Numerical methods for the heat equation

$$\begin{cases} \text{PDE}: \ \partial_t u(x,t) - \Delta u(x,t) = f(x,t) & (x,t) \in \Omega \times (0,T) \\ \text{b.c.}: \ u(x,t)\big|_{\partial\Omega} = g_D(x,t) & \text{"} \in \Gamma \times (0,T) \\ \text{i.c.}: \ u(x,0) = u_0(x) & x \in \Omega \end{cases}$$

We interpret $u(x,t)$ as a function of t which maps a time t to some function $\tilde{u}(x)$, $t \longmapsto \tilde{u}(x)$ $\quad \tilde{u}(t)(x) = u(x,t)$. $\quad u(t,\cdot) = u(t)$

$$\text{PDE}: \quad \frac{d}{dt} u(t) + \underset{\uparrow \text{ linear mapping}}{A} u(t) = f(t) \Rightarrow \text{ODE-setting where } -\Delta = A$$

$$\Rightarrow \quad \frac{d}{dt} u(t) = \tilde{f}(t, u(t)) = f(t) - Au(t).$$

- Try to use numerical methods for ODEs now.
- Implicit Euler:

$$\begin{cases} \tau = \frac{T}{N} & N \in \mathbb{N}, \quad t_n = n \cdot \tau. \\ \text{for } n = 1,2 \ldots \ ; \text{ compute} \\ \frac{u(t_{n+1}) - u(t_n)}{\tau} = \tilde{f}(t_{n+1}, u(t_{n+1})) = f(t_{n+1}) - Au(t_{n+1}). \end{cases}$$

- Combine with FEM

For each t_n, we consider the PDE :

$$u(x,t_{n+1}) - \tau\Delta u(x,t_{n+1}) = \tau f(x,t_{n+1}) + \tau u(x,t_n)$$

$\circledast_1$
$$\begin{cases} (Id - \tau\Delta)u(x,t_{n+1}) = \quad " \quad " . \\ \text{b.c} \quad u(x,t_{n+1}) = \underbrace{g(x,t_{n+1})}_{g^{n+1}} \text{ on } \partial\Omega \end{cases}$$

$\circledast_2$ Weak formulation   Find $u(t^{n+1}) \in \underbrace{H^1_{g^{n+1}}(\Omega)}_{V_{g^{n+1}}}$ s.t $\forall$ $v \in \underbrace{H^1_0(\Omega)}_{V_0}$ it holds

$$\underbrace{\int_\Omega u(t_{n+1})\, v \, dx}_{=:(u(t_n),v)_\Omega} + \tau \underbrace{(\nabla u(t_{n+1}), \nabla v)_\Omega}_{a(u,v)} = \tau (f^{n+1}, v)_\Omega + (u(t^n), v)_\Omega$$

Discrete weak formulation

$\circledast_3$ · Use FEM to solve $\circledast$, leading to

Find $u_h(t_{n+1}) \in V_{h,g^{n+1}}$ s.t. $\forall$ $v_h \in V_{h,0}$ it holds that

$$(u_h(t_{n+1}), v_h)_\Omega + \tau\, a(u_h(t_{n+1}), v_h) = \dots$$

$\circledast_4$. Linear algebra formulation.

$\{\varphi_i\}_{i=1}^{W=\dim V_{h,g^{n+1}}}$ $\{$ to simplify notation, $g=0\}$.

$$u_h(t_{n+1}) = \sum_{j=1}^{W} U_j^{n+1}\varphi_j(x) \quad , \quad u_h(t_n) = \sum_{j=1}^{W} U_j^n \varphi_j$$

- For $i = 1 \dots W$:

$$\underbrace{\sum_{j=1}^{W} U_j^{n+1} \overbrace{(\varphi_j, \varphi_i)_\Omega}^{M_{ij}}}_{(u_h^{n+1}, \varphi_i)_\Omega} + \tau \sum_{j=1}^{W} U_j^{n+1} \underbrace{a(\varphi_j,\varphi_i)}_{=: A_{ij}} = \tau \underbrace{(f^{n+1}, \varphi_i)}_{b_i^{n+1}} + \overbrace{\sum_{j=1}^{W} U_j^n \underbrace{(\varphi_j,\varphi_i)_\Omega}_{M_{ij}}}^{(u_h(t_n), \varphi_i)_\Omega}$$

$$U^{n+1} := (U_j^{n+1})_{j=1}^{W} \quad , \quad U^n :=$$

- Initial conditions. $\quad I_h u_0(x) = u_h^0(x) = \sum_{j=1}^{W} U_j^0 \varphi_j(x) \qquad U^0 = (U_j^0)_{j=1}^{W}$.

For $n = 1, 2 \dots$ $\forall i$ solve

$$MU^{n+1} + \tau A U^{n+1} = \tau b^{n+1} + MU^n.$$

General theta method

- Covers backward Euler, forward Euler and Crank-Nicolson method.

Given u^n and $\tau_n = t^{n+1} - t^n$ (for simplicity we assume a fixed time step $\tau = \tau_n \; \forall n$),

find u^{n+1} s.t.

$$\boxed{M\left(\frac{u_{n+1} - u_n}{\tau}\right) + \theta A u^{n+1} + (1-\theta) A u^n = \theta f^{n+1} + (1-\theta) f^n}$$

- Recall that θ-methods can be interpreted as trying to approximate u at $t_n + \theta\tau$, so the rhs of 1) could be replaced by $f^{n+\theta} = f(t_n + \theta \tau)$.

- θ-methods reduces to   Advantages | Disadvantages:

$\theta = 0$ explicit/forward Euler: First order in time ⊖, cheap to compute ⊕ but not A-stable ⊖, leads to severe time-step restrictions ⊖

$\theta = \frac{1}{2}$ Crank-Nicolson: Second order in time ⊕, implicit, A-stable ⊕ but not strongly A-stable $-/+$ / not very dissipative; very popular scheme

$\theta = 1$ implicit/backward Euler: First order ⊖, strongly A-stable ⊕, but ⊖ very dissipative ⊕⊖, not well-suited to compute stationary flows.

A quick glimpse at numerical methods for parabolic OCP

- State equation (SE)

distributed control

$$\partial_t y - \nabla\cdot(\mathcal{K}\nabla y) = \beta\,\omega \quad \text{in } \Omega\times(0,T) =: Q \quad \text{space-time cylinder}$$

$$-\mathcal{K}\partial_n y = 0 \quad \text{on } \partial\Omega\times(0,T)$$

$$y(0) = y_0 \quad \text{on } \Omega\times\{0\}$$

- Cost functional

distributed observation

$$J(y,\omega) = \frac{1}{2}\underbrace{\int_0^T\int_\Omega \left(y - y_d^Q\right)^2 dx\,dt}_{= \|y-y_d^Q\|_Q^2} + \frac{\gamma_1}{2}\overbrace{\int_\Omega \left(y(T) - y_d^\Omega\right)^2 dx}^{\|y(T)-y_d^\Omega\|_\Omega^2} + \frac{\gamma_2}{2}\overbrace{\int_0^T\int_\Omega \omega^2\,dx\,dt}^{\|\omega\|_Q^2}$$

end-time observations

- Example of a parabolic OCP: minimize $J(y,\omega)$ subject to SE and $\omega \in \mathcal{U}_{ad} \subseteq L^2(Q)$.

- Adjoint equation (AE)

$$-\partial_t p - \nabla\cdot(\mathcal{K}\nabla y) = \bar{y} - y_d^Q \quad \text{in } Q$$

$$-\mathcal{K}\partial_n p = 0 \quad \text{on } (0,T)\times\partial\Omega$$

$$p(T) = \gamma_1\left(\bar{y}(T) - y_d^\Omega\right)$$

$$\left(p(T),\varphi_i\right) = \gamma_1\left(\bar{y}_{\bar{\omega}}(T) - y_d^\Omega, \varphi_i\right)$$

$$\mathcal{M}\vec{p} = \gamma\mathcal{M}Y - , Y_d$$

- Optimality conditions

$$\left(\gamma_2\bar{\omega} + \beta p, \omega - \bar{\omega}\right)_Q \geq 0$$

$$\int_0^T\int_\Omega \left(\gamma_2\bar{\omega} + \beta p\right)\left(\omega - \bar{\omega}\right) dx\,dt \geq 0 \quad \forall \omega \in \mathcal{U}_{ad}$$

- Riesz representation of $f'(\omega)$ $\quad \left(f(\omega) = J(y(\omega),\omega)\right)$

$$\nabla f(\omega) = \gamma_2\omega + \alpha p$$

$$\left(y_d^\Omega, \varphi_i\right) = \left(\pi y_d^\Omega, \varphi_i\right)$$

· Optimize then discretize (OtD) approach (for $\gamma_1 = 0$, no end-time observation, unconstrained w)

OC: $(\gamma_3 \bar{w} + \beta p, w)_Q = 0 \iff (\gamma_2 \bar{w}(t) + \beta p(t), w(t))_\Omega = 0$ for a.e. t in $(0,T)$.

$\iff \gamma_2 \bar{w} + \beta p = 0$ for a.e. (x,t) in Q.

· We discretize (SE), (AE), (OC) using FEM and then backward Euler.

· FEM: $V_h \subseteq V = H^1(\Omega)$, $\{\varphi_i\}_{i=1}^{N_V}$ basis functions

$U_h \subseteq U = L^2(\Omega)$ $\{\psi_i\}_{i=1}^{N_U}$. Simplify by assuming $U_h = V_h$. $\psi_i = \varphi_i$

$$y_h(x,t) = \sum_{j=1}^{N_V} y_j(t)\,\varphi_j(x), \quad u_h(x,t) = \sum_{j=1}^{N_V} u_j(t)\,\varphi_j(x), \quad \vec{y}(t) = (y_i(t))_{i=1}^{N}, \ \vec{w}$$

· SE (discrete): $(\partial_t y_h, \varphi_i)_\Omega + (\kappa \nabla y_h, \nabla \varphi_i)_\Omega = (\beta u_h, \varphi_i)$ $i = 1, \dots N_V$.

· SE (algebraic): $M \dot{\vec{y}} + A \vec{y} = \beta M \vec{w}$ $\vec{y}(0) = (y_i^0)_{i=1}^{N}$

$M_{ij} = (\varphi_j, \varphi_i)_\Omega$ $A_{ij} = (\kappa \nabla \varphi_j, \nabla \varphi_i)_\Omega$ $V_h \ni J_h y(0) = \sum_{j=1}^{N} y_j^0 \varphi_j$

coefficients from an interpolation / projection of y_d^Q.

· AE (algebraic) $-M \dot{\vec{p}} + A^T \vec{p} = M(\vec{y} - \vec{y}_d^Q)$ $M \vec{p}(T) = M(\gamma \vec{y} - \vec{y}_d^\Omega)$

· OC (algebraic): $\gamma_2 M \vec{w} + \beta M^T p = 0$ for $t \in (0,T)$

$\vec{y}_d^\Omega$ collects dof-coeff from L^2-projection.

- Discretize in time using Backward Euler: $\tau = \frac{T}{N}$

- SE $\quad M\vec{y}^{\,n+1} + \tau A\vec{y}^{\,n+1} = \tau\beta M\vec{w}^{\,n+1} + M\vec{y}^{\,n} \qquad n = 0, \dots, N_t$

$$\left(\frac{M}{\tau} + A\right)\vec{y}^{\,n+1} - \frac{M}{\tau}\vec{y}^{\,n} - \beta M\vec{w}^{\,n+1} = 0$$

- Collect everything in big matrix system:

$$\underbrace{\begin{pmatrix} \frac{M}{\tau}+A & & & \\ -\frac{M}{\tau} & \frac{M}{\tau}+A & & \\ & \ddots & \ddots & \\ & & -\frac{M}{\tau} & \frac{M}{\tau}+A \end{pmatrix}}_{\mathcal{K} \in \mathbb{R}^{N_v N_t \times N_v N_t}} \underbrace{\begin{pmatrix} \vec{y}_1 \\ \vdots \\ \vec{y}_{N_t} \end{pmatrix}}_{\vec{Y} \in \mathbb{R}^{N_v \cdot N_t}} - \beta \begin{pmatrix} M & & \\ & \ddots & \\ & & M \end{pmatrix} \begin{pmatrix} \vec{w}^{\,1} \\ \vdots \\ \vec{w}^{\,N_t} \end{pmatrix} = \begin{pmatrix} M\vec{y}_0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

- How large is this system. $\quad A, M \in \mathbb{R}^{N_v \times N_v}$.

- Same discretization (BE) for AE: $-M\dot{\vec{p}} + A\vec{p} = M(\vec{y} - \vec{y}_d^a)$

- Time-discretization $\frac{M}{\delta}(\vec{p}^n - \vec{p}^{n+1}) + A^T\vec{p}^n = M(\vec{y}^n - (\vec{y}_d^a)^n)$ $\quad n = N-1, \dots, 1, 0.$

$$\begin{pmatrix} \frac{M}{\delta} + A^T & -\frac{M}{\delta} & & \\ & \ddots & \ddots & \\ & & \frac{M}{\delta} + A^T & -\frac{M}{\delta} \\ & & & \frac{M}{\delta} + A^T \end{pmatrix} \begin{pmatrix} \vec{p}^0 \\ \vdots \\ p^{N-1} \end{pmatrix} + \begin{pmatrix} M & & \\ & \ddots & \\ & & M \end{pmatrix} \begin{pmatrix} \vec{y}^0 \\ \vdots \\ \vec{y}^{N-1} \end{pmatrix}$$

$$= \begin{pmatrix} M(\vec{y}_d^a)^0 \\ \vdots \\ M(\vec{y}_d^a)^{N-1} \end{pmatrix}$$

$3 N_y \cdot N_t$

- OC: $\alpha_2 \begin{pmatrix} \alpha & & \\ & \ddots & \\ & & \alpha \end{pmatrix} \begin{pmatrix} \vec{u}^1 \\ \vdots \\ \vec{u}^{N_t} \end{pmatrix} + \beta \begin{pmatrix} M^T & & \\ & \ddots & \\ & & M^T \end{pmatrix} \begin{pmatrix} \vec{p}^1 \\ \vdots \\ \vec{p}^{N_t} \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}$

Projected gradient method

- Minimize reduced functional $f(u) := J(y(u), u)$ using the projected gradient method as before

Algorithm

- Initial control u_1
- For $k = 1, 2, \ldots$
  - Compute state y_k solving the forward heat equation
  - " adjoint state p_k solving the backward equation
  - Compute new descent direction $d_k = -\nabla f(u_k)$ via Riesz representation

$$d_k = -(\gamma_2 u_k + \beta p_k)$$

$$(\gamma_2 \bar{u} + \beta p, u - \bar{u})_Q \geq 0$$

  - Find appropriate step length α_k by trying to solve

$$\Rightarrow (\gamma_2 \bar{u}(x,t) + \beta p(x,t))(u(x,t) - \bar{u}(x,t)) \geq 0 \text{ for a.e. } (x,t) \in Q.$$

$$f(P_{U_{ad}}(u_k + \alpha_k d_k)) = \min_{\alpha > 0} f(P_{U_{ad}}(u_k + \alpha d_k)).$$

e.g. using a suitable backtracking algorithm.

- Note: as before for (time-dependent) box-constraints $\xi_0(x,t) \leq u(x,t) \leq \xi_1(x,t)$, OC translates to

$$\bar{u}(x,t) = P_{[\xi_0(x,t), \xi_1(x,t)]}\left(-\frac{\beta}{\gamma_2} p(x,t)\right).$$