits_{i=1}^{m} \prod\limits_{j=1 j\neq i}^{m} \omega_j} \tag{11}$$

The most general kind of boundary conditions for second-order partial differential equations is called *mixed* and subsumes all other types of boundary conditions:

$$u_{|\partial\Omega_1} = \varphi; \qquad \left(\frac{\partial u}{\partial n} + hu\right)_{|\partial\Omega_2} = \psi \tag{12}$$

The corresponding most general solution structure can be written in the form of

$$u = \omega_1\Phi + \frac{\omega_1\omega_2}{\omega_1+\omega_2}\left(\psi + \omega_2\Phi - h\omega_1\Phi - h\varphi - D_1^{\omega_2}\left(\omega_1\Phi\right) - D_1^{\omega_2}\left(\varphi\right)\right) + \varphi, \tag{13}$$

where

$$D_1^{\omega}\left(f\right) = \frac{\partial\omega}{\partial x}\frac{\partial}{\partial x}\left(f\right) + \frac{\partial\omega}{\partial y}\frac{\partial}{\partial y}\left(f\right) \tag{14}$$

is a differential operator in the direction of the internal normal to the boundary $\partial\Omega$. This operator assumes that function $\omega$ is normalized[1], i.e.

$$\frac{\partial\omega_i}{\partial\nu}_{|\partial\Omega_i} = 1; \qquad \frac{\partial^k\omega_i}{\partial\nu^k}_{|\partial\Omega_i} = 0; \qquad k = 2, 3, \ldots \tag{15}$$

Sometimes it is more convenient to view structure (13) as a sum $u = u_1 + u_0$ of homogeneous and inhomogeneous parts:

$$u_1 = \omega_1\Phi + \frac{\omega_1\omega_2}{\omega_1+\omega_2}\left(\omega_2\Phi - h\omega_1\Phi - D_1^{\omega_2}\left(\omega_1\Phi\right)\right); \qquad u_0 = \frac{\omega_1\omega_2}{\omega_1+\omega_2}\left(\psi - h\varphi - D_1^{\omega_2}\left(\varphi\right)\right) + \varphi$$

In its most general form, the structure of the solution (13) is a single function composed from normalized functions $\omega_i$ defining the portions of the domain's boundary, prescribed and possibly interpolated known functions $h$, $\psi$, $\varphi$, and the function $\Phi$ that is determined by its assumed form and the chosen solution procedure.

[1] In other words, it behaves as a distance function near the boundary.

### 2.4 RFM for deforming domains

In abstract terms, the general solution structure can be written in the form

$$u = B[\Phi], \tag{16}$$

where $B = B(\omega, \{\omega_i\}, \{\varphi_i\})$ is an operator that depends on geometry of the domain $\omega \geq 0$ and boundary conditions $\varphi_i$ prescribed on boundary portions defined by $\omega_i = 0$. Remarkably, this operator $B$ does *not* depend on the differential equation; rather it encodes all given geometric information and acts on the coefficients $\{C_i\}$ of the basis functions $\Phi$ to approximate the differential equation. In this sense, the solution structure (16) satisfies the boundary conditions exactly, and is approximated by discretizing the underlying functional space.

This generalization of RFM inherits all computational advantages of the RFM for homogeneous problems that we identified above, including independence from spatial discretization of the domain, and ability to independently choose and modify geometric boundary conditions and basis functions. In particular, suppose $B[\Phi]$ is a constructed solution structure for some $(\omega, \{\omega_i\}, \{\varphi_i\})$, and there is a known transformation

$$(\omega, \{\omega_i\}, \{\varphi_i\}) \mapsto (\omega', \{\omega_i'\}, \{\varphi_i'\}). \tag{17}$$

Then obtaining the solution structure $B'[\Phi]$ for the transformed problems is a matter of a simple syntactic substitution given by equation (17). Notice that the basis $\Phi$ with undetermined coefficients $\{C_i\}$ does *not* need to be modified, although it could be modified if so desired – in both type and cardinality.

We now see that RFM is perfectly suited for modeling and simulation of problems with time-varying geometry and boundary conditions. Once the solution structure $B_0[\Phi]$ is identified for the initial boundary conditions (3) at time $t = 0$, the solution structure $B_t[\Phi]$ at any future time is obtained by formulating the changes in geometry and boundary conditions as functions of time as $(\omega(t), \{\omega_i(t)\}, \{\varphi_i(t)\})$. Provided that the solution procedure is available to solve the quasi-steady problem at time $t = 0$, then the *same* solution procedure can be applied at any future time step. Below, we detail the main steps of RFM for the engine combustion chamber example introduced in section 1 and illustrated in Figure 1.

## 3 Example: Heat Transfer in Engine Combustion Chamber

### 3.1 Assumed Mathematical Model

Nonsteady state heat conduction in the internal combustion chamber (refer to Figure 1) is modeled by a two-dimensional axisymmetric Poisson equation:

$$\lambda \bigtriangledown^2 T = c_\rho \rho \frac{\partial T}{\partial t} \tag{18}$$

where $\lambda$ is thermal conductivity, $c_\rho$ is specific heat, $\rho$ is material density, $T$ is the metal temperature, and $t$ is time. The equation holds in the cylinder walls and in the piston, subject to the mixed boundary conditions:

$$T_{|\partial\Omega_1} = T_{oil}; \qquad \left(-\lambda \frac{\partial T}{\partial n}\right)_{|\partial\Omega_2} = q, \tag{19}$$

where $q$ is the heat flux generated by the combustion process. For illustration purposes, we simplify the more realistic boundary conditions from [12] and assume that $q$ remains constant.

To complete the formulation of the boundary value problem we need to prescribe the initial conditions. Any known distribution of the temperature field can be chosen for that purpose. Without loss of generality, we can assume that the initial temperature distribution occurred sufficiently far in advance and therefore satisfies the steady state version of equation (18):

$$\lambda \bigtriangledown^2 T = 0. \tag{20}$$

### 3.2 Description of the combustion chamber by implicit functions

In order to construct the solution structure for this problem, we first need to describe the geometry of the domain using appropriate implicit functions $\omega_i$. Specifically, in Figure 4 the boundary of the chamber $\Omega$ is divided in two parts: $\partial\Omega_1$

with Dirichlet boundary conditions, and interior boundary of the chamber $\partial\Omega_2$ with Neumann boundary conditions (recall Figure 1). Both boundaries are moving. At any given time instance, the solution structure for this problem is given by Equation (13) above and requires functions $\omega_1$ and $\omega_2$ whose zero sets correspond to points of $\partial\Omega_1$ and $\partial\Omega_2$ respectively. These functions are easily constructed using theory of $R$-functions as follows.

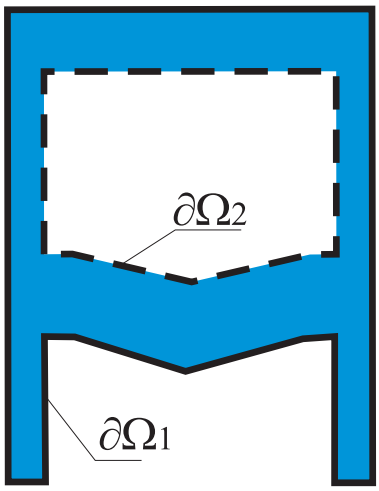



Figure 4: Boundaries $\partial\Omega_1$ and $\partial\Omega_2$ of the combustion chamber correspond to two different types of boundary conditions on the chamber $\Omega$

Figure 5(a) shows a geometric parameterization of the engine combustion chamber used as an example in this section. The details (angles, gaps, etc.) are exaggerated for illustration purposes. Using these parameters, we define a number of simple geometric primitives shown in Figure 5(b):

- Two vertical strips $f_i,\ i = 1, 2$ are defined by :

$$f_i = \frac{R_i^2 - x^2}{2R_i} \geq 0$$

- The horizontal strip:

$$f_3 = \left( \left( \frac{H_2}{2} \right)^2 - \left( y - \frac{H_2}{2} \right)^2 \right) H_2^{-1} \geq 0$$

- Three horizontal linear halfplanes:

$$f_4 = H_1 - y \geq 0; \qquad f_6 = H - y \geq 0; \qquad f_7 = y - (H - t) \geq 0;$$

- and four inclined linear halfplanes $f_8, f_9, f_{10}, f_{11}$:

$$f_8, f_9 = \frac{-R_p y \pm d \cdot x + R_p (H - d)}{\sqrt{R_p^2 + d^2}} \geq 0; \qquad f_{10}, f_{11} = \frac{R_p y \pm d \cdot x - R_p (H - t - d)}{\sqrt{R_p^2 + d^2}} \geq 0$$

Notice that all primitives $f_i$ are normalized in this example.

These primitives can be combined using set operations $\cap$ and $\cup$ to define the two-dimensional regions $\Omega_1$ and $\Omega_2$. In our example, $\Omega_1$ is the region bounded by the exterior walls where the oil temperature $T_{oil}$ is prescribed, and $\Omega_2$ defines the (unbounded) region extending from the interior walls of the combustion chamber with the known heat flux $q$. Syntactically replacing the set operations by the corresponding $R$-functions yields inequalities defining $\Omega_1$ and $\Omega_2$ respectively as:

$$\omega_1 = (f_2 \wedge_0 f_3) \vee_0 (- (f_1 \wedge_0 (- (f_7 \vee_0 (f_{10} \wedge_0 f_{11}))))) \geq 0 \tag{21}$$

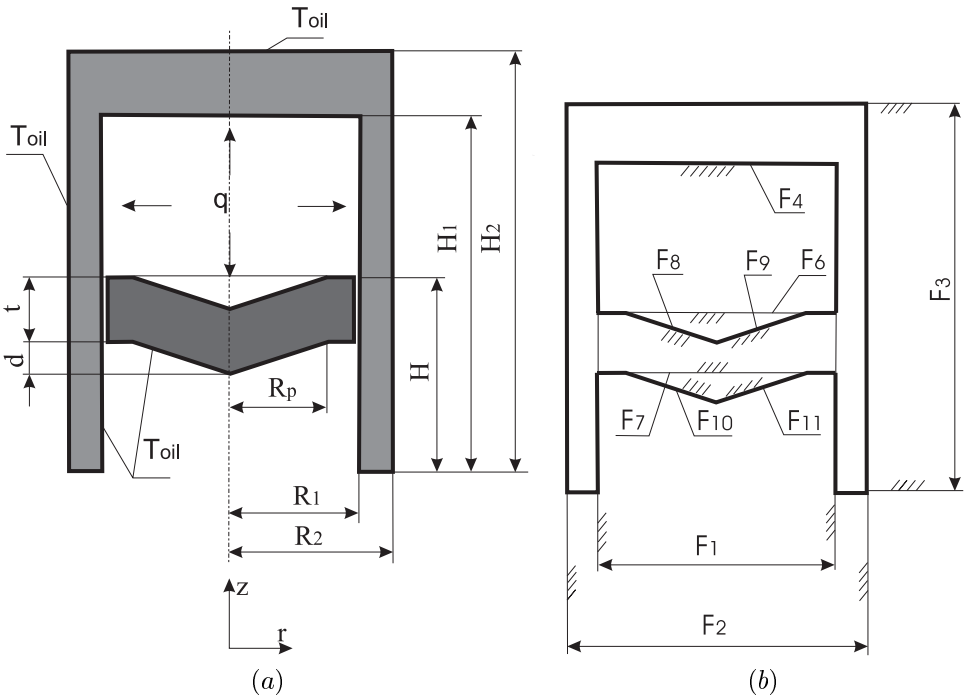



Figure 5: Geometrical primitives used to describe the engine combustion chamber

$$\omega_2 = (-(f_1 \wedge_0 f_4)) \vee_0 (f_6 \wedge_0 (f_8 \vee_0 f_9)) \geq 0, \tag{22}$$

where operations $\wedge_0$ , $\vee_0$ are the $R$-functions defined by (8). Both functions $\omega_1$ and $\omega_2$ are parameterized by $H$ which measures the piston's position at any given time (see Figure 5). Substituting the well known kinematic description of a slider-crank mechanism[33]

$$H(t) = r\cos\Theta(t) + l\left(1 - \frac{r^2}{l^2}\sin^2\Theta(t)\right)^{\frac{1}{2}} \tag{23}$$

into the defining equations yields the required representations of $\omega_1(t), \omega_2(t)$. The functions $\omega_1$ and $\omega_2$ are plotted in Figures 6 for two distinct positions of the piston.

### 3.3 Solution Structure

An appropriate structure for our example problem is obtained by applying expression (13) with our specific boundary conditions:

$$T = \omega_1\Phi + \frac{\omega_1\omega_2}{\omega_1+\omega_2}\left(-\frac{q}{\lambda} + \omega_2\Phi - D_1^{\omega_2}(\omega_1\Phi) - D_1^{\omega_2}(T_{oil})\right) + T_{oil}, \tag{24}$$

where mixed boundary conditions are prescribed on boundaries of the regions defined by implicit functions $\omega_1 = 0$ and $\omega_2 = 0$ respectively.

To complete the solution structure, we also need to choose an approximation for the unknown function $\Phi$ in the form of equation (6). This approximation can be achieved by any system of linearly independent basis functions $\{\chi_i\}$, because their choice does not depend on the geometry of domain or boundary conditions. In our case, we chose to approximate $\Phi$ by a linear combination of bicubic B-splines over a uniform rectangular grid $(50 \times 50)$, as illustrated in Figure 7. The attractive computational properties of B-splines are well known [8].

As we discussed in the previous section, expression (24) can be also viewed as an operator $B$ acting on the unknown function $\Phi$. $B$ is clearly dependent on the geometry of the problem as represented by functions $\omega_1$ and $\omega_2$ and the boundary conditions $q$ and $T_{oil}$. If dependence of these variables on time is known, we immediately obtain the

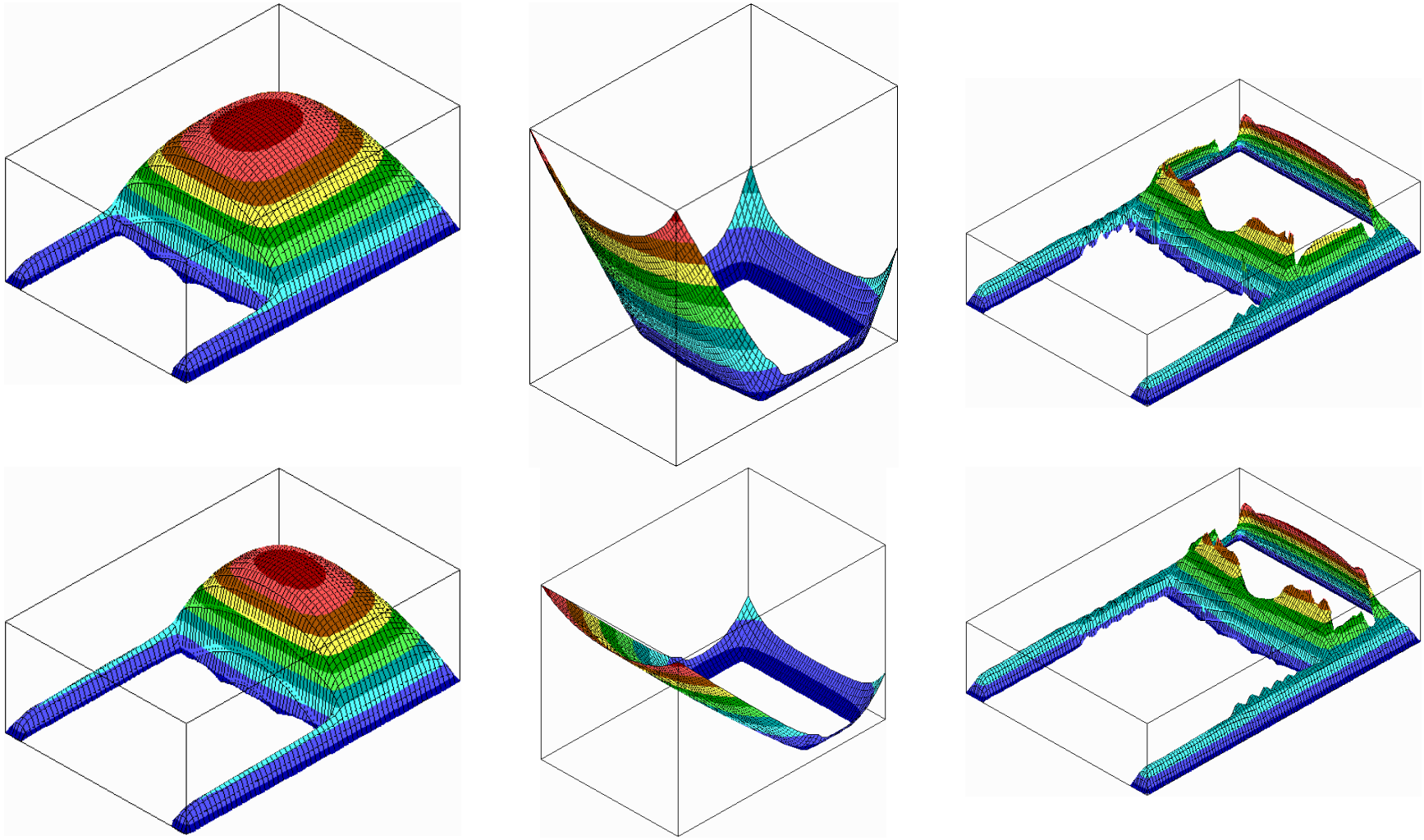


Figure 6: The two rows correspond to the two piston positions. From the left to right: $\omega_1 = 0$ defines the boundaries where $T_{oil}$ is prescribed; $\omega_2 = 0$ models boundaries where the heat flux $q$ is known; computed quasi-steady temperature distribution.

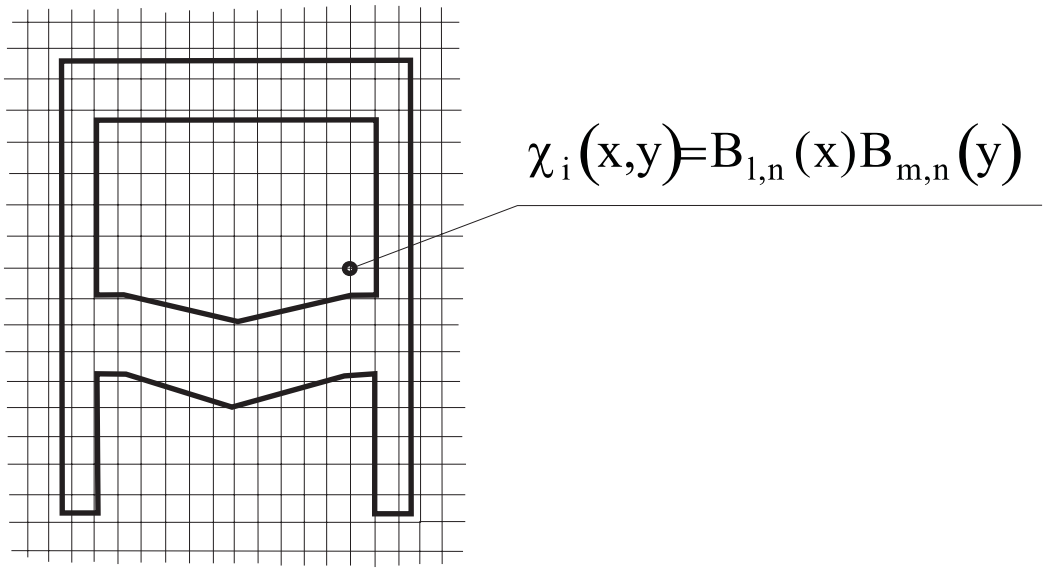



Figure 7: The same rectangular grid of B-splines may be used to compute approximate solutions over any geometric domain

time-dependent solution structure as

$$T\left(t\right)=B_t\left(\omega_1\left(t\right),\omega_2\left(t\right),q(t),T_{oil}(t)\right)\left[\Phi\right]. \tag{25}$$

In our simplified model, $q$ and $T_{oil}$ remain constant, while functions time dependence of $\omega_1(H(t))$ and $\omega_2(H(t))$ is completely determined by the law of motion $(23)$, yielding the time-dependent solution structure $B_t[\Phi]$ corresponding to the expression (24).

## 3.4 Solution procedure

The above solution structure defines the space of functions that satisfy the specified boundary and the initial equation, but not any particular differential equation. A particular function is obtained for every choice of the unknown coefficients $\{C_i\}$, and we want to choose them in such a way as to approximate the exact solution to the given differential equation as closely as possible.

Because equation (18) is a non-steady equation, the coefficients $\{C_i\}$ are themselves functions of time; furthermore, their values at any time instant are dependent on the earlier solutions. To solve the problem, we will first discretize by time, and then we will solve the quasi-steady problem at every time step.

### 3.4.1 Discretization by time

The discrete (in time) version of the differential equation (18) has the following form:

$$\lambda\bigtriangledown^2 T_k=c_\rho\rho\frac{T_k-T_{k-1}}{\Delta t} \tag{26}$$

where $T_k$ and $T_{k-1}$ are temperature distributions on $k$th and $(k-1)$th time step. Applying an implicit finite-difference scheme [16] to equation (26) yields:

$$\lambda\bigtriangledown^2 T_k-c_\rho\rho\frac{T_k}{\Delta t}=-c_\rho\rho\frac{T_{k-1}}{\Delta t} \tag{27}$$

At time $t_k$ equation (27) describes a quasi-steady problem for the unknown temperature field $T_k$. Note that the right side of the equation contains the (previously computed) temperature distribution $T_{k-1}$ at the previous time instant,

but it is applied to the quasi-steady problem at time $t_k$. Since the value of function $T_{k-1}$ on $k$th time step is given by $B_{t_{k-1}}[\Phi_{k-1}]$, solving equation (27) requires knowledge of both solution structures $B_{t_{k-1}}$ and $B_{t_k}$, or at least the transformation from the current structure $B_{t_k}$ to the previous structure $B_{t_{k-1}}$. In our case, this relationship is determined by the change $H(t_k) - H(t_{k-1})$ in value of the motion parameter $H$.

### 3.4.2 Quasi-steady solution

A variety of numerical methods can be used to solve (approximately) equation (27) for the unknown coefficients $C_i$. We will use a variational method with the least square approximation. This requires first reducing the given boundary value problem to a problem with homogeneous boundary conditions as follows. First we separate the sought solution $T_k$ into two parts:

$$T_k = T_k^1 + T_k^0, \tag{28}$$

where

$$T_k^1 = \sum_{i=1}^{n} C_i B\left(\omega_1\left(t_k\right), \omega_2\left(t_k\right), 0, 0\right)[\chi_i] = \sum_{i=1}^{n} C_i \xi_i^k$$

is a linear combination of the basis functions $\{\xi_i\}$ satisfying the homogeneous boundary conditions, and

$$T_k^0 = B\left(\omega_1\left(t_k\right), \omega_2\left(t_k\right), q, T_{oil}\right)[0]$$

satisfies the given nonhomogeneous boundary conditions.

Substituting equation (28) into (27) yields a quasi-steady differential equation in $T_k^1$ with homogeneous boundary conditions:

$$\lambda \bigtriangledown^2 T_k^1 - c_\rho \rho \frac{T_k^1}{\Delta t} = -c_\rho \rho \frac{T_{k-1} - T_k^0}{\Delta t} - \lambda \bigtriangledown^2 T_k^0 \tag{29}$$

Note that the right side of the equation contains only known quantities, including the approximate non-homogeneous solution $T_k^0$ and temperature distribution $T_{k-1}$ on the previous time step. Solving this equation entails computing the coefficients $\{C_i\}$ of the basis $\{\xi_i\}$ that approximates the homogeneous solution $T_k^1$ and therefore the temperature field $T_k$.

To approximate equation (29) using least-square method, we will minimize the functional:

$$F = \int\int_{\Omega} \left(\lambda \bigtriangledown^2 T_k^1 - c_\rho \rho \frac{T_k^1}{\Delta t} - \left(-c_\rho \rho \frac{T_{k-1} - T_k^0}{\Delta t} - \lambda \bigtriangledown^2 T_k^0\right)\right)^2 d\Omega \tag{30}$$

Differentiating functional (30) with respect to the unknown coefficients $C_i$ we obtain a system of linear algebraic equations $[a_{ji}]\,[C_i] = [b_j]$ with the coefficients computed as follows:

$$\begin{aligned} a_{ji} &= \int\int_{\Omega} \left(\lambda \bigtriangledown^2 \xi_i - c_\rho \rho \tfrac{\xi_i}{\Delta t}\right)\left(\lambda \bigtriangledown^2 \xi_j - c_\rho \rho \tfrac{\xi_j}{\Delta t}\right) d\Omega; \\ b_j &= \int\int_{\Omega} \left(-c_\rho \rho \tfrac{T_{k-1} - T_k^0}{\Delta t} - \lambda \bigtriangledown^2 T_k^0\right)\left(\lambda \bigtriangledown^2 \xi_j - c_\rho \rho \tfrac{\xi_j}{\Delta t}\right) d\Omega \end{aligned} \tag{31}$$

### 3.4.3 Stepping through time

Solving the linear system (31) and substituting the value of coefficients $C_i$ into the solution structure (24) yields an approximate solution $T_k$ to the quasi-steady problem at time $t_k$. We begin by solving the quasi-steady equation (20) for the initial temperature distribution $T_0$.

At the next time step $t_{k+1}$, we compute new position of the piston $H$ from equation (23); this automatically updates the definitions of functions $\omega_1$ and $\omega_2$ and the solution structure $B_{t+1}$ at time $t_{k+1}$. At this point we solve the quasi-steady problem again, and repeat the entire process. Figure 8 shows the computed temperature distributions for six different piston positions.

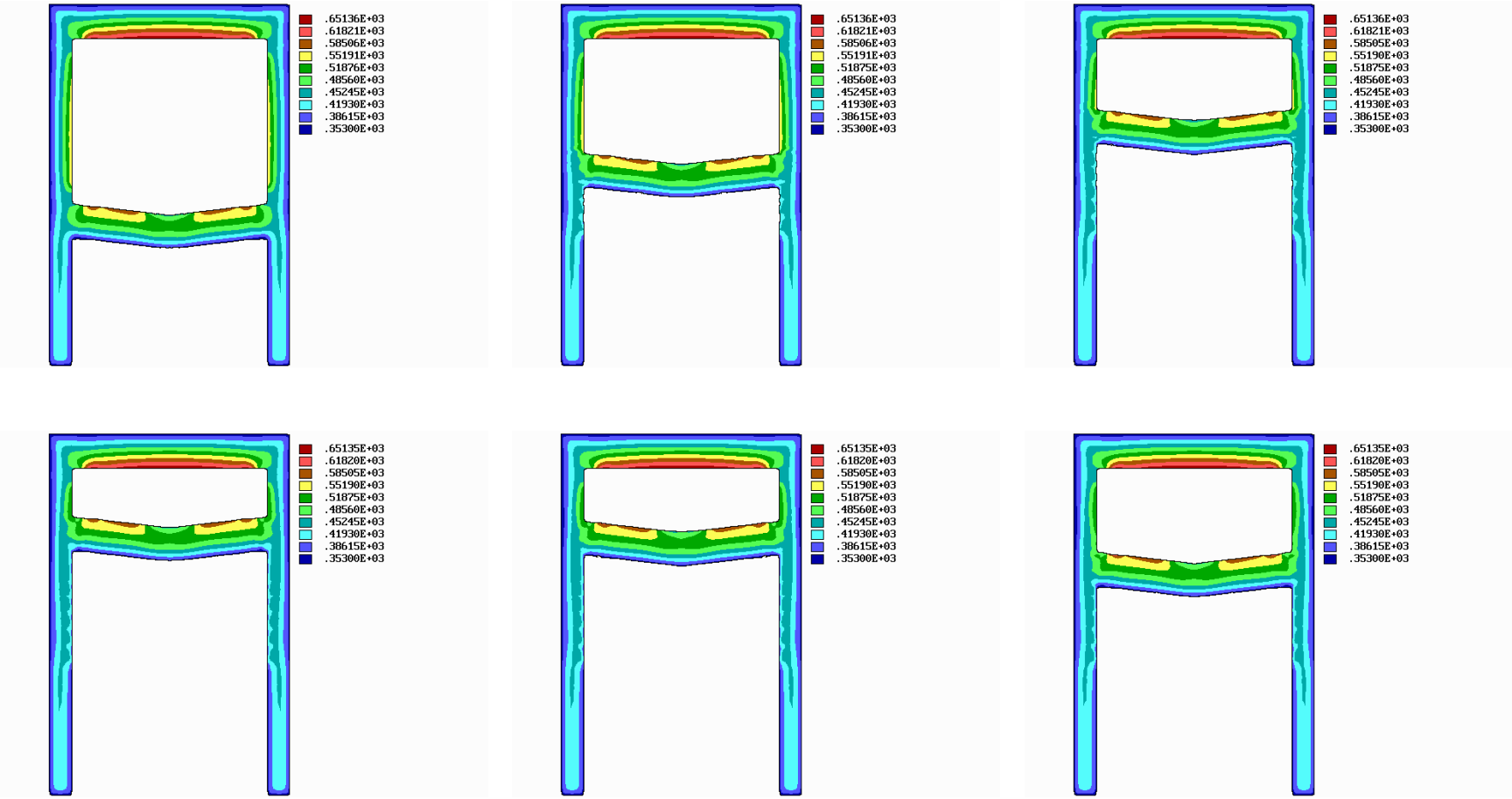



Figure 8: Computed quasi-steady temperature fields for six different positions of the piston

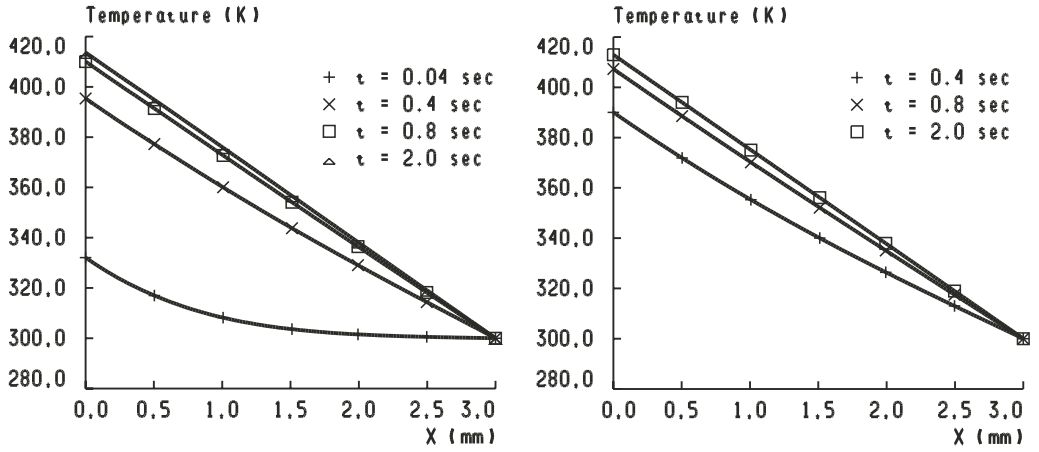



Figure 9: Temperature distributions computed using different time step

### 3.4.4 Validation

The complete solution procedure has been validated on a simple one-dimensional problem that was also used in [13]. Figure 9 shows heat conduction prediction in one-dimensional 3 mm thick slab with thermal conductivity $\lambda = 53.1W \cdot m^{-1}K^{-1}$, density $\rho = 7870kg \cdot m^{-3}$ and specific heat $c_\rho = 447J \cdot kg^{-1}K^{-1}$. The slab is initially at uniform temperature of $T = 300K$ which is maintained at the right end of the slab. A steady heat flux of $q = 2MW \cdot m^{-2}$ is applied at the left side beginning at $t = 0$.

The boundaries of the one-dimensional slab are described by two functions $\omega_1 = 0.003 - x \geq 0$; $\omega_2 = x \geq 0$, that are substituted in the derived solution structure (25). Figures 9(a) and (b) show temperature distribution predicted by the described RFM procedure for two different time steps: 0.01 s in Figure 9(a) and 0.1 s in Figure 9(b). The results are essentially identical to those computed in [13] using one order of magnitude smaller time steps. Another indication of the convergent solution is given in Figure 10 which shows sensitivity of the solution at time $t = 0.8$ s with respect to the number $nx$ of the basis functions in $\Phi$ and the time step $\Delta t$.

# 4 Automation Issues

The above example illustrates the necessary ingredients in applying RFM to problems with time-varying geometry and/or boundary conditions. Successful application of the method requires:

- constructing the functions $\omega, \omega_i$ describing the domain and/or its (partial) boundaries of the domain;
- assembling the solution structure dictated by the given boundary conditions and chosen basis functions;
- modeling motion or deformation of the domain and/or boundary conditions;
- discretizing the boundary value problem by time and solving the corresponding quasi-steady boundary value problem at every time step using RFM.

Each of these tasks can be automated to a large degree and applied in a reasonably general setting. We now discuss some of the technical issues related to these tasks.

## 4.1 Constructing implicit functions

The scope and applicability of RFM is largely determined by the ability to construct implicit functions for geometric domains. Fortunately, the required functions exist for virtually all geometric objects of interest in engineering [29],

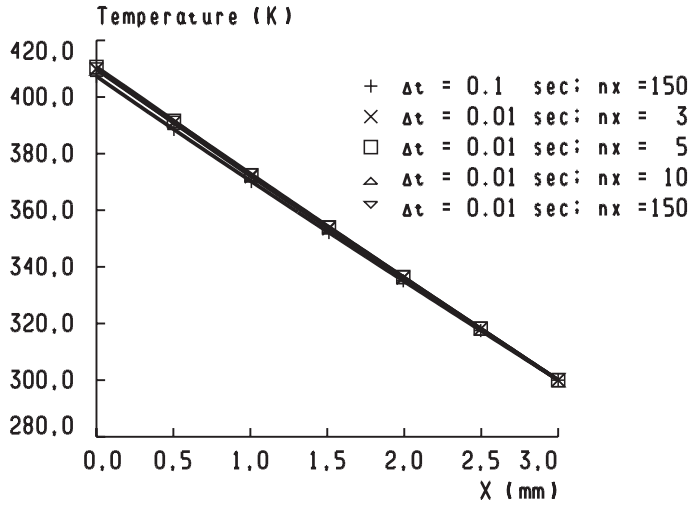



Figure 10: Convergence test — temperature distributions computed at the same time point using different time step and degrees of freedom

and their construction can be completely automated using the theory of $R$-functions [22, 28, 31]. Any geometric object that can be represented by a Boolean expression on some geometric equalities and inequalities can be also described by a *single* real valued function. Most engineering applications deal with semi-analytic subsets of Euclidean space that admit such Boolean representations by definition [29].

Geometric domain and its boundaries are usually modeled by one of many unambiguous geometric representation schemes [19]. Many of the popular representation schemes rely on Boolean operations in one form or another. For example, Constructive Solid Geometry (CSG) representations use regularized set operations on solid primitives or unbounded halfspaces. The implicit functions $\omega_1$ and $\omega_2$ for the engine combustion chamber in Section 3 have been constructed (in this case, manually) in two steps: first a CSG expression was constructed for the desired domain $\Omega_i$ combining the geometric primitive regions $f_j \geq 0$, $j = 1, \ldots, 11$; then each CSG set operation (union, intersection, complement) is replaced by the corresponding $R$-functions (8). Currently, the boundary representations (b-rep) appear to be the predominant method for representing geometric objects in commercial modeling systems. From the set-theoretic point of view, boundary representation consists from one or more piecewise smooth surfaces (curves in two dimensions) constructed as a union of faces; each face is a trimmed subset of a smooth surface. This logical description implies that it is also possible to construct the implicit functions corresponding to the boundary representation directly, without constructing CSG representations.

Thus, both CSG and boundary representations represent complex geometric information using Boolean operations on primitive geometric objects, and are suitable for alternative real-function description using the theory of $R$-functions. Detailed study of the required constructions, resulting implicit functions, and their differential properties can be found in [31]. Completely automatic construction of the corresponding implicit functions is feasible, but involves automatic construction of the required Boolean expressions and may put additional constraints on the form of such expressions to assure differential properties of the constructed functions [30]. Implicit functions can be also constructed directly by sweeping and by numerous other methods used in computer graphics and animation [34, 15]. The reader is referred to [5] for the recently updated survey and bibliography.

## 4.2 Assembling the solution structure

A solution structure in RFM defines the assumed form of the approximate solution to the formulated problem. The general form of solution structures for most boundary value problems are already known [22] and are basically a matter of table look-up, based on the type of given boundary conditions. We note however that the solution structures are generally not unique; additional criteria for choosing a best suited solution structure may involve computational considerations and/or *a priori* knowledge of physical phenomena. Thus, a particular solution structure is further de-

termined by: (1) construction of functions that exactly satisfy the problem's boundary conditions, as discussed above; and (2) selection of the functional basis with desired computational and convergence properties. In the examples of this paper, a uniform grid of splines was chosen as a functional basis. Other bases may also have advantageous numerical, computational, or dimensional properties. RFM has also been used with multivariate polynomials, Chebyshev, and Fourier polynomials[25, 26].

## 4.3 Motions and deformations

The cause of change in geometry and boundary conditions is significant from both physical and computational point of view. If geometric changes are caused by some external conditions that do not depend on the field itself, then the law of geometric changes is known *a priori*. This is the case in our example, where the changes in engine chamber's geometry are effected by the piston motion per equation (23). We modeled the time-varying geometry by parameterizing the constructed functions $\omega_1$ and $\omega_2$ in terms of the time-dependent motion parameter $H(t)$. More generally, suppose we use $R$-functions to construct a function

$$\omega = \omega\left(x', y', z', a'_0, \ldots, a'_k\right)$$

such that $\omega \geq 0$ defines some geometric domain, and this function is naturally parameterized in terms of spatial parameters $x', y', z'$ and some engineering parameters $a'_0, \ldots, a'_k$. Then the deforming or moving geometric domain will be described by some time varying function $\omega(t)$ that is obtained from the static function $\omega$ through an appropriate time-dependent coordinate transformation $M(t)$.

$$\left[x', y', z', a'_0, \ldots, a'_k\right]^T = M(t)\left[x, y, z, a_0, \ldots, a_k\right]^T \tag{32}$$

Similar deformation techniques have been used extensively in modeling and computer graphics [2, 3, 35]. The coordinate transformation $M(t)$ may specify rigid motions, local and/or global deformations, and in general do not have to be invertible.

A more challenging class of deformations involves geometric changes that cannot be prescribed *a priori* because they depend on the properties of the (changing) field. For example, implicit functions have been used to model solidification in metal castings, where the speed of solidification (and therefore the moving boundaries of the solidified metal) is determined by an experimentally verified law that takes into account total casting volume, surface area, and heating and cooling conditions at the boundaries [20]. A large class of deformations may be described by prescribing a speed function at every point of the "evolving front" that may depend on local and global properties of the boundaries, as well as other external conditions [27]. Constructing a time-dependent function $\omega(t)$ for such problems is more difficult because it may require discretizing by time and relying on numerical approximations.

## 4.4 Solution procedure

The time discretization procedure we used in the engine example generalizes in a straightforward fashion. For example, a parabolic differential equation can be written in a form

$$Au = f + \frac{\partial u}{\partial t},$$

where $A$ is a spatial differential operator. Applying the same discretization procedure, we get:

$$Au_k - \frac{u_k}{\Delta t} = f - \frac{u_{k-1}}{\Delta t}; \qquad \Delta t = t_k - t_{k-1}$$

Once again, we can represent the sought solution $u$ as a sum $u = u^1 + u^0$, where $u^1$ satisfies the homogeneous and $u^0$ satisfies the non-homogeneous boundary conditions. In general, this separation into homogeneous and non-homogeneous portions is easy and does not require symbolic computations. The general form corresponding to Equation (29) becomes a quasi-steady equation:

$$Au^1_k - \frac{u^1_k}{\Delta t} = f - Au^0_k + \frac{u^0_k - u_{k-1}}{\Delta t} \tag{33}$$

All terms on the right side of the equation are known functions at the time $t_k$. But note that the solution $u_{k-1}$ is only available at time $t_{k-1}$; since every point $p \in \Omega$ moved according to some deformation $M$, function $u_{k-1}$ must

be evaluated at point $M^{-1}(p)$ at time $t_k$. Evaluating $u_{k-1}$ may be significantly more complicated if $M(t)$ is not invertible.

Equation (33) can be solved using various variational methods, such as the one used in our example. Any numerical procedure used in RFM will require the following steps:

**Differentiation** Because the structure of the solution includes the unevaluated functions describing the geometry and boundary conditions, the differential operator $A$ cannot be applied *a priori*. Instead, automatic differentiation must be used to compute the integrands at points throughout the domain. Efficient numerical (but exact) differential techniques are known [18, 23, 32]. Our example problem required computation of the third derivatives resulting from application of Laplacian operator to the solution structure (24), which already contains the first derivatives.

**Integration** Numerical integration over the domain may be performed in a variety of ways, for example by integrating over the boundary of the domain [14], or using various spatial sampling methods [16, 11]; because the solution structure already satisfies the boundary conditions exactly, this sampling does not lead to the usual difficulties associated with spatial discretizations (such as finite element meshing) and can be computed by almost any commercially available modeling system. When a regular B-spline grid is used to approximate the solution, this amounts to integrating only over those squares that intersect the domain. In this case robustness problems may arise when a square is just "touching" the boundary of the domain. These may be resolved by appropriate perturbation of the grid since it does not have to conform to the domain.

**Solving** The type (dimension, structure) of the resulting linear system depends only on the choice of the basis functions, and not on the geometry of the domain. If $\{\xi\}$ is a sequence of (global) multivariate polynomials, the resulting matrices are full and may quickly become ill-conditioned [22]. For B-splines, it is well known that the matrices are banded and can be solved quickly and robustly [8].

# 5 Conclusion

It is worth noting that modeling of motions and deformations is an active area of research in geometric modeling and computational geometry [3, 17, 6, 35]. Specifying, controlling, and analyzing motions and deformations often lead to challenging problems that are not properly addressed by the existing computational techniques aimed at dealing with static geometry [17]. The dependence of many numerical methods on a particular spatial discretization significantly inhibits development of effective and automatic methods for analyzing and simulating physical behavior of deforming domains.

RFM offers an attractive alternative to classical methods, because it relies on discretization and approximation techniques that do not conform to the particular geometric domain or spatial discretization. As a result, RFM offers a degree of automation that is simply not possible with other methods, including the possibility of meshfree analysis directly from solid models and the ability to quickly and easily modify boundary conditions or even the governing equations. This flexibility comes at a price: automatic differentiation and integration over arbitrary geometric domain introduce significant computational overhead when compared to the traditional methods. On the other hand, the exact enforcement of boundary conditions appears to result in rapid convergence of the solution (recall Figure 10) and requires a smaller number of degrees of freedom. Formal computational properties of RFM require further study and careful analysis.

Deforming or moving problems with partial differential and integral equations are among the hardest; the methods proposed in this paper are generally applicable to a broad class of problems with time-varying geometries and boundary conditions, but the detailed steps may be significantly different. An important class of such problems that may also be suitable for RFM, but is not discussed in this paper, is that of problems with evolving fronts.

# Acknowledgments

This work was supported in part by the National Science Foundation grants DMI-9502728, DMI-9522806 and CCR-9813507. All computations described in this paper (including generation of all images) were performed in system Polye [24, 23, 26]. The authors are grateful to Earlin Lutz and Jovan Zagajac for reading the earlier draft and providing a number of useful suggestions.

# References


[1] *Proceedings of the 6th International Meshing Roundtable*. Sandia National Laboratories, Park City, Utah, October 13-15, 1997.

[2] A. H. Barr. Global and local deformation of solid primitives. *Computer Graphics*, 18(3):21–30, 1984.

[3] D. Bechmann. Space deformation models survey. *Computer and Graphics*, 18(4):571 –586, 1994.

[4] T. Belytschko, Y. Krongauz, D. Organ, M. Fleming, and P. Krysl. Meshless methods: An overview and recent developments. *Comput. Methods Appl. Mech. Engrg.*, 139(1–2):3–47, 1996.

[5] J. Bloomenthal. *Introduction to Implicit Surfaces*. Morgan Kaufmann Publishers, 1997.

[6] S.-W. Cheng and H. Edelsbrunner. Design and analysis of planar shape deformation. In *14th Annual ACM Symposium on Computational Geometry*, Minneapolis, MN, 1998.

[7] J. J. Cox and D. C. Anderson. Single model formulation that link engineering analysis with geometric modeling. In J. Turner, J. Pegna, and M. Wozny, editors, *Product Modeling for Computer-Aided Design and Manufacturing*. Elsevier Science Publishers, North-Holland, 1991.

[8] Carl de Boor. *A Practical Guide to Splines*. Springer-Verlag, 1978.

[9] L. V. Kantorovich and V. I. Krylov. *Approximate Methods of Higher Analysis*. Interscience Publishers, 1958.

[10] I. Yu. Kharrik. On approximation of functions that have zero values and derivatives on domain boundary by special functions. *Siberian mathematical journal*, 4(2):408–425, 1963.

[11] Y. T. Lee and A. A. G. Requicha. Algorithms for computing the volume and other integral properties of solids: Parts I and II. *Communications of the ACM*, 25(9):635–650, 1982.

[12] Y. Liu and R. D. Reitz. Multidimensional modeling of engine combustion chamber surface temperatures. In *Advances in SI and Diesel Engine Modeling (SP-1276), SAE-971594*, pages 57–71, May 1997.

[13] Y. Liu and R. D. Reitz. Modeling of heat conduction within chamber walls for multidimensional internal combustion engine simulation. *Int. J. Heat Mass Transfer*, 41(6-7):859–869, 1998.

[14] E. D. Lutz. *Numerical Methods for Hypersingular and Near-Singular Boundary Integrals in Fracture Mechanics*. PhD thesis, Cornell University, Computer Science Department, May 1991.

[15] A. Pasko, V. Adzhiev, A. Sourin, and V. Savchenko. Function representation in geometric modeling: Concepts, implementation and applications. *The Visual Computer*, 11(8):429–446, 1995.

[16] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C*. Cambridge University Press, second edition, 1992.

[17] S. Raghothama and V. Shapiro. Necessary conditions for boundary representation variance. In *Proceedings of the Thirteenth ACM Symposium on Computational Geometry*, June 4–6 1997.

[18] L. B. Rall and G. F. Corliss. An introduction to automatic differentiation. In M. Berz, C. Bischof, G. Corliss, and A. Griewank, editors, *Computational Differentiation, Procs Second International Workshop on Computational Differentiation*. SIAM, 1996.

[19] A. A. G. Requicha. Representations for rigid solids: Theory, methods, and systems. *ACM Computing Surveys*, 12(4):437–464, December 1980.

[20] V. Rvachev, T. Sheiko, V. Shapiro, and J. Uicker. Implicit function modeling of solidification in metal casting. *Transaction of ASME, Journal of Mechanical Design*, 119:466–473, December 1997.

[21] V. L. Rvachev. *Methods of Logic Algebra in Mathematical Physics*. Naukova Dumka, 1974. In Russian.

[22] V. L. Rvachev. *Theory of $R$-functions and Some Applications*. Naukova Dumka, 1982. In Russian.

[23] V. L. Rvachev and G. P. Manko. *Automation of Programming for Boundary Value Problems*. Naukova Dumka, 1983. In Russian.

[24] V. L. Rvachev, G. P. Manko, and A. N. Shevchenko. The $R$-function approach and software for the analysis of physical and mechanical fields. In J. P. Crestin and J. F. McWaters, editors, *Software for Discrete Manufacturing*, Paris, 1986. North-Holland.

[25] V. L. Rvachev and T. I. Sheiko. $R$-functions in boundary value problems in mechanics. *Applied Mechanics Reviews*, 48(4):151–188, 1996.

[26] V. L. Rvachev and A. N. Shevchenko. *Problem-oriented languages and systems for engineering computations*. Tekhnika, Kiev, 1988. In Russian.

[27] J.A. Sethian. *Level Set Methods: Evolving Interfaces in Geometry, Fluid Mechanics, Computer Vision and Material Sciences*. Cambridge University Press, 1996.

[28] V. Shapiro. Theory of $R$-functions and applications: A primer. Tech. Report TR91-1219, Computer Science Department, Cornell University, Ithaca, NY, 1991.

[29] V. Shapiro. Real functions for representation of rigid solids. *Computer-Aided Geometric Design*, 11(2):153–175, 1994.

[30] V. Shapiro. Well-formed set representations of solids. *International Journal on Computational Geometry and Applications*, 1997. accepted for publication.

[31] V. Shapiro and I. Tsukanov. Implicit functions with guaranteed differential properties. In *Solid Modeling '99, Proc. Fifth Symposium on Solid Modeling and Applications*, June 1999.

[32] A.N. Shevchenko and V.N. Rokityanskaya. Automatic differentiation of functions of many variables. *Cybernetics and System Analysis*, 32(5):709–723, 1996.

[33] J. E. Shigley and Jr. J. J. Uicker. *Theory of Machines and Mechanisms*. McGraw-Hill, Inc., second edition, 1995.

[34] A. Sourin and A. Pasko. Function representation for sweeping by a moving solid. In *Proc. Third Symposium on Solid Modeling Foundations and CAD/CAM applications*, May 17-19 1995.

[35] Savchenko V. and Pasko A. Transformation of functionally defined shapes by extended space mappings. *The Visual Computer*, 14(5-6):257–270, 1998.

[36] T. C. Wilson, J. A. Talbert, and J. J. Cox. Modeling primitives: an object oriented formulation of boundary value problems in a solid geometric modeling context. In J. Rossignac, J. Turner, and G. Allen, editors, *Second Symposium on Solid Modeling and Applications*, pages 441–448, Montreal, Canada, May 19-21, 1993.

[37] J. Zagajac. A fast method for estimating discrete field values in early engineering design. *IEEE Transactions on Visualization and Computer Graphics*, 2(1):35–43, 1996.