PETR 5313: CRN 38950, Fall 2017
Numerical Application in Petroleum Engineering,
Lesson 11: PDE

Ekarit Panacharoensawad, PhD
Terry Fuller Petroleum Engineering Research Building Room 236
ekarit.panacharoensawad@ttu.edu

Cite as: Panacharoensawad, E. (2017) "Numerical application in petroleum engineering, Lesson 11: PDE", presentation slide for Texas Tech PETR 5313 Fall 2017

Outline

- Finite difference (to a certain depth)
  - Partial derivative approximation
  - Type of PDE
  - Non-dimensionalize of PDE
  - 1D heat equation
    - Forward in time/central in space
    - Backward in time/central in space
    - Crank-Nicolson
  - 2D heat equation
    - Backward in time (forward scheme is for homework)
- Finite element (introduction to FEniCS)

Partial derivative approximation: Forward scheme

Forward finite difference first order approximation

$$\left.\frac{\partial f(x,y,z)}{\partial z}\right|_{\text{at x,y,z}} \approx \frac{f(x,y,z+\varepsilon)-f(x,y,z)}{\varepsilon}$$

Forward finite difference formulas (uniform grid spacing)

Derivative	Accuracy	*Points* 0	1	2	3	4	5
1	1	–1	1				
1	2	–3/2	2	–1/2			
1	3	–11/6	3	–3/2	1/3		
1	4	–25/12	4	–3	4/3	–1/4	
2	1	1	–2	1			
2	2	2	–5	4	–1		
2	3	35/12	–26/3	19/2	–14/3	11/12	
2	4	15/4	–77/6	107/6	–13	61/12	–5/6

Partial derivative approximation: Backward scheme

From forward finite difference table, give all odd derivative coefficient with the opposite sign to get backward scheme.

Backward finite difference formulas (uniform grid spacing)

Derivative	Accuracy	*Points* 0	−1	−2	−3	−4	−5
1	1	1	−1				
1	2	3/2	−2	1/2			
1	3	11/6	3	3/2	−1/3		
1	4	25/12	−4	3	−4/3	1/4	
2	1	1	−2	1			
2	2	2	−5	4	−1		
2	3	35/12	−26/3	19/2	−14/3	11/12	
2	4	15/4	−77/6	107/6	−13	61/12	−5/6

Partial derivative approximation: Central scheme

The table below is for uniform grid spacing.

For second derivative, use h^2 as the denominator (previous tables too)

Derivative	Accuracy	Points −4	−3	−2	−1	0	1	2	3	4
1	2				−1/2	0	1/2			
1	4			1/12	−2/3	0	2/3	−1/12		
1	6		−1/60	3/20	−3/4	0	3/4	−3/20	1/60	
1	8	1/280	−4/105	1/5	−4/5	0	4/5	−1/5	4/105	−1/280
2	2				1	−2	1			
2	4			−1/12	4/3	−5/2	4/3	−1/12		
2	6		1/90	−3/20	3/2	−49/18	3/2	−3/20	1/90	
2	8	−1/560	8/315	−1/5	8/5	−205/72	8/5	−1/5	8/315	−1/560

https://en.wikipedia.org/wiki/Finite_difference_coefficient

Type of PDE

Elliptic – time-independent

Hyperbolic – time dependent and wavelike

Parabolic – time-dependent and diffusive

More formal definition

$$a\frac{\partial^2 u}{\partial x_1^2} + 2b\frac{\partial^2 u}{\partial x_1 \partial x_2} + c\frac{\partial^2 u}{\partial x_2^2} + d\frac{\partial u}{\partial x_1} + e\frac{\partial u}{\partial x_2} + fu = g$$

a,b,c,d,e,f, and g are functions of x_1 and x_2

Elliptic PDE $b^2 - 4ac < 0$

Example: Poisson's equation (e.g. Gauss's law for electricity)

$$\nabla^2 u = f(x)$$

Example: Laplace equation (e.g. steady state heat transfer)

$$\nabla^2 u = 0$$

Solution is smooth, even if IC and BC are rough. One value at BC affect all solution points

Parabolic PDE $b^2 - 4ac = 0$

Example: Heat Equation (transient heat conduction in solid)

$$\frac{\partial T}{\partial t} = \alpha \nabla^2 T \qquad \alpha = \frac{k}{\rho \hat{C}_p}$$

This gives one-way communication in time (the past impacts the future, but the future does not impact the past)

$$\Delta f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} + \frac{\partial^2 f}{\partial z^2}$$

Cartesian / Cylindrical

$$\Delta f = \frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial f}{\partial r}\right) + \frac{1}{r^2}\frac{\partial^2 f}{\partial \varphi^2} + \frac{\partial^2 f}{\partial z^2}$$

https://en.wikipedia.org/wiki/Laplace_operator

Hyperbolic PDE $b^2 - 4ac > 0$

Example: Wave equation

$$\frac{\partial^2 u}{\partial t^2} = c^2 \frac{\partial^2 u}{\partial x^2}$$

The disturbance in BC cause wave to travel (propagate) at speed C. Not every points feel the disturbance all at once (Elliptic PDE is equivalent to every point feel the disturbance in BC, all at once).

http://mathworld.wolfram.com/EllipticPartialDifferentialEquation.html
https://en.wikipedia.org/wiki/Partial_differential_equation#Equations_of_second_order

Steps for solving PDE Numerically

0) Optional (but suggested) - Non-dimensionalize the PDE

1) Discretize the domain into discrete point in time and space

2) Use finite difference formula transform all partial derivatives into algebraic expressions as a function of point value

3) Arrange all algebraic equations into matrix form

4) Use numerical method in solving system of linear equation to get solution at each point

5) Visualize the solution at time-slice or space-slice

Solving 1-D Heat Equation

In this example, we will solve 1 dimension transient heat transfer copper. For 3 cm 1D copper bar, what will happen if we attach it to a hot heat source (at 50 C), if the copper bar temperature is at 20 C, initially.

$$\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial x^2}$$

$$\text{Initial condition: At } t = 0, \quad T = 20, \qquad \forall T \in \Omega; \qquad \Omega = [0, 0.03]$$

$$\text{or} \qquad T_i = 20$$

$$\text{Boundary condition1: At } x = 0m., T = 50, \qquad \text{at any time}$$

$$\text{or} \quad T_0 = 50$$

1D Heat Equation: BC/Variable definition

Boundary condition2: At $x = 0.03m.,\ \ \dfrac{\partial T}{\partial x} = 0, \quad$ at any time

Assume no heat loss at $x = 0.03m$

T is temperature in degree Celsius $[\,C\,]$

T_i is the initial temperature

T_0 is the temperature at $x = 0$

1D Heat Equation: BC/Variable definition...

x is a distance in centimeter $[m]$

a is x at the left boundary

b is x at the right boundary

α is thermal diffusivity of copper $[m^2/s]$

$$\alpha = \frac{k}{\rho \hat{C}_p}$$

For copper, $\alpha = 1.11 \times 10^{-4}\ m^2/s$

t is time in second $[s]$

Non-dimensionalize Process

The dimensionless PDE form can be used for other question that has the same dimensionless form

Define: $\tilde{T} = \frac{T - T_i}{T_0 - T_i}$ $\qquad \tilde{x} = \frac{x - a}{b - a} = \frac{x - 0}{0.03 - 0}$

This way, the solution will be in between 0 and 1

$$\frac{\partial}{\partial x} = \frac{1}{b - a} \frac{\partial}{\partial \tilde{x}} \qquad \frac{\partial \tilde{T}}{\partial t} = \frac{1}{T_0 - T_i} \frac{\partial T}{\partial t}$$

$$\frac{\partial \tilde{T}}{\partial \tilde{x}} = \frac{\partial \tilde{T}}{\partial x} \frac{\partial x}{\partial \tilde{x}} = \frac{b - a}{T_0 - T_i} \frac{\partial T}{\partial x}$$

Non-dimensionalize Process...

$$\frac{\partial}{\partial \tilde{x}}\frac{\partial \tilde{T}}{\partial \tilde{x}} = \frac{\partial}{\partial \tilde{x}}\frac{b-a}{T_0-T_i}\frac{\partial T}{\partial x} = \frac{(b-a)^2}{T_0-T_i}\frac{\partial^2 T}{\partial x^2}$$

Then, the relationship between dimensional/less are

$$(T_0-T_i)\frac{\partial \tilde{T}}{\partial t} = \frac{\partial T}{\partial t}$$ and $$\frac{T_0-T_i}{(b-a)^2}\frac{\partial^2 \tilde{T}}{\partial \tilde{x}^2} = \frac{\partial^2 T}{\partial x^2}$$

Substituting into the governing equation, gives

$$(T_0-T_i)\frac{\partial \tilde{T}}{\partial t} = \alpha\frac{T_0-T_i}{(b-a)^2}\frac{\partial^2 \tilde{T}}{\partial \tilde{x}^2}$$ or $$\frac{(b-a)^2}{\alpha}\frac{\partial \tilde{T}}{\partial t} = \frac{\partial^2 \tilde{T}}{\partial \tilde{x}^2}$$

Non-dimensionalize Process...

By defining the dimensionless time as, $\tilde{t} = \frac{\alpha}{(b-a)^2} t$

We have $$\frac{\partial \tilde{T}}{\partial \tilde{t}} = \frac{\partial^2 \tilde{T}}{\partial \tilde{x}^2}$$

With the dimensionless form, BCs become,

At $x = 0, \ \tilde{x} = 0, \ T = T_0 \quad \text{or} \quad \tilde{T} = 1$

At $x = b, \ \tilde{x} = 1, \ \frac{\partial T}{\partial x} = 0$

IC becomes

$$\tilde{T} = 0 \quad \text{at} \quad \tilde{t} = 0$$

Discretized form of PDE

Use forward difference in time, we have

$$\frac{\partial \tilde{T}}{\partial \tilde{t}} \approx \frac{\tilde{T}_{n+1} - T_n}{\Delta t}$$

Use Central difference in space, we have

$$\frac{\partial^2 \tilde{T}}{\partial \tilde{x}^2} \approx \left. \frac{T_{i-1} - 2T_i + T_{i+1}}{(\Delta \tilde{x})^2} \right|_{@n}$$

Forward scheme can be unstable, the time step is restricted by CFL condition

CFL Condition (Courant-Friedrichs-Lewy Condition)

CFL condition is necessary but not sufficient for the convergence, meaning that even if the step size satisfies CFL condition, it can be still unstable

For heat equation: CFL Condition becomes

$$\frac{\Delta\tilde{t}}{(\Delta\tilde{x})^2} < C$$ With C = 1, I got unstable result, let's use C = 0.5

This means that for $\Delta\tilde{x} = 0.1$ We need $\Delta\tilde{t} = 0.0045$

https://webcourse.cs.technion.ac.il/236861/Winter2007-2008/ho/WCFiles/numerical_schemes.pdf
https://en.wikipedia.org/wiki/Talk%3ACourant%E2%80%93Friedrichs%E2%80%93Lewy_condition
https://people.maths.ox.ac.uk/trefethen/4all.pdf

Discretized form of PDE...

$$\left.\frac{\tilde{T}_{n+1} - \tilde{T}_n}{\Delta \tilde{t}}\right|_{@i} = \left.\frac{\tilde{T}_{i-1} - 2\tilde{T}_i + \tilde{T}_{i+1}}{(\Delta \tilde{x})^2}\right|_{@n}$$

$$\tilde{T}_{n+1} = \left.\left(\tilde{T}_i + \Delta \tilde{t}\frac{\tilde{T}_{i-1} - 2\tilde{T}_i + \tilde{T}_{i+1}}{(\Delta \tilde{x})^2}\right)\right|_{@n}$$

$$\left.\tilde{T}_{n+1}\right|_{@i} = \left.\left(\frac{\Delta \tilde{t}}{(\Delta \tilde{x})^2}\tilde{T}_{i-1} + \left(1 - \frac{2\Delta \tilde{t}}{(\Delta \tilde{x})^2}\right)\tilde{T}_i + \frac{\Delta \tilde{t}}{(\Delta \tilde{x})^2}\tilde{T}_{i+1}\right)\right|_{@n}$$

Need 3 points at n, to get 1 point at n+1 (for forward scheme)

Discretization of Boundary Conditions

Dirichlet Boundary Condition

- Constant T boundary condition

$$\text{At } \tilde{x} = 0, \text{ we have, } \tilde{T} = 1$$

$$\text{Thus, we have } T_{i=0} = 1 \text{ at every } n$$

Neumann Boundary Condition

- Constant flux at the boundary

$$\text{At } \tilde{x} = 1, \quad \frac{\partial \tilde{T}}{\partial \tilde{x}} = 0$$

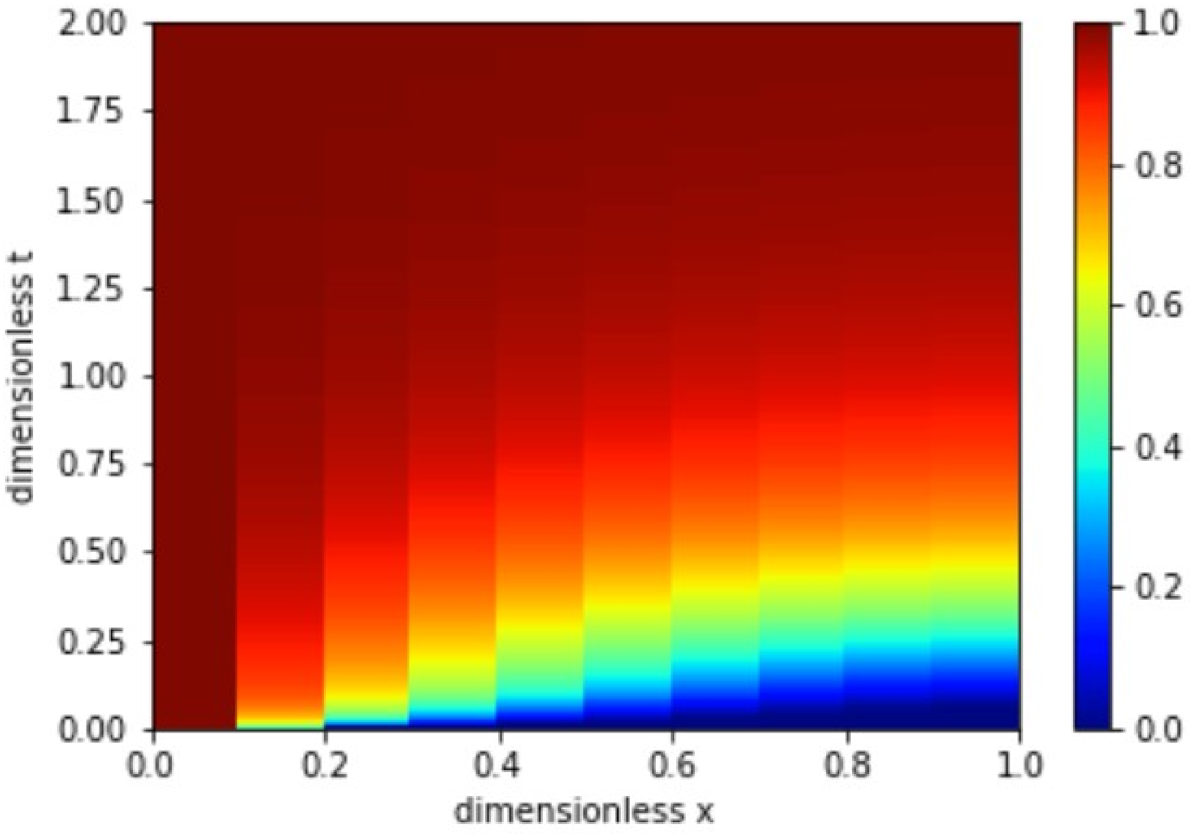
From backward scheme, we have

$$\frac{-\tilde{T}_{B-1} + \tilde{T}_B}{\tilde{x}} = 0 \qquad \tilde{T}_{B-1} = \tilde{T}_B$$

Where B is the boundary point

1D-Heat Equation: Dimensionless Result

Forward Euler (forward finite difference in time) result



For smaller x-grid spacing at the same t-grid spacing, the result is not stable.

1D-Heat Equation: Dimensional Result

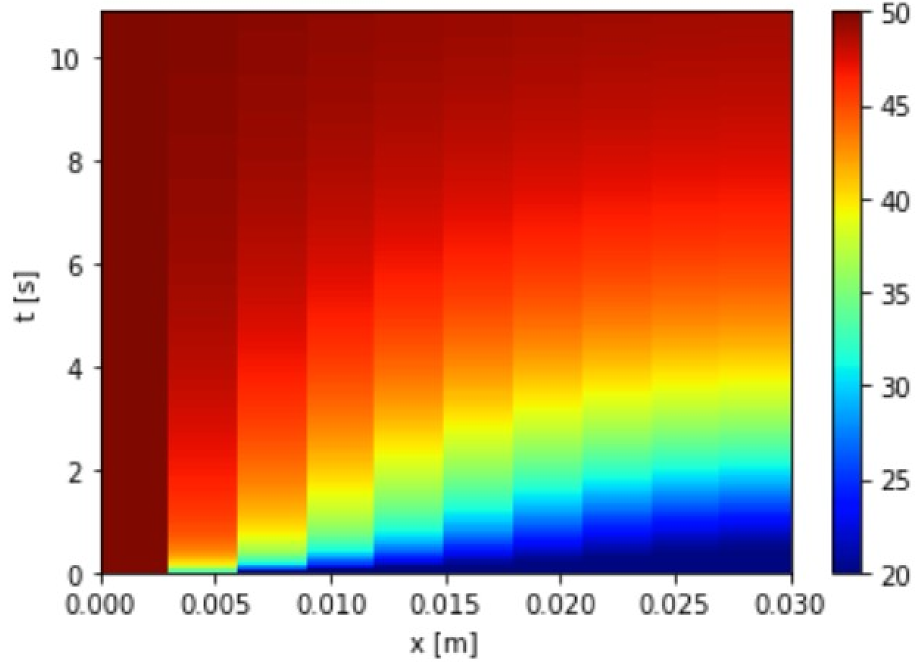
The result can be obtained by using the relationships between dimensionless and dimensional variable

$$\text{For copper, } \alpha = 1.11 \times 10^{-4}\ m^2/s$$

$$\tilde{t} = \frac{1.11 \times 10^{-4}}{(0.03)^2} t$$

$$\tilde{x} = \frac{x-a}{b-a} = \frac{x}{0.03}$$

$$\tilde{T} = \frac{T-T_i}{T_0-T_i} = \frac{T-20}{50-20}$$



Backward finite difference in time

Unconditionally stable (we can use smaller x-grid spacing)

$$\frac{\partial \tilde{T}}{\partial \tilde{t}} \approx \left.\frac{\tilde{T}_{n+1} - \tilde{T}_n}{\Delta \tilde{t}}\right|_{@i}$$

$$\frac{\partial^2 \tilde{T}}{\partial \tilde{x}^2} \approx \left.\frac{\tilde{T}_{i-1} - 2\tilde{T}_i + \tilde{T}_{i+1}}{(\Delta \tilde{x})^2}\right|_{@n+1}$$

$$\left.\frac{\tilde{T}_{n+1} - \tilde{T}_n}{\Delta \tilde{t}}\right|_{@i} = \left.\frac{\tilde{T}_{i-1} - 2\tilde{T}_i + T_{i+1}}{(\Delta \tilde{x})^2}\right|_{@n+1}$$

Backward finite difference in time...

Solve system of linear equation to get T_{n+1}

$$\left.\frac{\tilde{T}_{n+1}-\tilde{T}_n}{\Delta\tilde{t}}\right|_{@i} = \left.\frac{\tilde{T}_{i-1}-2\tilde{T}_i+T_{i+1}}{(\Delta\tilde{x})^2}\right|_{@n+1}$$

$$\left(-r\tilde{T}_{i-1}+(1+2r)\tilde{T}_i-r\tilde{T}_{i+1}\right)_{n+1} = \left(\tilde{T}_i\right)_n$$

$$\left(-r\tilde{T}_{i-1}+q\tilde{T}_i-r\tilde{T}_{i+1}\right)_{n+1} = \left(\tilde{T}_i\right)_n$$

$$\text{where } r=\frac{\Delta\tilde{t}}{(\Delta\tilde{x})^2} \text{ and } q=1+2r$$

3 points at n+1 link to 1 point at n!

BC/Matrix Form

BC: at $x = 0, \quad \tilde{T} = 1$ (Dirichlet BC)

BC: at $x = 0, \quad \tilde{T}_{mx} = \tilde{T}_{mx-1}$ (Backward - Neumann BC)

$$\underbrace{\begin{array}{ccccccc} \tilde{T}_0 & & & & & & \\ -r\tilde{T}_0 & +q\tilde{T}_1 & -r\tilde{T}_2 & & & & \\ & -r\tilde{T}_1 & +q\tilde{T}_2 & -r\tilde{T}_3 & & & \\ & & -r\tilde{T}_2 & +q\tilde{T}_3 & -r\tilde{T}_4 & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & -r\tilde{T}_{mx-2} & +q\tilde{T}_{mx-1} & -r\tilde{T}_{mx} \\ & & & & & -\tilde{T}_{mx-1} & +\tilde{T}_{mx} \end{array}}_{at\ n+1} \begin{array}{cc} = & 1 \\ = & \tilde{T}_{1n} \\ = & \tilde{T}_{2n} \\ = & \tilde{T}_{3n} \\ = & \vdots \\ = & \tilde{T}_{mx-1,n} \\ = & 0 \end{array}$$

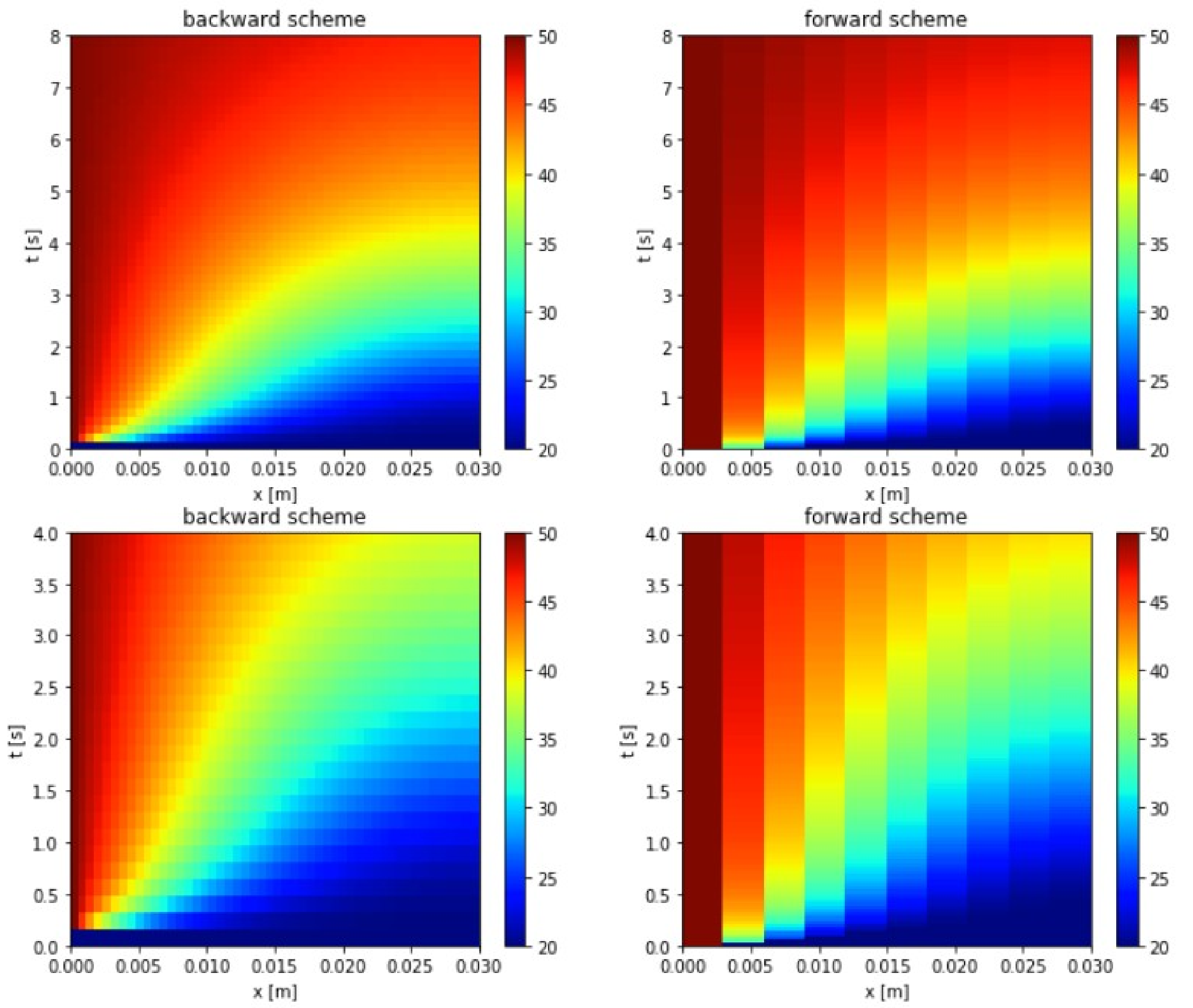
Backward Finite Difference: Matrix Form

Just need Thomas Algorithm to solve system of linear eqns

$$\begin{bmatrix} 1 & & & & & & \\ -r & q & -r & & & & \\ & -r & q & -r & & & \\ & & -r & q & -r & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & -r & q & -r \\ & & & & & -1 & 1 \end{bmatrix} \underbrace{\begin{bmatrix} \tilde{T}_0 \\ \tilde{T}_1 \\ \tilde{T}_2 \\ \tilde{T}_3 \\ \vdots \\ \tilde{T}_{mx-1} \\ \tilde{T}_{mx} \end{bmatrix}}_{at\ n+1} = \begin{bmatrix} 1 \\ \tilde{T}_{1n} \\ \tilde{T}_{2n} \\ \tilde{T}_{3n} \\ \vdots \\ \tilde{T}_{mx-1,n} \\ 0 \end{bmatrix}$$

Results

Which one is the backward scheme?



Crank-Nicolson (second order in time)

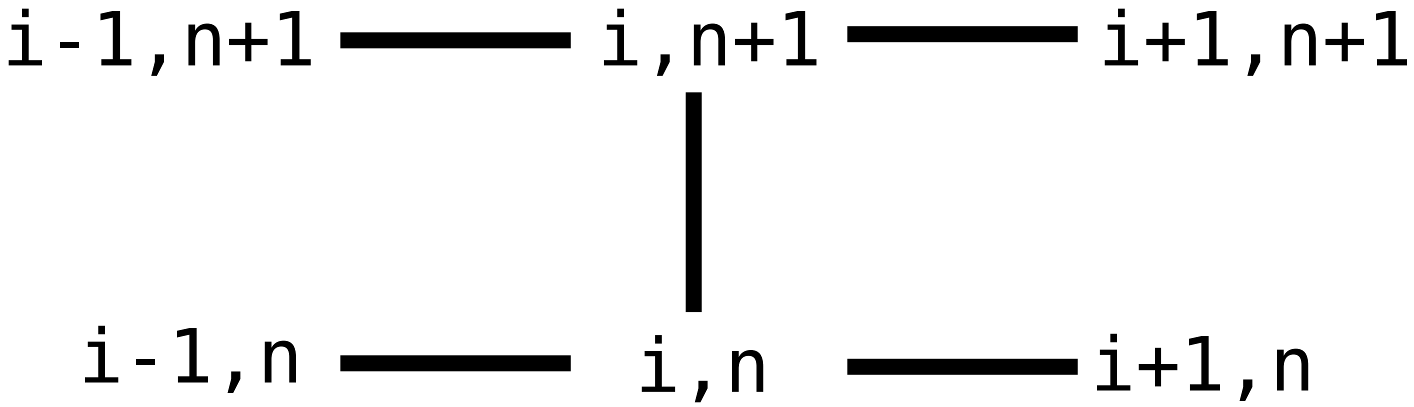
$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = F_i^n \left(u,\ x,\ t,\ \frac{\partial u}{\partial x},\ \frac{\partial^2 u}{\partial x^2} \right) \qquad \text{(forward Euler)}$$

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = F_i^{n+1} \left(u,\ x,\ t,\ \frac{\partial u}{\partial x},\ \frac{\partial^2 u}{\partial x^2} \right) \qquad \text{(backward Euler)}$$

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = \qquad \text{(Crank–Nicolson)}$$

$$\frac{1}{2} \left[F_i^{n+1} \left(u,\ x,\ t,\ \frac{\partial u}{\partial x},\ \frac{\partial^2 u}{\partial x^2} \right) + F_i^n \left(u,\ x,\ t,\ \frac{\partial u}{\partial x},\ \frac{\partial^2 u}{\partial x^2} \right) \right]$$

Crank-Nicolson scheme



We need 3 points at n and 3 points at n+1 at the same time to predict 1 point at i and n+1

Crank-Nicolson is unconditionally stable, but can contain some oscillation if $\dfrac{\alpha \Delta t}{(\Delta x)^2} > 0.5$

Crank-Nicolson Scheme

$$\frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2}$$

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} =$$

$$\frac{\alpha}{2(\Delta x)^2}\left(\left(u_{i+1}^{n+1} - 2u_i^{n+1} + u_{i-1}^{n+1}\right) + \left(u_{i+1}^{n} - 2u_i^{n} + u_{i-1}^{n}\right)\right)$$

Crank-Nicolson is the average between forward and backward scheme

Crank-Nicolson Scheme

Dropping the tilde sign (but still means dimensionless), we get

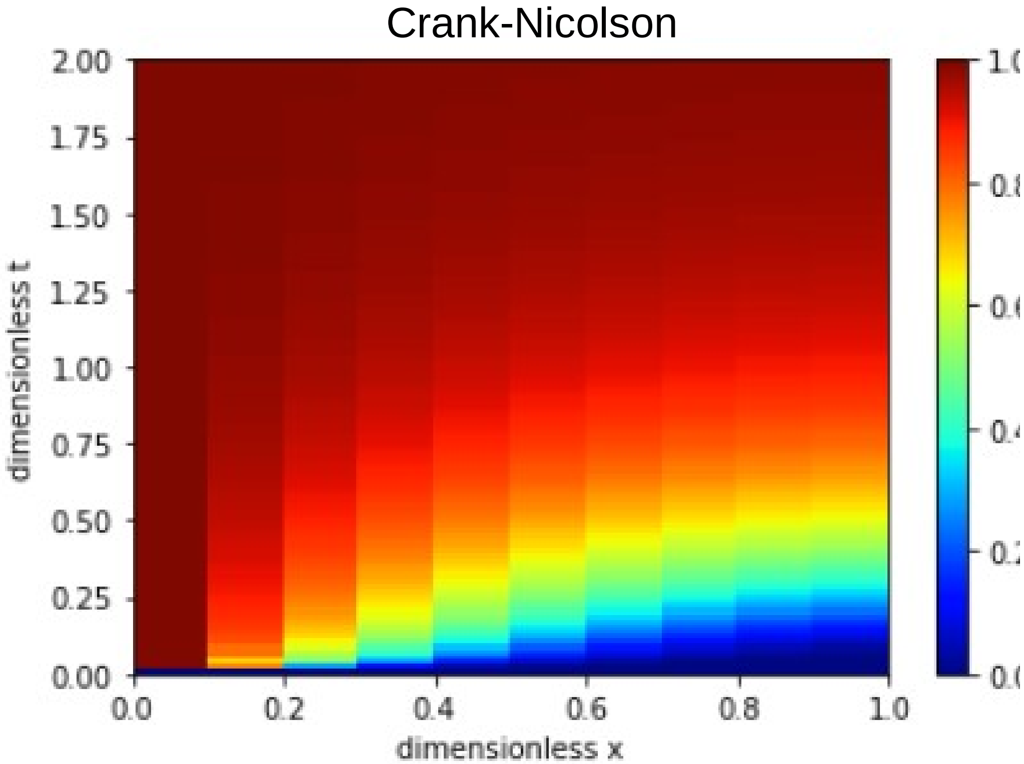
$$
\begin{bmatrix}
1 & & & & & & \\
-r & q & -r & & & & \\
 & -r & q & -r & & & \\
 & & -r & q & -r & & \\
 & & & \ddots & \ddots & \ddots & \\
 & & & & -r & q & -r \\
 & & & & 0.5 & -2 & 1.5
\end{bmatrix}
\underbrace{\begin{bmatrix} T_0 \\ T_1 \\ T_2 \\ T_3 \\ \vdots \\ T_{mx-1} \\ T_{mx} \end{bmatrix}}_{at\ n+1}
=
\begin{bmatrix}
1 & & & & & & \\
r & p & r & & & & \\
 & r & p & r & & & \\
 & & r & p & r & & \\
 & & & \ddots & \ddots & \ddots & \\
 & & & & r & p & r \\
0 & \cdots & & & & \cdots & 0
\end{bmatrix}
\underbrace{\begin{bmatrix} 1 \\ T_1 \\ T_2 \\ T_3 \\ \vdots \\ T_{mx-1} \\ T_{mx} \end{bmatrix}}_{at\ n}
$$

Note that RHS is just a column matrix that can be evaluated right away (everything is defined at n, not n+1)

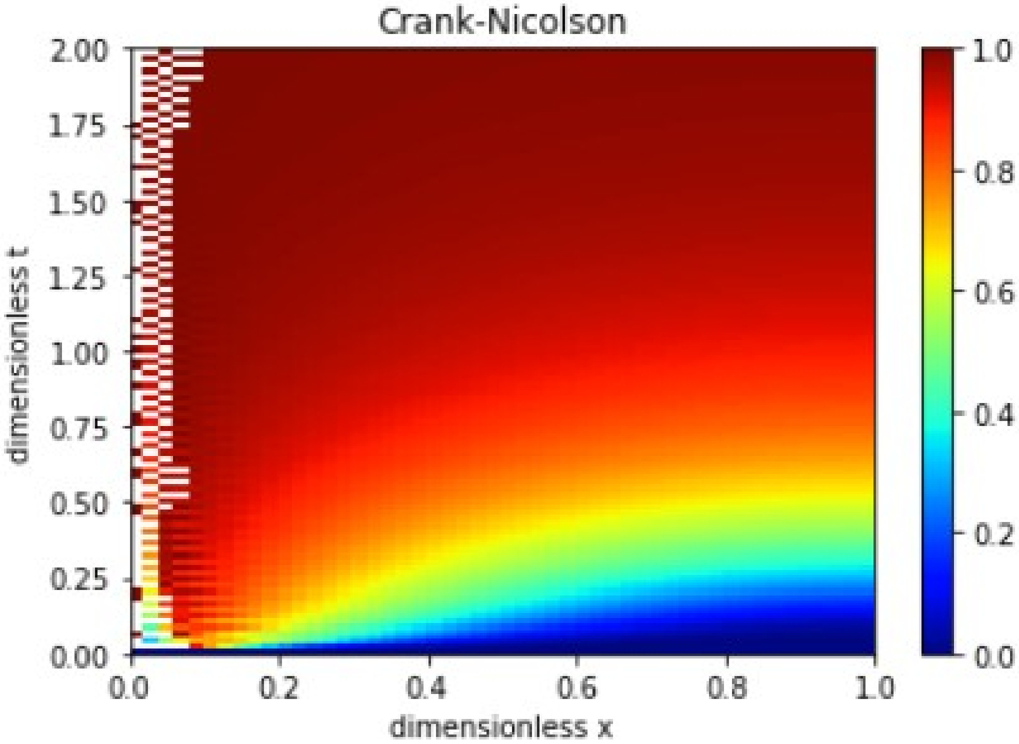
Crank-Nicolson with/without oscillation

Some oscillation occur when CFL is not satisfied

When CFL is satisfied



When CFL is not satisfied



2D Transient Heat Transfer

$$\frac{\partial \tilde{T}}{\partial \tilde{t}} = \nabla^2 \tilde{T}$$

Dropping tilde sign but still means dimensionless, we get

$$\frac{\partial T}{\partial t} = \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2}$$

Discretized form

$$\frac{\partial^2 T}{\partial x^2} \approx \frac{T_{i-1} - 2T_i + T_{i+1}}{(\Delta x)^2} \qquad \frac{\partial^2 T}{\partial y^2} \approx \frac{T_{j-1} - 2T_j + T_{j+1}}{(\Delta y)^2}$$

2D Transient Heat Transfer...

Let $\Delta y = \Delta x$

We have

$$(T_{n+1} - T_n)_{i,j} =$$
$$r(T_{i-1} - 2T_i + T_{i+1})_{j,n+1} + r(T_{j-1} - 2T_j + T_{j+1})_{i,n+1}$$

Rearranging (put n on rhs, n+1 on lhs), we have

$$\underbrace{(1 + 4r)T_{i,j} - rT_{i-1,j} - rT_{i+1,j} - rT_{i,j-1} - rT_{i,j+1}}_{n+1} = \underbrace{T_{i,j}}_{n}$$

2D Transient Heat Transfer...

By defining $q = 1 + 4r$, we have

$$\underbrace{qT_{i,j} - rT_{i-1,j} - rT_{i+1,j} - rT_{i,j-1} - rT_{i,j+1}}_{n+1} = \underbrace{T_{i,j}}_{n}$$

We need 5 points (up/down/left/right/center) at n+1 and 1 point at n, for the equation for the point at i,n+1

We need to put it into a matrix form! How?

2D Transient heat: BC

$$\left.\frac{\partial T}{\partial x}\right|_{x=0} = 0 \quad \text{or} \quad T_{0,j} = T_{1,j}$$

$$\left.\frac{\partial T}{\partial x}\right|_{x=1} = 0 \quad \text{or} \quad T_{N,j} = T_{N-1,j}$$

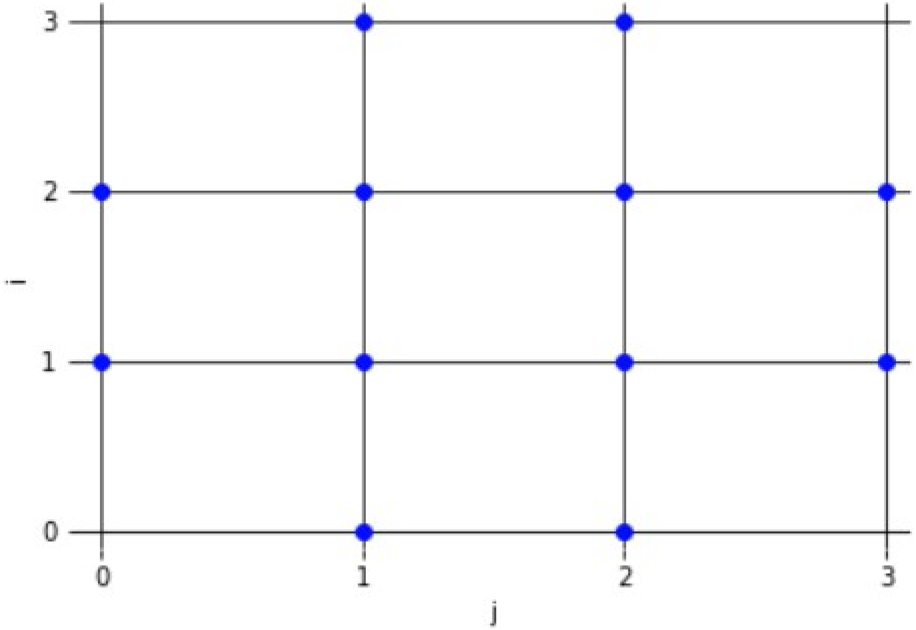
$$\left.\frac{\partial T}{\partial y}\right|_{y=0} = 0 \quad \text{or} \quad T_{i,0} = T_{i,1}$$

$$\left.\frac{\partial T}{\partial y}\right|_{y=1} = 0 \quad \text{or} \quad T_{i,N} = T_{i,N-1}$$

No heat loss at any end

Starts with 4x4 mesh

We do not need (and must not have) equation for the corner point



How many times do we use the equation below?

$$\underbrace{(1+4r)T_{i,j} - rT_{i-1,j} - rT_{i+1,j} - rT_{i,j-1} - rT_{i,j+1}}_{n+1} = \underbrace{T_{i,j}}_{n}$$

2D Transient Heat: Matrix Form (4x4 mesh)

$$
\begin{bmatrix}
1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-r & 0 & -r & q & -r & 0 & 0 & -r & 0 & 0 & 0 & 0 \\
0 & -r & 0 & -r & q & -r & 0 & 0 & -r & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -r & 0 & 0 & -r & q & -r & 0 & -r & 0 \\
0 & 0 & 0 & 0 & -r & 0 & 0 & -r & q & -r & 0 & -r \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1
\end{bmatrix}
\begin{bmatrix}
T_{0,1} \\ T_{0,2} \\ T_{1,0} \\ T_{1,1} \\ T_{1,2} \\ T_{1,3} \\ T_{2,0} \\ T_{2,1} \\ T_{2,2} \\ T_{2,3} \\ T_{3,1} \\ T_{3,2}
\end{bmatrix}_{n+1}
=
\begin{bmatrix}
0 \\ 0 \\ 0 \\ T_{1,1} \\ T_{1,2} \\ 0 \\ 0 \\ T_{2,1} \\ T_{2,2} \\ 0 \\ 0 \\ 0
\end{bmatrix}_{n}
$$

2D Transient Heat: Matrix Form (NxN mesh)

- The discretized PDE in matrix form can be written in the block tri-diagonal matrix
- We must not have corner points, because we have no equation for them
  - In this example, we exclude the corner points

$$\begin{bmatrix} \vec{A}_{00} & \vec{A}_{01} & & & & & \\ \vec{A}_{10} & \vec{A}_{11} & \vec{A}_{12} & & & & \\ & \vec{A}_{21} & \vec{A}_{22} & \vec{A}_{23} & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & \vec{A}_{m-2,m-3} & \vec{A}_{m-2,m-2} & \vec{A}_{m-2,m-1} & \\ & & & & \vec{A}_{m-1,m-2} & \vec{A}_{m-1,m-1} & \vec{A}_{m-1,m} \\ & & & & & \vec{A}_{m,m-1} & \vec{A}_{m,m} \end{bmatrix} \begin{bmatrix} \vec{T}_{i,j=0} \\ \vec{T}_{i,j=1} \\ \vec{T}_{i,j=2} \\ \vdots \\ \vec{T}_{i,j=m-2} \\ \vec{T}_{i,j=m-1} \\ \vec{T}_{i,j=m} \end{bmatrix} = \begin{bmatrix} \vec{B}_{i,j=0} \\ \vec{B}_{i,j=1} \\ \vec{B}_{i,j=2} \\ \vdots \\ \vec{B}_{i,j=m-2} \\ \vec{B}_{i,j=m-1} \\ \vec{B}_{i,j=m} = [\vec{0}] \end{bmatrix}$$

2D Transient Heat: Sub-Matrix

$$\vec{A}_{00} = \begin{bmatrix} 1 & & \\ & \ddots & \\ & & 1 \end{bmatrix}$$

$\vec{A}_{00}$ is (N-2) x (N-2), to exclude the cornor points

Let’s go back a take a quick look on 4x4 case

2D Transient Heat: Sub-Matrix

$$\vec{A}_{01} = \begin{bmatrix} 0 & -1 & & & & 0 \\ 0 & & -1 & & & 0 \\ & & & \ddots & & \\ 0 & & & & & 0 \\ 0 & & & & -1 & 0 \end{bmatrix} = \vec{A}_{m,m-1}$$

$\vec{A}_{01}$ is (N-2) x (N), The first and the last column is to exclude the connection to the corner points (the one above the corner point)

2D Transient Heat: Sub-Matrix

$$\vec{A}_{10} = \begin{bmatrix} 0 & \cdots & 0 \\ -r & & \\ & \ddots & \\ & & -r \\ 0 & \cdots & 0 \end{bmatrix} = \vec{A}_{m-1,m}$$

Only for non-BC points near the edge

$\vec{A}_{10}$ is N x (N-2), the first and the last row of zeros is for left and right Neumann condition where the bottom point is not involved. This sub-matrix gives part of the equation for the non-BC points.

2D Transient Heat: Sub-Matrix (diagonal terms)

$$\vec{A}_{ii} = \begin{bmatrix} 1 & -1 & & & & & \\ -r & q & -r & & & & \\ & -r & q & -r & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & -r & q & -r \\ & & & & 1 & -1 \end{bmatrix} \quad i \in \{1, 2, 3, ..., m-1\}$$

$\vec{A}_{ii}$ is NxN where the first and the last row is for BC at the left and right edge points. This sub-matrix is just part of the equation for non-BC points.

2D Transient Heat: Sub-Matrix (off-diagonal)

$$\vec{A}_{21} = \begin{bmatrix} 0 & & \cdots & & 0 \\ & -r & & & \\ \vdots & & \ddots & & \vdots \\ & & & -r & \\ 0 & & \cdots & & 0 \end{bmatrix}$$

This matrix can be described as $\vec{A}_{ij}$

where $i = j-1, j+1$ for $j \in [2, m-2]$.

For $j = 1$ and $m-1$, use $\vec{A}_{10}$ and $\vec{A}_{m,m-1}$ instead

Creating the coefficient matrix: 2D Backward Heat

Involved Libraries

- scipy.sparse.diags: This is to create sparse sub-matrix that have diagonal and off-diagonal terms
- scipy.linalg.block_diag: This is to put several sub-matrix into a block diagonal matrix (put several block into the diagonal parts)

Creating the coefficient matrix: 2D Backward Heat...

Involved Libraries

- Scipy.sparse.bsr_matrix (block compressed row): This is to combine each off-diagonal blocks (A_{12}) togeters.
- scipy.sparse.linalg.splu: This is to create LU decomposition object for sparse matrix. Once object is created, object.solve(RHS) will give the answer

User define function: We need a function to change (back and forth) from NxN T-array to N**2 – 4 T-column vector.

2D Backward Heat: Calculation Step

0: Create initial condition for temperature, T-matrix

1: Create coefficient matrix

2: Do sparse LU decomposition of the coefficient matrix

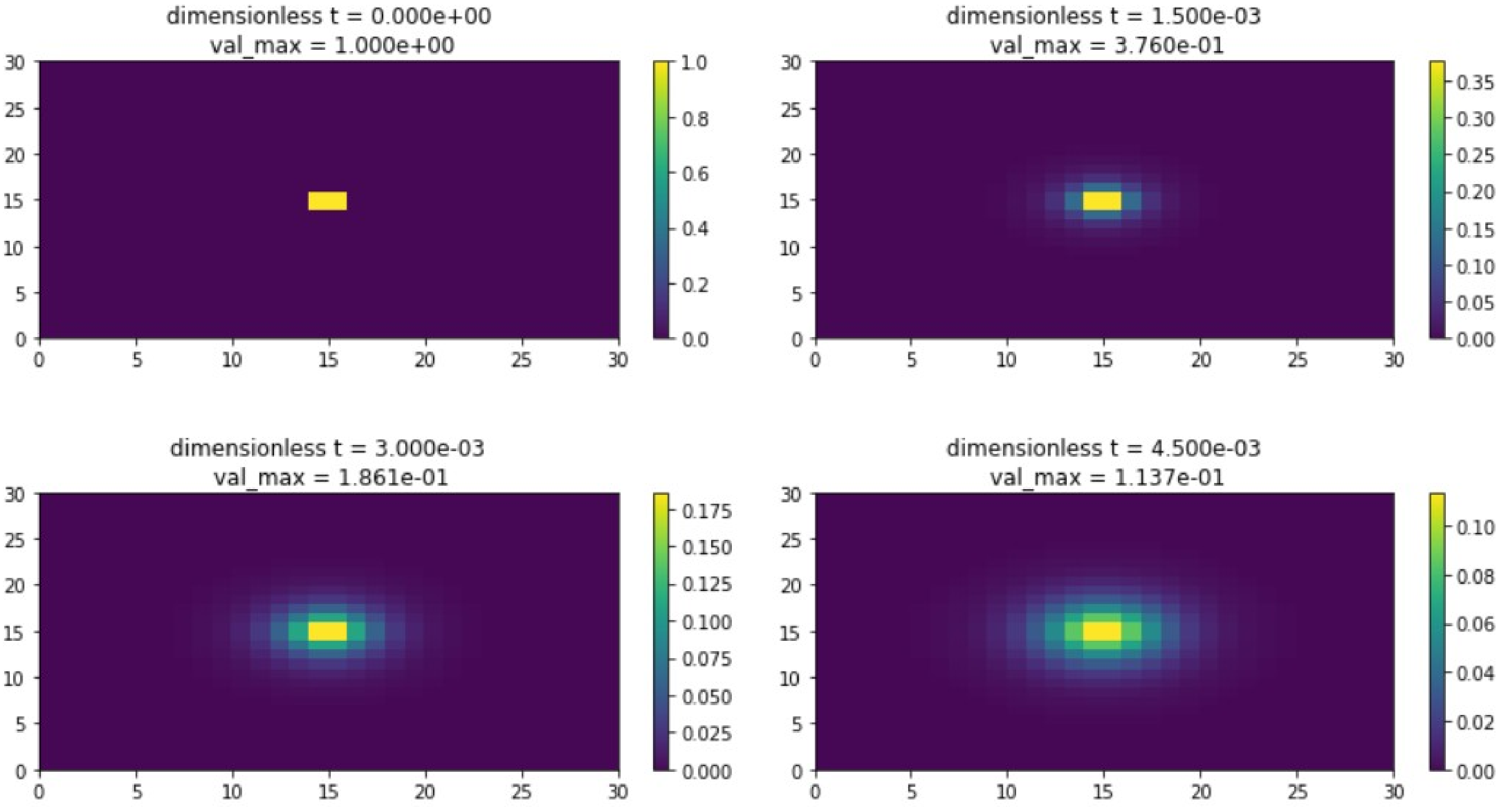
3: For-loop

- 3.1: Create RHS matrix
- 3.2: LU.solve(RHS) to get T-matrix
- 3.3: Store T-matrix and move to the next time step

4: Show result as an animation or 2D plot at each time slice

Result 2D transient heat (Relative view)

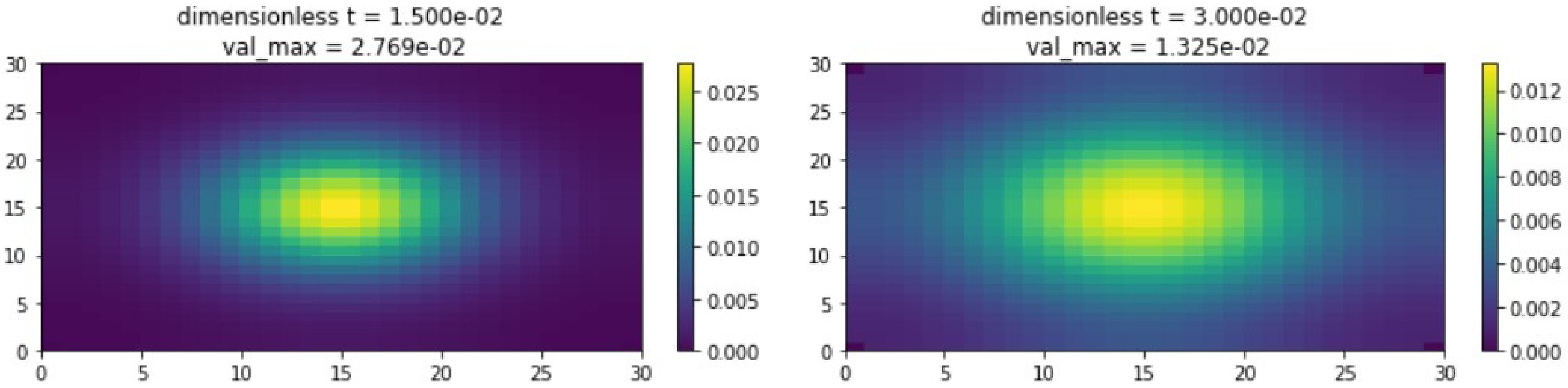
Maximum value decreases with time (but color is still yellow)



Even at 1 time step (top-right picture), many cell around the hot cell feel the heat

Result 2D transient heat (Relative view)

This is a backward scheme, so we can discretize more in space without the need to progress slowly in time.



See animation from L11_absolute_T.mp4   and   L11_relative_T.mp4

FEniCS: Finite Element Method to solve PDE

Outline

- Brief view on FEM/FEniCS

To run FEniCS smoothly, either

- Install Ubuntu virtual machine on windows or
- Install Ubuntu as a separate OS
- Then, install FEniCS on Ubuntu/virtual machine with Ubuntu

https://fenicsproject.org/pub/tutorial/html/._ftut1004.html
https://fenicsproject.org/qa/7316/print-values-numerical-solution-neumann-boundary-problem
https://fenicsproject.org/qa/6450/get-field-value-at-mesh-vertex

FEM (FEniCS): Example

$$-\nabla^2 u(x) = -6 \qquad -\frac{\partial^2 u}{\partial x^2} - \frac{\partial^2 u}{\partial y^2} = -6$$

$$\text{BC (Dirichlet) on every boundary: } 1 + x^2 + 2\,y^2$$

$$\Omega = [0,1] \times [0,1]$$

Once we define the problem (together with BC), we need to get the variational form

Multiplying the PDE by a function v, integrating the resulting equation over the domain Ω , and perform integration by parts of terms with second-order derivatives.

FEM with Dirichlet Boundary Condition:

$$
\begin{aligned}
-\Delta u(\mathbf{x}) &= f(\mathbf{x}), \qquad && \mathbf{x} \text{ in } \Omega \\
u(\mathbf{x}) &= u_D(\mathbf{x}), && \mathbf{x} \text{ on } \partial\Omega
\end{aligned}
$$

Δ is Laplace operator or ∇^2

$u(\mathbf{x})$ is unknown function

Ω is spatial domain

$\partial\Omega$ is the boundary of Ω

$$-\Delta u(\mathbf{x}) = f(\mathbf{x})$$

By transforming PDE into variational problem, we get,

$$-\int_\Omega (\Delta u) v dx = \int_\Omega f v dx$$ v is a test function

From integration by parts, we have

$$\nabla \cdot \left(a\vec{b}\right) = a\left(\nabla \cdot \vec{b}\right) + \vec{b} \cdot (\nabla a)$$

For $a = v$ and $b = \nabla u$, we have

$$\nabla \cdot \left(v(\nabla u)\right) = v(\nabla^2 u) + (\nabla u) \cdot (\nabla v)$$

Rearranging, we have

$$-v(\nabla^2 u) = (\nabla u) \cdot (\nabla v) - \nabla \cdot \left(v(\nabla u)\right)$$

FEM: Variational Form

From

$$-v(\nabla^2 u) = (\nabla u)\cdot(\nabla v) - \nabla\cdot\big(v(\nabla u)\big)$$

or

$$-\int_\Omega v(\nabla^2 u)dx = \int_\Omega \nabla u\cdot\nabla v dx - \int_\Omega \nabla\cdot(v\nabla u)dx$$

From divergence theorem

(change from volume to surface integration),

$$\iiint_\Omega (\nabla\cdot\vec{F})\,dV = \oint_{\partial\Omega} (\vec{F}\cdot\hat{n})\,dS$$

FEM: Variational Form...

$$-\int_{\Omega} v(\nabla^2 u)dx = \int_{\Omega} \nabla u \cdot \nabla v dx - \oint_{\partial\Omega} (v\nabla u) \cdot \hat{n} dS$$

$$\nabla u \cdot \hat{n} = \frac{\partial u}{\partial n}$$

n is the unit normal vector pointing outward, so we have

$$-\int_{\Omega} v(\nabla^2 u)dx = \int_{\Omega} \nabla u \cdot \nabla v dx - \oint_{\partial\Omega} \frac{\partial u}{\partial n} v dS$$

http://sfepy.org/doc-devel/solving_pdes_by_fem.html#sec-solving-pdes-fem
https://en.wikipedia.org/wiki/Integration_by_parts

FEM: Variational Form...

At the boundary, we have

$$u(\mathbf{x}) = u_D(\mathbf{x})$$

By requring v to vanish (be zero) at $\partial\Omega$, we have

$$\int_\Omega \nabla u \cdot \nabla v dx = \int_\Omega f v dx$$

This is the weak form or variational form of the BVP

FEM: Test/Trial space

This equation should hold for all test function v in some suitable space $\hat{V}$ (so-called test space). The solution u is in function space V (so-called trial space). The test space $\hat{V}$ and the trial space can be defined as

$$V = \{v \in H^1(\Omega) : v = u_D \text{ on } \partial\Omega\}$$

$$\hat{V} = \{v \in H^1(\Omega) : v = 0 \text{ on } \partial\Omega\}$$

$H^1(\Omega)$ is the Sobolev space containing v such that

v^2 and $|\nabla v|^2$ have finite integrals over Ω (functions are continuous)

FEM - FEniCS

After obtaining the variational form of

$$\int_{\Omega} \nabla u \cdot \nabla v dx = \int_{\Omega} f v dx \ \forall v \in \hat{V}$$

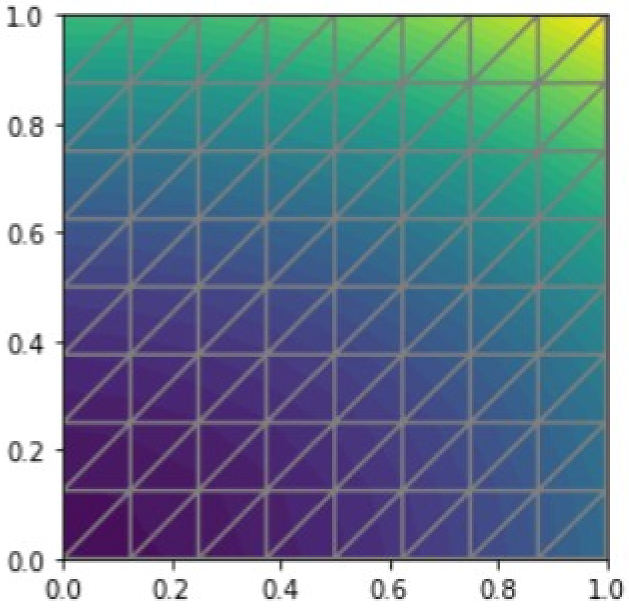
Then, FEniCS can solve it by using

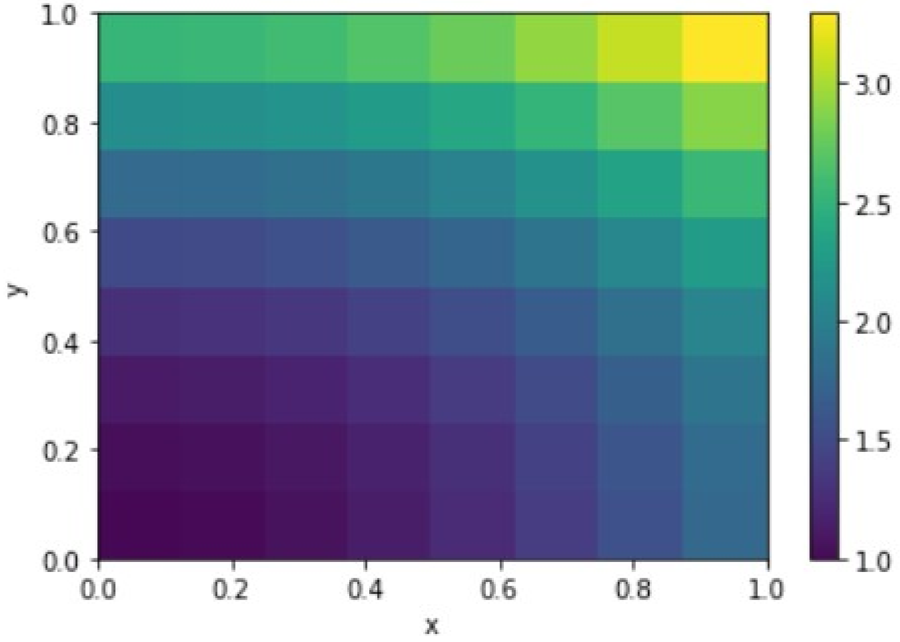
```
a = fn.dot(fn.grad(u), fn.grad(v))*fn.dx
L = f*v*fn.dx
u = fn.Function(V)
fn.solve(a == L, u, bc)
```

FEniCS Solution for the example question

$$-\nabla^2 u(x) = -6 \qquad -\frac{\partial^2 u}{\partial x^2} - \frac{\partial^2 u}{\partial y^2} = -6 \qquad \Omega = [0,1] \times [0,1]$$

$\text{BC (Dirichlet) on every boundary: } 1 + x^2 + 2\,y^2$

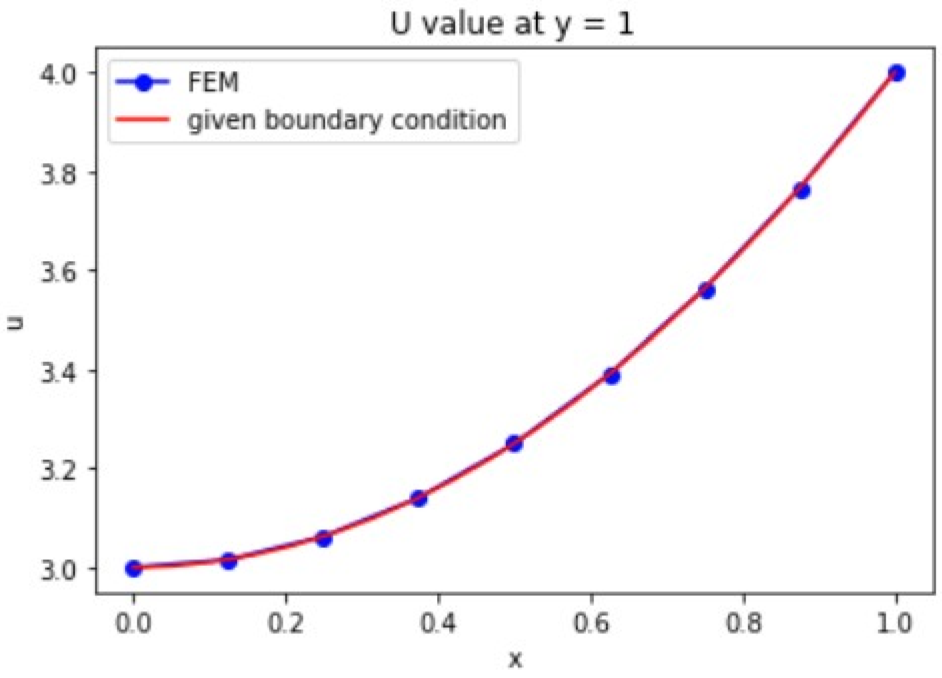


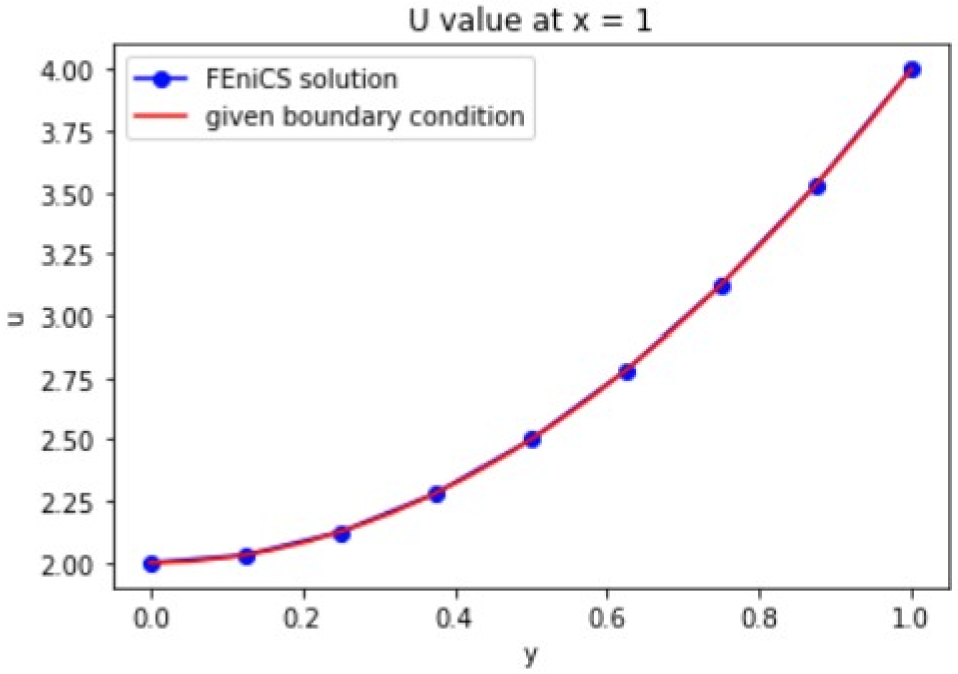


https://fenicsproject.org/pub/tutorial/html/._ftut1004.html

FEniCS Solution: Check answer on BC

FEniCS answer match with the given boundary conditions





Check if LHS == RHS

At x = 0.5 and y = 0, calculate LHS and check if it is -6

- Use 8 points forward scheme to estimate $\frac{\partial^2 u}{\partial y^2}$

$$\frac{\partial^2 u}{\partial y^2} = 4.00000000000027285$$

- Use 9 points central scheme to estimate $\frac{\partial^2 u}{\partial x^2}$

$$\frac{\partial^2 u}{\partial x^2} = 1.99999999999999432$$

$$\therefore -\frac{\partial^2 u}{\partial x^2} - \frac{\partial^2 u}{\partial y^2} == -6$$

https://en.wikipedia.org/wiki/Finite_difference_coefficient