

A kernel-based method for parabolic equations with nonlinear convection terms

James F. Epperson

Department of Mathematical Sciences, University of Alabama in Huntsville, Huntsville, AL 35899, United States

Received 27 March 1990

Revised 4 January 1991

Abstract

Epperson, J.F., A kernel-based method for parabolic equations with nonlinear convection terms, *Journal of Computational and Applied Mathematics* 36 (1991) 275–288.

We present an algorithm for parabolic equations with (possibly nonlinear) convective terms, based on our earlier work using linear semigroup methods. Here the semigroup is used to decouple the convective term to allow for a two-stage solution process. The method is shown to be asymptotically stable and second-order accurate, with the stability being independent of the mesh parameters and the convective term.

Keywords: Green's functions, numerical methods, convective diffusion, partial differential equations.

1. Introduction

In this note we present and analyze methods for parabolic problems of the form

$$\begin{aligned}u_t &= Lu + V(u)_x, & x \in \Omega, \quad t > 0, \\u(x, t) &= 0, & x \in \Gamma = \partial\Omega, \quad t > 0, \\u(x, 0) &= u_0(x) & x \in \Omega,\end{aligned}\tag{1}$$

where L is a second-order symmetric uniformly elliptic linear differential operator, and $\Omega \subset \mathbb{R}^n$, $n \geq 1$, is an open bounded domain with a smooth boundary Γ . As the examples will show, the assumption of homogeneous boundary data is for simplicity only.

The method uses a linear semigroup approach to decouple the convection term from the diffusion operator by using the kernel or Green's function for the operator $\partial_t - L$ to formally solve (1) in an integral form, then approximate the kernel operation and the resulting integration. This is in some sense a development of the short note [5], as well as [6], but it also is motivated in part by the continuing development of the method of modified characteristics [2,3].

The paper is organized as follows. In Section 2 we present the method as applied to the simplest problem in order to show the salient features of the algorithm. Stability and accuracy are established (under very mild hypotheses) in Theorems 3.3, 3.5, 3.6 of Section 3. These results

imply that the method is stable — in the sense that the approximate solution decays to zero as $t \rightarrow \infty$ — for any time step k and spatial step h , independent of the size of the convective term. While the method is second-order accurate in both space and time, the multiplying constant in the error estimate can still be quite large when the convective term is large compared to the ratio k/h .

Example computations are given in Section 4. These show that the kernel method performs modestly better than the traditional Crank–Nicolson differencing, with slightly less “ripple” near steep gradients. The principal advantage of the kernel method lies in the rapid convergence of iterations for problems with nonlinear convective terms. Further work to improve the performance of the method for problems with sharp fronts, including the development of a strongly stable variant, is in progress.

2. The model problem

Consider a simple one-dimensional heat equation with constant diffusion coefficient and nonlinear convective term:

$$\begin{aligned} u_t &= au_{xx} + V(u)_x, \quad 0 < x < 1, \\ u(0, t) &= u(1, t) = 0, \\ u(x, 0) &= u_0(x). \end{aligned} \tag{2}$$

As in [6], we let G be the Green’s function for the linear part of (2), i.e.,

$$\varphi(x, t) = \int_0^1 G(x, t; \xi) \varphi_0(\xi) d\xi \tag{3}$$

if and only if

$$\begin{aligned} \varphi_t &= a\varphi_{xx}, \quad 0 < x < 1, \\ \varphi(0, t) &= \varphi(1, t) = 0, \\ \varphi(x, 0) &= \varphi_0(x). \end{aligned} \tag{4}$$

Define the kernel integration (3) as $\mathcal{G}(\tau)\varphi_0 = \varphi$, i.e.,

$$(\mathcal{G}(\tau)\varphi_0)(x, \tau) = \int_0^1 G(x, \tau; \xi) \varphi_0(\xi) d\xi,$$

so that $\mathcal{G}(\tau)\varphi_0$ is a solution of the homogeneous linear heat equation having φ_0 as the initial data.

It is straightforward, then, to solve (2) in terms of \mathcal{G} by the variation of parameters formula:

$$u(x, t) = (\mathcal{G}(t)u_0)(x, t) + \left(\int_0^t \mathcal{G}(t-s)V(u(\cdot, s))_x ds \right)(x, t).$$

Alternately, we can write the solution at time t in terms of the solution at time $t - \Delta t$ as follows:

$$u(x, t) = (\mathcal{G}(\Delta t)u(\cdot, t - \Delta t))(x, t) + \left(\int_{t-\Delta t}^t \mathcal{G}(t-s)V(u(\cdot, s))_x ds \right)(x, t).$$

If we carefully use the trapezoid rule on the time integral, keeping in mind the δ -function behavior of G as the second argument goes to 0, then we get that

$$\begin{aligned} u(x, t) - \frac{1}{2}\Delta t V(u(x, t))_x \\ = (\mathcal{G}(\Delta t)(u(\cdot, t - \Delta t) + \frac{1}{2}\Delta t V(u(\cdot, t - \Delta t))_x))(x, t) + \epsilon(t), \end{aligned} \quad (5)$$

where the truncation error $\epsilon(t)$ is defined by

$$\epsilon(t) = -\frac{1}{12}\Delta t^3 \frac{\partial^2}{\partial s^2} \left\{ \int_0^1 G(x, t-s; \xi) V(u(\xi, s))_x d\xi \right\} \Big|_{s=\eta},$$

for $\eta \in [t - \Delta t, t]$.

Converting (5) to a practical computational algorithm simply requires some means of approximating the kernel G , or, more fundamentally, the operator \mathcal{G} . While direct approximation of the kernel and/or the associated integration is worth considering [1,4,5] it is also the case that any standard finite-difference/element approximation of (4) also provides a means of approximating \mathcal{G} . This is what we will consider here; for simplicity of exposition, we will restrict ourselves to the finite-difference case.

Given a grid of points $\{x_i = ih \mid 0 \leq i \leq N, h > 0\}$ on $[0, 1]$ and a time step $k = \Delta t > 0$, we can compute approximate solutions of (4) by solving the linear system associated with the difference approximation. If $v^{(0)} \in \mathbb{R}^{N-1}$ is the vector of discrete initial values, then the approximate solution after a single time step is $v^{(1)} = \mathcal{G}_{h,k} v^{(0)}$, where $\mathcal{G}_{h,k}$ is the solution operator for the linear system. For the usual Crank–Nicolson based schemes, $\mathcal{G}_{h,k}$ is an $(N-1) \times (N-1)$ matrix defined by

$$\mathcal{G}_{h,k} = (I + \mu K)^{-1}(I - \mu K),$$

where K is symmetric and positive definite, and $\mu = ak/(2h^2)$. Consequently, we can approximate solutions of the original problem (2) by solving the set of nonlinear equations

$$u_i^{n+1} - \frac{1}{2}k \left(\frac{V(u_{i+1}^{n+1}) - V(u_{i-1}^{n+1})}{2h} \right) = g_i^{n+1}, \quad (6)$$

where u_i^{n+1} approximates $u(x_i, (n+1)k)$ and $g^{n+1} \in \mathbb{R}^{N-1}$ is defined according to

$$g^{n+1} = \mathcal{G}_{h,k} v^n, \quad (7)$$

for

$$v_i^n = u_i^n + \frac{1}{2}k \left(\frac{V(u_{i+1}^n) - V(u_{i-1}^n)}{2h} \right). \quad (8)$$

The algorithm can therefore be stated quite succinctly as follows.

Given an initial vector u^0 ,

- (1) form v^n from u^n according to (8);
- (2) compute $g^{n+1} = \mathcal{G}_{h,k} v^n$;
- (3) solve the nonlinear system (6) to get u^{n+1} ;
- (4) return to (1).

(A)

Note that we can carry out this algorithm using the same linear equation solver (to compute v^n) that we could use for a problem governed by the linear part of (2). Unlike the previous work

[5,6], we do have to solve a nonlinear *system* at each time step; but it is worth noting that the right side vector g^{n+1} is independent of u_i^{n+1} for all i , thus simplifying the iteration substantially. In addition, note that (A) is immediately extendable to the more general problem (1) in an obvious manner.

3. Error analysis

We now develop the stability and error analysis for the algorithm (A). To simplify the exposition, we will consider only the one-dimensional case; the results extend to multiple dimensions with only notational changes. To begin with, we develop some useful notation.

Given a set of values $\{\varphi_i, 0 \leq i \leq N\}$ defined on our grid, we denote the discrete L^2 norm of this set by

$$\|\varphi_i\|_{2,h}^2 = \sum_{i=0}^N h \varphi_i^2.$$

Since this is proportional to the usual Euclidean vector norm, it can be made consistent with the usual operator 2-norm, i.e.,

$$\|Aw\|_2 \leq \|A\|_2 \|w\|_2 = h^{-1} \|A\|_2 \|w\|_{2,h}.$$

The vector of exact solution values at time t_n will be denoted by $U(t_n) = (u(x_1, t_n), \dots, u(x_{N-1}, t_n))^T$. Similarly, the vector of approximate solution values at time t_n will be denoted by $U_{h,k}(t_n) = (u_1^n, \dots, u_{N-1}^n)^T$.

For $w \in \mathbb{R}^{N-1}$, $S_V: \mathbb{R}^{N-1} \rightarrow \mathbb{R}^{N-1}$ is defined by

$$(S_V(w))_i = \begin{cases} V(w_{i+1}), & i = 1, \\ V(w_{i+1}) - V(w_{i-1}), & 2 \leq i \leq N-2, \\ -V(w_{i-1}), & i = N-1. \end{cases}$$

Therefore, the numerical method (6), (7) can be written in vector form as follows:

$$U_{h,k}(t_{n+1}) - \alpha S_V(U_{h,k}(t_{n+1})) = \mathcal{G}_{h,k}(U_{h,k}(t_n) + \alpha S_V(U_{h,k}(t_n))), \quad (9)$$

where $\alpha = k/(4h)$.

We now make some assumptions about the exact solution u and the discrete scheme used to solve the system (7).

(H₁) $u \in C^{4,2}$, $V \in C^3$ and $V(0) = 0$.

(H₂) There exists a positive constant C_V such that $|\mathcal{V}(x, t, s)| \leq C_V$ for all (x, t, s) , where

$$\mathcal{V}(x, t, s) = \frac{\partial^2}{\partial s^2} \left\{ \int_0^1 G(x, t-s; \xi) V(u(\xi, s))_x d\xi \right\}.$$

(H₃) There exists a positive constant C_G such that

$$\|\mathcal{G}_{h,k}\|_2 = \gamma \leq 1 - C_G k,$$

as $k \rightarrow 0$, so long as $k = Ch$ for some positive constant C .

(H₄) The matrix $\mathcal{G}_{h,k}$ represents a second-order accurate method in the sense that

$$\|(\mathcal{G}(k)\varphi)(\cdot, t+k) - (\mathcal{G}_{h,k}\Phi(t))\|_{2,h} \leq Ck(h^2 + k^2)N_{2,h}(\varphi),$$

where $\Phi(t) = (\varphi(x_1, t), \dots, \varphi(x_{n-1}, t))^T$ for φ an arbitrary smooth function; $C > 0$ is a positive constant independent of h , k and φ ; and $N_{2,h}(\varphi)$ depends only upon the discrete L_2 norm of φ and its derivatives.

The last two assumptions are in fact true for the usual Crank–Nicolson finite-difference (or finite-element) approximations to linear parabolic equations, as we shall now show.

Lemma 3.1. *Let $\mathcal{G}_{h,k}$ be the solution operator for a Crank–Nicolson central difference approximation to the initial/boundary value problem (4); then (H₃) and (H₄) both hold.*

Proof. We have

$$\mathcal{G}_{h,k} = (I + \mu K)^{-1}(I - \mu K),$$

where $K = \text{tridiag}(-1, 2, -1)$ and $\mu = ak/(2h^2)$. Since K is symmetric and positive definite, it can be orthogonally diagonalized, hence we quickly get, for k sufficiently small,

$$\|\mathcal{G}_{h,k}\|_2 = \max_i \left(\frac{1 - \mu\lambda_i}{1 + \mu\lambda_i} \right) = 1 - \min_i \left(\frac{2\mu\lambda_i}{1 + \mu\lambda_i} \right) = 1 - \left(\frac{2\mu\lambda_{\min}}{1 + \mu\lambda_{\min}} \right),$$

where the $\{\lambda_i\}$ are the eigenvalues of K , and $\lambda_{\min} > 0$ is the smallest one. But $h^{-2}\lambda_i$ is an approximation to the i th eigenvalue of the differential operator $-\partial_{xx}$; moreover, the min/max characterization of the eigenvalues guarantees that

$$h^{-2}\lambda_{\min} \geq \Lambda_{\min},$$

where Λ_{\min} is the smallest eigenvalue of $-\partial_{xx}$. Therefore

$$\|\mathcal{G}_{h,k}\|_2 \leq 1 - \left(\frac{2k\Lambda_{\min}}{2 + k\Lambda_{\min}} \right),$$

which establishes (H₃).

To complete the lemma, we simply observe that the truncation error for Crank–Nicolson is well known to be $\mathcal{O}(h^2 + k^2)$, and that (H₃) constitutes a stability estimate; (H₄) follows immediately. \square

Remark 3.2. The proof of this lemma is not as dependent on the one-dimensional nature of the problem (or the simple form of the PDE) as might appear to be the case. Essentially, all that is required is that the PDEs have the form $u_t = Lu + V(u)_x$ where $-L$ is a positive definite elliptic operator over a “reasonable” domain Ω . In this case the eigenvalues of K will still be positive, the min/max characterization is still valid, and the same result follows.

We are now in a position to establish the stability and accuracy of (A), as follows.

Theorem 3.3 (truncation error). *Let u be the exact solution of (1); then the vector $U(t_{n+1})$ satisfies*

$$U(t_{n+1}) - \alpha S_V(U(t_{n+1})) = \mathcal{G}_{h,k}(U(t_n) + \alpha S_V(U(t_n))) + E(t_{n+1}), \quad (10)$$

where $E(t_{n+1}) = (\epsilon_i^{n+1}, \dots, \epsilon_{N-1}^{n+1})^T$ is the truncation error which can be bounded according to

$$\|E(t_{n+1})\|_\infty \leq Ck(h^2 + k^2)N_{2,h}(u),$$

where $C > 0$ is independent of h and k , and $N_{2,h}(\cdot)$ is as above.

Proof. In (5), if we approximate $(V(u))_x$ with the usual central difference formula and introduce the vector notation $U(t_n)$ for $\{u(x_i, t_{n+1})\}$, we quickly get

$$U(t_{n+1}) - \alpha S_V(U(t_{n+1})) = \mathcal{G}(U(t_n) + \alpha S_V(U(t_n))) + E_1(t_{n+1}),$$

where E_1 is the vector of truncation errors (from the trapezoid rule and from the difference approximation to $(V(u))_x$) and \mathcal{G} is understood to be operating on each component of its argument vector. If we replace \mathcal{G} by the matrix $\mathcal{G}_{h,k}$, then we get

$$U(t_{n+1}) - \alpha S_V(U(t_{n+1})) = \mathcal{G}_{h,k}(U(t_n) + \alpha S_V(U(t_n))) + E(t_{n+1}),$$

where $E(t_{n+1}) = (e_1(t_{n+1}), \dots, e_{N-1}(t_{n+1}))^T$ with

$$\begin{aligned} e_i(t_{n+1}) = & (\mathcal{G}(k)u)(x_i, t_{n+1}) - (\mathcal{G}_{h,k}U(t_{n+1}))_i \\ & - \frac{1}{12}k^3 \frac{\partial^2}{\partial s^2} \left\{ \int_0^1 G(x_i, t_{n+1} - s; \xi) V(u(\xi, s))_x d\xi \right\} \Big|_{s=\eta} \\ & - kh^2(V(u(\xi_i, \tau_n)))_{xxx}, \end{aligned}$$

for $\xi_i \in (x_{i-1}, x_{i+1})$ and $\tau_n \in (t_n, t_{n+1})$. It therefore follows from (\mathbb{H}_1) , (\mathbb{H}_2) and (\mathbb{H}_4) that

$$|e_i(t_{n+1})| \leq Ck(h^2 + k^2)N_{2,h}(u) + \frac{1}{12}C_V k^3,$$

for all i , which completes the proof. \square

We now turn our attention to establishing the stability of the method; a preliminary lemma is first necessary, however.

Lemma 3.4. Let R be a skew matrix, i.e., $R^T = -R$. Then

- (i) $\|I + R\|_2 = \max_j (1 + r_j^2)^{1/2},$
- (ii) $\|(I + R)^{-1}\|_2 = \min_j (1 + r_j^2)^{1/2},$

where $\{ir_j\}$ are the eigenvalues of R .

Proof. This is a direct computation using the definition of $\|\cdot\|_2$ for matrices and the fact that R is skew. We look at the first case, only. Define A as

$$A = (I + R)^T(I + R),$$

so that

$$\rho(A)^{1/2} = \|I + R\|_2.$$

But the skewness of R implies that $A = I - R^2$ from which (i) follows immediately. \square

This lemma will be used towards the end of the stability theorem, which we now state and prove.

Theorem 3.5 (stability). *Assume bounded initial data in (2) and that the convective term satisfies (\mathbb{H}_1) . Then the algorithm **(A)** is stable in the sense that, for all time steps k and spatial steps h , the approximate solution satisfies $U_{h,k}(t_n) \rightarrow 0$, as $n \rightarrow \infty$.*

Proof. From (9), using the Mean Value Theorem and the assumption that $V(0) = 0$, we have

$$U_{h,k}(t_{n+1}) = \mathcal{P}_n U_{h,k}(t_n), \quad (11)$$

where \mathcal{P}_n is the matrix

$$\mathcal{P}_n = (I - \alpha R_{n+1})^{-1} \mathcal{G}_{h,k}(I + \alpha R_n),$$

for

$$R_n = \text{tridiag}(-V'(\mu_{i-1}^n), 0, V'(\mu_{i+1}^n)),$$

where $V'(\mu_i^n)u_i^n = V(u_i^n)$, by the Mean Value Theorem. Note that R_n is skew; this is the crucial property which is preserved in the more complicated multidimensional case.

Typically, stability comes from proving that the spectral radius $\rho(\mathcal{P}_n)$ is less than one for all n . In this case, however, we will look not at the growth of the individual \mathcal{P}_n , but at the entire product $\prod_j \mathcal{P}_j$.

From (11) it follows that

$$U_{h,k}(t_n) = \{\mathcal{P}_n \mathcal{P}_{n-1} \cdots \mathcal{P}_0\} U_{h,k}(0).$$

This can be written quite simply as

$$\{\mathcal{P}_n \mathcal{P}_{n-1} \cdots \mathcal{P}_0\} = (I - \alpha R_{n+1})^{-1} \{P_n P_{n-1} \cdots P_0\} (I + \alpha R_0),$$

where each P_n is defined by

$$P_n = \mathcal{G}_{h,k}(I + \alpha R_n)(I - \alpha R_n)^{-1}.$$

But, for R_n skew, $(I + \alpha R_n)(I - \alpha R_n)^{-1}$ is orthogonal (this is the Cayley transformation), thus

$$\|P_n\|_2 \leq \gamma < 1,$$

for all n . Therefore

$$\|\mathcal{P}_n \mathcal{P}_{n-1} \cdots \mathcal{P}_0\|_2 \leq \gamma^n \|(I - \alpha R_{n+1})^{-1}\|_2 \|(I + \alpha R_0)\|_2.$$

In order to use Lemma 3.4 to bound the remaining matrix norms we must first observe that the hypotheses on V , together with Gerschgorin's Theorem, imply that the eigenvalues of R_0 are bounded above by $V'_{\max} = \max_i |V'(\mu_i^0)|$, where, recall, $V'(\mu_i^0)u_i^0 = V(u_i^0)$ by the Mean Value Theorem. Therefore,

$$\|(I + \alpha R_0)\|_2 \leq 1 + \alpha V'_{\max}.$$

Similarly, we can bound the other matrix norm above by one:

$$\|(I - \alpha R_{n+1})^{-1}\|_2 \leq 1,$$

so that we finally have

$$\|(I - \alpha R_{n+1})^{-1}\|_2 \|(I + \alpha R_0)\|_2 \leq 1 + \alpha V'_{\max}.$$

This establishes that $\|\mathcal{P}_n \mathcal{P}_{n+1} \cdots \mathcal{P}_0\|_2 \leq C_R \gamma^n$ for

$$C_R = 1 + \alpha V'_{\max}.$$

Although $C_R \gg 1$ is possible (quite likely, in fact), we still have stability in the required sense since the γ^n term will go to zero and C_R is bounded for all n . This completes the proof. \square

Theorem 3.6 (error estimate). *Under the assumptions (\mathbb{H}_1) – (\mathbb{H}_4) above, the set of discrete values $\{u_i^n\}$ converges to $\{u(x_i, t_n)\}$, in the sense that*

$$\|u(x_i, t_n) - u_i^n\|_{2,h} \leq C(h^2 + k^2)M_{2,h}(u),$$

where $M_{2,h}(u)$ depends only upon discrete L_2 norms of u and its derivatives, and C is bounded for all n .

Proof. Subtracting (9) and (10) gives

$$\begin{aligned} U(t_{n+1}) - U_{h,k}(t_{n+1}) - \alpha S_V(U(t_{n+1})) + \alpha S_V(U_{h,k}(t_{n+1})) \\ = \mathcal{G}_{h,k}(U(t_n) - U_{h,k}(t_n) + \alpha S_V(U(t_n)) - \alpha S_V(U_{h,k}(t_n))) + E(t_{n+1}). \end{aligned}$$

As before, it follows from the Mean Value Theorem and the definition of S_V that, for all n ,

$$S_V(U(t_n)) - S_V(U_{h,k}(t_n)) = R_n(U(t_n) - U_{h,k}(t_n)),$$

where

$$R_n = \text{tridiag}(-V'(v_{i-1}^n), 0, V'(v_{i+1}^n)),$$

for

$$V(u(x_i, t_n)) - V(u_i^n) = V'(v_i^n)(u(x_i, t_n) - u_i^n).$$

Again, note that R_n is skew. Writing the error as

$$e_{h,k}(t_n) = U(t_n) - U_{h,k}(t_n),$$

we now have the recursion

$$(I - \alpha R_{n+1})e_{h,k}(t_{n+1}) = \mathcal{G}_{h,k}(I + \alpha R_n)e_{h,k}(t_n) + E(t_{n+1}).$$

Solving this in the usual way yields the cumbersome expression:

$$(I - \alpha R_n)e_{h,k}(t_n) = A_n(I - \alpha R_0)e_{h,k}(0) + \sum_{j=1}^n B_{n,j}E(t_j),$$

where

$$A_n = (\mathcal{G}_{h,k} Q_{n-1} \mathcal{G}_{h,k} Q_{n-2} \cdots Q_1 \mathcal{G}_{h,k}),$$

for

$$Q_n = (I + \alpha R_n)(I - \alpha R_n)^{-1},$$

which is, again, orthogonal, and

$$B_{n,j} = \begin{cases} I, & n = j, \\ (\mathcal{G}_{h,k} Q_{n-i} \mathcal{G}_{h,k} Q_{n-1} \cdots \mathcal{G}_{h,k} Q_{n-j}), & n \neq j. \end{cases}$$

Therefore, taking discrete 2-norms we have

$$\|e_{h,k}(t_n)\|_{2,h} \leq C_1 \gamma^n \|e_{h,k}(0)\|_{2,h} + C_2 \left\{ \max_i \|E(t_i)\|_{2,h} \right\} \sum_{j=0}^{n-1} \gamma^j,$$

where

$$C_1 = \|(I - \alpha R_n)^{-1}\|_2 \|I + \alpha R_0\|_2 \leq 1 + \alpha V'_{\max}$$

and

$$C_2 = \|(I - \alpha R_n)^{-1}\|_2 \leq 1,$$

the bounds being produced by the same argument as was used in Theorem 3.5. Finally, we note that, summing the geometric series and using (H_3) ,

$$\begin{aligned} \left\{ \max_i \|E(t_i)\|_{2,h} \right\} \sum_{j=0}^{n-1} \gamma^j &\leq Ck(h^2 + k^2) N_{2,h}(u) \frac{1 - \gamma^n}{1 - \gamma} \\ &\leq C(h^2 + k^2) N_{2,h}(u). \end{aligned}$$

This is sufficient to complete the proof. \square

Remark 3.7. While Theorem 3.5 gives us stability without conditions on h and k , the presence of the constant C_1 in the error estimate above indicates that a large convective term can still cause large errors in the approximation. Therefore, in order to achieve convergence to the exact solution we require that the ratio $\omega = k/h$ be bounded uniformly as $h, k \rightarrow 0$. Note, however, that this bound need not depend on the size of $V'(\cdot)$. Note also that this constant is only multiplying the error due to the initial condition and is therefore subject to the greatest decay.

4. Examples

In this section we present the results of some simple example computations for the method outlined above. All computations were done on a Sun 4/260 workstation using NCAR Graphics for the (approximate) solution profiles.

Example 4.1. Largely for the purpose of demonstrating that we obtain the predicted accuracy, we consider first a very simple linear example, i.e.,

$$\begin{aligned} u_t &= u_{xx} + V u_x, \quad 0 < x < \pi, \\ u(0, t) &= u(\pi, t) = 0, \\ u(x, 0) &= e^{-Vx/2} \sin x, \end{aligned}$$

which has exact solution

$$u(x, t) = e^{-t} e^{-V(x/2 + Vt/4)} \sin x.$$

For $V = -1$, Fig. 1 gives the solution profile at $t = \frac{1}{2}\pi$, using $h = \frac{1}{64}\pi$ and $k = \Delta t = \frac{1}{128}\pi$. The L^2 error for this profile is approximately $1.44 \cdot 10^{-7}$; the L^∞ error is $3.17 \cdot 10^{-4}$. As h and k were halved, the errors decreased by the expected factor of 4.

Example 4.2. We next consider a case of Burger's equation:

$$u_t = \alpha u_{xx} - uu_x, \quad 0 < x < \pi,$$

$$u(0, t) = u(\pi, t) = 0,$$

$$u(x, 0) = u_0(x),$$

where α is a parameter. The nonlinear system corresponding to (6) was solved with a nonlinear Gauss–Seidel method, i.e., we used the iteration

$$u_i^{n+1,j+1} - \frac{1}{4}k \left(\frac{(u_{i+1}^{n+1,j})^2 - (u_{i-1}^{n+1,j+1})^2}{2h} \right) = g_i^{n+1}.$$

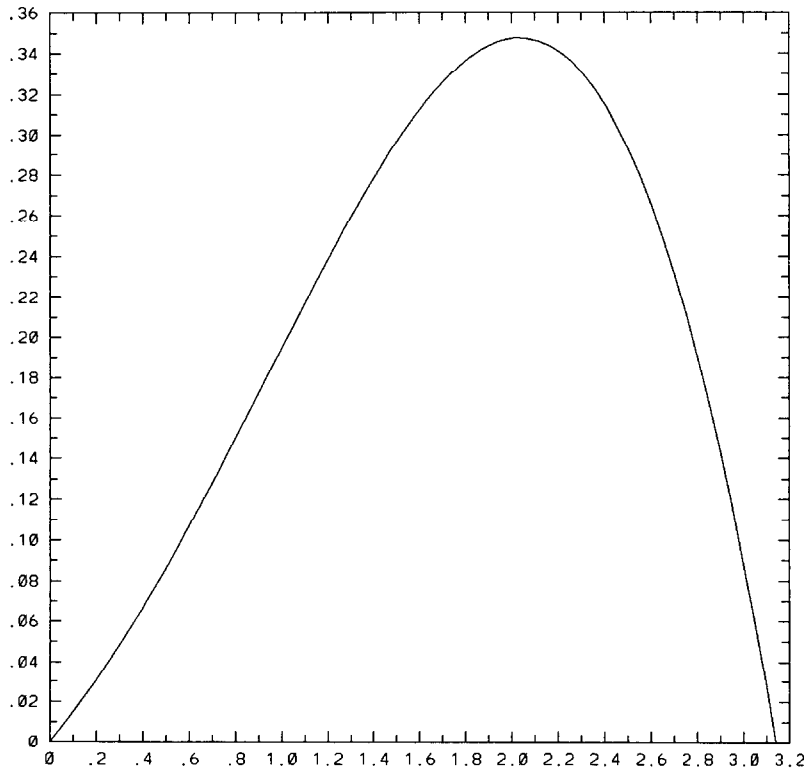


Fig. 1. Simple linear test case; $h = \frac{1}{64}\pi$, $k = \frac{1}{2}h$, $t = \frac{1}{2}\pi$.

Table 1

Average number of iterations per time step for algorithm (A) vs. ordinary Crank–Nicolson (CN), using a Gauss–Seidel type iteration

N	(A)	CN
8	4.6	9.4
16	4.7	14.1
32	4.5	21.2
64	4.2	31.8
128	3.9	46.4

For comparison, we also solved the problem using ordinary Crank–Nicolson, also with a nonlinear Gauss–Seidel iteration. Table 1 gives the average number of iterates per time step for both cases, using $h = \pi/N$, $k = \frac{1}{2}h$, and $t_{\max} = \frac{1}{2}\pi$. While it is true that the performance of both iterations could be improved by more sophisticated procedures, this marked difference in convergence for the same type of iteration does indicate that the kernel method, because of the separation of the diffusive and convective parts of the operator, will converge faster.

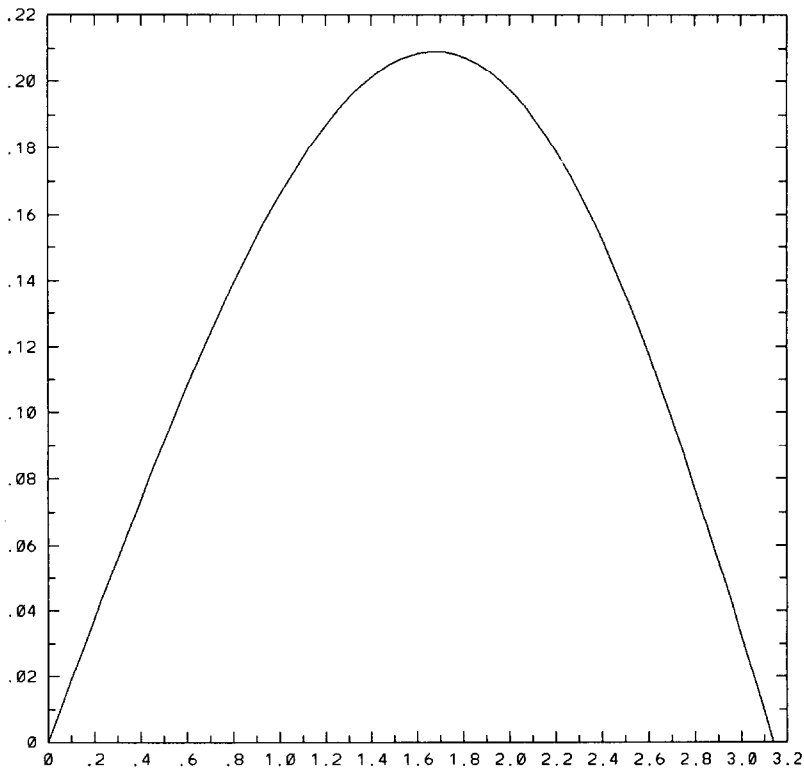


Fig. 2. Burgers equation test case; $h = \frac{1}{64}\pi$, $k = \frac{1}{2}h$, $t = \frac{1}{2}\pi$.

For the special choice of initial data

$$u_0(x) = \frac{2\alpha \sin x}{2 + \cos x},$$

the problem has the exact solution

$$u(x, t) = \frac{2\alpha \exp(-\alpha t) \sin x}{2 + \exp(-\alpha t) \cos x}.$$

Figure 2 gives the solution profile for this particular example at $t = \frac{1}{2}\pi$ for $h = \frac{1}{64}\pi$, $k = \frac{1}{128}\pi$, and $\alpha = 1.0$. When h and k were successively halved, the error again decreased by the expected factor of 4.

Example 4.3. We use this example to show how to accommodate inhomogeneous boundary data. Consider the problem

$$u_t = u_{xx} + Vu_x, \quad 0 < x < \pi,$$

$$u(0, t) = e^{-t}, \quad u(\pi, t) = 0,$$

$$u(x, 0) = e^{-Vx/2} \cos\left(\frac{1}{2}x\right),$$

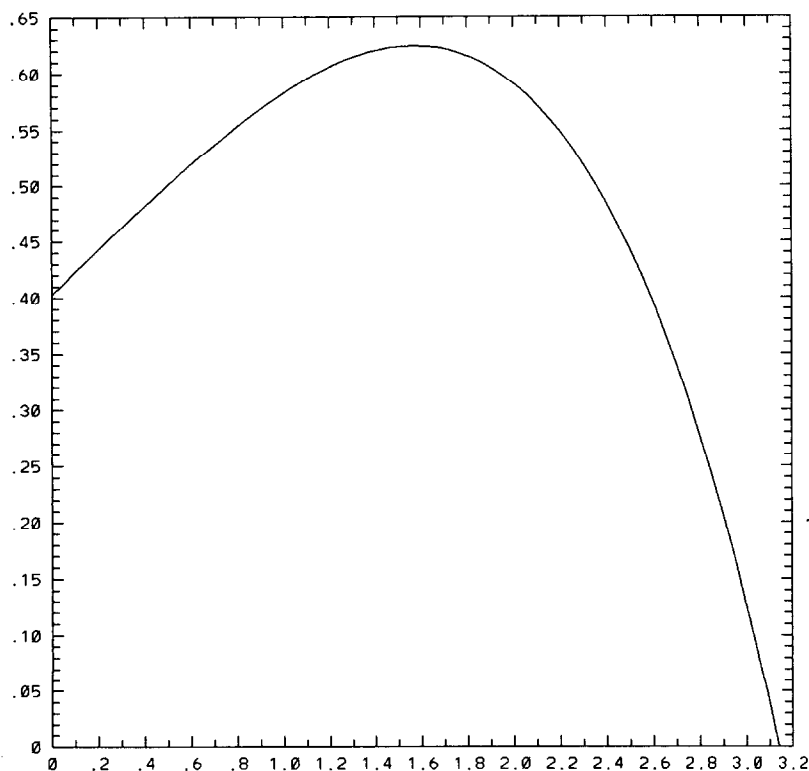


Fig. 3. Inhomogeneous B/C test case; $h = \frac{1}{64}\pi$, $k = \frac{1}{2}h$, $t = \frac{1}{2}\pi$.

which has the exact solution

$$u(x, t) = e^{-rt} e^{-Vx/2} \cos\left(\frac{1}{2}x\right),$$

so long as $r = \frac{1}{4}(V^2 + 1)$. In this case we still solve the system of equations (6), except that the right-hand side vector is computed by approximating a single step solution of the problem

$$g_s = g_{xx}, \quad 0 < x < \pi, \quad t - \Delta t < s < t,$$

$$g(0, s) = e^{-rs}, \quad g(\pi, s) = 0,$$

$$g(x, t - \Delta t) = u(x, t - \Delta t) + \frac{1}{2}kVu_x(x, t - \Delta t).$$

Figure 3 gives the solution profile for $t = \frac{1}{2}\pi$, using $h = \frac{1}{64}\pi$ and $k = \frac{1}{128}\pi$, with $V = -1$. For this profile, the L^2 error was approximately $3.04 \cdot 10^{-8}$, and the L^∞ error was $1.43 \cdot 10^{-4}$. Again, the error decreased by the expected factor of 4 when h and k were halved.

Example 4.4. Consider now the linear convection-diffusion problem

$$u_t = u_{xx} - Vu_x, \quad 0 < x < \infty,$$

$$u(0, t) = 1, \quad u(\infty, t) = 0,$$

$$u(x, 0) = 0,$$

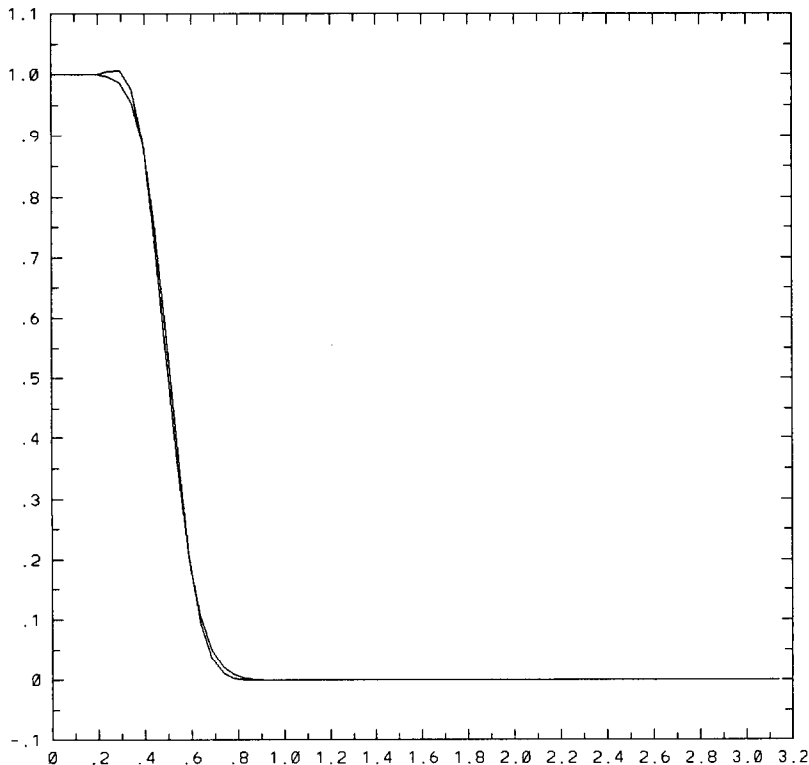


Fig. 4. Convection dominated example; $h = \frac{1}{64}\pi$, $k = \frac{1}{312}V$, $V=100$.

Table 2
 L^2 error at $t = t_{\max}$ for algorithm (A), using $h = \pi/N$,
 $k = 1/(4NV)$, $V = 100$

N	L^2 error
8	$1.26 \cdot 10^{-1}$
16	$3.40 \cdot 10^{-2}$
32	$4.03 \cdot 10^{-3}$
64	$2.74 \cdot 10^{-4}$
128	$1.03 \cdot 10^{-4}$

whose exact solution is given by

$$u(x, t) = \frac{1}{2} \operatorname{erfc}\left(\frac{x - Vt}{2\sqrt{t}}\right) + \frac{1}{2} e^{Vx} \operatorname{erfc}\left(\frac{x + Vt}{2\sqrt{t}}\right).$$

For large V this solution is essentially a traveling near-shock moving from left to right. As such, it is a good example for performance evaluation of numerical schemes for convective diffusion solvers. We used the exact solution at $x = \pi$ to provide boundary data on a finite interval, and solved the resulting problem for a sequence of values of k and h , for $1/(4V) < t \leq 1/(2V)$, using $V = 100$. Table 2 shows the L^2 error for each case. Figure 4 shows the approximate solution profile for $t = \pi/(2V)$, using the $h = \frac{1}{64}\pi$ data. The “ripple” effect is marginally less than when the same problem was solved using Crank–Nicolson with the same data.

Remark on work in progress

It appears possible to modify (A) to produce a strongly stable, second-order accurate method that requires only the solution of scalar nonlinear equations at each time step. This work was done while the present paper was in the review process, and should be published soon.

References

- [1] C. Brooks, Computational Green’s function method for nonlinear heat equations, Master’s Thesis, Univ. Georgia, 1987.
- [2] C.N. Dawson, T.F. Russell and M.F. Wheeler, Some improved error estimates for the modified method of characteristics, *SIAM J. Numer. Anal.* **26** (1989) 1487–1512.
- [3] J. Douglas and T.F. Russell, Numerical methods for convection-dominated diffusion problems based on combining the method of characteristics with finite element or finite difference methods, *SIAM J. Numer. Anal.* **19** (5) (1982) 871–885.
- [4] J. Epperson, Efficient computation of heat equation kernels, working paper.
- [5] J. Epperson, On the use of Green’s functions for approximating nonlinear parabolic PDEs, *Appl. Math. Lett.* **2** (3) (1989) 293–296.
- [6] J. Epperson, Semi-group linearization for nonlinear parabolic equations, *Numer. Methods Partial Differential Equations* **7** (1991) 147–163.