

Finite Differences and Collocation Methods for the Solution of the two Dimensional Heat Equation

Jules Kouatchou*
NASA Goddard Space Flight Center
Code 931
Greenbelt, MD 20771

Abstract

In this paper we combine finite difference approximations (for spatial derivatives) and collocation techniques (for the time component) to numerically solve the two dimensional heat equation. We employ respectively a second-order and a fourth-order schemes for the spatial derivatives and the discretization method gives rise to a linear system of equations. We show that the matrix of the system is non-singular. Numerical experiments carried out on serial computers, show the unconditional stability of the proposed method and the high accuracy achieved by the fourth-order scheme.

Key words: Heat equation, collocation methods, finite difference, fourth-order scheme.

Mathematical Subject Classification: 65M06, 65N12.

1 Introduction

We consider the two dimensional heat equation:

$$\frac{\partial u}{\partial t}(x, y, t) = \alpha^2 \left(\frac{\partial^2 u}{\partial x^2}(x, y, t) + \frac{\partial^2 u}{\partial y^2}(x, y, t) \right), \quad (x, y, t) \in \Omega \times [0, \infty) \quad (1)$$

where $\Omega = [0, 1] \times [0, 1]$, and with the initial condition

$$u(x, y, 0) = \psi(x, y), \quad (x, y) \in \Omega,$$

and the boundary conditions

$$u(0, y, t) = f_0(y, t), \quad u(1, y, t) = f_1(y, t), \quad u(x, 0, t) = g_0(x, t), \quad \text{and} \quad u(x, 1, t) = g_1(x, t) \quad \text{for} \quad t \geq 0.$$

*kouatch@pecos1.gsfc.nasa.gov, the author is also affiliated with Morgan State University and his work was supported by NASA under the grant No. NAGS-3508

We assume that f_0 , f_1 , g_0 and g_1 are smooth functions in the variable t , i.e., their first derivatives with respect to t exist and are continuous.

If (1) is discretized with standard or high-order finite difference approximations, the resulting method leads to a stability condition. For a reasonable number of mesh points in the spatial direction, this typically requires a very small time step to satisfy the stability requirement. Even if the approximations produce implicit methods, the computational complexity considerably increases, especially if high-order formulas are employed [5]. In addition, these techniques, when implemented on parallel computers can only allow the parallelization in space, i.e., at each time step, spatial grid points are partitioned and assigned to processors; the solution is then computed before we move to the next time step.

Recently, Jézéquel [3] combined the standard finite difference approximation for the spatial derivative and collocation technique for the time component to numerically solve the one dimensional heat equation. The method (called implicit collocation method) is unconditionally stable. Its principle is as follows: after discretization in space of the problem, the solution is approximated at each spatial grid point by a polynomial depending on time. The resulting derivation produces a linear system of equations. The order of the method is in space the order of difference approximation and in time the degree of the polynomial.

In this paper, we extend Jézéquel's work [3] to the two dimensional heat equation. In addition to the spatial discretization with the standard second-order formula, we also present discretization based on a fourth-order formula. For the two formulas, we show that the matrix arising from the system of equations is non-singular and we present their respective accuracies. Our numerical experiments are carried out on a serial computer.

An outline of the paper is as follows. In Section 2, we explain the basic principle behind the implicit collocation method. Section 3 presents the derivation of the system of linear equations when the fourth-order and the second-order formulas are respectively utilized for the spatial derivatives. Numerical results are given in Section 4. In Section 5, we discuss some

issues related to the implicit collocation method. Finally, some conclusions are formulated in Section 6.

2 Principle of the Implicit Collocation Technique

The idea behind the technique can be described as follows:

1. We start with a time dependent partial differential equation (PDE).
2. The PDE is discretized in space, giving rise to a system of ordinary differential equations with unknown functions at each spatial grid point.
3. The implicit collocation method consists of approximating at each spatial grid point the solution by a polynomial that depends on time. To solve the PDE with implicit collocation method is to determine the coefficients of all polynomials.
4. Depending of the PDE, we obtain a linear or non-linear system of equations (where the unknowns are the coefficients) that can be solved by a direct or iterative method.
5. Once the coefficients of the polynomials are determined, the approximated solution of the PDE is computed on a given time interval that depends on the degree of the polynomials.

One of the main advantages of the implicit collocation method is that if it is efficiently implemented on distributed memory computers, the parallelization is carried out both across time and space [3].

In the next section we use this description to derive the system of equations. The main presentation focuses on the fourth-order finite difference approximations for spatial derivatives.

3 Derivation of the System of Equations

3.1 Discretization Procedure

Let $h = 1/n$ and Δt be the spatial and time mesh-widths respectively. We can subdivide the spatial domain and consider the time step as follows:

$$\begin{aligned} x_i = ih, \quad y_j = jh, \quad i, j = 0, 1, \dots, n \\ t_k = k\Delta t, \quad k = 0, 1, \dots \end{aligned}$$

For simplicity we write the approximated solution of u and its time derivative at the spatial grid points (x_i, y_j) as:

$$U_{i,j}(t) = u(x_i, y_j, t), \quad \text{and} \quad U'_{i,j}(t) = \frac{\partial u}{\partial t}(x_i, y_j, t).$$

At any given time t , if we use the discretization of the steady state Poisson equation with a fourth-order (FO) scheme [1], we can approximate the spatial derivatives of (1). We obtain for any grid point (x_i, y_j) , $i, j = 1, \dots, n-1$:

$$\begin{aligned} & \frac{1}{2} [U'_{i+1,j}(t) + U'_{i,j+1}(t) + U'_{i-1,j}(t) + U'_{i,j-1}(t) + 8U'_{i,j}(t)] \\ = & \frac{\alpha^2}{h^2} [4(U_{i+1,j}(t) + U_{i,j+1}(t) + U_{i-1,j}(t) + U_{i,j-1}(t)) \\ & + (U_{i+1,j+1}(t) + U_{i-1,j+1}(t) + U_{i-1,j-1}(t) + U_{i+1,j-1}(t)) - 20U_{i,j}(t)], \end{aligned} \quad (2)$$

with the conditions

$$\begin{aligned} U_{i,j}(0) &= \psi(x_i, y_j), \\ U_{0,j}(t) &= f_0(y_j, t), \quad U'_{0,j}(t) = \frac{\partial f_0}{\partial t}(y_j, t), \\ U_{n,j}(t) &= f_1(y_j, t), \quad U'_{n,j}(t) = \frac{\partial f_1}{\partial t}(y_j, t), \\ U_{i,0}(t) &= g_0(x_i, t), \quad U'_{i,0}(t) = \frac{\partial g_0}{\partial t}(x_i, t), \\ U_{i,n}(t) &= g_1(x_i, t), \quad U'_{i,n}(t) = \frac{\partial g_1}{\partial t}(x_i, t). \end{aligned}$$

Eq. 2 is a system of $(n-1) \times (n-1)$ ordinary differential equations and for any value of t , it is fourth-order in space.

Remark 1 In case the spatial derivative in (1) is discretized with the standard second-order (SO) finite difference approximation, we obtain the system:

$$U'_{i,j}(t) = \frac{\alpha^2}{h^2} [U_{i+1,j}(t) + U_{i,j+1}(t) + U_{i-1,j}(t) + U_{i,j-1}(t) - 4U_{i,j}(t)]. \quad (3)$$

Here the conditions on the time derivatives on boundary points are not employed.

Now it remains to introduce the concept of implicit collocation methods in our derivations.

Let $P_{i,j}(t)$ be the polynomial of degree r satisfying the system (2) at the spatial grid point (x_i, y_j) and at times $t_k = k\Delta t$ ($k = 0, \dots, r-1$). Then for any $i, j = 1, \dots, n-1$ and $k = 0, \dots, r-1$, we have

$$P_{i,j}(t_k) = a_{i,j,r}t_k^r + a_{i,j,r-1}t_k^{r-1} + \dots + a_{i,j,1}t_k + a_{i,j,0}.$$

The coefficients $a_{i,j,0}$ are determined from the initial condition:

$$a_{i,j,0} = P_{i,j}(0) = U_{i,j}(0) = \psi(x_i, y_j).$$

To solve the system (2) by the collocation method is to determine the coefficients $a_{i,j,0}, a_{i,j,1}, \dots, a_{i,j,r}$, for $i, j = 1, \dots, n-1$. After some algebraic manipulations (see [3] for details on the one dimensional heat equation) we obtain the linear system of $r \times (n-1) \times (n-1)$ equations

$$AX = S, \quad (4)$$

where A is a block-tridiagonal matrix given by

$$A = \text{tri}[A_{j-1}, A_j, A_{j+1}]_{n-1}.$$

A_{j-1} , A_j and A_{j+1} are square matrices (with $r \times (n-1)$ rows) defined as

$$\begin{aligned} A_{j-1} &= \text{tri} \left[-E, \frac{1}{2} \frac{h^2}{\alpha^2} E' - 4E, -E \right]_{n-1}, \\ A_j &= \text{tri} \left[\frac{1}{2} \frac{h^2}{\alpha^2} E' - 4E, 4 \frac{h^2}{\alpha^2} E' + 20E, \frac{1}{2} \frac{h^2}{\alpha^2} E' - 4E \right]_{n-1}, \\ A_{j+1} &= \text{tri} \left[-E, \frac{1}{2} \frac{h^2}{\alpha^2} E' - 4E, -E \right]_{n-1}. \end{aligned}$$

The subscript $n - 1$ determines the number of block-rows. E and E' are $r \times r$ nonsymmetric

matrices

$$E = \begin{pmatrix} t_0^r & t_0^{r-1} & \cdots & t_0 \\ t_1^r & t_1^{r-1} & \cdots & t_1 \\ \vdots & \vdots & & \vdots \\ t_{r-1}^r & t_{r-1}^{r-1} & \cdots & t_{r-1} \end{pmatrix}, \quad E' = \begin{pmatrix} rt_0^{r-1} & (r-1)t_0^{r-2} & \cdots & 2t_0 & 1 \\ rt_1^{r-1} & (r-1)t_1^{r-2} & \cdots & 2t_1 & 1 \\ \vdots & \vdots & & \vdots & \vdots \\ rt_{r-1}^{r-1} & (r-1)t_{r-1}^{r-2} & \cdots & 2t_{r-1} & 1 \end{pmatrix},$$

the vector X of $r \times (n - 1) \times (n - 1)$ unknowns and the right hand side S are

$$X = \begin{pmatrix} \begin{pmatrix} a_{1,1,r} \\ \vdots \\ a_{1,1,1} \end{pmatrix} \\ \begin{pmatrix} a_{2,1,r} \\ \vdots \\ a_{2,1,1} \end{pmatrix} \\ \vdots \\ \begin{pmatrix} a_{n-2,n-1,r} \\ \vdots \\ a_{n-2,n-1,1} \end{pmatrix} \\ \begin{pmatrix} a_{n-1,n-1,r} \\ \vdots \\ a_{n-1,n-1,1} \end{pmatrix} \end{pmatrix},$$

$$S = \begin{pmatrix} \begin{pmatrix} 4(a_{2,1,0} + a_{1,2,0}) + a_{2,2,0} - 20a_{1,1,0} + 4(U_{1,0}(t_0) + U_{0,1}(t_0)) \\ + U_{0,2}(t_0) + U_{0,0}(t_0) + U_{2,0}(t_0) - \frac{1}{2} \frac{h^2}{\alpha^2} (U'_{1,0}(t_0) + U'_{0,1}(t_0)) \\ \vdots \\ 4(a_{2,1,0} + a_{1,2,0}) + a_{2,2,0} - 20a_{1,1,0} + 4(U_{1,0}(t_{r-1}) + U_{0,1}(t_{r-1})) \\ + U_{0,2}(t_0) + U_{0,0}(t_{r-1}) + U_{2,0}(t_{r-1}) - \frac{1}{2} \frac{h^2}{\alpha^2} (U'_{1,0}(t_{r-1}) + U'_{0,1}(t_{r-1})) \\ \vdots \\ \begin{pmatrix} 4(a_{i+1,j,0} + a_{i,j+1,0} + a_{i-1,j,0} + a_{i,j-1,0}) + a_{i+1,j+1,0} \\ + a_{i-1,j+1,0} + a_{i-1,j-1,0} + a_{i+1,j-1,0} - 20a_{i,j,0} \\ \vdots \\ 4(a_{i+1,j,0} + a_{i,j+1,0} + a_{i-1,j,0} + a_{i,j-1,0}) + a_{i+1,j+1,0} \\ + a_{i-1,j+1,0} + a_{i-1,j-1,0} + a_{i+1,j-1,0} - 20a_{i,j,0} \\ \vdots \end{pmatrix} \\ \begin{pmatrix} 4(a_{n-2,n-1,0} + a_{n-1,n-2,0}) + a_{n-2,n-2,0} - 20a_{n-1,n-1,0} + 4(U_{n-1,n}(t_0) + U_{n,n-1}(t_0)) \\ + U_{n,n-2}(t_0) + U_{n,n}(t_0) + U_{n-2,n}(t_0) - \frac{1}{2} \frac{h^2}{\alpha^2} (U'_{n-1,n}(t_0) + U'_{n,n-1}(t_0)) \\ \vdots \\ 4(a_{n-2,n-1,0} + a_{n-1,n-2,0}) + a_{n-2,n-2,0} - 20a_{n-1,n-1,0} + 4(U_{n-1,n}(t_{r-1}) + U_{n,n-1}(t_{r-1})) \\ + U_{n,n-2}(t_{r-1}) + U_{n,n}(t_{r-1}) + U_{n-2,n}(t_{r-1}) - \frac{1}{2} \frac{h^2}{\alpha^2} (U'_{n-1,n}(t_{r-1}) + U'_{n,n-1}(t_{r-1})) \end{pmatrix} \end{pmatrix}$$

Remark 2 *With the SO scheme, the matrix A is still block-tridiagonal and*

$$\begin{aligned} A_{j-1} &= \text{diag}[-E]_{n-1}, \\ A_j &= \text{tri} \left[-E, \frac{h^2}{\alpha^2} E' + 4E, -E \right]_{n-1}, \\ A_{j+1} &= \text{diag}[-E]_{n-1}. \end{aligned}$$

The vector of unknowns remains the same and the right hand side S does not involve time derivatives at boundary points.

Remark 3 *A is a matrix with bandwidth equal to $(2n - 2)r$ and $(2n + 1)r$ for the SO and FO spatial schemes respectively. The block structure of the matrix A for the SO or FO scheme, is similar to the one obtained from the discretization of the two dimensional steady-state Poisson equation with the SO or FO scheme. In the latter, instead of having the block matrices E and E' , we have constant coefficients.*

Remark 4 *To obtain the solution, the coefficients $a_{i,j,k}$ ($i, j = 1, \dots, n - 1$ and $k = 1, \dots, r$) are first evaluated and then the approximated values $U_{i,j}(t_k)$ ($i, j = 1, \dots, n - 1$ and $k = 1, \dots, r - 1$) are calculated. Each of these two steps can be carried out in parallel. In addition, the collocation method does not only consist of determining the $U_{i,j}(t_k)$ only, but also calculates coefficients of polynomials approaching the solution in the interval $[t_0, t_{r-1}]$.*

3.2 Non-Singularity of the Matrix A

If we focus on the FO spatial approximation, Eq. 4 has a solution only if the matrix A is non-singular. To determine the non-singularity of A , it is enough to study its block-diagonal elements.

Let $D = 4\frac{h^2}{\alpha^2}E' + 20E$ be the block-diagonal of the matrix A_i defined above. We can state the following theorem.

Theorem 1 *The determinant of the matrix D is zero if and only if two values t_{l-1} and t_{m-1} are identical.*

Proof: D is a $r \times r$ matrix where

$$D_{i,j} = \left(4 \frac{h^2}{\alpha^2} (r+1-j) + 20t_{i-1} \right) t_{i-1}^{r-j} \geq 0, \quad i, j = 1, \dots, r.$$

For a given i , it is impossible that $D_{i,j} = 0$ for all values of j . Indeed the coefficients of the last column of the matrix D are all non zero: $D_{i,r} = 4 \frac{h^2}{\alpha^2} + 20t_{i-1} \neq 0$. D can be seen as a Vandermonde-like matrix. Since all the t_{i-1} are all different, D is a non-singular matrix. \square

Conjecture 1 *The determinant of the matrix A is non zero and the linear system (4) has a unique solution.*

The same conclusion can be obtained if we employ the SO spatial scheme.

4 Numerical Experiments

Consider Eq. 1 with the conditions

$$\alpha = \frac{1}{\pi}, \quad u(x, y, 0) = \sin \pi x + \sin \pi y,$$

$$f_0(y, t) = f_1(y, t) = e^{-t} \sin \pi y, \quad g_0(x, t) = g_1(x, t) = e^{-t} \sin \pi x.$$

The exact solution is given by $u(x, y, t) = (\sin \pi x + \sin \pi y)e^{-t}$.

For the second-order (SO) and the fourth-order (FO) spatial discretizations respectively, the implicit collocation technique was implemented on an SGI O2 workstation in Fortran 77. To solve the linear system of equations, we used the decomposition algorithm for inverting asymmetric and indefinite matrices proposed by Luo [4]. The method is easy to program, requires only the storage of the matrix and the right hand side of Eq. 4 and was chosen for its parallelization potential.

To test the accuracy and stability of the implicit collocation technique, we present for $\Delta t = 0.1$ and $\Delta t = 0.01$ and for different values of r and n , the absolute maximum error (obtained by comparing the true solution with the approximated one) achieved in the interval $[0, (r-1)\Delta t]$. We compare errors obtained with the second-order (SO) and the fourth-order

(FO) spatial schemes. We do not report the timing results because the two schemes give the same elapsed time. This is due to the fact that in the implementation of Luo's algorithm, we did not take advantage of the respective matrix bandwidths; even if we did, the difference in elapsed times would be negligible since their bandwidths are comparable.

To provide an idea on how large the system of equations is, we first present in Table 1, the dimension of the matrix A in (4) for different values of n and r .

n	$r = 3$	$r = 4$	$r = 5$
4	27×27	36×36	45×45
8	147×147	196×196	245×245
16	675×675	900×900	1125×1125

Table 1: Dimension of the matrix A for different values of n and r .

We report in Tables 2, 3 and 4 absolute maximum errors for $r = 3, 4, 5$ respectively when the spatial mesh-width varies. The maximum error in the interval $[0, (r - 1)\Delta t]$ was

n	$\Delta t = 0.1$		$\Delta t = 0.01$	
	SO	FO	SO	FO
4	1.61(-2)	5.35(-4)	1.97(-3)	6.27(-5)
8	4.14(-3)	3.33(-5)	5.01(-4)	3.90(-6)
16	1.04(-3)	1.89(-6)	1.25(-4)	2.43(-7)

Table 2: Maximum error obtained with the second-order (SO) and fourth-order (FO) spatial schemes for different values of Δt when $r = 3$.

obtained for $t = (r - 1)\Delta t$ and the error increases as t does within the interval. This result is consistent with the one achieved by Jézéquel while solving the one dimensional heat equation using the second order scheme and collocation technique [3].

We observe that for given r and Δt , SO indeed produces solutions of second-order accuracy and FO of fourth-order accuracy. In addition, with FO the accuracy is far more better. These findings are consistent with the ones obtained by Gupta *et al.* [2] and Zhang *et al.* [6] while solving the steady state Poisson equation with the two schemes.

n	$\Delta t = 0.1$		$\Delta t = 0.01$	
	SO	FO	SO	FO
4	2.14(-2)	7.10(-4)	2.93(-3)	9.35(-5)
8	5.52(-3)	4.28(-5)	7.44(-4)	5.80(-6)
16	1.39(-3)	2.16(-6)	1.86(-4)	3.61(-7)

Table 3: Maximum error obtained with the second-order (SO) and fourth-order (FO) spatial schemes for different values of Δt when $r = 4$.

n	$\Delta t = 0.1$		$\Delta t = 0.01$	
	SO	FO	SO	FO
4	2.51(-2)	8.34(-4)	3.86(-3)	1.23(-4)
8	6.49(-3)	5.10(-5)	9.82(-4)	7.66(-6)
16	1.63(-3)	3.18(-6)	2.46(-4)	4.76(-7)

Table 4: Maximum error obtained with the second-order (SO) and fourth-order (FO) spatial schemes for different values of Δt when $r = 5$.

5 Discussion

5.1 Degree of the Polynomials

The order of the implicit collocation technique is the order of the difference scheme in space and the order of the polynomial in time (namely r). The question that arises is how do we choose r in order to obtain high accurate approximated solutions of (1)? By increasing the value of r , we do not only increase the length of the time interval where the solution is to be found but also the size of the linear system of equations. In [3], Jézéquel studied the numerical validity of the coefficients of the polynomials. She defined the optimal degree r_{opt} of the polynomials to be the highest integer for which all the coefficients of the polynomials and the approximated solution remain significant. She found that the degree r_{opt} increases as the time and spatial mesh sizes increase but r_{opt} is not arbitrary large.

5.2 Merit of the Method

Here we attempt to computationally compare the implicit collocation method and other standard implicit methods (arising from classical finite difference approximations in both space and time).

Assume that the spatial domain has $(n-1) \times (n-1)$ interior grid points. The derivations with a standard implicit method (SI) produces a system of equations with $N_{si} = (n-1) \times (n-1)$ unknowns whereas with the implicit collocation (IC) method, we obtain a system with $N_{ic} = r \times (n-1) \times (n-1)$ unknowns, where r is the degree of the polynomials.

Let Δt_{ic} be the time step used for IC. The solution using IC, can be determined at any point in the interval $[0, (r-1)\Delta t_{ic}]$. Let $(r-1)m$ be the number of equidistant points where the solution is to be computed in this time interval. To determine the solution at the same points of the interval with SI, $(r-1)m$ time iterations must be carried out. The corresponding time step is $\Delta t_{si} = \Delta t_{ic}/m$.

The implementation of IC requires the solution of a linear system of N_{ic} equations; its computational cost is $C_{ic} \approx N_{ic}^3$. For SI, we need to find the solution of a linear system of N_{si} equations $(r-1)m$ times; the cost is then $C_{si} \approx (r-1)mN_{si}^3$. If we assume that C_{ic} and C_{si} are equal, then $r^3 = (r-1)m$ or $m = r^2 + r + 1/(r-1)$. We can conclude that if $m \leq r^2 + r + 2$, then SI is computationally less expensive than IC, and if $m > r^2 + r + 2$, IC is cheaper.

To summarize, the implicit collocation method is beneficial with respect to standard implicit methods, if for given r and Δt , the number of equidistant time points in the interval $[0, (r-1)\Delta t]$ is at least greater than $r^3 + r$.

6 Conclusions

We have carried out the numerical approximation of the two dimensional heat equation by using finite difference schemes (for spatial derivatives) and implicit collocation technique (for

time component). The proposed method is unconditionally stable. We employed second-order and a fourth-order spatial schemes respectively and showed that both have the same computational complexity and that the fourth-order one clearly produces more accurate solutions.

The main advantage of the implicit collocation technique is not only its stability condition but also the fact that it can be implemented on distributed memory computers where the parallelization strategy is performed both across time and space. In future works, we plan to implement the method on parallel computers and to extend our analysis to the three dimensional heat equation.

Acknowledgment: I am grateful to Fabienne Jézéquel and Jun Zhang for their useful advice. I would like to also thank Milton Halem, Nancy Palm and Eugene DeLoatch for their support.

References

- [1] M.M. Gupta, A fourth-order Poisson solver, *J. Comp. Phys.*, **55**, p. 166-172 (1984).
- [2] M.M. Gupta, J. Kouatchou and J. Zhang, Comparison of second and fourth order discretizations for multigrid Poisson solvers, *J. Comp. Phys.*, **132**, p. 226-232 (1997).
- [3] F. Jézéquel, A validated parallel across time and space solution of the heat transfer equation, *Applied Numerical Mathematics*, **31**, p. 65-79 (1999).
- [4] J. Luo, A new class of decomposition for inverting asymmetric and indefinite matrices, *Computers Math. Applic.*, **24** (4), p. 95-104 (1993).
- [5] W.F. Spitz and G.F. Carey, Extension of high-order compact schemes to time dependent problems, submitted for publication.

- [6] J. Zhang, J. Kouatchou and M. Othman, On cyclic reduction and finite difference schemes, submitted for publication.