A Comparative Study of Compact Finite Volume Methods for the 2-D Diffusion Equation with Finite Difference ADI and SOR*

B. N. Borah1, R. E. White2, A. J. Kyrillidis1, S. Sankarlingham1

1. NC State University, Raleigh, NC
2. NC A&T State University, Greensboro, NC

Abstract

Recently developed Compact Finite Difference scheme (CPT) is applied to two dimensional diffusion equations. The relative merits of CPT-ADI are investigated with other computational schemes such as finite difference method - ADI (FDM-ADI) and FDM-SOR. The numerical results obtained from these three approaches are compared to known analytical solutions. The primary interest of this study lies on vectorization and parallel processing. According to our results shown in tables 1 CPT-ADI is found to be superior scheme with regards to accuracy, than both FDM-ADI and FDM-SOR. It is also fastest algorithm than both FDM-ADI and FDM-SOR as it is evident from CPU times.

1. Introduction

We shall describe briefly the compact finite difference method (CPT) for one dimensional steady state problem. The extension to 2-D problem may be easily done. The underlying physics behind this approach lies in solving the differential equation in isolation from its neighboring subintervals (i.e. compactly). Then, extend the solution in the large by means of continuity conditions for the flux and temperature across the boundaries of the contiguous subintervals (See Rose [1]).

We consider here one dimensional steady diffusion problem:

(1) \[ D \cdot E \cdot u' = f \]
(2) \[ f' = u \]
(3) \[ B. C. \quad f = g \]

for \( X \in I \) and \( I = [x_-, x_+] \)

Divide the interval \( I \) into \( m \) nonoverlapping subintervals:

\[ I_j = \left\{ x \mid \frac{x_j - 1/2}{2} \leq x \leq \frac{x_j + 1/2}{2} \right\} \]

with center points \( x_j \), \( j = 1/2, 3/2, \ldots, M - 1/2 \) and interior endpoints \( x_j \), \( j = 1, 2, \ldots, M - 1 \). We shall adopt the finite difference notations \( \Delta x_j = x_j + 1/2 - x_j - 1/2 \), \( h_j = \Delta x/2 \) and \( u(i) = u_i \). We also denote:

- \( I_c = \{ 1/2, 3/2, \ldots, M - 1/2 \} \) (center points)
- \( I_e = \{ 1, 2, \ldots, M - 1 \} \) (interior endpoints)

Fig. 1.

The discrete equations for (1) and (2) can be written as, for \( j \in I_c \),

(4) \[ u_j + \frac{1}{2} - u_j - \frac{1}{2} = \Delta x_j f_j \]

(5a) \[ \phi_j + \frac{1}{2} - \phi_j - \frac{1}{2} = h_j u_j + \frac{1}{2} \]

and

(5b) \[ \phi_j - \phi_j - \frac{1}{2} = h_j u_j - \frac{1}{2} \]

Next, it is required that both \( u \) and \( f \) be continuous across every endpoint common to two intervals:

\[ [u]_i = [\phi]_i = 0 \quad \text{for} \quad i \in I_e. \]
Using this continuity condition, \([u]\) = 0 in terms of \(\phi\) we have

\[
\phi_i + \Phi_i - \Phi_i - \frac{1}{h_i} \cdot \frac{1}{2} = \frac{\phi_i + \frac{1}{h_i} - \phi_i}{\frac{1}{2}}
\]

and in the case of equal intervals, which reduces to

\[
\phi_i = \frac{\phi_i + \frac{1}{h_i} + \phi_i - \frac{1}{h_i}}{2}, \quad i \in I_c \text{ (endpoints)}
\]

Now from (4) and (5) we get,

\[
\begin{align*}
\phi_j + \frac{1}{2} - &\phi_j + \frac{1}{2} = 2 h_j^2 f_j, \quad i \in I_c \\
\phi_j + \frac{1}{2} - &\phi_j + \frac{1}{2} = 0, \quad i \in I_c
\end{align*}
\]

The equations (7) and the boundary conditions lead to determined system of algebraic equations for the values of \(\phi\). These equations lead to a tridiagonal system of equations which is, therefore, solved by Thomas algorithm.

The extension to two dimensional scheme may be easily done. Two dimensional stencil is shown in Fig. 2.

Fig. 2.

Consider the two dimensional diffusion equation \(u_t = u_{xx} + u_{yy}\). The compact ADI method is given by equations (8) - (10):

\[
\begin{align*}
\frac{u_i^{n+\frac{1}{2}} - u_{i+1,j}^{n+\frac{1}{2}}}{\tau} &= F_{i,j}^{n+\frac{1}{2}} + G_{i,j}^{n+\frac{1}{2}} \\
\frac{u_{i,j}^{n+\frac{1}{2}} - u_{i+1,j}^{n+\frac{1}{2}}}{\tau} &= F_{i,j}^{n+\frac{1}{2}} + G_{i,j}^{n+\frac{1}{2}}
\end{align*}
\]

where \(F_{i,j}\) and \(G_{i,j}\) are standard finite difference expressions for \(u_{xx}\) and \(u_{yy}\) respectively and \(\tau = \Delta t/2\). Using the standard finite difference scheme we get the following equations

\[
\begin{align*}
-u_{i-1,j}^{n+\frac{1}{2}} + (\alpha + 2) u_{i,j}^{n+\frac{1}{2}} - u_{i+1,j}^{n+\frac{1}{2}} &= 0, \\
-u_{i,j-1}^{n+\frac{1}{2}} + (\alpha + 2) u_{i,j}^{n+\frac{1}{2}} - u_{i,j+1}^{n+\frac{1}{2}} &= 0
\end{align*}
\]

\(i, j = 1, 3, \ldots, 2 M - 1, \quad \alpha = 4 (\Delta x)^2, \quad \Delta x = \Delta y / \Delta t\)

\[
\begin{align*}
-u_{i,j}^{n+\frac{1}{2}} + (\alpha + 2) u_{i,j}^{n+\frac{1}{2}} - u_{i+1,j}^{n+\frac{1}{2}} &= 0, \\
-u_{i,j}^{n+\frac{1}{2}} + (\alpha + 2) u_{i,j}^{n+\frac{1}{2}} - u_{i-1,j}^{n+\frac{1}{2}} &= 0
\end{align*}
\]

and the continuity conditions are

\[
\begin{align*}
-u_{i-1,j}^{n+\frac{1}{2}} + 2 u_{i,j}^{n+\frac{1}{2}} - u_{i+1,j}^{n+\frac{1}{2}} &= 0, \\
-u_{i,j-1}^{n+\frac{1}{2}} + 2 u_{i,j}^{n+\frac{1}{2}} - u_{i,j+1}^{n+\frac{1}{2}} &= 0
\end{align*}
\]

The two algebraic systems (9), (9a), and (10), (10a) can be solved by the tridiagonal algorithm.

2. Numerical experiments

One of the primary objectives of this project is to vectorization of 1-D and 2-D diffusion problems. We vectorize the 2-D heat equation for the Cray Y-MP using red-black SOR algorithm.

We consider one analytical solutions for the following diffusion equation and compare them with respective
numerical solutions. The results are recorded in tables 1.

The two dimensional diffusion equation is 
\[ u_t = u_{xx} + u_{yy} + f(x,y,t), \quad 0 < x < 1, \quad 0 < y < 1, \quad t > 0. \]
The following analytical solutions are considered:

\[ u(x,y,t) = 100(x(1-x))(y(y-1))e^{-t}, \quad f(x,y,t) = 200(y^2 + x^2) - 100(xy^2 - x^2 e^{-t}) \]

All the problems mentioned above are associated with Dirichlet boundary conditions.
The CPT-ADI is compared with FDM-ADI and FDM-SOR. The functional errors in the method is O(h^2) as expected and derivative error is of O(h). The result is shown in table 1.

### FDM-ADI \( \Delta t = \Delta x^2 \)

<table>
<thead>
<tr>
<th>No. of Hx=Hy cells</th>
<th>No. of iters</th>
<th>CPU Time (sec)</th>
<th>Max Fct Error</th>
<th>Max Derivative Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>128</td>
<td>0.76</td>
<td>4.592E-03</td>
<td>1.01704E-02</td>
</tr>
<tr>
<td>32</td>
<td>512</td>
<td>6.15</td>
<td>1.1469E-03</td>
<td>5.24588E-02</td>
</tr>
<tr>
<td>64</td>
<td>2048</td>
<td>49.48</td>
<td>2.8667E-04</td>
<td>2.66432E-02</td>
</tr>
</tbody>
</table>

### CPT-ADI \( \Delta t = \Delta x \)

<table>
<thead>
<tr>
<th>No. of Hx=Hy cells</th>
<th>No. of iters</th>
<th>CPU Time (sec)</th>
<th>Max Fct Error</th>
<th>Max Derivative Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>16</td>
<td>1.183</td>
<td>5.356E-03</td>
<td>5.33550E-03</td>
</tr>
<tr>
<td>32</td>
<td>32</td>
<td>9.26</td>
<td>1.3391E-03</td>
<td>1.3390E-03</td>
</tr>
<tr>
<td>64</td>
<td>64</td>
<td>73.23</td>
<td>3.3479E-03</td>
<td>3.3477E-03</td>
</tr>
</tbody>
</table>

### FDM-SOR \( \Delta t = \Delta x \)

<table>
<thead>
<tr>
<th>No. of Hx=Hy cells</th>
<th>No. of iters</th>
<th>CPU Time (sec)</th>
<th>Max Fct Error</th>
<th>Max Derivative Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>240 ( \omega = 1.65 )</td>
<td>2.10</td>
<td>3.5050E-03</td>
<td>5.869E-01</td>
</tr>
<tr>
<td>32</td>
<td>960 ( \omega = 1.80 )</td>
<td>36.73</td>
<td>2.155E-03</td>
<td>2.940E-01</td>
</tr>
<tr>
<td>64</td>
<td>3392 ( \omega = 1.85 )</td>
<td>473.18</td>
<td>9.5737E-04</td>
<td>1.469E-01</td>
</tr>
</tbody>
</table>

3. Concluding remarks

The x, y domains are divided into 8, 16, 32 and 64 cells and in each case, the starting time is taken to be 0 and the final time is 1. The relation between time step and space step varies among FDM-ADI, FDM-SOR and CPT-ADI schemes. We assume all the material constants are equal to unity and \( \Delta x = \Delta y \). In case of FDM-SOR and CPT-ADI, \( \Delta x = \Delta y = \Delta t \) and however in case of FDM-ADI, \( \Delta t = (\Delta x)^2 \).

This is the disadvantage for FDM-ADI. For example, when \( \Delta x = 1.5625E-2, \Delta t = 4.883E-4 \) and it would take 2048 iterations to reach time 1. However, CPT-ADI and FDM-SOR are free from this difficulty. But SOR also has a different disadvantage. Each time step, FDM-SOR takes large number of iterations to converge to a preassigned level. With 64 cells CPT-ADI requires only 64 iterations to reduce the error level to 3.34792E-03, where as FDM-ADI requires 2048 iterations to reduce the error level to 2.86671E-04 and FDM-SOR requires 3,392 iterations to reduce the error level to 9.57374E-04.

As CPT algorithm requires more points evaluation per iteration, it is obvious that CPT requires more CPU time than FDM-SOR or FDM-ADI.

Acknowledgments: Thanks to M. E. Rose for his many valuable suggestions for the completion of this paper. Also many thanks go to A. Nachman of AFOSR.

References