

DICACIM programme A.Y. 2023–24

Numerical Methods for Partial Differential Equations

Numerical solution of parabolic problems

Paola Gervasio

DICATAM, Università degli Studi di Brescia (Italy)

uniBS, May, 2024



UNIVERSITÀ
DEGLI STUDI
DI BRESCIA

Convergence results for the θ -method

If u_0 , f , and u are sufficiently regular, it holds:

$$\|u(t^n) - u_h^n\|_{L^2(\Omega)}^2 + 2\alpha\Delta t \sum_{k=1}^n \|u(t^k) - u_h^k\|_{H^1(\Omega)}^2 \leq C(u_0, f, u)(\Delta t^{q(\theta)} + h^{2r})$$

where

$$q(\theta) = \begin{cases} 2 & \text{if } \theta \neq 1/2 \\ 4 & \text{if } \theta = 1/2, \end{cases}$$

and r is the local polynomial degree.

It follows that, if h is taken small enough (such that $h^{2r} < \Delta t^{q(\theta)}$), it holds

$$\|u(T) - u_h^{N_t}\|_{L^2(\Omega)} \sim \begin{cases} \Delta t & \text{Euler } (\theta = 0, \theta = 1) \\ \Delta t^2 & \text{Crank-Nicolson } (\theta = 1/2) \end{cases}$$



Problem 1 ($d = 1$)

Set $\Omega = (0, 1)$ and $T = 1$.

1. **Approximate** the solution u of the parabolic problem

$$\begin{cases} \frac{\partial u}{\partial t} - \frac{\partial^2 u}{\partial x^2} = \sin(x)(\cos(t) - \sin(t)) & (x, t) \in \Omega \times (0, T) \\ u(0, t) = 0, u(1, t) = \sin(1) \cos(t) & t \in (0, T) \\ u(x, 0) = \sin(x) & x \in \Omega \end{cases}$$

by FEM- \mathbb{P}_1 in space and θ -method in time, with $\theta \in \{0, 0.5, 1\}$.

2. The exact solution is $u(x, t) = \sin(x) \cos(t)$, **verify the convergence order of the θ -method** by using three values for Ne (number of elements): $Ne = 125$ ($h = 0.008$), $Ne = 250$ ($h = 0.004$), $Ne = 500$ ($h = 0.002$).



FEM1d/fem_1d_heat_solver.m

```
>> help fem_1d_heat_solver
fem_1d_solver: solve  $du/dt - \mu d^2u/dx^2 + \sigma u = f$ 
                in  $\Omega$  x  $(t_0, T)$ 
with Dirichlet and/or Neumann boundary conditions
by either P1-fem or P2-fem on a uniform grid.

[nodes, un1]=fem_1d_heat_solver(geom, problem_data, ...
                                parameters)
```

```
Input: geom: struct with fields:
        geom.xa, geom.xb = end-points of  $\Omega$ 
        geom.t0, geom.tf = end-points of  $(t_0, T)$ 
problem_data: struct with coefficients and
              functions data
              .mu =  $\mu$  (positive constant)
```

...

```
Output: nodes = column array with the nodes of the mesh
        un1 = column array of the numerical solution
              at the final time T
```



Solution.

Write a script to:

1. define the geometry,
2. define the problem data,
3. define the discretization parameters,
4. call `fem_1d_heat_solver`,
5. plot the solution,
6. compute the errors by `fem_1d_errors`.

Different runs:

- $\theta = 1$ (BE), $N_e = 125$, $N_t = [20, 40, 80, 160]$
- $\theta = 1$ (BE), $N_e = 250$, $N_t = [20, 40, 80, 160]$
- $\theta = 1$ (BE), $N_e = 500$, $N_t = [20, 40, 80, 160]$
- $\theta = 1/2$ (CN), $N_e = 125$, $N_t = [20, 40, 80, 160]$
- $\theta = 1/2$ (CN), $N_e = 250$, $N_t = [20, 40, 80, 160]$
- $\theta = 1/2$ (CN), $N_e = 500$, $N_t = [20, 40, 80, 160]$
- $\theta = 0$ (FE), $N_e = 10$, $N_t = 500, 600$, $\Delta t < ch^2?$



Problem 2 ($d = 1$)

We consider a homogeneous, three meters long aluminium bar with uniform section. We are interested in simulating the evolution of the temperature in the bar starting from a suitable initial condition, by solving the heat equation

$$\rho C_p \frac{\partial u}{\partial t} - \nabla \cdot (k \nabla u) = f \quad \text{in } \Omega \times (t_0, T).$$

If we impose adiabatic conditions on the lateral surface of the bar (i.e. homogeneous Neumann conditions), and Dirichlet conditions at the end sections of the bar, the temperature only depends on the axial space variable (denoted by x). Thus the problem can be modeled by the one-dimensional heat equation with $f = 0$, completed by the initial condition at $t = t_0$ and by Dirichlet boundary conditions at the endpoints of the reduced computational domain $\Omega = (0, L)$ ($L = 3\text{m}$). Pure aluminium has thermal conductivity $k = 237 \text{ W/(m K)}$, density $\rho = 2700\text{kg/m}^3$ and specific heat capacity $c = 897 \text{ J/(kg K)}$, then its thermal diffusivity is $\mu = 9.786 \cdot 10^{-5} \text{ m}^2/\text{s}$. Finally we consider the initial condition $T(x, 0) = 500 \text{ K}$ if $x \in (1, 2)$, 250 K otherwise and the Dirichlet boundary conditions $T(0, t) = T(3, t) = 250 \text{ K}$. Simulate the evolution for 2000 seconds and $N_t = 100$, with $\theta = 0, 1, 0.5$ and compare the three solutions.

CN: small oscillation in correspondence of the discontinuity of the initial datum

FE: huge spurious oscillations occur;

BE: smooth solution.



$d \geq 2$: MATLAB pdetool

1. **create** a PDEModel object
2. define the **geometry** (Ω),
3. define the **data**: coefficientis, source term, initial condition,
4. define the **boundary conditions**,
5. built the **mesh** (\mathcal{T}_h),
6. **solve** the discrete problem (compute u_h^n),
7. **plot** the solution.

Remark. Solvepde calls the function ode15s for time discretization. ode15s implements adaptive multistep methods for stiff equations: you do not need to set Δt , it is computed automatically, in order to guarantee that the error w.r.t. the exact solution is about 10^{-3} . To change this value, for example to 10^{-4} :
`model.SolverOptions.RelativeTolerance = 1e-04;`
before to solve.



The heat equation

The heat equation reads:

$$\left\{ \begin{array}{ll} \rho C_p \frac{\partial u}{\partial t} - \nabla \cdot (k \nabla u) = f & \text{in } \Omega \times (t_0, T) \\ u = g_D & \text{on } \partial\Omega_D \times (t_0, T) \\ \mu \mathbf{n} \cdot \nabla u = g_N & \text{on } \partial\Omega_N \times (t_0, T) \\ \mu \mathbf{n} \cdot \nabla u + \alpha u = \alpha u_{ext} & \text{on } \partial\Omega_R \times (t_0, T) \\ u = u^0 & \text{in } \Omega \times \{t_0\} \end{array} \right. \quad \begin{array}{l} g_D = \text{external temperature} \\ g_N = \text{heat flux} \\ \text{convective heat transfer} \end{array}$$

data:

- ρ = mass density of the material,
- C_p = specific heat capacity,
- k = thermal conductivity,
- f = internal heat source,
- α = heat transfer coefficient,
- u_{ext} = temperature of the surrounding medium.



Problem 3

Let us consider a region $\Omega = R_1 \setminus (R_2 \cup R_3)$ of the plane with $R_1 = (0, 1) \times (0, 2)$, $R_2 = (0.3, 0.4) \times (0.3, 0.9)$, and $R_3 = (0.5, 0.6) \times (1.1, 1.7)$, that represent the vertical section of a piece of medium with two cavities.

The mass density of the medium is 7.2 kg/m^3 , the specific heat capacity is $C_p = 500 \text{ J/(kg K)}$, and the thermal conductivity is $k = 50 \text{ W/(m K)}$.

The left wall of the medium is in contact with a body at constant temperature $u = 373\text{K}$, the right wall is in contact with air at a constant temperature $u_{ext} = 293\text{K}$ (heat transfer coefficient is $\alpha = 25\text{W/(m}^2 \text{ K)}$), no heat exchange occurs through the other walls.

Simulate the evolution of the temperature inside the medium along the time interval $(0, 20)$ sec, provided that the initial temperature is $u_0 = 273.15\text{K}$ in the whole medium.



Set the geometry

```
model=createpde; % object PDEModel
R1=[3,4,0,1,1,0,0,0,2,2]';
R2=[3,4,0.3,0.4,0.4,0.3,0.3,0.3,0.9,0.9]';
R3=[3,4,0.5,0.6,0.6,0.5,1.1,1.1,1.7,1.7]';
gd=[R1,R2,R3];
ns=(char('R1','R2','R3'))';
sf='R1-R2-R3';
g=decsg(gd,sf,ns);
% import the geometry into the model
geometryFromEdges(model,g)
% draw the geometry
figure(1); clf
pdegplot(model,'EdgeLabels','on')
```



Data of the problem

Recall that the general equation reads:

$$m \frac{\partial^2 u}{\partial t^2} + d \frac{\partial u}{\partial t} - \nabla \cdot (c \nabla u) + au = f$$

and that the boundary conditions are:

$$\begin{aligned} \text{Dirichlet :} & \quad hu = r && \text{on } \partial\Omega_D \\ \text{Neumann :} & \quad \mathbf{n} \cdot (c \nabla u) + qu = g && \text{on } \partial\Omega_N \end{aligned}$$

```
% coefficients
rho=7.2; % kg/m^3
Cp=500; % J/(kg K)
k=50; % W/(m K)
specifyCoefficients(model,'m',0,'d',rho*Cp,'c',k,...
                    'a',0,'f',0);
% boundary conditions
applyBoundaryCondition(model,'dirichlet',...
                      'Edge',7,'u',373.15); % left
applyBoundaryCondition(model,'neumann',...
                      'Edge',1,'q',25,'g',25*293); % right
applyBoundaryCondition(model,'neumann',...
                      'Edge',1,'q',0,'g',0); % other edges
% initial condition
setInitialConditions(model,273.15);
```



Mesh and solution

```
% mesh
generateMesh(model, 'Hmax', 0.05, ...
                'GeometricOrder', 'quadratic');

% set a list of time-steps at which you want
% to draw the solution
tlist=0:1:20; % seconds

% solve
results=solvepde(model, tlist);

% compute the gradient of the temperature
[gradx, grady]=evaluateGradient(results);
```



Plot

```
% fixed bounds for the colorbar
clim=[min(min(results.NodalSolution)),...
      max(max(results.NodalSolution))];
figure(2); clf
nt=size(tlist,2); % number of time steps
for n=1:nt
    % extract the temperature at time t=tlist(n)
    u=results.NodalSolution(:,n);
    % plot
    pdeplot(model, 'XYData',u, 'Contour', 'on',...
            'FlowData', [-gradx(:,n), -grady(:,n)]);
    set(gca, 'CLim', clim)
    colormap('parula')
    set(gca, 'Xlim', [0,1], 'YLim', [0,2],...
          'PlotBoxAspectRatio', [1,2,1])
    title(['t=', num2str(tlist(n))]);
    pause(0.2)
end
```

