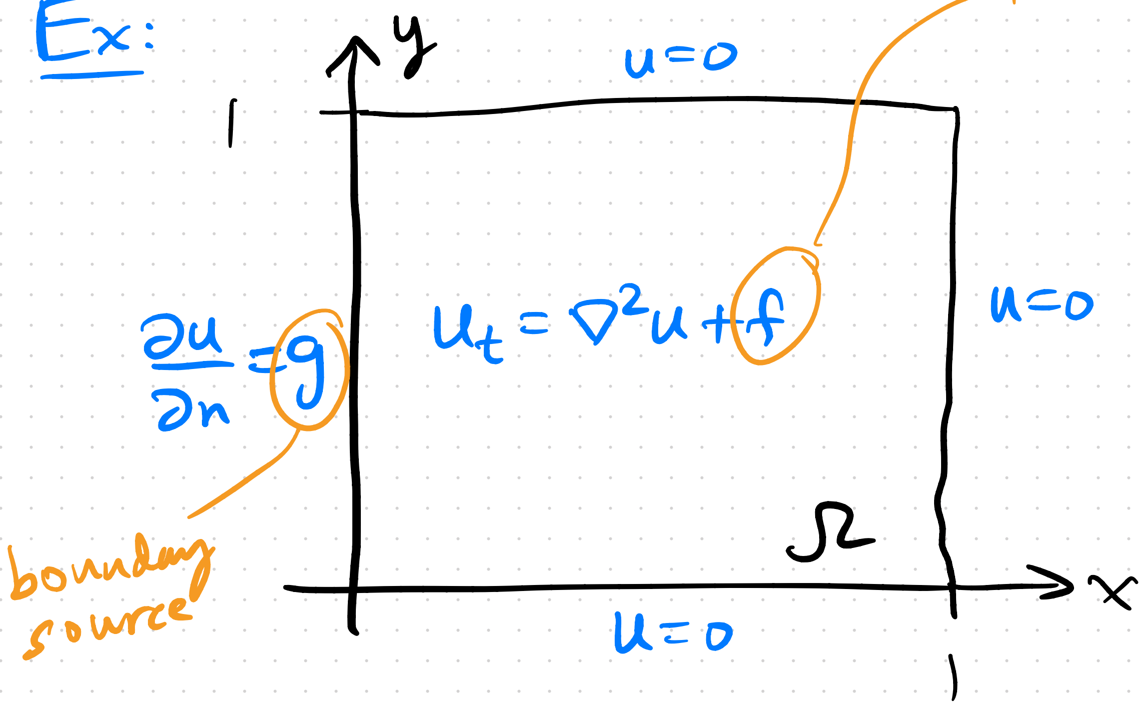
Solve time-dependent heat equation

29 Feb. 2024

Ex:



model of heat conduction in a plate, with heat flowing in on left side

initial condition: $u(t, 0) = 0$

what this example will demonstrate/derive:

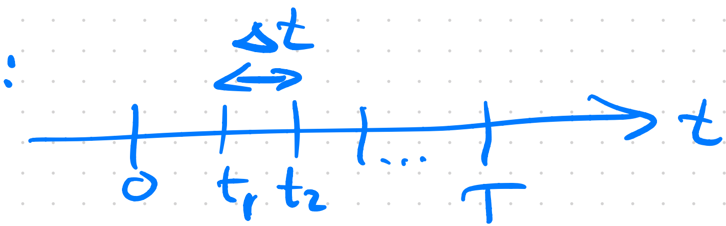
1. weak forms for explicit (forward Euler) and implicit (backward Euler) time stepping
2. basic how-to for time-stepping
3. that explicit stepping has stability issues!
4. how the mass matrix and stiffness matrix are behind the scenes

forward Euler, and its weak form

$$u_t = \nabla^2 u + f$$

for simplicity: assume $f = f(x,y)$, $g=g(y)$ do not depend on t

- discretize time $t \in [0,T]$:



$$t_n = n\,\Delta t$$

- finite-difference for $u_t = \frac{\partial u}{\partial t}$:

$$\frac{u^n - u^{n-1}}{\Delta t} = \nabla^2 u^{n-1} + f$$

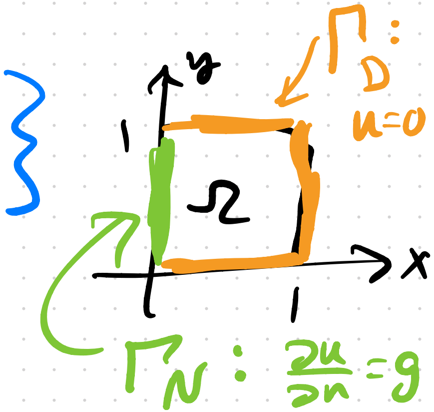
for $\quad u^n(x,y) \approx u(t_n, x, y)$

- clear denominators, multiply by v, and integrate:

$$\int_\Omega u^n v = \int_\Omega u^{n-1} v + \Delta t \int_\Omega (\nabla^2 u^{n-1})\, v + \Delta t \int_\Omega f v$$

- assume u^n, u^{n-1}, v are in

$$H^1_D(\Omega) = \{ w \in H^1(\Omega) : w|_{\Gamma_D} = 0 \}$$



- apply product rule and div. thm:

$$\int_\Omega u^n v = \int_\Omega u^{n-1} v + \Delta t \left(\int_{\partial\Omega} v \nabla u^{n-1} \cdot \hat{n}\, ds - \int_\Omega \nabla u^{n-1} \cdot \nabla v \right)$$

$$+ \Delta t \int_\Omega f v$$

- apply b.c.s to get weak form:

$$F^e := \int_\Omega u^n v - \int_\Omega u^{n-1} v + \Delta t \int_\Omega \nabla u^{n-1} \cdot \nabla v - \Delta t \int_\Omega f v - \Delta t \int_{\Gamma_N} g v = 0$$

practically Firedrake! ... see py/29feb/stepper.py

- at each time step we will solve this for u^n, starting with known $u^0 = u(0,x,y)$, the initial condition

backward Euler, and its weak form

$$u_t = \nabla^2 u + f$$

- finite-difference u_t:

$$\frac{u^n - u^{n-1}}{\Delta t} = \nabla^2 u^n + f$$

$$\iff \quad u^n = u^{n-1} + \Delta t\, \nabla^2 u^n + \Delta t\, f$$

what's changed versus forward Euler?

- multiply by v and integrate:

$$\int_\Omega u^n v = \int_\Omega u^{n-1} v + \Delta t \int_\Omega (\nabla^2 u^n) v + \Delta t \int_\Omega f v$$

- assume u^n, u^{n-1}, v are in $H^1_D(\Omega)$, apply product rule and div. thm:

$$\int_\Omega u^n v = \int_\Omega u^{n-1} v + \Delta t \left(\int_{\partial\Omega} v \nabla u^n \cdot \hat{n} \, ds - \int_\Omega \nabla u^n \cdot \nabla v \right) + \Delta t \int_\Omega f v$$

- apply b.c.s to get final weak form!

$$\left. \begin{aligned} F^i := \int_\Omega u^n v - \int_\Omega u^{n-1} v + \Delta t \int_\Omega \nabla u^n \cdot \nabla v \end{aligned} \right\} \text{compare } F^e$$

$$+ \Delta t \int_\Omega f v + \Delta t \int_{\Gamma_N} g v = 0$$

demo code

py/29feb/stepper.py

do ~~demo!~~

- produces Paraview files
  1. result.pvd (steps $u^0, u^1, \ldots, u^N$ suitable for animation)
  2. sources.pvd (f, g for visualization)
- play with $m =$ (spatial resolution), $N =$ (# of timesteps), $\Delta t =$ (timestep duration)

Under the hood ... why unstable?

- assume sources f, g are zero for simplicity
- explicit weak form:

$$F^e = \int_\Omega u^n v - \int_\Omega u^{n-1} v + \Delta t \int_\Omega \nabla u^{n-1} \cdot \nabla v$$

- recall φ_j is the hat function at node (x_j, y_j)

def:

$$M_{ij} = \int_\Omega \varphi_i \varphi_j$$ is the <u>mass matrix</u>

$$A_{ij} = \int_\Omega \nabla \varphi_i \cdot \nabla \varphi_j$$ is the <u>stiffness matrix</u>

- Firedrake's assembly process turns $F^e == 0$ into

$$M\vec{u}^n - M\vec{u}^{n-1} + \Delta t\, A\vec{u}^{n-1} \overset{\circledast}{=} 0$$

where $\vec{u}^n \in \mathbb{R}^q$, $\quad q = (\#$ of nodes in mesh$)$

- of course: $\vec{u}^n \cong u(t_n, x, y)$ new values

$\vec{u}^{n-1} \cong u(t_{n-1}, x, y)$ old values

- so solve(F==0, unew, ...) in explicit case solves linear system $\circledast$ for $\vec{u}^n$

- so forward/backward Euler become matrix iterations:

$$F^e = 0 \iff M\vec{u}^n - M\vec{u}^{n-1} + \Delta t A\vec{u}^{n-1} = 0$$

$$\iff \vec{u}^n = \underbrace{(I - \Delta t M^{-1}A)}_{= Q^e} \vec{u}^{n-1}$$

$$F^i = 0 \iff M\vec{u}^n - M\vec{u}^{n-1} + \Delta t A\vec{u}^n = 0$$

$$\iff \vec{u}^n = \underbrace{(I + \Delta t M^{-1}A)^{-1}}_{= Q^i} \vec{u}^n$$

<u>lemma:</u> the iteration $\vec{w}^n = Q\vec{w}^{n-1}$ will cause some mode (some vector $\vec{w}^0$) to explode exponentially if and only if there is an eigenvalue of Q with magnitude exceeding 1:

$$\left(\begin{array}{l}\vec{w}^n \text{ can} \\ \text{explode} \\ \text{exponentially}\end{array}\right) \iff \left(\begin{array}{l}\text{there is } \vec{x} \neq 0 \text{ so that} \\ Q\vec{x} = \lambda\vec{x}, \\ \text{with } |\lambda| > 1\end{array}\right)$$

- this explains, quantitatively, our instability:
(explicit time-stepping is observed to be unstable)

$$\Longleftrightarrow \left(Q^e \text{ has } |\lambda| > 1 \right)$$

$$\Longleftrightarrow \left(I - \Delta t M^{-1} A\right) \vec{x} = \lambda \vec{x} \quad \& \quad |\lambda| > 1$$

$$\Longleftrightarrow M^{-1} A \vec{x} = \left(\frac{1-\lambda}{\Delta t}\right) \vec{x} \quad \& \quad |\lambda| > 1$$

$$\Longleftrightarrow M^{-1} A \vec{x} = \alpha \vec{x} \quad \& \quad |1 - \alpha \Delta t| > 1$$

$$\Longleftrightarrow \quad \underbrace{M^{-1}A\vec{x} = \alpha\vec{x} \quad \& \quad 1-\alpha\Delta t < -1}$$

uses fact that $M^{-1}A$ is similar to an SPD matrix, so α is real

$\Longleftrightarrow$ $M^{-1}A$ has an eigenvalue α so that

$$\alpha > 2/\Delta t$$

- note eigenvalues of $M^{-1}A$ are entirely determined by the spatial mesh

- by contrast:

  eigenvalues of $Q^{i} = (I + \Delta t \, M^{-1} A)^{-1}$

  are all less than 1

- so: implicit stepping is stable for any $\Delta t > 0$