Numerical Solutions to PDEs
Lecture Notes #13
— Systems of PDEs in Higher Dimensions —
The Alternating Direction Implicit Method

Peter Blomgren,
⟨blomgren.peter@gmail.com⟩

Department of Mathematics and Statistics
Dynamical Systems Group
Computational Sciences Research Center
San Diego State University
San Diego, CA 92182-7720
http://terminus.sdsu.edu/

Spring 2017
Outline

1. Recap
   - Previously...

2. The Alternating Direction Implicit Method
   - Introduction
   - Crank-Nicolson / ADI on a 2D Square

3. ADI Algorithms
   - Peaceman-Rachford
   - D’Yakonov
   - Boundary Conditions for ADI Schemes

4. Implementing ADI Methods
   - Peaceman-Rachford
   - The Mitchell-Fairweather Scheme
   - Mixed ($u_{xy}$) Derivative Terms
Previously

We started looking at multi-dimensional hyperbolic and parabolic problems, first via vector-valued problems with one time and one space dimension, and then to full multi-space dimensional problems.

In terms of definitions, nothing much changed — the concepts of convergence, consistency, stability and order of accuracy are the same.

However, some of the analysis becomes quite challenging. — For instance, we end up needing to bound $n$th powers of amplification matrices $\|G^n\| \leq C_T$.

In order to be able to say anything useful we have to make simplifying assumptions, e.g simultaneous diagonalizability.

We looked at time-split schemes as a practical way to route around some (size / complexity) of the computational challenges. (Stability and Boundary Conditions are a different story...)

Peter Blomgren, (blomgren.peter@gmail.com) Systems of PDEs in $n$D: The ADI Method — (3/22)
The Alternating Direction Implicit (ADI) method is particularly useful for solving **parabolic equations** on rectangular domains, but can be generalized to other situations.

Given a parabolic equation, \( u_t = \nabla \circ (B \nabla u) \),

\[
\begin{bmatrix}
\frac{\partial}{\partial x} & \frac{\partial}{\partial y}
\end{bmatrix}
\begin{bmatrix}
b_{11} & b_{12} \\
b_{12} & b_{22}
\end{bmatrix}
\begin{bmatrix}
\frac{\partial}{\partial x} \\
\frac{\partial}{\partial y}
\end{bmatrix}
u = b_{11}u_{xx} + 2b_{12}u_{xy} + b_{22}u_{yy},
\]

for which \( b_{11}, b_{22} > 0 \) and \( b_{12}^2 < b_{11} \cdot b_{22} \) for parabolicity; and constant (for now).

Initially, we will consider the case \( b_{12} = 0 \) (no mixed derivative), on a square domain...
Crank-Nicolson on a Square

If we use the Crank-Nicolson schemes (for 2 spatial dimensions), we end up having to invert a penta-diagonal matrix in each iteration:

**Figure:** [LEFT] The matrix which must be inverted in each Crank-Nicolson iteration. If we trade storage of the LU-factorization [CENTER, RIGHT] for speed, then here with $6 \times 6$ interior points, we end up needing more than 4 times the storage. For $100 \times 100$ interior points, the requirement jumps from 49,600 matrix entries, to just over 2,000,000 (a factor of 40). The band-width grows linearly in $n$, and the LU-factorization fills in the whole bandwidth. In 3D the story gets even worse — with $n \times n \times n$ interior points, the bandwidth is $n^2$...
If we use the Crank-Nicolson schemes (for 3 spatial dimensions), we end up having to invert a hepta-diagonal matrix in each iteration:

Figure: [Left] The matrix which must be inverted in each Crank-Nicolson iteration. If we trade storage of the LU-factorization [Center, Right] for speed, then here with $6 \times 6 \times 6$ interior points, we end up needing more than 10 times the storage. For $20^3$ ($30^3$) interior points, the requirement jumps from 53,600 (183,600) matrix entries, to just over 6,000,000 (47,000,000) — a factor of 114 (256). The band-width grows quadratically $O(n^2)$, and the LU-factorization fills in the whole bandwidth. $\text{LU time}_{\text{Matlab}} = 8.5s$ (143.6s).
The ADI method reduces an $n$-dimensional problem to a sequence of $n$ one-dimensional problems. We here present the idea in 2D...

Let $A_1$ and $A_2$ be two linear operators, e.g.

$$A_1 u = b_1 \frac{\partial^2}{\partial x^2} u, \quad A_2 u = b_2 \frac{\partial^2}{\partial y^2} u.$$ 

For the argument to make sense, we must require that we have efficient (convenient) ways of solving the equations

$$w_t = A_i w, \quad i = 1, 2,$$

with $A_1$, and $A_2$ as above and a Crank-Nicolson step, these solutions are given by inversion of tri-diagonal matrices.
The ADI method will give us a way to solve the combined equation

\[ u_t = A_1 u + A_2 u, \]

using the available 1D-solvers as building blocks.

Crank-Nicolson applied to the combined equation gives us

\[
\frac{u^{n+1} - u^n}{k} = \frac{1}{2} \left[ A_1 u^{n+1} + A_1 u^n \right] + \frac{1}{2} \left[ A_2 u^{n+1} + A_2 u^n \right] + O(k^2).
\]

Which, with some rearrangement can be written

\[
\left[ I - \frac{k}{2} A_1 - \frac{k}{2} A_2 \right] u^{n+1} = \left[ I + \frac{k}{2} A_1 + \frac{k}{2} A_2 \right] u^n + O(k^3).
\]
Now, we notice that

\[(1 \pm A_1)(1 \pm A_2) = 1 \pm A_1 \pm A_2 + A_1 A_2.\]

By adding and subtracting $k^2 A_1 A_2 u^{[*]}$ on both sides of the Crank-Nicolson expression we get

\[
\left[ I - \frac{k}{2} A_1 - \frac{k}{2} A_2 + \frac{k^2}{4} A_1 A_2 \right] u^{n+1} = \left[ I + \frac{k}{2} A_1 + \frac{k}{2} A_2 + \frac{k^2}{4} A_1 A_2 \right] u^n \\
+ \frac{k^2}{4} A_1 A_2 \left[ u^{n+1} - u^n \right] + O \left( k^3 \right). 
\]
We can factor this, and use the fact that $u^{n+1} = u^n + \mathcal{O}(k)$ to embed the last term on the right-hand-side into the $\mathcal{O}(k^3)$-term:

$$
\begin{bmatrix}
  I - \frac{k}{2} A_1 \\
  I - \frac{k}{2} A_2
\end{bmatrix}
\begin{bmatrix}
  u^{n+1}
\end{bmatrix}
= 
\begin{bmatrix}
  I + \frac{k}{2} A_1 \\
  I + \frac{k}{2} A_2
\end{bmatrix}
\begin{bmatrix}
  u^n + \mathcal{O}(k^3)
\end{bmatrix}.
$$

Now, if we want to advance the solution numerically, we can discretize this equation, and here when $A_1 = b_1 u_{xx}$, $A_2 = b_2 u_{yy}$, the matrices corresponding to $I - k/2 A_i$ will be tridiagonal and can be inverted quickly using the Thomas algorithm.

We get the discretized ADI scheme

$$
\begin{bmatrix}
  I - \frac{k}{2} A_{1,h} \\
  I - \frac{k}{2} A_{2,h}
\end{bmatrix}
\begin{bmatrix}
  v^{n+1}
\end{bmatrix}
= 
\begin{bmatrix}
  I + \frac{k}{2} A_{1,h} \\
  I + \frac{k}{2} A_{2,h}
\end{bmatrix}
\begin{bmatrix}
  v^n
\end{bmatrix}.
$$
There are several approaches to solving the ADI scheme, one commonly used approach is the Peaceman-Rachford algorithm, which also explain the origin of the name alternating direction implicit method:

\[
\begin{align*}
\left[ I - \frac{k}{2} A_{1,h} \right] v^{n+1/2} &= \left[ I + \frac{k}{2} A_{2,h} \right] v^n, \\
\left[ I - \frac{k}{2} A_{2,h} \right] v^{n+1} &= \left[ I + \frac{k}{2} A_{1,h} \right] v^{n+1/2}.
\end{align*}
\]

In the first half-step, the $x$-direction is implicit, and the $y$-direction explicit, and in the second half-step the roles are reversed.

Is this scheme equivalent to the ADI scheme we derived?!? — It looks quite different!
We have,
\[
\begin{bmatrix}
I - \frac{k}{2} A_1, h
\end{bmatrix} v^{n+1/2} = \begin{bmatrix}
I + \frac{k}{2} A_2, h
\end{bmatrix} v^n,
\]
\[
\begin{bmatrix}
I - \frac{k}{2} A_2, h
\end{bmatrix} v^{n+1} = \begin{bmatrix}
I + \frac{k}{2} A_1, h
\end{bmatrix} v^{n+1/2}.
\]

Hence,
\[
\begin{bmatrix}
I - \frac{k}{2} A_1, h
\end{bmatrix} \begin{bmatrix}
I - \frac{k}{2} A_2, h
\end{bmatrix} v^{n+1} = \begin{bmatrix}
I - \frac{k}{2} A_1, h
\end{bmatrix} \begin{bmatrix}
I + \frac{k}{2} A_1, h
\end{bmatrix} v^{n+1/2}
\]
\[
= \begin{bmatrix}
I + \frac{k}{2} A_1, h
\end{bmatrix} \begin{bmatrix}
I - \frac{k}{2} A_1, h
\end{bmatrix} v^{n+1/2} = \begin{bmatrix}
I + \frac{k}{2} A_1, h
\end{bmatrix} \begin{bmatrix}
I + \frac{k}{2} A_2, h
\end{bmatrix} v^n.
\]

Note that we do not need $A_1, h A_2, h = A_2, h A_1, h$ for this to hold.
The D’Yakonov scheme is a direct splitting of the ADI scheme we originally derived:

\[
\begin{align*}
\left[ I - \frac{k}{2} A_1, h \right] v^{n+1/2} &= \left[ I + \frac{k}{2} A_1, h \right] \left[ I + \frac{k}{2} A_2, h \right] v^n \\
\left[ I - \frac{k}{2} A_2, h \right] v^{n+1} &= v^{n+1/2},
\end{align*}
\]

Other ADI-type schemes can be derived starting with other basic schemes (we worked from Crank-Nicolson), e.g. the **Douglas-Rachford** method (Strikwerda pp. 175–176) is derived based on backward-time central-space.
Boundary Conditions for ADI Schemes

Here, we consider Dirichlet boundary conditions \( u = \beta(t,x,y) \) specified at the boundary, in the context of the Peaceman-Rachford scheme

\[
\begin{align*}
\left[ I - \frac{k}{2} A_{1,h} \right] v^{n+1/2} & = \left[ I + \frac{k}{2} A_{2,h} \right] v^n, \\
\left[ I - \frac{k}{2} A_{2,h} \right] v^{n+1} & = \left[ I + \frac{k}{2} A_{1,h} \right] v^{n+1/2}.
\end{align*}
\]

The correct boundary conditions for the half-step quantity is given by

\[
v^{n+1/2} = \frac{1}{2} \left[ I + \frac{k}{2} A_{2,h} \right] \beta^n + \frac{1}{2} \left[ I - \frac{k}{2} A_{2,h} \right] \beta^{n+1}.
\]

Where did that come from?!? — Flip the second equation in the scheme, add the two, and solve for \( v^{n+1/2} \)... And it makes sense, “half” the condition comes from the past, and “half” from the future.
We consider Peaceman-Rachford on a grid, where
\((x_\ell, y_m) = (\ell \Delta x, m \Delta y), \ell = 0, \ldots, L, m = 0, \ldots, M\). We let
\(\mu_x = k/\Delta x^2, \mu_y = k/\Delta y^2\). Further, we let \(v_{\ell,m}\) denote the full-step quantity, and \(w_{\ell,m}\) denote the half-step quantity; if we are not interested in saving the results for all \(t = kn\), we can overwrite these quantities...

We get, the first half-stage

\[- \left[ \frac{b_1 \mu_x}{2} \right] w_{\ell-1,m} + \left[ 1 + b_1 \mu_x \right] w_{\ell,m} - \left[ \frac{b_1 \mu_x}{2} \right] w_{\ell+1,m} \]

\[= \left[ \frac{b_2 \mu_y}{2} \right] v_{\ell,m-1} + \left[ 1 - b_2 \mu_y \right] v_{\ell,m} + \left[ \frac{b_2 \mu_y}{2} \right] v_{\ell,m+1},\]

for \(\ell = 1, \ldots, L - 1\), and \(m = 1, \ldots, M - 1\).
Implementing ADI Methods

Figure: “Active” points in the first half-step, the interior points are active both for the old $v$-layer and the $w$-layer which is being computed. Also, the boundary values at the top $v_{\ell,M}$ and bottom $v_{\ell,0}$ boundaries are active, and so are $w_{0,m}$ (left) and $w_{L,m}$ (right).
If we enumerate our grid-points in the following (lexicographical) way

then we get \((M - 1)\) tridiagonal systems (one for each “row”), with \((L - 1)\) unknowns.
We also need the missing boundary conditions for \( w \)

\[
\begin{align*}
w_{0,m} &= \left[ \frac{b_2 \mu_y}{4} \right] \beta_{0,m-1}^n + \left[ \frac{1 - b_2 \mu_y}{2} \right] \beta_{0,m}^n + \left[ \frac{b_2 \mu_y}{4} \right] \beta_{0,m+1}^n \\
&\quad - \left[ \frac{b_2 \mu_y}{4} \right] \beta_{0,m-1}^{n+1} + \left[ \frac{1 + b_2 \mu_y}{2} \right] \beta_{0,m}^{n+1} - \left[ \frac{b_2 \mu_y}{4} \right] \beta_{0,m+1}^{n+1}.
\end{align*}
\]

\[
\begin{align*}
w_{L,m} &= \left[ \frac{b_2 \mu_y}{4} \right] \beta_{L,m-1}^n + \left[ \frac{1 - b_2 \mu_y}{2} \right] \beta_{L,m}^n + \left[ \frac{b_2 \mu_y}{4} \right] \beta_{L,m+1}^n \\
&\quad - \left[ \frac{b_2 \mu_y}{4} \right] \beta_{L,m-1}^{n+1} + \left[ \frac{1 + b_2 \mu_y}{2} \right] \beta_{L,m}^{n+1} - \left[ \frac{b_2 \mu_y}{4} \right] \beta_{L,m+1}^{n+1}.
\end{align*}
\]

For \( m = 1, \ldots, M - 1 \) (\( m = 0 \), and \( m = M \) are not needed).
The second half-stage is given by

\[- \left[ \frac{b_2 \mu_y}{2} \right] v_{\ell, m-1} + \left[ 1 + b_2 \mu_y \right] v_{\ell, m} - \left[ \frac{b_2 \mu_y}{2} \right] v_{\ell, m+1} \]

\[= \left[ \frac{b_1 \mu_x}{2} \right] w_{\ell-1, m} + \left[ 1 - b_1 \mu_x \right] w_{\ell, m} + \left[ \frac{b_1 \mu_x}{2} \right] - w_{\ell+1, m}, \]

for \( \ell = 1, \ldots, L - 1, \) and \( m = 1, \ldots, M - 1. \)

With the correct grid-ordering, we get \((L - 1)\) tridiagonal systems of size \((M - 1)\).

Boundary conditions for \( v \) are given at time-level \((n + 1)\).
Implementing ADI Methods

Figure: “Active” points in the second half-step [left], and the appropriate enumeration order of the grid-points [right].
In Strikwerda (pp. 180–181), there is a discussion of the Mitchell-Fairweather scheme, which is an ADI scheme which is second order in time, and fourth order accurate in space:

\[
\begin{align*}
\left[ 1 - \frac{1}{2} \left( b_1 \mu_x - \frac{1}{6} \right) h^2 \delta_x^2 \right] v^{n+1/2} &= \left[ 1 + \frac{1}{2} \left( b_2 \mu_y + \frac{1}{6} \right) h^2 \delta_y^2 \right] v^n, \\
\left[ 1 - \frac{1}{2} \left( b_2 \mu_y - \frac{1}{6} \right) h^2 \delta_y^2 \right] v^{n+1} &= \left[ 1 + \frac{1}{2} \left( b_1 \mu_x + \frac{1}{6} \right) h^2 \delta_x^2 \right] v^{n+1/2},
\end{align*}
\]

with Dirichlet boundary conditions for \( v^{n+1/2} \)

\[
v^{n+1/2} = \frac{1}{2b_1 \mu_x} \left\{ \left( b_1 \mu_x + \frac{1}{6} \right) \left[ 1 + \frac{1}{2} \left( b_2 \mu_y + \frac{1}{6} \right) h^2 \delta_y^2 \right] \beta^n \right. \\
+ \left. \left( b_1 \mu_x - \frac{1}{6} \right) \left[ 1 - \frac{1}{2} \left( b_2 \mu_y - \frac{1}{6} \right) h^2 \delta_y^2 \right] \beta^{n+1} \right\}.
\]
It has been shown that no ADI scheme involving only the time levels $n + 1$ and $n$ can be second-order accurate when $b_{12} \neq 0$ (i.e. when we have mixed derivatives).

A second-order accurate modification of the Peaceman-Rachford scheme is given by

\[
\begin{align*}
\left[ 1 - \frac{k}{2} b_{11} \delta^2_x \right] v^{n+1/2} &= \left[ 1 + \frac{k}{2} b_{22} \delta^2_y \right] v^n + k b_{12} \delta_{0x} \delta_{0y} \left[ \frac{3}{2} v^n - \frac{1}{2} v^{n-1} \right], \\
\left[ 1 - \frac{k}{2} b_{22} \delta^2_y \right] v^{n+1} &= \left[ 1 + \frac{k}{2} b_{11} \delta^2_x \right] v^{n+1/2} + k b_{12} \delta_{0x} \delta_{0y} \left[ \frac{3}{2} v^n - \frac{1}{2} v^{n-1} \right],
\end{align*}
\]

with Dirichlet boundary conditions for $v^{n+1/2}$

\[
v^{n+1/2} = \frac{1}{2} \left( 1 + \frac{k}{2} b_{22} \delta^2_y \right) \beta^n + \frac{1}{2} \left( 1 - \frac{k}{2} b_{22} \delta^2_y \right) \beta^{n+1}.
\]