A collocation method for parabolic quasilinear problems on general domains (1)

U. Ascher (2) and J. B. Rosen (3)

ABSTRACT

A collocation method is described which obtains approximate solutions to quasilinear parabolic problems on a general two-dimensional domain. The method is best suited for obtaining robust solutions to smooth problems with the accuracy required in most engineering applications. The solution is obtained in terms of a finite element, B-spline basis. An interactive computer graphics system is used for both problem formulation and the subsequent display of selected results. The theoretical basis for the method is discussed, and some typical computational results are presented.

1. INTRODUCTION

A collocation method is described for efficiently obtaining approximate solutions to quasilinear parabolic problems. The method is most suitable when solutions to smooth problems on two-dimensional domains are needed with an accuracy sufficient for most practical applications. An interactive graphics system is used for both problem formulation and display of results.
Let $\Omega \subset E_n$ be an open bounded domain, $n \geqslant 1$, with $\partial\Omega$ its boundary. Consider the initial-boundary value parabolic problem

$$\begin{aligned} u_t &= a(x,u)\Delta u + b(t,x,u,u_x) && (t,x) \in (0,.T) \times \Omega, \\ u(t,x) &= g(t,x) && (t,x) \in (0,T) \times \partial\Omega, \quad (1.1) \\ u(0,x) &= u_0(x) && x \in \bar{\Omega} = \Omega \cup \partial\Omega, \end{aligned}$$

where $x = (x_1, ..., x_n)$, $u_x = (u_{x_1}, ..., u_{x_n})$,

$\Delta u = \sum_{j=1}^{n} u_{x_j x_j}$, $T > 0$, $a(x,u)$ is a positive scalar function, $g(t,x)$ and $u_0(x)$ are given data functions.
In order to solve (1.1) approximately, let
$S = \text{span}\ \{\phi_1, ..., \phi_m\}$ be a finite element approximation space on $\bar{\Omega}$. One is then looking for a map
$v:[0,T] \to S$,
i.e., an approximate solution of the form

$$v(\alpha;t,x) = \sum_{i=1}^{m} \alpha_i(t)\phi_i(x). \qquad (1.2)$$

The method is described as follows.
(a) The initial-value function u_0 is first approximated in the discrete ℓ_1 norm using linear programming. This determines the initial approximation

$$v(\alpha;0,x) = \sum_{i=1}^{m} \alpha_i(0)\phi_i(x) \qquad (1.3)$$

and a set of collocation points, both in the interior and on the boundary of the domain, at which the approximate solution exactly satisfies the initial data. The accuracy of this initial approximation gives information prior to solving the problem, on the adequacy of the number m of basis functions ϕ_i being used.

(b) The time-dependent solution is then carried forward by satisfying the differential equation on the interior collocation points and the boundary data on the boundary collocation points. This leads to a set of initial-value ordinary differential equations for $\alpha_i(t)$, $i = 1,...,m$, which are solved approximately via a discretization in the time variable t (e.g., using a predictor-corrector scheme). This is, of course, a version of the so-called line method [20].
The functions ϕ_i are conveniently chosen so that they form a "smooth" B-spline product basis. This choice leads, in fact, to a finite element method. Let h be the maximal diameter of the elements induced by the space S, and r the degree of the space. Then, provided that the choice of the collocation points is stable, the method is convergent and the error is $O(h^{r-1})$ (for the time-continuous approximation).

(1) This research was supported by the National Science Foundation Grant No. GJ32552, and the University of Minnesota Computing Center.

(2) U. Ascher, now at Computer Science Department, University of British Columbia, Vancouver, Canada.

(3) J. B. Rosen, Computer Science Dept., University of Minnesota, Minneapolis, Minn 55455, USA.

This is to be compared with the more favorable error of $0(h^{r+1})$ often achievable with Galerkin or least squares procedures, using the same approximation space. On the other hand, collocation methods are more economical and straightforward to use, since they require no integrations (cf. [7], [16], [11]). The method suggested here is most efficient, with respect to other methods, when very accurate solutions are not required, or when the problem to be solved involves a complicated nonlinear term $b(t,x,u,u_x)$. This is true because the number of function evaluations required is relatively small. Also, the method handles nonzero boundary data $g(t,x)$ with no need for further modification. (Such a modification is required in the case of a Galerkin procedure [4], [6]). It should also be emphasized that the method can handle a rather general spatial domain Ω with no difficulty.

In section 2 we consider the ℓ_1-approximation procedure which provides us with the initial approximation and the collocation points. The procedure assures, under normal circumstances, the local existence of the collocation approximation.

It should be remarked that in general many sets of collocation points exist which would yield an approximation converging with the same order $0(h^{r-1})$ as achieved by our selection of points. The ℓ_1-approximation here is just a device for producing a "good" set of collocation points in an automatic manner for an arbitrarily shaped domain. We try to provide, in the remainder of section 2, the basis for the conjecture that the collocation points produced by the ℓ_1-approximation process are "good".

In section 3 we present the collocation method and prove its convergence, assuming that a stable set of points has been selected. Selection of a stable set of points depends primarily on the choice of weights in the initial approximation, and we discuss this choice of weights in section 4.

Our method, for two spatial variables, has been implemented in an interactive computer graphics system [17], [3], [15]. This contributes significantly to the flexibility and ease of use of the method. In section 4 we briefly describe the system and its use in order to solve, in an interactive environment, boundary value problems on general domains.

A substantial amount of computational testing was carried out using the method described and the interactive graphics system developed for this purpose. Some typical numerical results obtained are summarized in section 4. A detailed presentation of numerical results is given in [1]. Based on these numerical experiments it is concluded that the computational method presented here is both practical and efficient for parabolic problems on a general two-dimensional domain.

2. THE INITIAL APPROXIMATION

We first describe how the approximation to the initial data is obtained. A similar approach has been used for solving linear boundary value problems [14].

Let Ω_ν and $\partial\Omega_\nu$ be two discrete grids of ν_I and ν_B points, respectively :

$$\Omega_\nu = \{x^1,\dots,x^{\nu_I}\} \subset \Omega \text{ and } \partial\Omega_\nu = \{x^{\nu_I+1},\dots,x^N\} \subset \partial\Omega,$$

where $N = \nu_I + \nu_B$, $N > m$. We call Ω_ν and $\partial\Omega_\nu$ the *minimization* grids on the interior and boundary, and finally let ω_0 and ω_1 be two weights for the interior and boundary, respectively.

For a function $f \in L_\infty(\bar{\Omega})$ we determine $\alpha^* = (\alpha_1^*,\dots\alpha_m^*)$ such that

$$v(\alpha^*; x) = \sum_{i=1}^{m} \alpha_i^* \phi_i(x)$$

is a best (discrete) ℓ_1-approximation to f on the minimization grids; i.e., α^* is determined as the solution to the minimization problem

$$\min_\alpha \{\omega_0 \|v(\alpha;\cdot) - f(\cdot)\|_{\Omega_\nu} + \omega_1 \| v(\alpha;\cdot) - f(\cdot)\|_{\partial\Omega_\nu}\} \tag{2.1}$$

where

$$\|\cdot\|_{\Omega_\nu} = \frac{1}{\nu_I} \sum_{\Omega_\nu} |\cdot| \, , \; \|\cdot\|_{\partial\Omega_\nu} = \frac{1}{\nu_B} \sum_{\partial\Omega_\nu} |\cdot| \tag{2.2}$$

A number of efficient linear programming algorithms [5], [13] have been suggested to solve (2.1). In the dual method, the required α^* is obtained as the dual solution vector to the following linear programming problem with bounded variables [14].

$$\max_\eta \{c\cdot\eta | A\eta = 0, \; -w_j \leqslant \eta_j \leqslant w_j, \; 1 \leqslant j \leqslant N\} \tag{2.3}$$

where

$$\eta = (\eta_1,\dots,\eta_N), \qquad A = \begin{bmatrix} \phi_1(x^1) & \dots & \phi_1(x^N) \\ \vdots & & \vdots \\ \phi_m(x^1) & & \phi_m(x^N) \end{bmatrix}.$$
$$c = (f(x^1),\dots,f(x^N)),$$
$$w_j = \begin{cases} \omega_0/\nu_I, & 1 \leqslant j \leqslant \nu_I, \\ \omega_1/\nu_B, & \nu_I < j \leqslant N, \end{cases} \tag{2.4}$$

For a reasonable choice of basis functions ϕ_i, $i=1,\dots,m$ and minimization grids Ω_ν and $\partial\Omega_\nu$, one has rank(A)= m. We shall henceforth assume this to be the case.

Using an appropriate version of the Simplex method for the solution, an optimal basis Φ of m columns of A is produced. Clearly a finite solution exists since $\eta = 0$ is feasible, and the objective function is bounded by $\omega_0\|f\|_{\Omega_\nu} + \omega_1\|f\|_{\partial\Omega_\nu}$. Let $\phi(x^j)$ denote the jth column of A. The m × m non-singular matrix Φ then consists of the columns $\phi(x^j)$ for $j \in J$, where J denotes the set of m selected columns.

It follows from the duality theory of linear programming that we have

$$v(\alpha^*;x^j) = f(x^j), \quad j \in J. \tag{2.5}$$

Thus the discrete ℓ_1-approximation process provides a set of points

$$\{x^j \mid j \in J\} \equiv \{\xi_1,\dots,\xi_m\}$$

which solve the following interpolation problem : Among all subsets of m points from $\Omega_\nu \cup \partial\Omega_\nu$, find one of these subsets such that the corresponding interpolant minimizes the bracketed quantity of (2.1).
At this stage we also have useful information on the adequacy of the basis functions ϕ_i, i=1,...,m, being used. If the error given by (2.1) in approximating the initial data is too large, then we must either increase m, or choose a better set of basis functions.
The interpolatory property exhibited for the discrete ℓ_1-approximation, appears also in best continuous (integral) L_p-approximations, $1 \leqslant P \leqslant \infty$, on an interval $I = [0,1]$ say, to continuous functions f. In particular, in the polynomial case

$$\phi_i(x) = x^{i-1} \quad i=1,\dots,m$$

with $f(x) = x^m$, the interpolating points are zeroes of the Chebyshev polynomial of the first kind, of the Legendre polynomial, or of the Chebyshev polynomial of the second kind, for the L_∞, L_2 or L_1 norm, respectively [12].
The points $\{\xi_1,\dots,\xi_m\} \equiv \{x^j \mid j \in J\}$ obtained above, are "best" interpolation points for f in the "discrete ℓ_1" sense. We wish to know if interpolation at these same points will still give a good approximation to other functions f.
Conditions were given in [2] under which interpolation at $\{\xi_1,\dots,\xi_m\}$ still produces a best ℓ_1-approximation to other functions. In practice, using tensor products of one-dimensional B-splines, it has been our experience that the interpolation points did not vary much from one discrete ℓ_1-approximation to another, even when the conditions for invariance given in [2] were violated (cf. [16]). More important, the interpolation points $\xi_1,\dots,\xi_m$, determined with a function f were also "good" (although not "best") for other functions tested. That is, for another function μ the error using $\xi_1,\dots,\xi_m$, was of the same order of magnitude as the error obtained for the best ℓ_1-approximation to μ.
The interpolation matrix

$$\Phi = [\phi_i(\xi_j)] \tag{2.6}$$

is the optimal basis matrix of the linear programming solution, and as such is invertible.

Let $v(\alpha;\cdot) = \sum_{i=1}^{m} \alpha_i\phi_i(\cdot) \in S$ and $\mu(\cdot) \in L_\infty(\bar\Omega)$. We can write, with $\tilde\alpha$ determined so that $v(\tilde\alpha;\cdot)$ interpolates $\mu(\cdot)$ on $\{\xi_1,\dots,\xi_m\}$,

$$v(\tilde\alpha;\xi_j) - v(\alpha;\xi_j) = \mu(\xi_j) - v(\alpha;\xi_j) \equiv e(\xi_j) \quad 1 \leqslant j \leqslant m.$$

Defining

$$\beta_i = \tilde\alpha_i - \alpha_i, \quad i = 1,\dots,m,$$

$$\beta = (\beta_1,\dots,\beta_m),\ e = [e(\xi_1),\dots,e(\xi_m)],$$

we have

$$\beta\Phi = e,$$

and

$$\beta = e\Phi^{-1}.$$

Thus,

$$\|D^k v(\tilde\alpha;\cdot) - D^k v(\alpha;\cdot)\|_{L_\infty(\bar\Omega)} = \|\sum_{i=1}^{m} \beta_i D^k\phi_i(\cdot)\|_{L_\infty(\bar\Omega)}$$

$$\leqslant \|e\|_\infty \|\Phi^{-1}\|_\infty \sum_{i=1}^{m} \|D^k\phi_i(\cdot)\|_{L_\infty(\bar\Omega)},$$

where

$$D^k = \frac{\partial^{|k|}}{\partial x_1^{k_1}\dots\partial x_n^{k_n}}, \quad |k| = k_1 + \dots + k_n. \tag{2.7}$$

So,

$$\|D^k v(\tilde\alpha;\cdot) - D^k\mu(\cdot)\|_{L_\infty(\bar\Omega)} \leqslant \|e\|_\infty \|\Phi^{-1}\|_\infty \sum_{i=1}^{m} \|D^k\phi_i(\cdot)\|_{L_\infty(\bar\Omega)}$$

$$+ \|D^k v(\alpha;\cdot) - D^k\mu(\cdot)\|_{L_\infty(\bar\Omega)} \tag{2.8}$$

The vector α is arbitrary in (2.8) and so we can choose it such that $v(\alpha;\cdot)$ is a best L_∞ approximation, say, to $\mu(\cdot)$ out of S. Inequality (2.8) then says that $v(\tilde\alpha;\cdot)$ approximates $\mu(\cdot)$ to the order of the best approximation possible out of S, if $\|\Phi^{-1}\|_\infty$ is bounded.
Consider now a spline-product space $S = S_r^h$ on $U = \mathop{X}_{i=1}^{n} [0,1]$, of degree r and uniform knot size h_i in each direction x_i, with $h = \max_{1\leqslant i\leqslant n} h_i$. Assume, without loss of generality, that $\bar\Omega \subset U$. Then the following theorem easily follows from the results of Schultz [18] and (2.8).

Theorem 2.1
Assume that Φ^{-1} is bounded independently of h :

$$\|\Phi^{-1}\|_\infty \leqslant M, \tag{2.9}$$

and let μ have all derivatives up to order r+1. Then there exists a constant K such that

$$\|D^k\mu(\cdot) - D^k v(\tilde\alpha;\cdot)\|_{L_\infty(\bar\Omega)} \leqslant K h^{r+1-|k|} \max_{\substack{|j|=r+1 \\ x\in\bar\Omega}} |D^j\mu(x)|, \quad 0 \leqslant |k| \leqslant r. \tag{2.10}$$

3. THE COLLOCATION SCHEME

Given the problem (1.1), we first obtain a set of collocation points $\{\xi_i\}_{i=1}^m$ and an ℓ_1-approximation

$$v(\alpha;0,x) = \sum_{i=1}^{m} \alpha_i(0)\phi_i(x) \tag{3.1}$$

to the initial data $f=u_0$, as described in section 2. If the collocation points are known in advance, then the above computation is not needed.
Of the collocation points, let $\xi_1,\dots,\xi_d \in \Omega$ and $\xi_{d+1},\dots,\xi_m \in \partial\Omega$, $0 < d < m$.
Define

$$P = \begin{bmatrix} \phi_1(\xi_{d+1}) & \dots & \phi_m(\xi_{d+1}) \\ \vdots & & \vdots \\ \phi_1(\xi_m) & & \phi_m(\xi_m) \end{bmatrix}, \quad Q = \begin{bmatrix} \phi_1(\xi_1) & \dots & \phi_m(\xi_1) \\ \vdots & & \vdots \\ \phi_1(\xi_d) & \dots & \phi_m(\xi_d) \end{bmatrix} \tag{3.2a}$$

i.e., $[Q^T:P^T] = \Phi$.

To simplify the presentation we assume that the coefficient $a(x,u)$ in (1.1) depends on x only, that is, $a = a(x)$. The more general case can be handled with some modification and is discussed in [1]. Then let

$$W = \text{diag }\{a(\xi_i)\}_{i=1,\dots,d};\ \bar{g}(t) = \{g(t,\xi_{d+1}),\dots,g(t,\xi_m)\}. \tag{3.2b}$$

Furthermore, define the vector $\bar{b} = \bar{b}(\alpha;t)$ with ith element given by

$$\bar{b}_i(\alpha;t) = b[t,\xi_i,v(\alpha;t,\xi_i),\ v_x(\alpha;t,\xi_i)],\ i=1,\dots,d, \tag{3.2c}$$

where v is defined by (1.2). Finally, define the $(d\times m)$ matrix R by applying the operator Δ to Q, i.e.,

$$(R)_{i,j} = \Delta\phi_j(\xi_i) \quad i=1,\dots,d;\ j = 1,\dots,m. \tag{3.2d}$$

Then the boundary collocation equations are

$$P\alpha(t) = \bar{g}(t) \quad \text{(m-d equations)}, \tag{3.3}$$

while the interior collocation equations are

$$Q\frac{d\alpha}{dt} = WR\alpha(t) + \bar{b}[\alpha(t);t] \quad \text{(d equations)}. \tag{3.4}$$

The initial values $\alpha(0)$ for (3.4) are given by the initial approximation (3.1).
The set of initial-value ordinary differential equations (3.3) - (3.4) is now solved approximately for $\alpha(t)$, using one of the well-known finite difference techniques [10], possibly with an error-controlled variable step size.
By differentiating (3.3) we can combine equations (3.3) - (3.4) to form a set of m equations

$$\Phi^T\frac{d\alpha}{dt} = G\alpha(t) + \sigma[\alpha(t)] + q(t); \quad \alpha(0) \text{ given}, \tag{3.5}$$

where Φ is defined in (2.6) and G,σ,q are defined in an obvious manner. Since Φ is invertible, a local existence of the approximate solution follows. The global existence will follow upon proof of convergence. This depends on a stability assumption, which we state next. A discussion of stability in the implementation of the computational method is given in section 4.

Stability assumption

The collocation procedure is said to be stable if
(a) The bound (2.9) holds.
(b) The matrix $(\Phi^T)^{-1}$ G has eigenvalues with real parts which are bounded from above independent of h, for each t.
The convergence theorem now follows.

Theorem 3.1

Let the functions $a(x,u)$ and $b(t,x,u,u_x)$ in (1.1) have bounded derivatives $a_u,b_u,b_{\hat{u}_x}$. Let u possess r+1 derivatives in the spatial variables, $r \geqslant 2$, and at least two in the time variable. Assume that the procedure is stable for our choice of collocation points and approximation space S_r^h. Then, for each $t \in [0,T]$,

$$\|u(t,\cdot) - v(\alpha;t,\cdot)\|_{L_\infty(\bar{\Omega})} = 0(h^{r-1}), \tag{3.6}$$

where α is determined by (3.3) - (3.4).

Proof

We estimate u - v through the function

$$\hat{u}(\hat{\alpha};t,x) = \sum_{i=1}^{m} \hat{\alpha}_i(t)\phi_i(x) \in S_r^h \tag{3.7}$$

which, at each time t, interpolates u on $\{\xi_1,\dots,\xi_m\}$.
Let

$$\eta = u - \hat{u}, \quad \rho = \hat{u} - v. \tag{3.8}$$

Then $\rho\in S_r^h$:

$$\rho(t,x) = \sum_{i=1}^{m} [\hat{\alpha}_i(t) - \alpha_i(t)]\,\phi_i(x) \equiv \sum_{i=1}^{m} \beta_i(t)\phi_i(x). \tag{3.9}$$

We may write

$$|u - v| \leqslant |\eta| + |\rho|.$$

The error of interpolation η and its derivatives up to order of at least $s = 2$ are bounded by (2.10), since theorem (2.1) is applicable. Thus we are left to treat ρ.
To simplify the proof we consider the case where b in (1.1) depends on t,x and u only, that is, $b = b(t,x,u)$. The proof is similar when b depends on u_x also. For any t, and at each $\xi_i \in \Omega$ we have

$$v_t = a(v)\,\Delta v + b(v),$$

and the same holds, of course, for u. Also, since $\hat{u}$ is the interpolant, we have $\hat{u} = u$, or $\eta = \eta_t = 0$, at each ξ_i, $i = 1,\dots, m$. Therefore, at $\xi_i \in \Omega$ we have

$$\rho_t = \rho_t + \eta_t = u_t - v_t = a(u)\Delta u + b(u) - a(v)\Delta v - b(v),$$

or

$$\rho_t = a(u)\Delta u - a(u)\Delta\hat{u} + a(u)\Delta\hat{u} - a(u)\Delta v + a(\hat{u})\Delta v - a(v)\Delta v + b(\hat{u}) - b(v).$$

Making use of the mean value theorem, we obtain

$$\rho_t = a(u)\Delta\eta + a(u)\Delta\rho + \Delta v\bar{a}_u\rho + \bar{b}_u\rho,$$

where $\bar{a}_u$ and $\bar{b}_u$ represent partial derivatives evaluated at intermediate values.

We now rearrange the terms, substitute the estimates from (2.10) for $\Delta\eta$ and, using the assumed boundedness of all the appearing quantities, arrive at an equation of the form

$$\rho_t - a_i \Delta\rho - p_i \rho = d_i ; \quad \rho(0) = 0, \qquad (3.10)$$

with $a_i = a(\xi_i, u)$, p_i bounded quantities and $d_i = 0(h^{r-1})$, for each $\xi_i \in \Omega$.

Also, trivially for the boundary collocation points,

$$\rho_t = \rho = 0 \quad \text{at } \xi_i \in \partial\Omega. \qquad (3.11)$$

By (3.7) we can combine equations (3.10) - (3.11) to form a set of m equations, similar to (3.5) :

$$\Phi^T \frac{d\beta}{dt} = G\beta(t) + \hat{P}\Phi^T \beta + \hat{d}; \quad \beta(0) = 0, \qquad (3.12)$$

where

$$\hat{P}_{ij} = \begin{cases} p_i, \ 1 \leqslant i = j \leqslant d \\ 0 \ , \text{otherwise}, \\ \quad (1 \leqslant i, j \leqslant m) \end{cases} \qquad \hat{d}_i = \begin{cases} d_i, \ 1 \leqslant i \leqslant d \\ 0 \ , d < i \leqslant m. \end{cases}$$

The matrix $\hat{G} \equiv G + \hat{p}\Phi^T$ is a perturbation matrix of G, with the perturbation term coming from the lower order terms in (1.1). Thus, from the stability assumption, for h small enough the matrix $(\Phi^T)^{-1} \hat{G}$ has eigenvalues with real parts which are bounded from above independently of h. Also, by (2.9),

$$\|(\Phi^T)^{-1}\hat{d}\|_\infty = 0(h^{r-1}).$$

By elementary stability analysis of the initial value problem (3.12), one now obtains

$$\|\beta(t)\|_\infty = 0(h^{r-1}), \quad 0 \leqslant t \leqslant T.$$

From here, trivially,

$$\|\rho(\beta;t,\cdot)\|_{L_\infty(\bar{\Omega})} = 0(h^{r-1}), \quad 0 \leqslant t \leqslant T,$$

and the theorem follows.

Q.E.D.

4. IMPLEMENTATION AND COMPUTATIONAL RESULTS

The method described above was implemented, for two spatial variables (n=2), using an interactive graphics system. This system is composed of a Digigraphics terminal with the CDC 1700 as a host computer, linked to a CDC Cyber 74 computer. It is capable of handling linear boundary value problems on general domains, according to the method described in [14] (including ℓ_1-approximation as a special case), quasilinear elliptic [1] and quasilinear parabolic problems, according to the method described here. A detailed description of the system is given in [17], [3].

When using the system, the user specifies the domain Ω of the problem, which is contained in the unit square, by drawing its boundary on the graphics screen using the light pen, or by specifying the vertices of a polygon. The boundary curve obtained by either method may be approximated by spline functions in parametric form, and then be replaced by the resulting smooth curve if the user so desires. A few examples of domains which have been used are shown in fig. 1.

The domain Ω and its boundary $\partial\Omega$ thus defined are represented internally by discrete point grids Ω_ν and $\partial\Omega_\nu$ (see section 2). The user specifies the desired grid densities (the two grids Ω_ν and $\partial\Omega_{\tilde{\nu}}$ are independent). Points in Ω_ν which are too close to $\partial\Omega$, and may thus cause trouble, can be deleted, if desired. Associated with each such "minimization grid" is an "evaluation grid" which is approximately four times as dense, and is used to measure the error in the approximate solution after the actual computation is performed.

The quasilinear parabolic problems which the system can now solve are of the form (1.1) with a=a(x). The scheme that is normally used to solve (3.3) - (3.4) is a predictor-corrector scheme, with a fixed time increment τ. Denoting a quantity $\Psi(t)$ at time $t=t_k=k\tau$ by Ψ^k, the unknown vector α^{k+1} is obtained from the known α^k (starting with the known value $\alpha^0=\alpha(0)$, obtained as described in section 2) according to the following scheme :

$$P\alpha^{k+1} = \bar{g}^{k+1} \qquad (4.1)$$

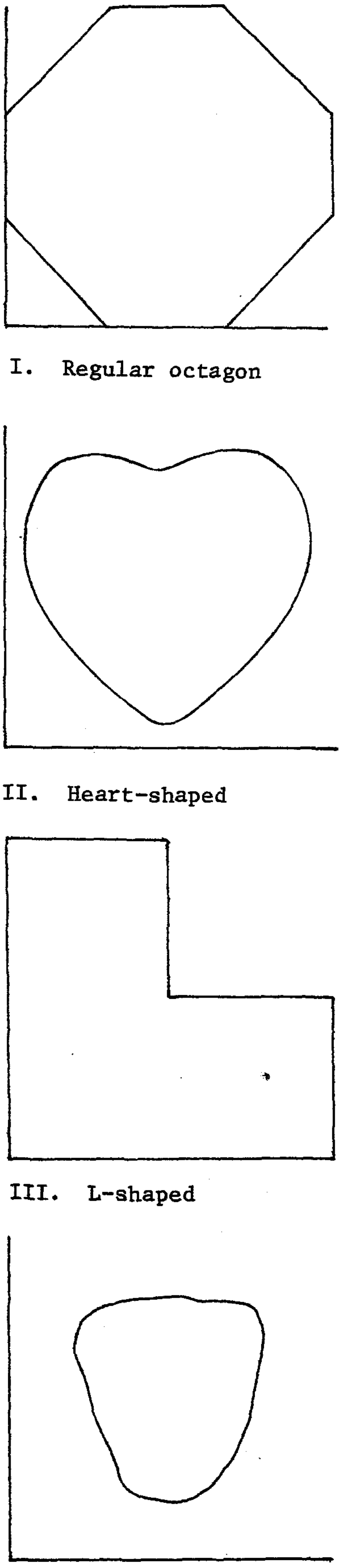
$$\begin{cases} (Q - \frac{1}{2}\tau WR)\tilde{\alpha}^{k+1} = (Q + \frac{1}{2}\tau WR)\alpha^k + \tau\bar{b}(\alpha^k;t_k) \\ (Q - \frac{1}{2}\tau WR)\alpha^{k+1} = (Q + \frac{1}{2}\tau WR)\alpha^k \\ \qquad\qquad + \frac{1}{2}\tau[\bar{b}(\tilde{\alpha}^{k+1};t_{k+1}) + \bar{b}(\alpha^k;t_k)] \end{cases} \qquad (4.2)$$

The approximate solution v is then given in the following convenient form on a set of discrete time levels t_k, $k=0,\dots,T/\tau$,

$$v(\alpha^k;t_k,x) = \sum_{i=1}^{m_1} \sum_{j=1}^{m_2} \alpha_{ij}^k \Psi_i(x_1)\Psi_j(x_2). \qquad (4.3)$$

The $\Psi_i(\cdot)$ form a "smooth" B-spline basis, with equally spaced knots, in each of the spatial directions, and $m_1 \cdot m_2 = m$.

After the user has completed the formulation of the problem via the graphics terminal, the data are prepared by the terminal's host computer and sent to the large scale computer (the Cyber 74) for execution. If suitable collocation points are not already known, the application program solves an ℓ_1-minimization problem (2.1), defined on the "minimization grids". The approximate solution is then carried forward in time on the collocation points using (4.1) and (4.2). Because of its smooth and convenient form the approximate solution thus obtained (and, if desired, its derivatives), can be displayed graphically in several



I. Regular octagon

II. Heart-shaped

III. L-shaped

IV. Hand drawn

Fig. 1. Examples of two-dimensional domains on which parabolic problems were solved.

ways, such as a contour plot or a selected cross-section plot. Furthermore, the evaluation grid may be used to compute quantities which allow error estimates to be made. In particular, the error in the initial approximation and on the boundary $\partial\Omega$ can be computed and displayed graphically. If the user is not satisfied with the results obtained, the problem can be modified interactivily to get an improved solution. For example, the number of basic functions can be increased (increase m, decrease h) in order to get a more accurate approximate solution. It is also rather easy to solve the same problem with modified initial and boundary data $u_0(x)$ and $g(t,x)$, once a solution has been obtained for one choice of these functions, say $\bar{u}_0$ and $\bar{g}$. Specifically the collocation points determined for $\bar{u}_0$ and $\bar{g}$ can be used for the new data, so that the initial ℓ_1-minimization is not required.

For a given domain and choice of basis functions, the matrices Φ and G appearing in equations (3.5) and (3.12) are determined by the selection of the collocation points $\{\xi_i\}_{i=1}^{m}$. These points are determined by the initial ℓ_1-approximation, and depend on the ratio $\gamma \equiv \omega_1/\omega_0$ of weighting factors used in (2.1). Increasing γ will emphasize the boundary error and therefore will tend to increase the number of boundary collocation points.

Since Φ and G depend on the ratio γ, the stability and convergence of the method will also depend on γ. A computational investigation has been carried out [1] for a variety of two-dimensional domains. This investigation shows that the best choice for γ is in the range $2 \leqslant \gamma \leqslant 4$. Typically this will result in the two terms of (2.1) being of approximately the same magnitude. So far, it has not been possible to show theoretically that stability and convergence can be guaranteed by a suitable choice of γ.

With the approximating functions $\Phi_1(x),\ldots,\Phi_m(x)$ being products of one-dimensional "smooth" B-spline functions

$\{\Phi_i(x) = \Psi_j(x_1)\Psi_k(x_2), 1 \leqslant j \leqslant \sqrt{m}, 1 \leqslant k \leqslant \sqrt{m}\}$,

the desirable number of points on the boundary is

$$4(\sqrt{m} - 1) \leqslant m - d \leqslant 4\sqrt{m}. \qquad (4.4)$$

For a rectangular domain, the maximum number of collocation points allowed on the boundary with the above product space is $4\sqrt{m} - \ell$, where ℓ is the number of corners which are collocation points (usually $\ell=4$), and for stability this number of collocation points should be achieved.

When solving a new problem the following procedure is recommended. First solve a test problem of a similar smoothness with known solution on the same domain and with the same equations (1.1), except that a known term $\hat{b}(t,x)$ is added to $b(t,x,u,u_x)$. The data $g(t,x)$ and $u_0(x)$ will also correspond to the known solution. The actual error in the approximate solution $v(\alpha;t,x)$ can then be computed on the evaluation grid. If the error is acceptably small, the given problem is then solved using the same basis functions and collocation points. On the other hand, if the test problem error is too large, the number of basis functions is increased until acceptable accuracy is obtained with the test problem, before attempting to solve the given problem.

This interactive graphics system has been successfully tested on a variety of parabolic problems in two space variables. We conclude with a brief discussion of some of these numerical results. For complete details on these numerical results see [1].

Problems were solved on 6 different domains, including the unit square. Four of these are shown in Fig. 1. A multiply connected domain is illustrated in Fig. 2, which also shows the minimization grid (heavy dots) and the evaluation grid (all dots). The differential operators in (1.1) were all of the form

$$u_t = L[u] + f(t,x,y)$$

and four different (elliptic) operators L[u] were used, with one linear and three quasilinear. In order to compare accuracy, problems with known solutions were constructed in most cases, by choosing $f(t,x,y) \equiv \bar{u}_t - L[\bar{u}]$ for any desired test function $\bar{u} = \bar{u}(t,x,y)$. A total of six such test functions were used.

To illustrate the numerical results obtained we briefly describe several typical cases.

For the first case the quasilinear elliptic operator $L[u] = \Delta u - 4au^3 + bu_x$ was used. For initial data given by $u_0(x,y,0) = 1/\sqrt{a}(x+y+c)$ and $f(x,y,t) \equiv 0$, this has the exact solution $\bar{u}(x,y,t) = 1/\sqrt{a}(x+y+bt+c)$. The parameter values a = E - 4, b = 0.2 and c = 0.1 were used. The basis consisted of 49 quintic B-splines with support on the unit square. A time step $\Delta t = 0.02$ was used. The error in the approximate solution v is measured in the discrete ℓ_1-norm and is given as a relative error, $RE = \|v - \bar{u}\|_1 / \|\bar{u}\|_1$. For the heart-shaped domain shown in Fig. 1, the relative error at t = 1 was RE = 1.90E - 5. The corresponding error at t = 0 (best ℓ_1-approximation to the initial data with this basis) was RE = 1.39E - 5.

For the second case described here, the same operator was used, but f(x,y,t) was determined so that (1.1) had the exact solution

$$\bar{u} = \log\,[2 + \cos(\pi x)\sin(\pi y)\exp(-c\pi^2 t)].$$

The same quintic spline basis was used as above, and the parameter value was taken as c = 0.2. For the heart-shaped domain with $\Delta t = 0.02$ the result was RE = 6.70E - 6 at t = 1, starting with an initial error of RE = 9.15E - 5 at t = 0. Thus the relative error decreased with time (as the solution becomes smoother). The identical problem was also solved on the unit square with the same spline basis and Δt. The result was RE = 9.45E - 6 at t = 1, with an inital error of RE = 3.24E - 4 at t = 0.

The smaller error on the irregular and nonconvex (but smaller) heart-shaped domain, shows clearly the advantage of this collocation method for irregular domains. The improved accuracy on the heart-shaped domain (as compared to the unit square domain) is due to the better approximation to the initial data which can be obtained using the same basis functions on a smaller domain.

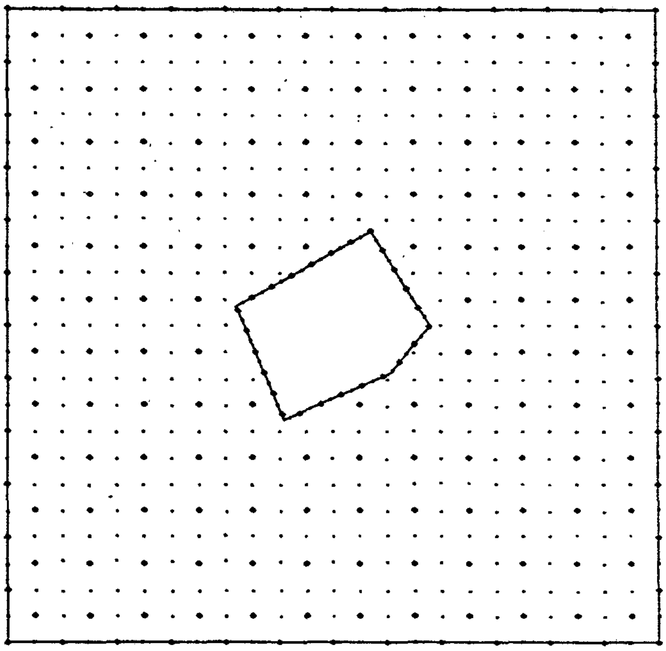


Fig. 2. Typical multiply-connected domain defined by a polygonal inner boundary.

The last case to be summarized here is one for which the general behavior has been investigated [9], but for which the exact time-dependent solution is not known. The operator here is $L[u] = \Delta u + \tau e^u$, with boundary condition u = 0 on $\partial\Omega$ and $u = u_0\,(x,y)$ as initial condition. For a specified domain Ω, we consider the corresponding elliptic (steady-state) problem L[u] = 0, with u = 0 on $\partial\Omega$. Then for each domain there is a value τ_c, such that for $\tau < \tau_c$ there exist two solutions to the elliptic problem; for $\tau = \tau_c$ there is one solution; and for $\tau > \tau_c$ there is no solution. Furthermore, the solution to the parabolic problem (1.1) will generally converge to one of the elliptic problem solutions as $t \to \infty$ (this depends on the initial data u_0), provided that $\tau \leqslant \tau_c$. On the other hand, for $\tau > \tau_c$ the solution to (1.1) will "blow up", that is $\|u(t)\|$ becomes unbounded in a finite time.

Numerical solutions were obtained for a circular domain (radius = 0.5), the unit square and the heart-shaped domain. While the parabolic problem solution is not known, the elliptic problem solution is known for the circular domain. For this case $\tau_c = 8.0$, and for $\tau = 7.9$ we know that $\|\bar{u}\|_\infty = 1.174$, where $\bar{u}$ solves the elliptic problem. The numerical solution to (1.1) with $u_0(r) = \sin\pi(r+1/2)$, where $r^2 = (x-1/2)^2 + (y - 1/2)^2$, was obtained with 49 quintic splines and $\Delta t = 0.02$. The "steady-state" solution was essentially reached at t = 2. The relative error at t = 2 was given by $\|v - \bar{u}\|_1 / \|\bar{u}\|_1 = 6.3E - 3$.

For the unit square and heart-shaped domains the value of τ_c was determined numerically by solving (1.1) for increasing values of τ. The maximum value of τ for which v did not blow up was taken as τ_c. The values obtained in this way were $\tau_c = 6.81$ for the unit square, and $\tau_c = 11.5$ for the heart-shaped domain.

Finally, numerical solutions were obtained to this parabolic problem on the unit square and heart-shaped domains for values of $\tau > \tau_c$, in order to follow in detail how the solution v(t,x,y) grows with time. This solution represents a simplified model of the self-ignition of a gas mixture, where v is the local temperature and the term τe^v represents the corresponding rate of heat generation. The interactive graphics system was used to give a 3-dimensional, hidden-line display showing v at each time step. The display was photographed to produce a 16 mm film showing the time history of the self-ignition as a dynamic graphical presentation [15].

A sequence of 20 frames from this film is shown in Fig. 3. This sequence shows the temperature (vertical axis) for the square domain (x-y axis) at equal time intervals $\Delta t = 0.03$, starting with t = 0.03. The initial temperature distribution was taken as v = 0. The parameter value used for this sequence was $\tau = 7.5 > \tau_c = 6.81$. The slow initial growth, and rapid final increase of temperature at the center of the domain is clearly shown. The final frame shown (t = 0.6) represents the largest value of t for which a bounded solution was obtained.

REFERENCES

1. ASCHER, U. : "A collocation method for parabolic and elliptic quasilinear problems on general domains", Ph. D. Thesis, Computer Science Dept., Univ. of Minnesota (1975).

2. ASCHER, U. : "On the invarance of the interpolation points of the discrete ℓ_1-approximation", *J. Approx. Th.* 24, 83-91 (1978).

3. ASCHER, U.; SAVITT, S. and ROSEN J. B. : "An interactive graphics system for boundary value problems : II", Tech. Report 76-2, Computer Science Dept., Univ. of Minnesota, Minneapolis (1976).

4. BABUSKA, I. : "Finite element method with penalty", Rept. BN-710, Univ. of Maryland (1971).

5. BARRODALE, I. and ROBERTS, F. D. K. : "An improved algorithm for discrete ℓ_1-linear approximation", SIAM J. Num. Anal 10, 839-848 (1973).

6. DENDY, J. E. : "Penalty Galerkin methods for partial differential equations", SIAM J. Num. Anal. 11, 604-636 (1974).

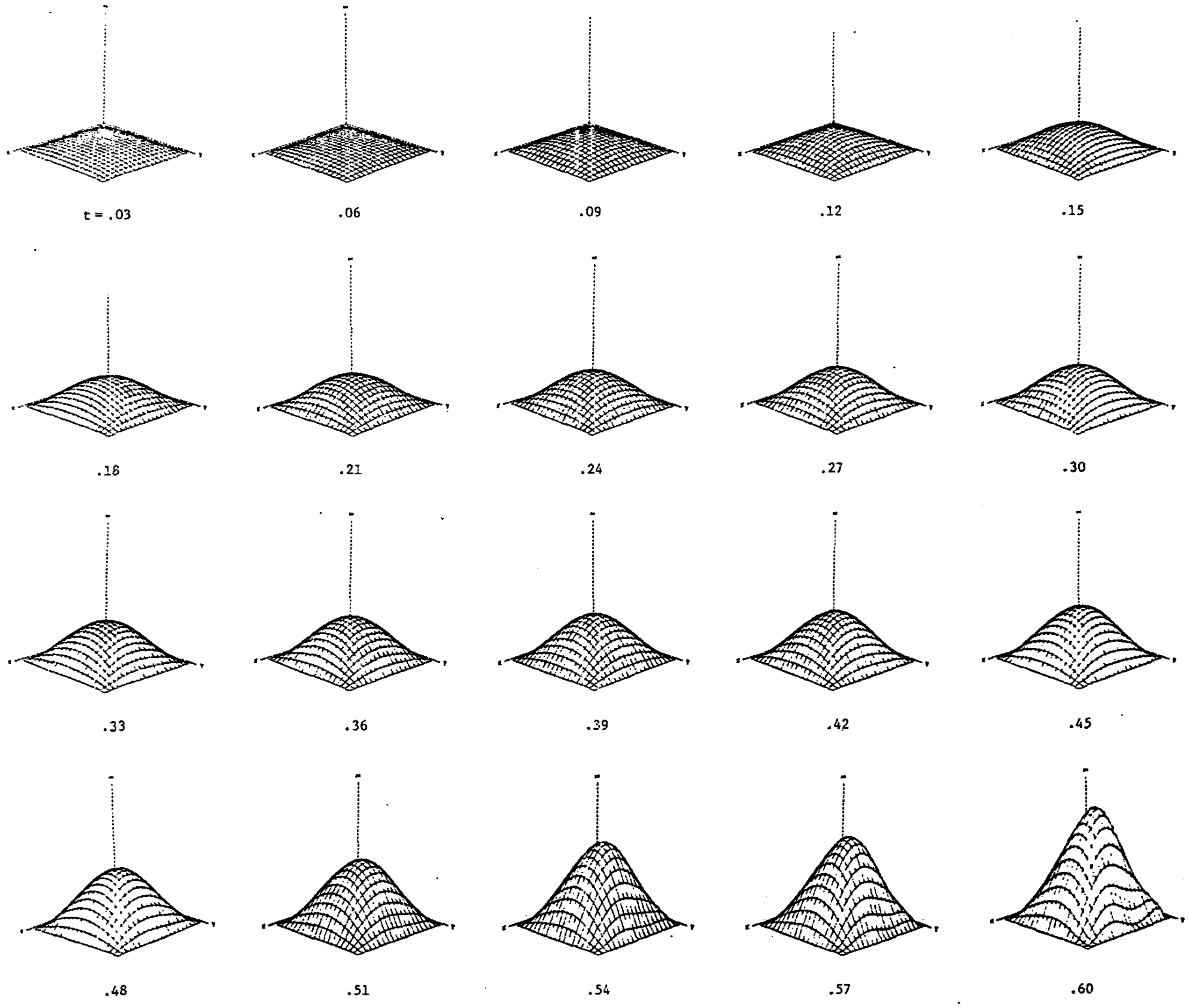


Fig. 3. Sequence representing temperature of self-igniting gas mixture on a square domain.

7. DOUGLAS, J. and DUPONT, T. : "A finite element collocation method for quasilinear parabolic equations", Math. Comp. 27, 17-28 (1973).

8. DOUGLAS, J. and DUPONT, T. : *Collocation methods for parabolic equations in a single space variable,* lecture notes in math., Springer-Verlag (1974).

9. FUJITA, H. : "On the nonlinear equations $\Delta u + e^u = 0$ and $v_t = \Delta v + e^v$", Bull. Amer. Math. Soc. 74, 132-135 (1969).

10. GEAR, C. W. : *Numerical initial value problems in ordinary differential equations,* Prentice-Hall (1971).

11. HOUSTIS, E. N.; LYNCH, R. E.; PAPATHEODOROU, T. S. and RICE, J. R. : "Evaluation of numerical methods for elliptic partial differential equations", Tech. Report 204, Computer Science Dept., Purdue Univ. (1976).

12. MASON, J. C. : "Orthogonal polynomial approximation methods in numerical analysis", in *Approximation theory,* A. Talbot (ed.), Acad. Press, 7-33 (1973).

13. RABINOWITZ, P. : "Applications of linear programming to numerical analysis", SIAM Review 10, 121-159 (1968).

14. ROSEN, J. B. : "Solution of boundary value problems by a weighted L_1 minimization", Proc. IFIP Congress 74, North Holland Publ. 456-460 (1974).

15. ROSEN, J. B.; ASCHER, U.; McCALL, E. and SAVITT, S. : "An interactive graphics system for numerical problems (16 mm sound film)", Computer Science Dept., Univ. of Minnesota, Minneapolis (1975).

16. RUSSELL, R. D. and VARAH, J. M. : "A comparison of global methods for linear two-point boundary value problems", Math. Comp. 29, 1007-1019 (1975).

17. SAVITT, S.; ASCHER, U. and ROSEN, J. B. : "An interactive graphics system for boundary value problems", Tech. Report 74-10 Computer Science Dept., Univ. of Minnesota (1974).

18. SCHULTZ, M. H. : "L^∞-multivariate approximation theory", SIAM J. Num. Anal. 6, 161-183 (1969).

19. SERBIN, S. M. : "A computational investigation of least squares and other projection methods for the approximate solution of boundary value problems", Ph. D. Thesis, Cornell Univ. (1971).

20. WALTER, W. : *Differential and integral inequalities,* Springer-Verlag (1970).