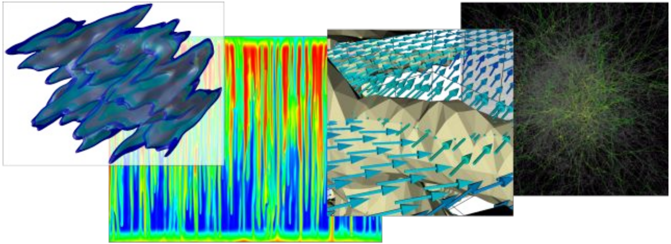
DUNE PDELab Tutorial 03

Conforming Finite Elements for a Nonlinear Heat Equation



Speaker:

Dominic Kempf
Scientific Software Center
Heidelberg University

PDE Problem

This tutorial extends the problem from tutorial 01 to the instationary setting:

$$
\begin{aligned}
\partial_t u - \Delta u + q(u) &= f && \text{in } \Omega \times \Sigma, \\
u &= g && \text{on } \Gamma_D \subseteq \partial\Omega, \\
-\nabla u \cdot \nu &= j && \text{on } \Gamma_N = \partial\Omega \setminus \Gamma_D, \\
u &= u_0 && \text{at } t = 0.
\end{aligned}
$$

Weak formulation: Find $u \in L_2(t_0, t_0 + T; u_g + V(t))$:

$$
\frac{d}{dt} \int_\Omega uv \, dx + \int_\Omega \nabla u \cdot \nabla v + q(u)v - fv \, dx + \int_{\Gamma_N} jv \, ds = 0 \qquad \begin{array}{l} \forall v \in V(t), \\ t \in \Sigma, \end{array}
$$

where $V(t) = \{v \in H^1(\Omega) \, : \, v = 0 \text{ on } \Gamma_D(t)\}$ and $H^1(\Omega) \ni u_g(t)|_{\Gamma_D} = g$.

Residual Forms

Introduce the residual forms: Find $u \in L_2(t_0, t_0 + T; u_g + V(t))$:

$$\frac{d}{dt} m^{\mathrm{L2}}(u, v) + r^{\mathrm{NLP}}(u, v) = 0 \quad \forall v \in V(t), t \in \Sigma.$$

with the *temporal residual form*

$$m^{\mathrm{L2}}(u, v) = \int_\Omega uv \, dx$$

and the *spatial residual form*

$$r^{\mathrm{NLP}}(u, v) = \int_\Omega \nabla u \cdot \nabla v + (q(u) - f)v \, dx + \int_{\Gamma_N} jv \, ds,$$

Method of Lines

1) Discretize in space with conforming finite elements, i.e. choose a finite-dimensional test space $V_h(t) \subset V(t)$.
2) Results in a system of ordinary differential equations (ODEs) for the time-dependent coefficient vector $z(t)$
3) Choose an appropriate method to integrate the ODE system
4) Other schemes can be implemented in PDELab as well, e.g. space-time methods

One Step θ Method

Subdivide time interval

$$\overline{\Sigma} = \{t^0\} \cup (t^0, t^1] \cup \ldots \cup (t^{N-1}, t^N]$$

Set $\Delta t^k = t^{k+1} - t^k$; Find $u_h^{k+1} \in U_h(t^{k+1})$ s.t.:

$$\frac{1}{\Delta t_k}(m_h^{\mathrm{L2}}(u_h^{k+1}, v; t^{k+1}) - m_h^{\mathrm{L2}}(u_h^k, v; t^k)) + \theta r_h^{\mathrm{NLP}}(u_h^{k+1}, v; t^{k+1}) + (1-\theta) r_h^{\mathrm{NLP}}(u_h^k, v; t^k) = 0 \quad \forall v \in V_h(t^{k+1})$$

Reformulated this formally corresponds in each time step to the nonlinear problem

$$\text{Find } u_h^{k+1} \in U_h(t^{k+1}) \text{ s.t.:} \quad r_h^{\theta,k}(u_h^{k+1}, v) + s_h^{\theta,k}(v) = 0 \quad \forall v \in V_h(t^{k+1})$$

where

$$r_h^{\theta,k}(u, v) = m_h^{\mathrm{L2}}(u, v; t^{k+1}) + \Delta t^k \theta r_h^{\mathrm{NLP}}(u, v; t^{k+1}),$$
$$s_h^{\theta,k}(v) = -m_h^{\mathrm{L2}}(u_h^k, v; t^k) + \Delta t^k (1-\theta) r_h^{\mathrm{NLP}}(u_h^k, v; t^k)$$

Runge-Kutta Methods

in Shu-Osher form:

1. $u_h^{(0)} = u_h^k$.
2. For $i = 1, \dots, s \in \mathbb{N}$, find $u_h^{(i)} \in u_{h,g}(t^k + d_i \Delta t^k) + V_h(t^{k+1})$:

$$\sum_{j=0}^{s} \Big[a_{ij} m_h \left(u_h^{(j)}, v; t^k + d_j \Delta t^k \right) + b_{ij} \Delta t^k r_h \left(u_h^{(j)}, v; t^k + d_j \Delta t^k \right) \Big] = 0 \qquad \forall v \in V_h(t^{k+1})$$

3. $u_h^{k+1} = u_h^{(s)}$.

An s-stage scheme is given by the parameters

$$A = \begin{bmatrix} a_{10} & \dots & a_{1s} \\ \vdots & & \vdots \\ a_{s0} & \dots & a_{ss} \end{bmatrix}, \quad B = \begin{bmatrix} b_{10} & \dots & b_{1s} \\ \vdots & & \vdots \\ b_{s0} & \dots & b_{ss} \end{bmatrix}, \quad d = (d_0, \dots, d_s)^T$$

Consider explicit and diagonally implicit schemes

Examples

- One step θ scheme (introduced above):

$$A = \begin{bmatrix} -1 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 1-\theta & \theta \end{bmatrix}, \quad d = (0,1)^T .$$

  Explicit/implicit Euler ($\theta \in \{0,1\}$), Crank-Nicolson ($\theta = 1/2$).
- Heun's second order explicit method

$$A = \begin{bmatrix} -1 & 1 & 0 \\ -1/2 & -1/2 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1/2 & 0 \end{bmatrix}, \quad d = (0,1,1)^T .$$

- Alexander's two-stage second order strongly S-stable scheme

$$A = \begin{bmatrix} -1 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & \alpha & 0 \\ 0 & 1-\alpha & \alpha \end{bmatrix}, \quad d = (0,\alpha,1)^T$$

  with $\alpha = 1 - \sqrt{2}/2$.
- Fractional step θ, three stage second order strongly A-stable

$$A = \begin{bmatrix} -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} \theta\theta' & 2\theta^2 & 0 & 0 \\ 0 & 2\theta\theta' & 2\theta^2 & 0 \\ 0 & 0 & \theta\theta' & 2\theta^2 \end{bmatrix}, \quad d = (0,\theta,1-\theta,1)^T$$

Note on Explicit Schemes

Example: Explicit Euler method ($\theta = 0$) results in: Find $u_h^{k+1} \in U_h(t^{k+1})$ s.t.:

$$m_h^{\mathrm{L2}}(u_h^{k+1}, v; t) - m_h^{\mathrm{L2}}(u_h^k, v; t) + \Delta t^k r_h^{\mathrm{NLP}}(u_h^k, v; t) = 0 \quad \forall v \in V_h(t^{k+1})$$

Appropriate spatial discretization results in diagonal mass matrix:

$$Dz^{k+1} = s^k - \Delta t^k q^k.$$

Requires stability condition for Δt^k.

Use the following algorithm:

i) While traversing the mesh assemble the vectors s^k and q^k separately and compute the maximum time step Δt^k.

ii) Form the right hand side $b^k = s^k - \Delta t^k q^k$ and "solve" the diagonal system $Dz^{k+1} = b^k$ (can be done in one step).

Extends to strong stability preserving Runge-Kutta methods

Realization in PDELab

1) The ini-file `tutorial03.ini` holds parameters controlling the execution.
2) Main file `tutorial03.cc` includes the necessary C++, DUNE and PDELab header files; contains `main` function; instantiates DUNE grid objects and calls the `driver` function
3) Function `driver` in file `driver.hh` instantiates the necessary PDELab classes and finally solves the problem.
4) File `nonlinearheatfem.hh` contains the local operator classes `NonlinearHeatFEM` and `L2` realizing the spatial and temporal residual forms.
5) File `problem.hh` contains a parameter class which encapsulates the user-definable part of the PDE problem.