Accuracy and Stability of a Predictor-Corrector Crank–Nicolson Method with Many Subdomains

Felix Kwok

Section de Mathématiques
Université de Genève

SciCADE, Toronto, July 11th, 2011

UNIVERSITÉ DE GENÈVE
FACULTÉ DES SCIENCES
Section de mathématiques

Model Problem

Suppose we want to solve

$$\begin{aligned} \frac{\partial u}{\partial t} &= \mathcal{L}u + f \qquad && \text{on } \Omega \subset \mathbb{R}^d, \\ u &= g && \text{on } \partial\Omega, \end{aligned}$$

where

$$\mathcal{L}u = \sum_{i,j} \frac{\partial}{\partial x_i}\left(a_{ij}(x)\frac{\partial u}{\partial x_j}\right) - c(x)u$$

is uniformly elliptic, i.e., $[a_{ij}]$ is uniformly s.p.d. and $c(x) \geq 0$.

Method of Lines

- Assume the spatial discretization

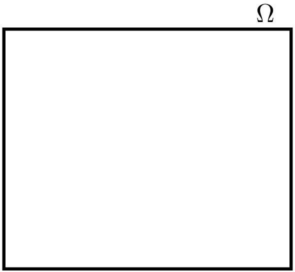
$$\mathcal{L}u \approx -\frac{1}{h^2}A\mathbf{u} + O(h^2),$$

where

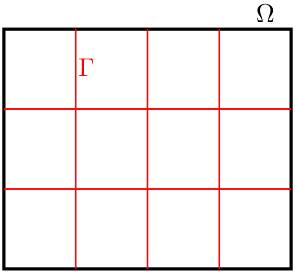
  - A is large and sparse
  - A is symmetric positive definite (due to Dirichlet boundary)
  - $\|A\|_2$ is independent of h (due to $\frac{1}{h^2}$ factor)

- If the time discretization is implicit, one must solve a large sparse linear system involving A at *every* time step

Domain Decomposition

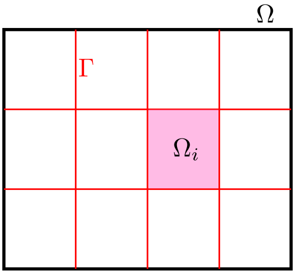


Domain Decomposition



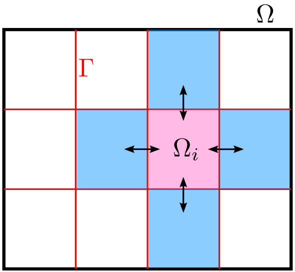
- Decompose Ω into several *subdomains* Ω_i

Domain Decomposition



- Decompose Ω into several *subdomains* Ω_i
- Define subdomain problems that can be solved in parallel

Domain Decomposition



- Decompose Ω into several *subdomains* Ω_i
- Define subdomain problems that can be solved in parallel
- Exchange interface data between subdomains
- Iterate to get consistent solution across subdomains

Types of DD algorithms

Example : Backward Euler

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

1. Domain decomposition in space : at each time step, solve

$$(I + \tfrac{\Delta t}{h^2} A)\mathbf{u}^{n+1} = F(\mathbf{u}^n, \mathbf{f}^n)$$

using domain decomposition in space only

Types of DD algorithms

Example : Backward Euler

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

2. Schwarz waveform relaxation : solve

$$(I + \tfrac{\Delta t}{h^2} A)\mathbf{u}_i^{n+1} = F_i(\mathbf{u}_i^n, \mathbf{u}_{i\pm 1}^n, \mathbf{f}^n)$$

on each Ω_i independently over *many time steps* and exchange interface data over the *entire time window*

- Must iterate to convergence

Types of DD algorithms

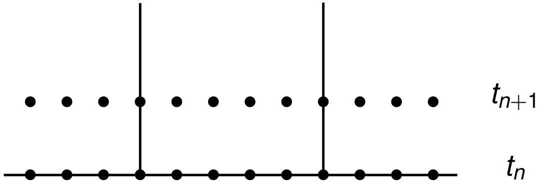
Example : Backward Euler

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

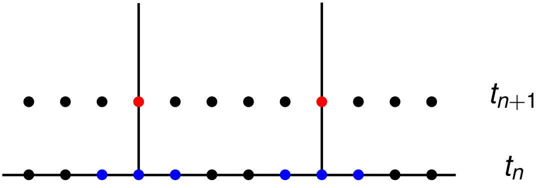
3. Predictor-corrector method : for each time step,
   - *Predict* interface values explicitly (Forward Euler)
   - Solve subdomain problems (Backward Euler) using predicted values
   - *Correct* interface values (Backward Euler) using interior values

- *Advantage :* No need to iterate to convergence !

Predictor-Corrector Euler



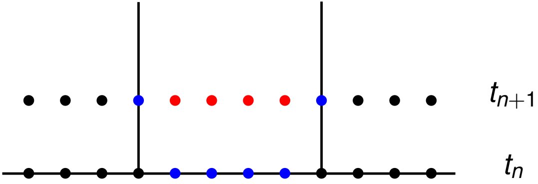
Predictor-Corrector Euler



1. *Predict* interface values $\mathbf{u}_\Gamma^*$ explicitly (Forward Euler)

$$\frac{\mathbf{u}_\Gamma^* - \mathbf{u}_\Gamma^n}{\Delta t} = -\frac{1}{h^2}\left(A_\Gamma \mathbf{u}_\Gamma^n + A_{\Gamma I}\mathbf{u}_I^n\right) + R_\Gamma \mathbf{f}$$

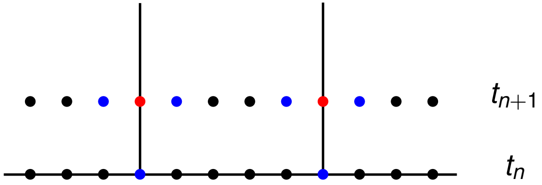
Predictor-Corrector Euler



2. Solve *independent* problems over Ω_i using $\mathbf{u}_\Gamma^*$

$$\frac{\mathbf{u}_I^{n+1/2} - \mathbf{u}_I^n}{\Delta t} = -\frac{1}{h^2}(A_I \mathbf{u}_I^{n+1/2} + A_{I\Gamma}\mathbf{u}_\Gamma^*) + R_I \mathbf{f}$$

Predictor-Corrector Euler



3. *Correct* $\mathbf{u}_\Gamma$ implicitly (Backward Euler) using interior values

$$\frac{\mathbf{u}_\Gamma^{n+1} - \mathbf{u}_\Gamma^n}{\Delta t} = -\frac{1}{h^2}(A_\Gamma \mathbf{u}_\Gamma^{n+1} + A_{\Gamma I}\mathbf{u}_I^{n+1/2}) + R_\Gamma \mathbf{f}$$

Predictor-Corrector Crank–Nicolson

- To derive Crank–Nicolson, make a time step to $t_{n+1/2}$ using backward Euler, then extrapolate :

$$\frac{\mathbf{u}^{n+1/2} - \mathbf{u}^{n}}{\Delta t/2} = -\frac{1}{h^2} A\mathbf{u}^{n+1/2} + \mathbf{f}$$

$$\mathbf{u}^{n+1} = 2\mathbf{u}^{n+1/2} - \mathbf{u}^{n}$$

Predictor-Corrector Crank–Nicolson

- To derive Crank–Nicolson, make a time step to $t_{n+1/2}$ using backward Euler, then extrapolate :

$$(I + \frac{k}{2}A)\mathbf{u}^{n+1/2} = \mathbf{u}^n + \frac{\Delta t}{2}\mathbf{f}$$

$$\mathbf{u}^{n+1} = 2\mathbf{u}^{n+1/2} - \mathbf{u}^n$$

(where $k = \Delta t/h^2$)

Predictor-Corrector Crank–Nicolson

- To derive Crank–Nicolson, make a time step to $t_{n+1/2}$ using backward Euler, then extrapolate :

$$(I + \frac{k}{2}A)\mathbf{u}^{n+1/2} = \mathbf{u}^n + \frac{\Delta t}{2}\mathbf{f}$$

$$(I + \frac{k}{2}A)\mathbf{u}^{n+1} = (I - \frac{k}{2}A)\mathbf{u}^n + \Delta t \cdot \mathbf{f}$$

(where $k = \Delta t/h^2$)

Predictor-Corrector Crank–Nicolson

- To derive Crank–Nicolson, make a time step to $t_{n+1/2}$ using backward Euler, then extrapolate :

$$(I + \frac{k}{2}A)\mathbf{u}^{n+1/2} = \mathbf{u}^n + \frac{\Delta t}{2}\mathbf{f}$$

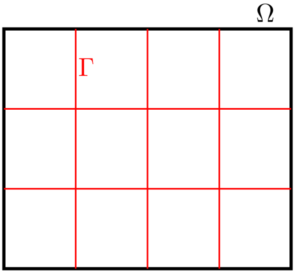
$$(I + \frac{k}{2}A)\mathbf{u}^{n+1} = (I - \frac{k}{2}A)\mathbf{u}^n + \Delta t \cdot \mathbf{f}$$

(where $k = \Delta t/h^2$)

- Do the same for the predictor-corrector version :

$$(I+\frac{k}{2}X_2A)(I+\frac{k}{2}X_1A)\mathbf{u}^{n+1} = (I-\frac{k}{2}X_2A)(I-\frac{k}{2}X_1A)\mathbf{u}^n+\Delta t\cdot\mathbf{f}$$

Predictor-Corrector Crank–Nicolson



$$(I + \frac{k}{2} X_2 A)(I + \frac{k}{2} X_1 A)\mathbf{u}^{n+1} = (I - \frac{k}{2} X_2 A)(I - \frac{k}{2} X_1 A)\mathbf{u}^{n} + \Delta t \cdot \mathbf{f}$$

- X_1 = projection onto Γ
- $X_2 = I - X_1$ = projection onto $\Omega \setminus \Gamma$

Predictor-Corrector Crank–Nicolson

- Introduced by Rempe and Chopp (SISC 2006)
- Used to simulate neural activity in branched structures
- Shown experimentally to have formal second order in time and in space
- What about simultaneous refinement ? Must analyze :
  - Stability
  - Accuracy

Stability

$$\underbrace{(I + \frac{k}{2}X_2A)(I + \frac{k}{2}X_1A)}_{B}\,\mathbf{u}^{n+1} = \underbrace{(I - \frac{k}{2}X_2A)(I - \frac{k}{2}X_1A)}_{C}\,\mathbf{u}^{n} + \Delta t \cdot \mathbf{f}$$

- Stability matrix :

$$B^{-1}C = (I + \frac{k}{2}X_1A)^{-1}(I + \frac{k}{2}X_2A)^{-1}(I - \frac{k}{2}X_2A)(I - \frac{k}{2}X_1A)$$

Stability

$$\underbrace{(I+\frac{k}{2}X_2A)(I+\frac{k}{2}X_1A)}_{B}\mathbf{u}^{n+1}=\underbrace{(I-\frac{k}{2}X_2A)(I-\frac{k}{2}X_1A)}_{C}\mathbf{u}^{n}+\Delta t\cdot\mathbf{f}$$

- Stability matrix :

$$B^{-1}C=(I+\frac{k}{2}X_1A)^{-1}(I+\frac{k}{2}X_2A)^{-1}(I-\frac{k}{2}X_2A)(I-\frac{k}{2}X_1A)$$

- Expect unconditional stability, but
  - Factors are non-symmetric
  - Factors do not commute

UNIVERSITÉ DE GENÈVE
FACULTÉ DES SCIENCES
Section de mathématiques

Stability

$$B^{-1}C = (I + \frac{k}{2}X_1A)^{-1}(I + \frac{k}{2}X_2A)^{-1}(I - \frac{k}{2}X_2A)(I - \frac{k}{2}X_1A)$$

Stability

$$A^{1/2}B^{-1}CA^{-1/2} = (I + \frac{k}{2}A^{1/2}X_1A^{1/2})^{-1}(I + \frac{k}{2}A^{1/2}X_2A^{1/2})^{-1} \cdot (I - \frac{k}{2}A^{1/2}X_2A^{1/2})(I - \frac{k}{2}A^{1/2}X_1A^{1/2})$$

- Symmetrize the factors

Stability

$$A^{1/2}B^{-1}CA^{-1/2} = (I+\frac{k}{2}M_1)^{-1}(I+\frac{k}{2}M_2)^{-1}(I-\frac{k}{2}M_2)(I-\frac{k}{2}M_1)$$

- Symmetrize the factors

Stability

$$\overbrace{A^{1/2}(I+\frac{k}{2}M_1)B^{-1}}^{S}\,\overbrace{C(I+\frac{k}{2}M_1)^{-1}A^{-1/2}}^{S^{-1}} =$$
$$(I+\frac{k}{2}M_2)^{-1}(I-\frac{k}{2}M_2)(I-\frac{k}{2}M_1)(I+\frac{k}{2}M_1)^{-1}$$

- Symmetrize the factors
- Move the first factor to the back

Stability

$$\|SB^{-1}CS^{-1}\|_2 = \|(I + \frac{k}{2}M_2)^{-1}(I - \frac{k}{2}M_2)\|_2 \cdot \|(I - \frac{k}{2}M_1)(I + \frac{k}{2}M_1)^{-1}\|_2$$

- Symmetrize the factors
- Move the first factor to the back
- Bound each piece by the 2-norm

Stability

$$\|SB^{-1}CS^{-1}\|_2 = \|(I + \frac{k}{2}M_2)^{-1}(I - \frac{k}{2}M_2)\|_2 \cdot \|(I - \frac{k}{2}M_1)(I + \frac{k}{2}M_1)^{-1}\|_2$$

- Each $M_i = A^{1/2}X_iA^{1/2}$ is symmetric, with eigenvalues ≥ 0
- $(I + \frac{k}{2}M_i)^{-1}(I - \frac{k}{2}M_i)$ is symmetric and has eigenvalues

$$\left|\frac{1 - k\lambda_i/2}{1 + k\lambda_i/2}\right| \leq 1$$

for all $k > 0 \implies$ unconditional stablity !

UNIVERSITÉ DE GENÈVE
FACULTÉ DES SCIENCES
Section de mathématiques

Local Truncation Error

- Would like to pick $\Delta t = h$ like in classical Crank–Nicolson

Local Truncation Error

- Would like to pick $\Delta t = h$ like in classical Crank–Nicolson
- The exact solution $u(\cdot, t_n)$ satisfies

$$Bu(\cdot, t_{n+1}) = Cu(\cdot, t_n) + \Delta t \cdot f(\cdot, t_{n+1/2}) + \rho_n$$

where

$$\rho_n = \Delta t \left[\frac{\Delta t^2}{4h^2} X_2 A X_1 \cdot \mathcal{L} u_t(\cdot, t_{n+1/2}) + O(\Delta t^2) + O(h^2) \right]$$

Local Truncation Error

- Would like to pick $\Delta t = h$ like in classical Crank–Nicolson
- The exact solution $u(\cdot, t_n)$ satisfies

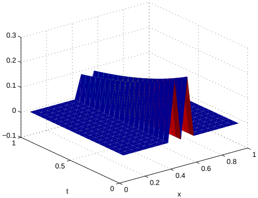
$$Bu(\cdot, t_{n+1}) = Cu(\cdot, t_n) + \Delta t \cdot f(\cdot, t_{n+1/2}) + \rho_n$$

where

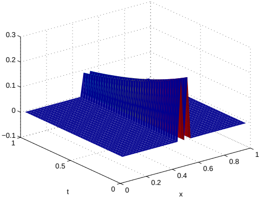
$$\rho_n = \Delta t \left[\frac{\Delta t^2}{4h^2} X_2 A X_1 \cdot \mathcal{L} u_t(\cdot, t_{n+1/2}) + O(\Delta t^2) + O(h^2) \right]$$

- If $\Delta t = h$, method is inconsistent !

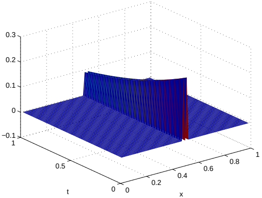
Local Truncation Error



$n = 20$



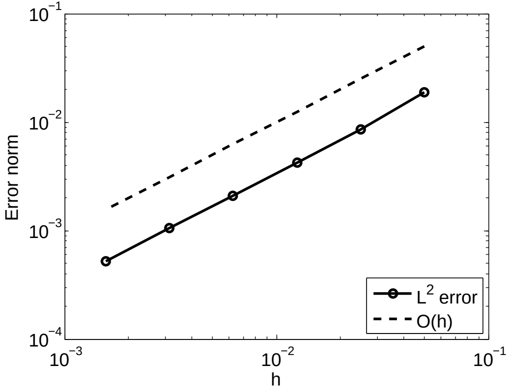
$n = 40$



$n = 80$

- Local truncation error does not decrease !

Global Error, $\Delta t = h$



- But, L^2 error shows first-order convergence !

Convergence Analysis

- Global error ε_n satisfies

$$\varepsilon_n = \sum_{j=1}^{n} (B^{-1}C)^{n-j} B^{-1} \rho_{j-1}$$

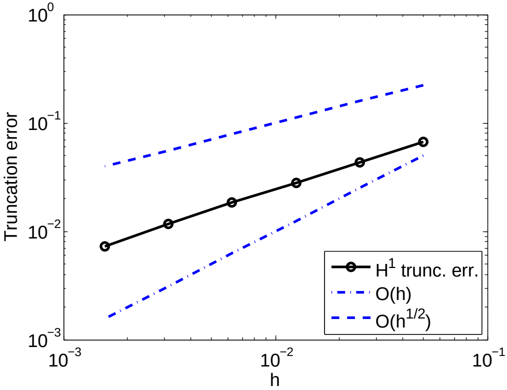
- Want to bound the discrete H^1 seminorm of ε_n :

$$C_1 h^{d-2} \|\varepsilon_n\|_A^2 \leq |\varepsilon_n|_{H^1(\Omega)}^2 \leq C_2 h^{d-2} \|\varepsilon_n\|_A^2$$

- Standard argument using stability shows

$$\|\varepsilon_n\|_A \leq \sum_{j=1}^{n} \|(I + \frac{k}{2} X_2 A)^{-1} \rho_{j-1}\|_A$$

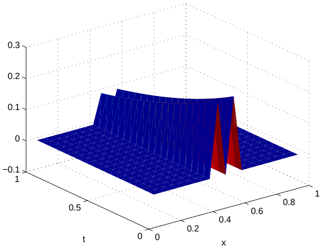
Convergence Analysis



- A-norm of $(I + \frac{k}{2}X_2 A)^{-1}\rho_0$ decays to zero, but not fast enough !

UNIVERSITÉ DE GENÈVE
FACULTÉ DES SCIENCES
Section de mathématiques

Another Approach



Observations :

- Local truncation error is a *smooth* function in time (but not in space !)
- The geometric series $\sum_{j=0}^{\infty}(B^{-1}C)^j$ converges

Another Approach

- Define $\delta_n = \rho_n - \rho_{n-1}$:

$$\rho_n = \tau\left[\frac{\tau^2}{4h^2} X_2 A X_1 \cdot \mathcal{L} u_t(\cdot, t_{n+1/2})\right]$$

$$\delta_n = \tau^{2}\left[\frac{\tau^2}{4h^2} X_2 A X_1 \cdot \mathcal{L} u_{tt}(\cdot, t_n)\right]$$

- Exchange the order of summation

$$\varepsilon_n = \sum_{j=1}^{n} (B^{-1}C)^{n-j} B^{-1} \underbrace{\sum_{l=0}^{j-1} \delta_l}_{\rho_{j-1}}$$

Another Approach

- Define $\delta_n = \rho_n - \rho_{n-1}$:

$$\rho_n = \tau \left[\frac{\tau^2}{4h^2} X_2 A X_1 \cdot \mathcal{L} u_t(\cdot, t_{n+1/2}) \right]$$

$$\delta_n = \tau^2 \left[\frac{\tau^2}{4h^2} X_2 A X_1 \cdot \mathcal{L} u_{tt}(\cdot, t_n) \right]$$

- Exchange the order of summation

$$\varepsilon_n = \sum_{l=0}^{n-1} \sum_{j=l+1}^{n} (B^{-1}C)^{n-j} B^{-1} \delta_l$$

Another Approach

- Define $\delta_n = \rho_n - \rho_{n-1}$:

$$\rho_n = \tau \left[\frac{\tau^2}{4h^2} X_2 A X_1 \cdot \mathcal{L} u_t(\cdot, t_{n+1/2}) \right]$$

$$\delta_n = \tau^{2} \left[\frac{\tau^2}{4h^2} X_2 A X_1 \cdot \mathcal{L} u_{tt}(\cdot, t_n) \right]$$

- Exchange the order of summation

$$\varepsilon_n = \sum_{l=0}^{n-1} (I - (B^{-1}C)^{n-l})(I - B^{-1}C)^{-1} B^{-1} \delta_l$$

Another Approach

- Define $\delta_n = \rho_n - \rho_{n-1}$:

$$\rho_n = \tau \left[\frac{\tau^2}{4h^2} X_2 A X_1 \cdot \mathcal{L} u_t(\cdot, t_{n+1/2}) \right]$$

$$\delta_n = \tau^2 \left[\frac{\tau^2}{4h^2} X_2 A X_1 \cdot \mathcal{L} u_{tt}(\cdot, t_n) \right]$$

- Exchange the order of summation

$$\varepsilon_n = \sum_{l=0}^{n-1} (I - (B^{-1}C)^{n-l})(\underbrace{B - C}_{=kA})^{-1} \delta_l$$

Another Approach

- Define $\delta_n = \rho_n - \rho_{n-1}$:

$$\rho_n = \tau \left[\frac{\tau^2}{4h^2} X_2 A X_1 \cdot \mathcal{L} u_t(\cdot, t_{n+1/2}) \right]$$

$$\delta_n = \tau^{2} \left[\frac{\tau^2}{4h^2} X_2 A X_1 \cdot \mathcal{L} u_{tt}(\cdot, t_n) \right]$$

- Exchange the order of summation

$$\varepsilon_n = \frac{1}{k} \sum_{l=0}^{n-1} \underbrace{(I - (B^{-1}C)^{n-l})}_{\|\cdot\| \leq 2} A^{-1} \delta_l$$

Convergence

- After some more manipulations, we get

$$\|A^{1/2}\varepsilon_n\|_2^2 \leq C\kappa(\mathcal{S}_\Gamma)h^{-(d-1)} \max_{0\leq j<n} \left(\tau^2 \|\mathcal{L}u_{tt}(\cdot, t_j)\|_\infty\right)^2$$

  where $\mathcal{S}_\Gamma$ is the Schur complement of A onto the interface Γ
- Under fairly general conditions,

$$\kappa(\mathcal{S}_\Gamma) = O(1/h)$$

- Thus, if $\Delta t = O(h^\alpha)$, then

$$|\varepsilon_n|_{H^1(\Omega)} \leq C\frac{\tau^2}{h} = O(h^{2\alpha-1}).$$

- Same estimate in $\|\cdot\|_{L^2}$ (using Poincaré inequality)

Convergence

$$|\varepsilon_n|_{H^1(\Omega)} \leq C\frac{\tau^2}{h} = O(h^{2\alpha-1})$$

- $\Delta t = h \ (\alpha = 1) \implies$ first order convergence
- $\Delta t = h^{3/2} \ (\alpha = 3/2) \implies$ second order convergence !
- Cannot go beyond second order due to $O(\Delta t^2 + h^2)$ terms away from the interface

1D Example

- Solve

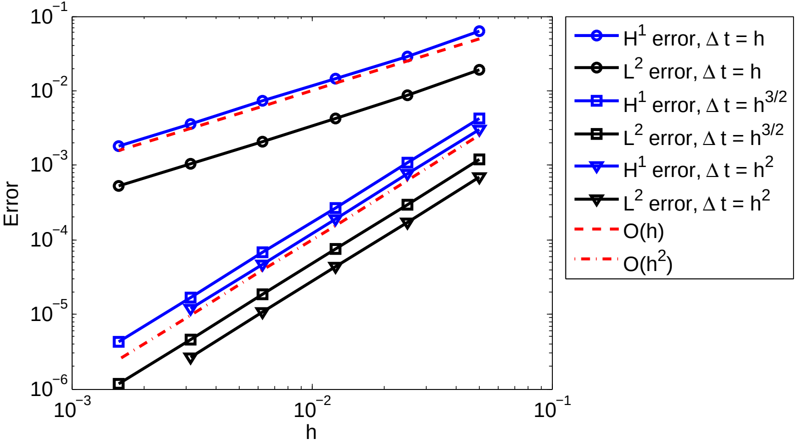
$$u_t = u_{xx} + f(x,t), \qquad u(x,0) = g(x)$$

  over $(x,t) \in (0,1) \times (0,1]$
- Choose $f(x,t)$ and $g(x)$ to have exact solution

$$u(x,t) = x\sin(\pi x)\sin(t)$$

- Two subdomains with interface at $x = 0.5$
- Choose $\Delta t = h^{\alpha}$ with $\alpha = 1, 1.5, 2$
- Measure error in L^2 norm and H^1 semi-norm

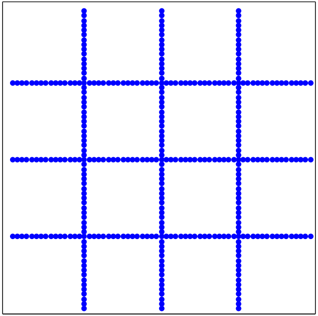
1D Example



$(n = 20, 40, 80, 160, 320, 640)$

UNIVERSITÉ DE GENÈVE
FACULTÉ DES SCIENCES
Section de mathématiques

2D Example

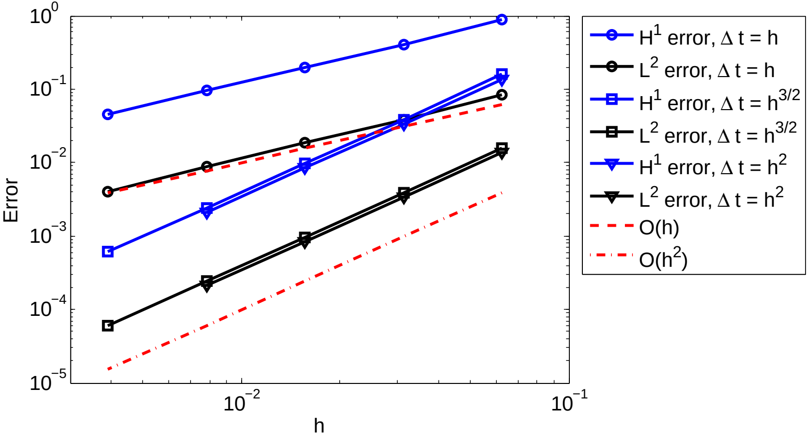


- Solve $u_t = \Delta u + f(x, t)$ on unit square, $t \in (0, 1]$
- Exact solution :

$$u(x, y, t) = sin(3\pi x)(1 - e^{2y})(1 - e^{y-1})\sqrt{1 + t}$$

UNIVERSITÉ DE GENÈVE
FACULTÉ DES SCIENCES
Section de mathématiques

2D Example



$(n = 16, 32, 64, 128, 256)$

UNIVERSITÉ DE GENÈVE
FACULTÉ DES SCIENCES
Section de mathématiques

Conclusion

- The Crank–Nicolson predictor-corrector method is
  - Unconditionally stable
  - Low order (or even inconsistent) near the interface
  - Converges with order $\min\{2\alpha - 1, 2\}$ for $\Delta t = O(h^{\alpha})$
- Ongoing work :
  - Influence of number of subdomains on convergence
  - Singular A (e.g., Neumann boundary)
  - Non-symmetric A (e.g., advection-diffusion equations)