Lecture 11

PDE and Programming (2)

Jung-Il Choi

School of Mathematics and Computing (Computational Science and Engineering)

YONSEI UNIVERSITY

Contents

Week 11

- **Multi-dimension**
  - Heat equation
    - ➔ Explicit method
    - ➔ Implicit methods in higher dimensions
    - ➔ Approximate factorization
    - ➔ Stability analysis
    - ➔ Alternating direction implicit methods (ADI)

Week 12

- **Poisson equation**
  - ➔ Iterative solution methods
    - ✓ Point Jacobi method
    - ✓ Gauss-Seidel method
    - ✓ Successive over relaxation method (SOR)
  - Non-linear PDEs

Multi-dimension

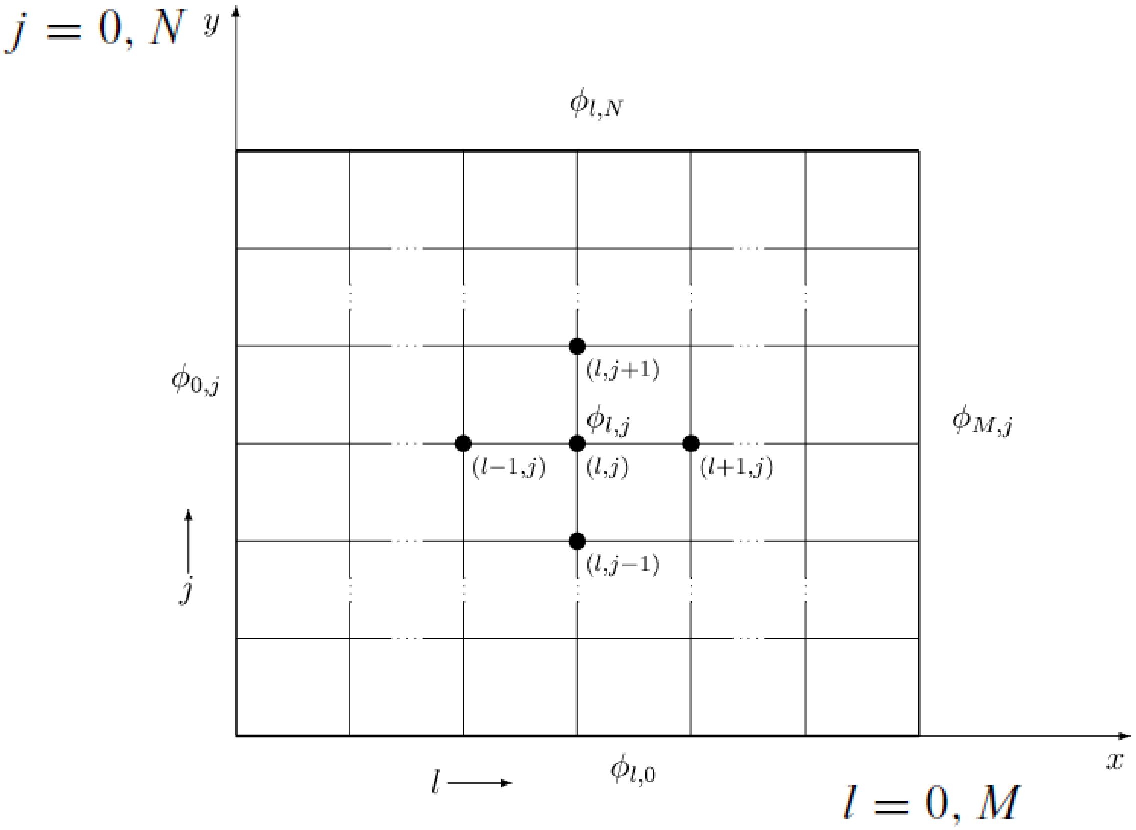
- **2D heat equation**
  - Consider the two-dimensional heat equation

$$\frac{\partial \phi}{\partial t} = \alpha \left(\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} \right)$$

  - Spatial-discretize with centered scheme and apply explicit Euler method.

$$\frac{\phi_{l,j}^{(n+1)} - \phi_{l,j}^{(n)}}{\Delta t} = \alpha \left[\frac{\phi_{l+1,j}^{(n)} - 2\phi_{l,j}^{(n)} + \phi_{l-1,j}^{(n)}}{\Delta x^2} + \frac{\phi_{l,j+1}^{(n)} - 2\phi_{l,j}^{(n)} + \phi_{l,j-1}^{(n)}}{\Delta y^2} \right]$$

$$l = 1, 2, \dots, M-1 \quad j = 1, 2, \dots, N-1, \quad n = 0, 1, 2, \cdots.$$



Multi-dimension

- **Stability properties**
  - Consider the solution of the form $\phi = \psi(t)e^{(ik_1x+ik_2y)}$

$$\frac{d\psi}{dt} = -\alpha\left(k_1'^2 + k_2'^2\right)\psi \qquad k_1'^2 = \frac{2}{\Delta x^2}[1-\cos(k_1\Delta x)]$$

$$k_2'^2 = \frac{2}{\Delta y^2}[1-\cos(k_2\Delta y)].$$

Since $-\alpha(k_1'^2 + k_2'^2)$ is real and negative

$$\Delta t \le \frac{2}{\alpha\left[\frac{2}{\Delta x^2}[1-\cos(k_1\Delta x)] + \frac{2}{\Delta y^2}[1-\cos(k_2\Delta y)]\right]}.$$

$$\Delta t \le \frac{1}{2\alpha\left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2}\right)}.$$

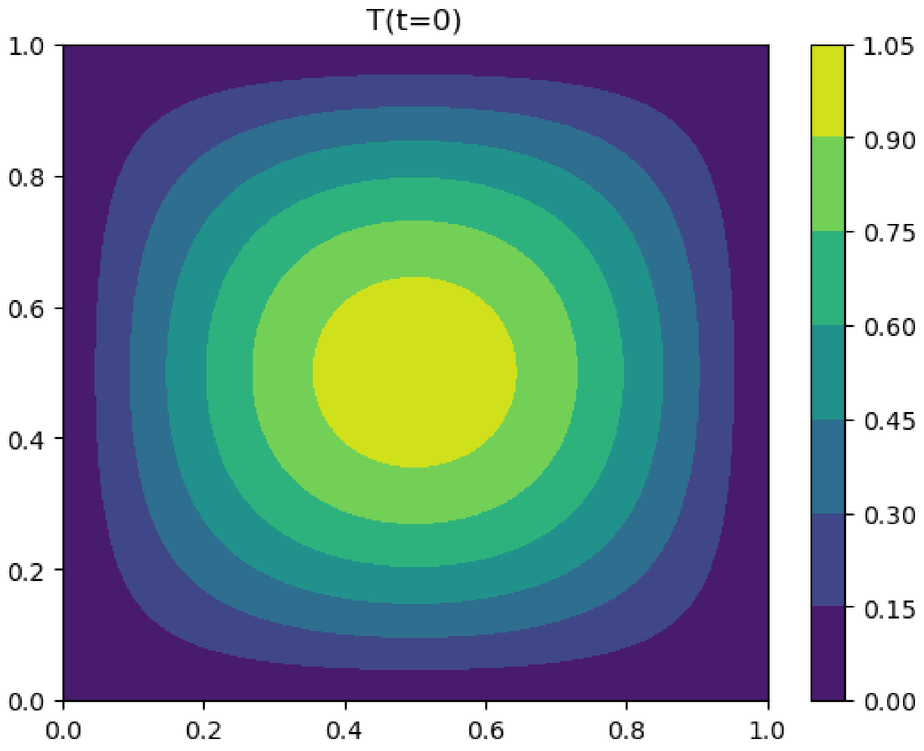
Assuming $\Delta x = \Delta y = h$, $\Delta t \le \frac{h^2}{4\alpha}$,

In the same way, we can get stability condition for the 3D case $\Delta t \le \frac{h^2}{6\alpha}$.

Multi-dimension

- **2D Heat equation example**
  - $\frac{\partial T}{\partial \mathrm{t}} = \alpha \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial x^2} \right), \quad \Omega = [0,1] \times [0,1]$

  - Initial condition $T(0, x, y) = \sin(\pi \mathrm{x})\sin(\pi \mathrm{y})$
  - Boundary condition $T_{\partial\Omega} = 0$



Multi-dimension

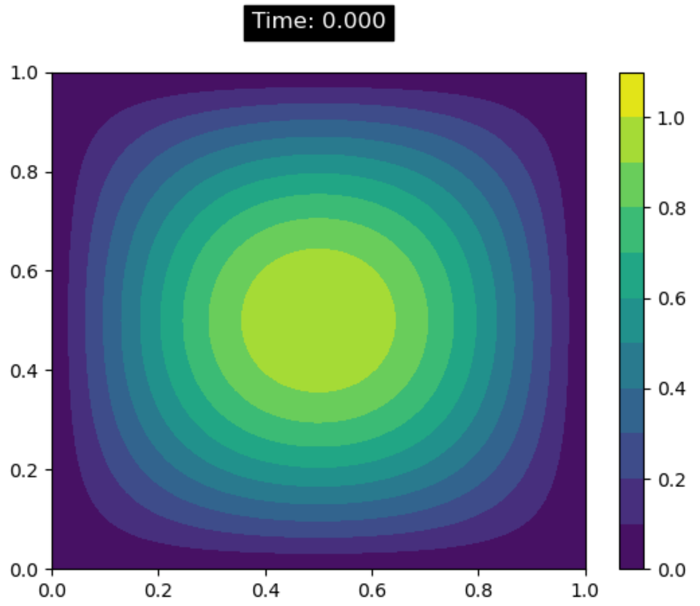
- **Example code**
  - In the same way as 1D heat equation case,

$$\text{Assuming } \Delta x = \Delta y = h, \quad \Delta t \le \frac{h^2}{4\alpha},$$

```python
def Euler(y0, f, dx, dy):
    rhs = f(y0,nx,ny,dx,dy)

    y = y0 + dt*rhs

    return y
```

```python
def heateq2D(T,nx,ny,dx,dy):
    alpha = 1
    dudt = np.zeros_like(T)
    nxm = nx-1; nxp = nx+1
    nym = ny-1; nyp = ny+1

    #central difference
    for i in range(1,nx):
        dudt[i,1:ny] = (T[i,0:nym] - 2*T[i,1:ny] + T[i,2:nyp])/dy**2
    for j in range(1,ny):
        dudt[1:nx,j]+= (T[0:nxm,j] - 2*T[1:nx,j] + T[2:nxp,j])/dx**2

    return dudt
```

```python
x0 = 0; xmax = 1; nx = 64;
y0 = 0; ymax = 1; ny = 64;

t0 = 0; dt = 0.00006;

nt = 2500;

alpha = 1.0;

x = np.linspace(x0,xmax,nx+1)
y = np.linspace(y0,ymax,ny+1)

dx = (xmax-x0)/nx
dy = (ymax-y0)/ny

T = np.zeros((nx+1,ny+1))

# initial condition
for i in range(1,nx):
    T[i,:] = np.sin(np.pi*x[i])*np.sin(np.pi*y[:])
# boundary condition
T[0,:] = 0; T[nx,:] = 0; T[:,0] = 0; T[:,ny] = 0;

for t in range(nt):
    time = t0 + (t+1)*dt

    T = Euler(T, heateq2D, dx, dy)
```



Implicit methods in higher dimensions

- **Implicit method**
  - Consider application of Crank-Nicolson scheme

$$\frac{\phi^{(n+1)} - \phi^{(n)}}{\Delta t} = \frac{\alpha}{2}\left[\frac{\partial^2 \phi^{(n+1)}}{\partial x^2} + \frac{\partial^2 \phi^{(n+1)}}{\partial y^2} + \frac{\partial^2 \phi^{(n)}}{\partial x^2} + \frac{\partial^2 \phi^{(n)}}{\partial y^2}\right]$$

assuming $\Delta x = \Delta y = h$,

$$\begin{aligned}\phi_{l,j}^{(n+1)} - \phi_{l,j}^{(n)} = {} & \frac{\alpha \Delta t}{2h^2}\left[\phi_{l+1,j}^{(n+1)} - 2\phi_{l,j}^{(n+1)} + \phi_{l-1,j}^{(n+1)} + \phi_{l,j+1}^{(n+1)} - 2\phi_{l,j}^{(n+1)} + \phi_{l,j-1}^{(n+1)}\right] \\ & + \frac{\alpha \Delta t}{2h^2}\left[\phi_{l+1,j}^{(n)} - 2\phi_{l,j}^{(n)} + \phi_{l-1,j}^{(n)} + \phi_{l,j+1}^{(n)} - 2\phi_{l,j}^{(n)} + \phi_{l,j-1}^{(n)}\right].\end{aligned}$$

Let $\beta = \alpha \Delta t / 2h^2$,

$$\begin{aligned}& -\beta\phi_{l+1,j}^{(n+1)} + (1+4\beta)\phi_{l,j}^{(n+1)} - \beta\phi_{l-1,j}^{(n+1)} - \beta\phi_{l,j+1}^{(n+1)} - \beta\phi_{l,j-1}^{(n+1)} \\ & = \beta\phi_{l+1,j}^{(n)} + (1-4\beta)\phi_{l,j}^{(n)} + \beta\phi_{l-1,j}^{(n)} + \beta\phi_{l,j+1}^{(n)} + \beta\phi_{l,j-1}^{(n)}.\end{aligned}$$

$$\phi_{l,j}^{(n+1)} \quad (l = 1,\ 2, \ldots,\ M-1; j = 1, 2, \ldots, N-1).$$

Implicit methods in higher dimensions

- **Matrix form**

$$-\beta\phi_{l+1,j}^{(n+1)} + (1+4\beta)\phi_{l,j}^{(n+1)} - \beta\phi_{l-1,j}^{(n+1)} - \beta\phi_{l,j+1}^{(n+1)} - \beta\phi_{l,j-1}^{(n+1)}$$
$$= \underbrace{\beta\phi_{l+1,j}^{(n)} + (1-4\beta)\phi_{l,j}^{(n)} + \beta\phi_{l-1,j}^{(n)} + \beta\phi_{l,j+1}^{(n)} + \beta\phi_{l,j-1}^{(n)}}_{F_{l,j}^{(n)}}.$$

$$\rightarrow \mathcal{A}\phi^{(n+1)} = F^{(n)}$$

unknown vector ϕ

$$\phi = \begin{bmatrix} \phi_{1,1} \\ \phi_{2,1} \\ \phi_{3,1} \\ \vdots \\ \phi_{M-1,1} \\ \phi_{1,2} \\ \phi_{2,2} \\ \phi_{3,2} \\ \vdots \\ \phi_{M-1,2} \\ \vdots \\ \vdots \\ \phi_{1,N-1} \\ \phi_{2,N-1} \\ \phi_{3,N-1} \\ \vdots \\ \phi_{M-1,N-1} \end{bmatrix}^{(n+1)}$$

block-tridiagonal form

$$\mathcal{A}\phi^{(n+1)} = \begin{bmatrix} B & C & & \\ A & B & C & \\ & \ddots & \ddots & \ddots \\ & & A & B \end{bmatrix} \begin{bmatrix} \phi_{l,1} \\ \phi_{l,2} \\ \vdots \\ \phi_{l,N-1} \end{bmatrix}^{(n+1)} \quad l = 1,2,\ldots,M-1$$

$$B = \begin{bmatrix} 1+4\beta & -\beta & & \\ -\beta & 1+4\beta & -\beta & \\ & \ddots & \ddots & \ddots \\ & & -\beta & 1+4\beta \end{bmatrix} \quad A, C = \begin{bmatrix} -\beta & & & \\ & -\beta & & \\ & & \ddots & \\ & & & -\beta \end{bmatrix}$$

ϕ is a vector with $(M-1)\times(N-1)$ unknown elements

Approximate factorization

- **Approximate factorization**
  - Consider application of the Crank-Nicolson method with the second-order spatial differencing to 2D heat equation

$$\frac{\phi^{(n+1)}-\phi^{(n)}}{\Delta t}=\frac{\alpha}{2}A_x\left[\phi^{(n+1)}+\phi^{(n)}\right]+\frac{\alpha}{2}A_y\left[\phi^{(n+1)}+\phi^{(n)}\right]+O(\Delta t^2)+O(\Delta x^2)+O(\Delta y^2) \quad \text{......(1)}$$

  - A_x and A_y are the difference operators representing the spatial derivatives

    ➔ For example,

$$A_x\phi=\frac{\phi_{i+1,j}-2\phi_{i,j}+\phi_{i-1,j}}{\Delta x^2} \quad i=1,2,\ldots,M-1 \quad j=1,2,\ldots,N-1.$$

a vector of length $(N-1)\times(M-1)$ with elements

  - Eq(1) can represent like

$$\left[I-\frac{\alpha\Delta t}{2}A_x-\frac{\alpha\Delta t}{2}A_y\right]\phi^{(n+1)}=\left[I+\frac{\alpha\Delta t}{2}A_x+\frac{\alpha\Delta t}{2}A_y\right]\phi^{(n)}+\Delta t[O(\Delta t^2)+O(\Delta x^2)+O(\Delta y^2)].$$

Approximate factorization

- **Approximate factorization**
  - Rearrange the previous equation into a partial factored form

$$\left(I-\frac{\alpha\Delta t}{2}A_x\right)\left(I-\frac{\alpha\Delta t}{2}A_y\right)\phi^{(n+1)}-\frac{\alpha^2\Delta t^2}{4}A_xA_y\phi^{(n+1)}$$

$$=\left(I+\frac{\alpha\Delta t}{2}A_x\right)\left(I+\frac{\alpha\Delta t}{2}A_y\right)\phi^{(n)}-\frac{\alpha^2\Delta t^2}{4}A_xA_y\phi^{(n)}+\Delta t[O(\Delta t^2)+O(\Delta x^2)+O(\Delta y^2)]$$

$$\left(I-\frac{\alpha\Delta t}{2}A_x\right)\left(I-\frac{\alpha\Delta t}{2}A_y\right)\phi^{(n+1)}=\left(I+\frac{\alpha\Delta t}{2}A_x\right)\left(I+\frac{\alpha\Delta t}{2}A_y\right)\phi^{(n)}$$

$$+\boxed{\frac{\alpha^2\Delta t^2}{4}A_xA_y(\phi^{(n+1)}-\phi^{(n)})}+\Delta t\left[O(\Delta t^2)+O(\Delta x^2)+O(\Delta y^2)\right].$$

$$\phi^{(n+1)}-\phi^{(n)}=O(\Delta t)$$

  - still second order accuracy

$$\boxed{\left(I-\frac{\alpha\Delta t}{2}A_x\right)\left(I-\frac{\alpha\Delta t}{2}A_y\right)\phi^{(n+1)}=\left(I+\frac{\alpha\Delta t}{2}A_x\right)\left(I+\frac{\alpha\Delta t}{2}A_y\right)\phi^{(n)}.}$$

the 3D heat equation

$$\left(I-\frac{\alpha\Delta t}{2}A_x\right)\left(I-\frac{\alpha\Delta t}{2}A_y\right)\left(I-\frac{\alpha\Delta t}{2}A_z\right)\phi^{(n+1)}$$
$$=\left(I+\frac{\alpha\Delta t}{2}A_x\right)\left(I+\frac{\alpha\Delta t}{2}A_y\right)\left(I+\frac{\alpha\Delta t}{2}A_z\right)\phi^{(n)}$$

Approximate factorization

- **Compare matrix form**
  - recall $$A_x\phi = \frac{\phi_{i+1,j} - 2\phi_{i,j} + \phi_{i-1,j}}{\Delta x^2}, \qquad A_y\phi = \frac{\phi_{i,j+1} - 2\phi_{i,j} + \phi_{i,j-1}}{\Delta x^2}$$
  - Original form: $$\left[I - \frac{\alpha\Delta t}{2}A_x - \frac{\alpha\Delta t}{2}A_y\right]\phi^{(n+1)} = \left[I + \frac{\alpha\Delta t}{2}A_x + \frac{\alpha\Delta t}{2}A_y\right]\phi^{(n)}$$

$$I - \frac{\alpha\Delta t}{2}A_x - \frac{\alpha\Delta t}{2}A_y \rightarrow \underbrace{\begin{bmatrix} B & C & & \\ A & B & C & \\ & \ddots & \ddots & \ddots \\ & & A & B \end{bmatrix}}_{(nx-1)\times(ny-1)} \left.\vphantom{\begin{bmatrix} B \\ A \\ \ddots \\ A \end{bmatrix}}\right\} (nx-1)\times(ny-1)$$

$$B = \begin{bmatrix} 1 + 2\frac{\alpha\Delta t}{2\Delta x^2} + 2\frac{\alpha\Delta t}{2\Delta y^2} & -\frac{\alpha\Delta t}{2\Delta x^2} & & \\ -\frac{\alpha\Delta t}{2\Delta x^2} & 1 + 2\frac{\alpha\Delta t}{2\Delta x^2} + 2\frac{\alpha\Delta t}{2\Delta y^2} & -\frac{\alpha\Delta t}{2\Delta x^2} & \\ & \ddots & \ddots & \ddots \\ & & -\frac{\alpha\Delta t}{2\Delta x^2} & 1 + 2\frac{\alpha\Delta t}{2\Delta x^2} + 2\frac{\alpha\Delta t}{2\Delta y^2} \end{bmatrix} \qquad A, C = \begin{bmatrix} -\frac{\alpha\Delta t}{2\Delta y^2} & & & \\ & -\frac{\alpha\Delta t}{2\Delta y^2} & & \\ & & \ddots & \\ & & & -\frac{\alpha\Delta t}{2\Delta y^2} \end{bmatrix}$$

Approximate factorization

- **Compare matrix form**
  - After apply approximate factorization

$$\left(I-\frac{\alpha\,\Delta t}{2}A_x\right)\left(I-\frac{\alpha\,\Delta t}{2}A_y\right)\phi^{(n+1)}=\left(I+\frac{\alpha\,\Delta t}{2}A_x\right)\left(I+\frac{\alpha\,\Delta t}{2}A_y\right)\phi^{(n)}.$$

$$I-\frac{\alpha\Delta t}{2}A_x=\underbrace{\begin{bmatrix} 1+2\frac{\alpha\Delta t}{2\Delta x^2} & -\frac{\alpha\Delta t}{2\Delta x^2} & & \\ -\frac{\alpha\Delta t}{2\Delta x^2} & 1+2\frac{\alpha\Delta t}{2\Delta x^2} & -\frac{\alpha\Delta t}{2\Delta x^2} & \\ \ddots & \ddots & \ddots & \\ & & 1+2\frac{\alpha\Delta t}{2\Delta x^2} & -\frac{\alpha\Delta t}{2\Delta x^2} \end{bmatrix}}_{(nx-1)}\left.\vphantom{\begin{bmatrix}1\\1\\1\\1\end{bmatrix}}\right\}(nx-1)$$

$$I-\frac{\alpha\Delta t}{2}A_y=\underbrace{\begin{bmatrix} 1+2\frac{\alpha\Delta t}{2\Delta y^2} & -\frac{\alpha\Delta t}{2\Delta y^2} & & \\ -\frac{\alpha\Delta t}{2\Delta y^2} & 1+2\frac{\alpha\Delta t}{2\Delta y^2} & -\frac{\alpha\Delta t}{2\Delta y^2} & \\ \ddots & \ddots & \ddots & \\ & & 1+2\frac{\alpha\Delta t}{2\Delta y^2} & -\frac{\alpha\Delta t}{2\Delta y^2} \end{bmatrix}}_{(nx-1)}\left.\vphantom{\begin{bmatrix}1\\1\\1\\1\end{bmatrix}}\right\}(nx-1)$$

Approximate factorization

- **Implementation**
  - Let, $$z = \left(I - \frac{\alpha\Delta t}{2}A_y\right)\phi^{(n+1)}.$$ (Intermediate solution vectors)
  - From $$\left(I - \frac{\alpha\Delta t}{2}A_x\right)\left(I - \frac{\alpha\Delta t}{2}A_y\right)\phi^{(n+1)} = \left(I + \frac{\alpha\Delta t}{2}A_x\right)\left(I + \frac{\alpha\Delta t}{2}A_y\right)\phi^{(n)} = rhs^{(n)} \Longrightarrow \left(I - \frac{\alpha\Delta t}{2}A_x\right)z = rhs^{(n)}$$

$$z_{i,j} - \left(\frac{\alpha\Delta t}{2}\right)\frac{z_{i-1,j} - 2z_{i,j} + z_{i+1,j}}{\Delta x^2} = rhs^{(n)}$$

$$-\frac{\alpha\Delta t}{2\Delta x^2}z_{i+1,j} + \left(1 + \frac{\alpha\Delta t}{\Delta x^2}\right)z_{i,j} - \frac{\alpha\Delta t}{2\Delta x^2}z_{i-1,j} = rhs^{(n)}$$

for each $j = 1, 2, \ldots, N-1$, a simple tridiagonal system is solved for $z_{i,j}$.
loop running over the index j.

$$\begin{bmatrix} 1+\frac{\alpha\Delta t}{\Delta x^2} & -\frac{\alpha\Delta t}{2\Delta x^2} & & \\ -\frac{\alpha\Delta t}{2\Delta x^2} & 1+\frac{\alpha\Delta t}{\Delta x^2} & -\frac{\alpha\Delta t}{2\Delta x^2} & \\ \ddots & \ddots & \ddots & \\ & & 1+\frac{\alpha\Delta t}{\Delta x^2} & -\frac{\alpha\Delta t}{2\Delta x^2} \end{bmatrix}\begin{bmatrix} z_{1,j} \\ z_{2,j} \\ \vdots \\ z_{nx-1,j} \end{bmatrix} = \begin{bmatrix} rhs_{1,j} + \frac{\alpha\Delta t}{2\Delta x^2}z_{0,j} \\ rhs_{2,j} \\ \vdots \\ rhs_{nx-1,j} + \frac{\alpha\Delta t}{2\Delta x^2}z_{nx,j} \end{bmatrix}$$

Boundary condition

Approximate factorization

- **Implementation**
  - Boundary treatment
    - ➔ When $i = 1 \; or \; nx$, boundary values for $\boldsymbol{z}$ are required in the form of $z_{0,j} \; or \; z_{nx,j}$.

$$z_{0,j} = \phi_{0,j}^{(n+1)} - \frac{\alpha \Delta t}{2} \frac{\phi_{0,j+1}^{(n+1)} - 2\phi_{0,j}^{(n+1)} + \phi_{0,j-1}^{(n+1)}}{\Delta y^2} \qquad j = 1, 2, \ldots, N-1.$$

  - Now, we have intermediate solution vector $\boldsymbol{z}$

$$\left(I - \frac{\alpha \Delta t}{2} A_y\right) \phi^{(n+1)} = z. \;\; \Rightarrow \;\; -\frac{\alpha \Delta t}{2\Delta y^2}\phi_{i,j+1}^{(n+1)} + \left(1 + \frac{\alpha \Delta t}{\Delta y^2}\right)\phi_{i,j}^{(n+1)} - \frac{\alpha \Delta t}{2\Delta y^2}\phi_{i,j-1}^{(n+1)} = z_{i,j}.$$

For each $i = 1, 2, \ldots, M-1$, a tridiagonal system of equations is solved for $\phi_{i,j}^{(n+1)}$
the loop is now over the index i.

$$\begin{bmatrix} 1+\frac{\alpha\Delta t}{\Delta y^2} & -\frac{\alpha\Delta t}{2\Delta y^2} & & \\ -\frac{\alpha\Delta t}{2\Delta y^2} & 1+\frac{\alpha\Delta t}{\Delta y^2} & -\frac{\alpha\Delta t}{2\Delta y^2} & \\ \ddots & \ddots & \ddots & \\ & & 1+\frac{\alpha\Delta t}{\Delta y^2} & -\frac{\alpha\Delta t}{2\Delta y^2} \end{bmatrix} \begin{bmatrix} \phi_{i,1} \\ \phi_{i,2} \\ \vdots \\ \phi_{i,ny-1} \end{bmatrix} = \begin{bmatrix} z_{i,1} \\ z_{i,2} \\ \vdots \\ z_{i,ny-1} \end{bmatrix}$$

Approximate factorization

- **Pseudo code**

function Approximate_Factorization(nx,ny,dx,dy,dt,alpha,T)

$nxm \leftarrow nx - 1,\ nym \leftarrow ny - 1$

$nxp \leftarrow ny + 1,\ nyp \leftarrow ny + 1$

$coef \leftarrow \frac{\alpha\Delta t}{2}$

$[A_x m, A_x c, A_x p] \leftarrow coef[\frac{1}{\Delta x^2}, -\frac{2}{\Delta x^2}, \frac{1}{\Delta x^2}]$

$[A_y m, A_y c, A_y p] \leftarrow coef[\frac{1}{\Delta y^2}, -\frac{2}{\Delta y^2}, \frac{1}{\Delta y^2}]$

$\#get\ rhs^{(n)}$

$for\ i = 1{:}\,nx$

$for\ j = 1{:}\,ny$

$f_{i,j} \leftarrow A_y m_j T_{i,j-1} + (1 + A_y c_j) T_{i,j} + A_y p_j T_{i,j+1}$

$end\ for$

$end\ for$

$for\ j = 1{:}\,ny$

$for\ i = 1{:}\,nx$

$rhs_{i,j} \leftarrow A_x m_i f_{i-1,j} + (1 + A_x c_i) f_{i,j} + A_x p_i f_{i+1,j}$

$end\ for$

$end\ for$

$\#\ boundary\ treatment$

$for\ j = 1{:}\,ny$

$rhs_{1,j} \leftarrow \frac{coef}{\Delta x^2}(A_y m_j T_{0,j-1} + (1 - A_y x_j) T_{0,j} + A_y m_j T_{0,j+1}$

$rhs_{nx-1,j} \leftarrow \frac{coef}{\Delta x^2}(A_y m_j T_{nx,j-1} + (1 - A_y x_j) T_{nx,j} + A_y m_j T_{nx,j+1}$

$end\ for$

$$rhs_{i,j} = \left(I + \frac{\alpha\Delta t}{2}A_x\right)\underbrace{\left(I + \frac{\alpha\Delta t}{2}A_y\right)\phi^{(n)}}_{f_{i,j}}$$

$$z_{0,j} = \phi_{0,j}^{(n+1)} - \frac{\alpha\Delta t}{2}\frac{\phi_{0,j+1}^{(n+1)} - 2\phi_{0,j}^{(n+1)} + \phi_{0,j-1}^{(n+1)}}{\Delta y^2} \qquad j = 1, 2, \ldots, N-1.$$

Approximate factorization

- **Pseudo code**

```
#get intermediate solution
for j = 1:ny
    z_j ← call TDMA(−A_x m, 1 − A_x c, −A_x p, rhs_j)
end for
#get solution
for i = 1:nx
    T_i ← call TDMA(−A_y m, 1 − A_y c, −A_y p, z_i)
end for

    return T
End function
```

$$\begin{bmatrix} 1+\frac{\alpha\Delta t}{\Delta x^2} & -\frac{\alpha\Delta t}{2\Delta x^2} & & \\ -\frac{\alpha\Delta t}{2\Delta x^2} & 1+\frac{\alpha\Delta t}{\Delta x^2} & -\frac{\alpha\Delta t}{2\Delta x^2} & \\ \ddots & \ddots & \ddots & \\ & & 1+\frac{\alpha\Delta t}{\Delta x^2} & -\frac{\alpha\Delta t}{2\Delta x^2} \end{bmatrix} \begin{bmatrix} z_{1,j} \\ z_{2,j} \\ \vdots \\ z_{nx-1,j} \end{bmatrix} = \begin{bmatrix} rhs_{1,j} + \frac{\alpha\Delta t}{2\Delta x^2} z_{0,j} \\ rhs_{2,j} \\ \vdots \\ rhs_{nx-1,j} + \frac{\alpha\Delta t}{2\Delta x^2} z_{nx,j} \end{bmatrix}$$

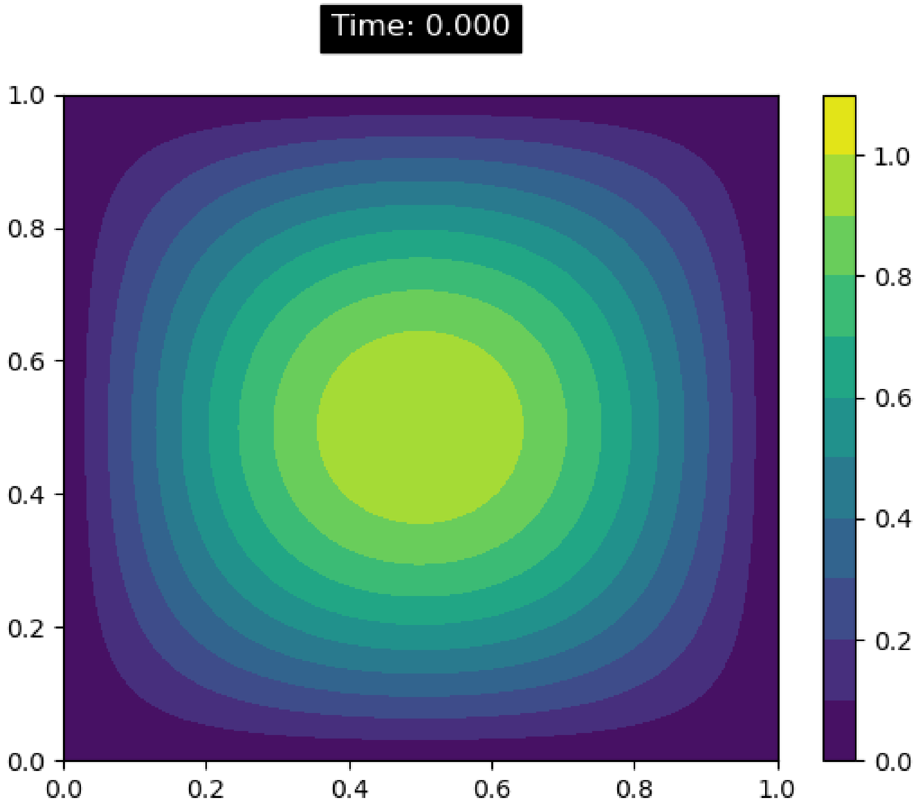
$$\begin{bmatrix} 1+\frac{\alpha\Delta t}{\Delta y^2} & -\frac{\alpha\Delta t}{2\Delta y^2} & & \\ -\frac{\alpha\Delta t}{2\Delta y^2} & 1+\frac{\alpha\Delta t}{\Delta y^2} & -\frac{\alpha\Delta t}{2\Delta y^2} & \\ \ddots & \ddots & \ddots & \\ & & 1+\frac{\alpha\Delta t}{\Delta y^2} & -\frac{\alpha\Delta t}{2\Delta y^2} \end{bmatrix} \begin{bmatrix} \phi_{i,1} \\ \phi_{i,2} \\ \vdots \\ \phi_{i,ny-1} \end{bmatrix} = \begin{bmatrix} z_{i,1} \\ z_{i,2} \\ \vdots \\ z_{i,ny-1} \end{bmatrix}$$

Approximate factorization

- **2D Heat equation example**

```python
x0 = 0; xmax = 1; nx = 64;
y0 = 0; ymax = 1; ny = 64;

t0 = 0; dt = 0.01;

nt = 15;

alpha = 1.0;

x = np.linspace(x0,xmax,nx+1)
y = np.linspace(y0,ymax,ny+1)

dx = (xmax-x0)/nx
dy = (ymax-y0)/ny

T = np.zeros((nx+1,ny+1))

# initial condition
for i in range(1,nx):
    T[i,:] = np.sin(np.pi*x[i])*np.sin(np.pi*y[:])
# boundary condition
T[0,:] = 0; T[nx,:] = 0; T[:,0] = 0; T[:,ny] = 0;

for t in range(nt):
    time = t0 + (t+1)*dt

    T = Approximate_Factorization(nx,ny,dx,dy,dt,alpha,T)
```

Compare with Euler method($\Delta t \leq 0.00006104$)
$(nx = ny = 64)$



Stability of factored scheme

- **Stability analysis**

$$\left(I-\frac{\alpha\Delta t}{2}A_x\right)\left(I-\frac{\alpha\Delta t}{2}A_y\right)\phi^{(n+1)}=\left(I+\frac{\alpha\Delta t}{2}A_x\right)\left(I+\frac{\alpha\Delta t}{2}A_y\right)\phi^{(n)}.$$

$$\phi_{lj}^{(n)}=\psi^n e^{ik_1x_l}e^{ik_2y_j}$$

$$\left(1+\frac{\alpha\Delta t}{2}k_1'^2\right)\left(1+\frac{\alpha\Delta t}{2}k_2'^2\right)\psi^{n+1}=\left(1-\frac{\alpha\Delta t}{2}k_1'^2\right)\left(1-\frac{\alpha\Delta t}{2}k_2'^2\right)\psi^n.$$

$$k_1'^2=\frac{2}{\Delta x^2}[1-\cos(k_1\Delta x)]$$

$$k_2'^2=\frac{2}{\Delta y^2}[1-\cos(k_2\Delta y)].$$

$$\left|\frac{\psi^{n+1}}{\psi^n}\right|=\left|\frac{\left(1-\frac{\alpha\Delta t}{2}k_1'^2\right)\left(1-\frac{\alpha\Delta t}{2}k_2'^2\right)}{\left(1+\frac{\alpha\Delta t}{2}k_1'^2\right)\left(1+\frac{\alpha\Delta t}{2}k_2'^2\right)}\right|\le 1$$

Unconditionally stable !

Alternating Direction Implicit Methods (ADI)

- **Alternating Direction Implicit Methods (ADI)**
  - Use operator splitting.
  - ADI scheme is an equivalent formulation of the factored scheme.

$$\frac{\partial \phi}{\partial t} = \alpha \left(\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} \right)$$

  - Consider half-step

$$\phi^{(n+1/2)} - \phi^{(n)} = \frac{\alpha \Delta t}{2} \left(\frac{\partial^2 \phi^{(n+1/2)}}{\partial x^2} + \frac{\partial^2 \phi^{(n)}}{\partial y^2} \right)$$

$$\phi^{(n+1)} - \phi^{(n+1/2)} = \frac{\alpha \Delta t}{2} \left(\frac{\partial^2 \phi^{(n+1/2)}}{\partial x^2} + \frac{\partial^2 \phi^{(n+1)}}{\partial y^2} \right)$$

$$\longrightarrow$$

$$\left(I - \frac{\alpha \Delta t}{2} A_x \right) \phi^{(n+1/2)} = \left(I + \frac{\alpha \Delta t}{2} A_y \right) \phi^{(n)}$$

$$\left(I - \frac{\alpha \Delta t}{2} A_y \right) \phi^{(n+1)} = \left(I + \frac{\alpha \Delta t}{2} A_x \right) \phi^{(n+1/2)}.$$

Alternating Direction Implicit Methods (ADI)

- **Alternating Direction Implicit Methods (ADI)**
  - $\phi^{(n+1/2)} = \left(I - \frac{\alpha\,\Delta t}{2} A_x\right)^{-1} \left(I + \frac{\alpha\,\Delta t}{2} A_y\right) \phi^{(n)},$ from $\left(I - \frac{\alpha\,\Delta t}{2} A_x\right) \phi^{(n+1/2)} = \left(I + \frac{\alpha\,\Delta t}{2} A_y\right) \phi^{(n)}$ ……(1)
  - Substituting Eq(1) into $\left(I - \frac{\alpha\,\Delta t}{2} A_y\right) \phi^{(n+1)} = \left(I + \frac{\alpha\,\Delta t}{2} A_x\right) \phi^{(n+1/2)}.$

    we can get $\left(I - \frac{\alpha\,\Delta t}{2} A_y\right) \phi^{(n+1)} = \left(I + \frac{\alpha\,\Delta t}{2} A_x\right) \left(I - \frac{\alpha\,\Delta t}{2} A_x\right)^{-1} \left(I + \frac{\alpha\,\Delta t}{2} A_y\right) \phi^{(n)}.$
  - Since the $(I + \frac{\alpha\Delta t}{2A_x})$ and $(I + \frac{\alpha\Delta t}{2A_y})$ operators commute,

$$\left(I - \frac{\alpha\,\Delta t}{2} A_x\right) \left(I - \frac{\alpha\,\Delta t}{2} A_y\right) \phi^{(n+1)} = \left(I + \frac{\alpha\,\Delta t}{2} A_x\right) \left(I + \frac{\alpha\,\Delta t}{2} A_y\right) \phi^{(n)}.$$

  - Boundary conditions are required for $\phi^{\left(n+\frac{1}{2}\right)}$

$$\phi^{(n+1/2)} - \frac{\alpha\,\Delta t}{2} A_x \phi^{(n+1/2)} = \left(I + \frac{\alpha\,\Delta t}{2} A_y\right) \phi^{(n)}$$

$$\phi^{(n+1/2)} + \frac{\alpha\,\Delta t}{2} A_x \phi^{(n+1/2)} = \left(I - \frac{\alpha\,\Delta t}{2} A_y\right) \phi^{(n+1)}.$$

$$\xrightarrow{\text{Adding these two equations}} \quad \phi_B^{(n+1/2)} = 1/2 \left[\left(I + \frac{\alpha\,\Delta t}{2} A_y\right) \phi_B^{(n)} + \left(I - \frac{\alpha\,\Delta t}{2} A_y\right) \phi_B^{(n+1)} \right]$$

Q&A Thanks for listening