A direct solver for time parallelization of wave equations

Laurence HALPERN
LAGA - Université Paris 13 and C.N.R.S.

6th Parallel in Time Workshop Monte Verita, Octobre 23, 2017

Joint work with Martin Gander (Genève), Johann Rannou and Juliette Ryan (ONERA)
PhD Thuy Thi Bich Tran

1. Introduction
2. The wave equation
   - An algorithm for the ODE
   - Error analysis
   - Diagonalization of B
   - Optimization of the algorithm
   - Application to an industrial case
3. Conclusion and Perspectives
4. Bibliography

Outline

1. Introduction

2. The wave equation
   - An algorithm for the ODE
   - Error analysis
   - Diagonalization of B
   - Optimization of the algorithm
   - Application to an industrial case

3. Conclusion and Perspectives

4. Bibliography

Parallelism and PDE: distribute the computation

How to solve in parallel $$\partial_t u - \Delta u = f$$

Time discretization

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

Elliptic problem: **Domain Decomposition in space** (Cai) , if

$$\Delta t \ll H^2$$

(H size of the subdomains) convergence <u>independent of the mesh</u>

Space-discretization

$$\partial_t U + A\,U = F$$

Parallelism across the method: Multistep methods.

Parallelism across the system: Waveform relaxation **(Domain Decomposition in space)**

Parallelism across the steps **(Domain Decomposition in time)**

Time discretization for the heat equation. 0-D

$$d_t u + au = 0, \quad u(0) = u_0, \quad t \in (0, T) \quad \Longleftrightarrow \quad u(t) = e^{-at} u_0.$$

Time discretization for the heat equation. 0-D

$$d_t u + au = 0, \quad u(0) = u_0, \quad t \in (0, T) \quad \Longleftrightarrow \quad u(t) = e^{-at} u_0.$$

$$\frac{u^n - u^{n-1}}{k_n} + au^n = 0, \quad u^0 = u_0, \quad \sum k_n = T$$

Time discretization for the heat equation. 0-D

$$d_t u + au = 0, \quad u(0) = u_0, \quad t \in (0, T) \quad \Longleftrightarrow \quad u(t) = e^{-at} u_0.$$

$$\frac{u^n - u^{n-1}}{k_n} + au^n = 0, \quad u^0 = u_0, \quad \sum k_n = T$$

$$\begin{pmatrix} \frac{1}{k_1} & & & \\ -\frac{1}{k_2} & \frac{1}{k_2} & & 0 \\ 0 & \ddots & \ddots & \\ & & -\frac{1}{k_N} & \frac{1}{k_N} \end{pmatrix} \begin{pmatrix} u^1 \\ u^2 \\ \vdots \\ u^N \end{pmatrix} + a \begin{pmatrix} u^1 \\ u^2 \\ \vdots \\ u^N \end{pmatrix} = \begin{pmatrix} \frac{1}{k_1} u^0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

$$(B + a\, I)\mathbf{u} = F, \quad \mathbf{u} = (u^1, \ldots u^N)$$

Time discretization for the heat equation. 0-D

$$d_t u + au = 0, \quad u(0) = u_0, \quad t \in (0, T) \quad \iff \quad u(t) = e^{-at} u_0.$$

$$\frac{u^n - u^{n-1}}{k_n} + au^n = 0, \quad u^0 = u_0, \quad \sum k_n = T$$

$$\begin{pmatrix} \frac{1}{k_1} & & & \\ -\frac{1}{k_2} & \frac{1}{k_2} & & 0 \\ 0 & \ddots & \ddots & \\ & & -\frac{1}{k_N} & \frac{1}{k_N} \end{pmatrix} \begin{pmatrix} u^1 \\ u^2 \\ \vdots \\ u^N \end{pmatrix} + a \begin{pmatrix} u^1 \\ u^2 \\ \vdots \\ u^N \end{pmatrix} = \begin{pmatrix} \frac{1}{k_1} u^0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

$$(B + a\, I)\mathbf{u} = F, \quad \mathbf{u} = (u^1, \ldots u^N)$$

$$B = SDS^{-1}, \quad S(D + aI)S^{-1}\mathbf{u} = F$$

Time discretization for the heat equation. 0-D

$$d_t u + au = 0, \quad u(0) = u_0, \quad t \in (0, T) \quad \Longleftrightarrow \quad u(t) = e^{-at} u_0.$$

$$\frac{u^n - u^{n-1}}{k_n} + au^n = 0, \quad u^0 = u_0, \quad \sum k_n = T$$

$$\begin{pmatrix} \frac{1}{k_1} & & & \\ -\frac{1}{k_2} & \frac{1}{k_2} & & 0 \\ 0 & \ddots & \ddots & \\ & & -\frac{1}{k_N} & \frac{1}{k_N} \end{pmatrix} \begin{pmatrix} u^1 \\ u^2 \\ \vdots \\ u^N \end{pmatrix} + a \begin{pmatrix} u^1 \\ u^2 \\ \vdots \\ u^N \end{pmatrix} = \begin{pmatrix} \frac{1}{k_1} u^0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

$$(B + a\, I)\mathbf{u} = F, \quad \mathbf{u} = (u^1, \dots u^N)$$

$$B = SDS^{-1}, \quad S(D + aI)S^{-1}\mathbf{u} = F$$

$$\boxed{(1)\ SG = F, \quad (2)\ (D + aI)\mathbf{v} = G, \quad (3)\ \widehat{\mathbf{u}} = S\,\mathbf{v}.}$$

Time discretization for the heat equation. d-D

$$\partial_t u - \Delta u = 0, \quad u(0) = u_0.$$

$$\frac{u^n - u^{n-1}}{k_n} - \Delta u^n = 0, \quad u^0 = u_0.$$

$$\underbrace{\begin{pmatrix} \frac{1}{k_1} I_x - \Delta & & & \\ -\frac{1}{k_2} I_x & \frac{1}{k_2} I_x - \Delta & & 0 \\ 0 & \ddots & \ddots & \\ & & -\frac{1}{k_N} I_x & \frac{1}{k_N} I_x - \Delta \end{pmatrix}}_{M} \cdot \underbrace{\begin{pmatrix} u^1 \\ u^2 \\ \vdots \\ u^N \end{pmatrix}}_{\mathbf{u}} = \underbrace{\begin{pmatrix} \frac{1}{k_1} u^0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}}_{F}$$

Resolution by multigrid, or iterative, or direct methods.

$$u = e^{-t\Delta} u_0.$$

Time-space discretization for the heat equation.

Discretization in space, M degrees of freedom.

$$M_h \mathbf{u}_h = F_h, \qquad \mathbf{u}_h \in (\mathbb{R}^M)^N.$$

$$M_h = \begin{pmatrix} \frac{1}{k_1} I_x - \Delta_h & & & \\ -\frac{1}{k_2} I_x & \frac{1}{k_2} I_x - \Delta_h & & 0 \\ 0 & \ddots & \ddots & \\ & & -\frac{1}{k_N} I_x & \frac{1}{k_N} I_x - \Delta_h \end{pmatrix}.$$

Time-space discretization for the heat equation.

Discretization in space, M degrees of freedom.

$$M_h \mathbf{u}_h = F_h, \qquad \mathbf{u}_h \in (\mathbb{R}^M)^N.$$

$$M_h = \begin{pmatrix} \frac{1}{k_1} I_x - \Delta_h & & & 0 \\ -\frac{1}{k_2} I_x & \frac{1}{k_2} I_x - \Delta_h & & \\ 0 & \ddots & \ddots & \\ & & -\frac{1}{k_N} I_x & \frac{1}{k_N} I_x - \Delta_h \end{pmatrix}.$$

$$M_h = B \otimes I_x + I_t \otimes (-\Delta_h), \quad B = \begin{pmatrix} \frac{1}{k_1} & & & 0 \\ -\frac{1}{k_2} & \frac{1}{k_2} & & \\ 0 & \ddots & \ddots & \\ & & -\frac{1}{k_N} & \frac{1}{k_N} \end{pmatrix}.$$

Direct method (Maday-Ronquist, CRAS 2007)

$$\underbrace{\left(B \otimes I_x + I_t \otimes (-\Delta_h)\right)}_{M_h} \mathbf{u}_h = F_h, \quad B = \begin{pmatrix} \frac{1}{k_1} & & & 0 \\ -\frac{1}{k_2} & \frac{1}{k_2} & & \\ 0 & \ddots & \ddots & \\ & & -\frac{1}{k_N} & \frac{1}{k_N} \end{pmatrix}.$$

Direct method (Maday-Ronquist, CRAS 2007)

$$\underbrace{\left(B \otimes I_x + I_t \otimes (-\Delta_h)\right)}_{M_h} \mathbf{u}_h = F_h, \quad B = \begin{pmatrix} \frac{1}{k_1} & & & 0 \\ -\frac{1}{k_2} & \frac{1}{k_2} & & \\ 0 & \ddots & \ddots & \\ & & -\frac{1}{k_N} & \frac{1}{k_N} \end{pmatrix}.$$

$$B = SDS^{-1}$$

Direct method (Maday-Ronquist, CRAS 2007)

$$\underbrace{(B \otimes I_x + I_t \otimes (-\Delta_h))}_{M_h} \mathbf{u}_h = F_h, \quad B = \begin{pmatrix} \frac{1}{k_1} & & & 0 \\ -\frac{1}{k_2} & \frac{1}{k_2} & & \\ 0 & \ddots & \ddots & \\ & & -\frac{1}{k_N} & \frac{1}{k_N} \end{pmatrix}.$$

$$B = SDS^{-1}$$

$$(S \otimes I_x)(D \otimes I_x + I_t \otimes (-\Delta_h))(S^{-1} \otimes I_x)\, \mathbf{u}_h = F_h$$

Direct method (Maday-Ronquist, CRAS 2007)

$$\underbrace{(B \otimes I_x + I_t \otimes (-\Delta_h))}_{M_h} \mathbf{u}_h = F_h, \quad B = \begin{pmatrix} \frac{1}{k_1} & & & 0 \\ -\frac{1}{k_2} & \frac{1}{k_2} & & \\ 0 & \ddots & \ddots & \\ & & -\frac{1}{k_N} & \frac{1}{k_N} \end{pmatrix}.$$

$$B = SDS^{-1}$$

$$(S \otimes I_x)(D \otimes I_x + I_t \otimes (-\Delta_h))(S^{-1} \otimes I_x)\,\mathbf{u}_h = F_h$$

$$\boxed{\begin{aligned} &(1) \quad (S \otimes I_x)\, G = F_h, \\ &(2) \quad (\tfrac{1}{k_n} - \Delta_h)\mathbf{v}^n = G^n, \quad 1 \le n \le N, \\ &(3) \quad \widehat{\mathbf{u}}_h = (S \otimes I_x)\,\mathbf{v} \end{aligned}}$$

Direct method (Maday-Ronquist, CRAS 2007)

$$\underbrace{\left(B \otimes I_x + I_t \otimes (-\Delta_h)\right)}_{M_h} \mathbf{u}_h = F_h, \quad B = \begin{pmatrix} \frac{1}{k_1} & & & 0 \\ -\frac{1}{k_2} & \frac{1}{k_2} & & \\ 0 & \ddots & \ddots & \\ & & -\frac{1}{k_N} & \frac{1}{k_N} \end{pmatrix}.$$

$$B = SDS^{-1}$$

$$(S \otimes I_x)(D \otimes I_x + I_t \otimes (-\Delta_h))(S^{-1} \otimes I_x)\,\mathbf{u}_h = F_h$$

$$\boxed{\begin{array}{ll} (1) & (S \otimes I_x)\, G = F_h, \\ (2) & (\frac{1}{k_n} - \Delta_h)\mathbf{v}^n = G^n, \quad 1 \le n \le N, \\ (3) & \widehat{\mathbf{u}}_h = (S \otimes I_x)\,\mathbf{v} \end{array}}$$

N equations in space can thus be solved independently on the processors. (2) is better conditioned than Δ_h, easily parallelized with OSM.

Direct method (Maday-Ronquist, CRAS 2007)

The method we have just proposed is first order in time, and since it requires that all the time steps are different, the accuracy will be related to the largest time step.

Direct method (Maday-Ronquist, CRAS 2007)

The method we have just proposed is first order in time, and since it requires that all the time steps are different, the accuracy will be related to the largest time step.

In order to make the method more efficient, we propose to use a higher order scheme in time with time steps $k_n = \rho^{n-1} k_1$, with ρ larger but close to 1, e.g. $\rho = 1.2$.

Direct method (Maday-Ronquist, CRAS 2007)

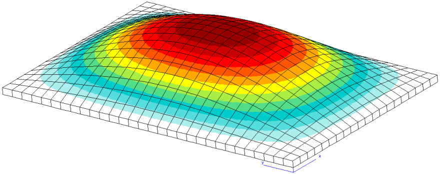
The method we have just proposed is first order in time, and since it requires that all the time steps are different, the accuracy will be related to the largest time step.

In order to make the method more efficient, we propose to use a higher order scheme in time with time steps $k_n = \rho^{n-1} k_1$, with ρ larger but close to 1, e.g. $\rho = 1.2$.

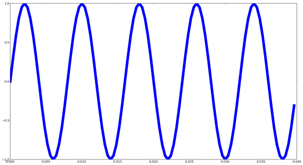
Note that, as can be expected, choosing ρ much closer to 1 may lead to instabilities due to numerical errors.

Error analysis (1)

We look for an exact solution of the form $U_z = \sin\left(\frac{\pi}{l}x\right)\sin\left(\frac{\pi}{L}y\right)\sin\left(\omega\pi t\right)$



(a) space solution on a $30 \times 20 \times 1$ mesh



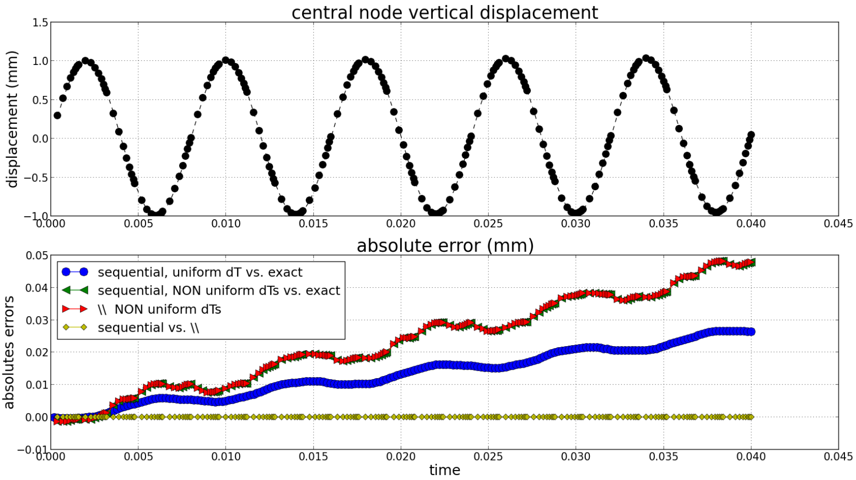
(b) time solution

we use a known solution to compare

1) the $\rho = 1$ sequential newmark scheme error
2) the $\rho < 1$ sequential newmark scheme error
3) the $\rho < 1$ parallel newmark scheme error
4) the introduced parallelism error (*i.e* (3) - (2))

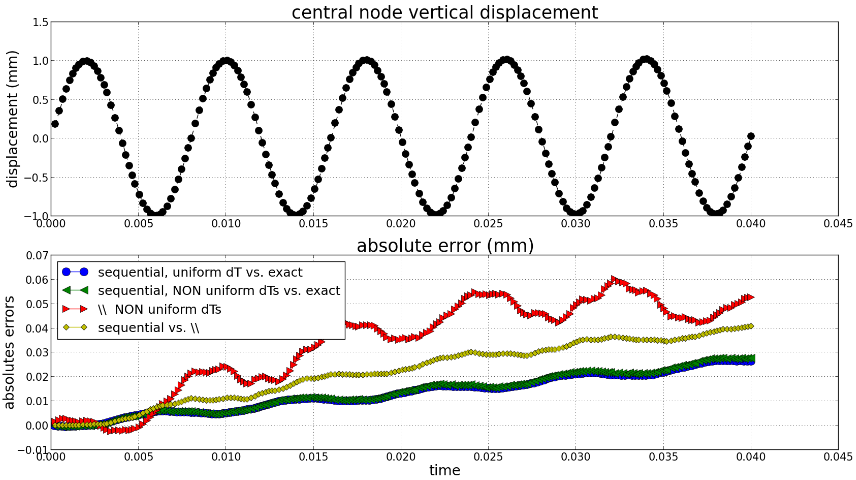
Error analysis (2)

$N_t = 8, \rho = 0.80 \quad \rightarrow \quad$ too much discretization error



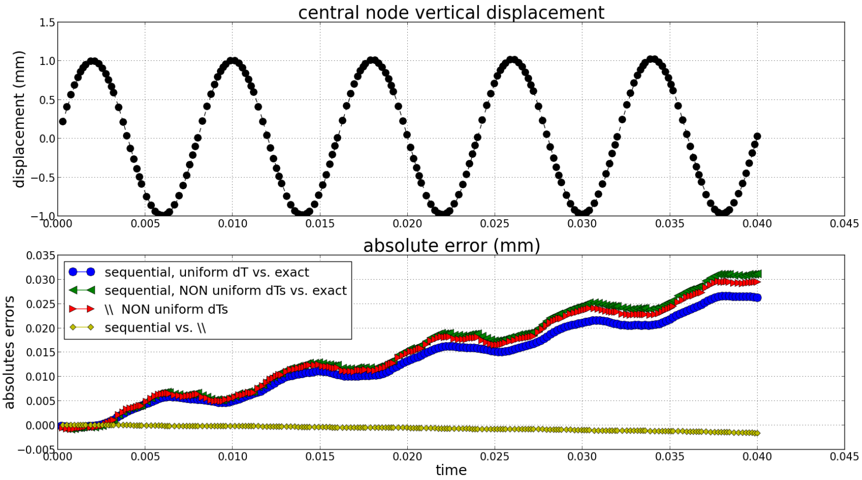
Error analysis (2)

$N_t = 8, \rho = 0.95 \quad \rightarrow \quad$ too much roundoff error



Error analysis (2)

$N_t = 8, \rho = 0.90 \quad \rightarrow \quad$ OK



Specifications

Choice of the timesteps $k_n = \rho^n k_1$, $\sum_{n=1}^{N} k_n = T$

$$(B \otimes I_x + I_t \otimes (-\Delta_h))\,\mathbf{u}_h = F_h, \quad B = SDS^{-1}, \quad D = diag(k_1, \dots, k_n).$$

$$\begin{array}{ll} (1) & (S \otimes I_x)\, G = F_h, , \\ (2) & (\frac{1}{k_n} - \Delta_h)\mathbf{v}^n = G^n, \quad 1 \leq n \leq N, \\ (3) & \widehat{\mathbf{u}}_h = (S \otimes I_x)\, V \end{array}$$

1. The timesteps have to be all different for B to be diagonalizable.

Specifications

Choice of the timesteps $k_n = \rho^n k_1$, $\sum_{n=1}^{N} k_n = T$

$$(B \otimes I_x + I_t \otimes (-\Delta_h))\,\mathbf{u}_h = F_h, \quad B = SDS^{-1}, \quad D = diag(k_1, \ldots, k_n).$$

$$\begin{array}{ll} (1) & (\hat{S} \otimes I_x)\, G = \mathbf{F}_h, \\ (2) & (\frac{1}{k_n} - \Delta_h)\mathbf{v}^n = G^n, \quad 1 \le n \le N, \\ (3) & \widehat{\mathbf{u}}_h = (\hat{S} \otimes I_x)\mathbf{v} \end{array}$$

1. The timesteps have to be all different for B to be diagonalizable.
2. The matrix S must be easy and cheap to invert (closed form is a must).

Specifications

Choice of the timesteps $k_n = \rho^n k_1$, $\sum_{n=1}^{N} k_n = T$

$$(B \otimes I_x + I_t \otimes (-\Delta_h))\, \mathbf{u}_h = F_h, \quad B = SDS^{-1}, \quad D = diag(k_1, \dots, k_n).$$

$$\begin{array}{ll} (1) & (\hat{S} \otimes I_x)\, G = \mathbf{F}_h, \\ (2) & (\frac{1}{k_n} - \Delta_h)\mathbf{v}^n = G^n, \quad 1 \leq n \leq N, \\ (3) & \widehat{\mathbf{u}}_h = (\hat{S} \otimes I_x)\mathbf{v} \end{array}$$

1. The timesteps have to be all different for B to be diagonalizable.
2. The matrix S must be easy and cheap to invert (closed form is a must).
3. The precision of the scheme can be affected.

Specifications

Choice of the timesteps $k_n = \rho^n k_1$, $\sum_{n=1}^{N} k_n = T$

$$(B \otimes I_x + I_t \otimes (-\Delta_h))\, \mathbf{u}_h = F_h, \quad B = SDS^{-1}, \quad D = diag(k_1, \dots, k_n).$$

$$\begin{array}{ll} (1) & (\hat{S} \otimes I_x)\, G = \mathbf{F}_h, \\ (2) & (\frac{1}{k_n} - \Delta_h)\mathbf{v}^n = G^n, \quad 1 \le n \le N, \\ (3) & \widehat{\mathbf{u}}_h = (\hat{S} \otimes I_x)\mathbf{v} \end{array}$$

1. The timesteps have to be all different for B to be diagonalizable.
2. The matrix S must be easy and cheap to invert (closed form is a must).
3. The precision of the scheme can be affected.
4. Therefore it is better to keep the time steps close to equidistant.

Specifications

Choice of the timesteps $k_n = \rho^n k_1$, $\sum_{n=1}^{N} k_n = T$

$$(B \otimes I_x + I_t \otimes (-\Delta_h))\,\mathbf{u}_h = F_h, \quad B = SDS^{-1}, \quad D = diag(k_1, \dots, k_n).$$

$$\begin{array}{ll} (1) & (\hat{S} \otimes I_x)\, G = \mathbf{F}_h, \\ (2) & (\frac{1}{k_n} - \Delta_h)\mathbf{v}^n = G^n, \quad 1 \le n \le N, \\ (3) & \widehat{\mathbf{u}}_h = (\hat{S} \otimes I_x)\mathbf{v} \end{array}$$

1. The timesteps have to be all different for B to be diagonalizable.
2. The matrix S must be easy and cheap to invert (closed form is a must).
3. The precision of the scheme can be affected.
4. Therefore it is better to keep the time steps close to equidistant.
5. Then the condition number of matrix S increases, deteriorating the results of steps (1) and (3).

Specifications

Choice of the timesteps $k_n = \rho^n k_1$, $\sum_{n=1}^{N} k_n = T$

$$(B \otimes I_x + I_t \otimes (-\Delta_h))\,\mathbf{u}_h = F_h, \quad B = SDS^{-1}, \quad D = diag(k_1, \ldots, k_n).$$

$$\begin{array}{ll} (1) & (\hat{S} \otimes I_x)\, G = \mathbf{F}_h, \\ (2) & (\frac{1}{k_n} - \Delta_h)\mathbf{v}^n = G^n, \quad 1 \leq n \leq N, \\ (3) & \widehat{\mathbf{u}}_h = (\hat{S} \otimes I_x)\mathbf{v} \end{array}$$

1. The timesteps have to be all different for B to be diagonalizable.
2. The matrix S must be easy and cheap to invert (closed form is a must).
3. The precision of the scheme can be affected.
4. Therefore it is better to keep the time steps close to equidistant.
5. Then the condition number of matrix S increases, deteriorating the results of steps (1) and (3).

QUANTIFY ?

Specifications

Choice of the timesteps $k_n = \rho^n k_1$, $\sum_{n=1}^{N} k_n = T$

$$(B \otimes I_x + I_t \otimes (-\Delta_h))\, \mathbf{u}_h = F_h, \quad B = SDS^{-1}, \quad D = diag(k_1, \ldots, k_n).$$

$$\begin{array}{ll} (1) & (\hat{S} \otimes I_x)\, G = \mathbf{F}_h, \\ (2) & (\frac{1}{k_n} - \Delta_h)\mathbf{v}^n = G^n, \quad 1 \le n \le N, \\ (3) & \widehat{\mathbf{u}}_h = (\hat{S} \otimes I_x)\mathbf{v} \end{array}$$

1. The timesteps have to be all different for B to be diagonalizable.
2. The matrix S must be easy and cheap to invert (closed form is a must).
3. The precision of the scheme can be affected.
4. Therefore it is better to keep the time steps close to equidistant.
5. Then the condition number of matrix S increases, deteriorating the results of steps (1) and (3).

QUANTIFY ? STRATEGIZE ?

Definitions

$$\boxed{u(t,\cdot)} = \mathcal{S}(t)u_0$$

$$(B \otimes I_x + I_t \otimes (-\Delta_h))\boxed{\mathbf{u}} = F_h$$

$$\begin{array}{ccc} \boxed{\mathbf{u}} & \longleftrightarrow & \mathcal{T} = (k_1, \ldots k_N) \\ \boxed{\overline{\mathbf{u}}} & \longleftrightarrow & \overline{\mathcal{T}} = (\overline{k}, \ldots, \overline{k}), \end{array}$$

$$(1) \quad (S \otimes I_x)G = \mathbf{F}_h,$$

$$(2) \quad \left(\frac{1}{k_n} - \Delta_h\right)\mathbf{u}^n = G^n, \quad 1 \leq n \leq N,$$

$$(3) \quad \boxed{\widehat{\mathbf{u}}} = (S \otimes I_x)\mathbf{v}.$$

Total error

$$\frac{|\mathcal{S}(T)u_0 - \hat{u}_N|}{|u_0|} \leq \underbrace{\frac{|\mathcal{S}(T)u_0 - \overline{u}_N|}{|u_0|}}_{\text{TRUNCATION ERROR WITH EQUAL TIME STEPS}} + \underbrace{\frac{|\overline{u}_N - u_N|}{|u_0|}}_{\text{ERROR DUE TO HETEROGENEOUS TIME STEPS}} + \underbrace{\frac{|u_N - \widehat{u}_N|}{|u_0|}}_{\text{ERROR DUE TO DIAGONALIZATION}}$$

Outline

Program for the wave equation

- The PDE $\ddot{u} - \Delta u = 0$.

Program for the wave equation

- The PDE $\ddot{u} - \Delta u = 0$.
- Work on the O.D.E. $\ddot{u} + a^2 u = 0$ (Fourier in space, $a = \|\xi\|$) with Crank-Nicolson scheme.

Program for the wave equation

- The PDE $\ddot{u} - \Delta u = 0$.
- Work on the O.D.E. $\ddot{u} + a^2 u = 0$ (Fourier in space, $a = \|\xi\|$) with Crank-Nicolson scheme.
    1. Evaluate the loss of precision produced by a set of $k_n = \rho^{n-1} k_1$ for $\boxed{\rho = 1 + \varepsilon}$.

Program for the wave equation

- The PDE $\ddot{u} - \Delta u = 0$.
- Work on the O.D.E. $\ddot{u} + a^2 u = 0$ (Fourier in space, $a = \|\xi\|$) with Crank-Nicolson scheme.
  1. Evaluate the loss of precision produced by a set of $k_n = \rho^{n-1} k_1$ for $\boxed{\rho = 1 + \varepsilon}$.
  2. write $(B + a^2 I)U = F$, and $B = SDS^{-1}$

Program for the wave equation

- The PDE $\ddot{u} - \Delta u = 0$.
- Work on the O.D.E. $\ddot{u} + a^2 u = 0$ (Fourier in space, $a = \|\xi\|$) with Crank-Nicolson scheme.
  1. Evaluate the loss of precision produced by a set of $k_n = \rho^{n-1} k_1$ for $\boxed{\rho = 1 + \varepsilon}$.
  2. write $(B + a^2 I)U = F$, and $B = SDS^{-1}$
  3. Find explicit forms for S and S^{-1}.

Program for the wave equation

- The PDE $\ddot{u} - \Delta u = 0$.
- Work on the O.D.E. $\ddot{u} + a^2 u = 0$ (Fourier in space, $a = \|\xi\|$) with Crank-Nicolson scheme.
  1. Evaluate the loss of precision produced by a set of $k_n = \rho^{n-1} k_1$ for $\boxed{\rho = 1 + \varepsilon}$.
  2. write $(B + a^2 I)U = F$, and $B = SDS^{-1}$
  3. Find explicit forms for S and S^{-1}.
  4. For given a and T, estimate the round-off error for the resolution of the diagonalized system.

Program for the wave equation

- The PDE $\ddot{u} - \Delta u = 0$.
- Work on the O.D.E. $\ddot{u} + a^2 u = 0$ (Fourier in space, $a = \|\xi\|$) with Crank-Nicolson scheme.
  1. Evaluate the loss of precision produced by a set of $k_n = \rho^{n-1} k_1$ for $\boxed{\rho = 1 + \varepsilon}$.
  2. write $(B + a^2 I)U = F$, and $B = SDS^{-1}$
  3. Find explicit forms for S and S^{-1}.
  4. For given a and T, estimate the round-off error for the resolution of the diagonalized system.
  5. For given a and T, equilibrate 1 and 4.

Program for the wave equation

- The PDE $\ddot{u} - \Delta u = 0$.
- Work on the O.D.E. $\ddot{u} + a^2 u = 0$ (Fourier in space, $a = \|\xi\|$) with Crank-Nicolson scheme.
  1. Evaluate the loss of precision produced by a set of $k_n = \rho^{n-1} k_1$ for $\boxed{\rho = 1 + \varepsilon}$.
  2. write $(B + a^2 I)U = F$, and $B = SDS^{-1}$
  3. Find explicit forms for S and S^{-1}.
  4. For given a and T, estimate the round-off error for the resolution of the diagonalized system.
  5. For given a and T, equilibrate 1 and 4.
- Apply to the P.D.E.

$$\boxed{\text{Perturbation analysis in } \varepsilon}$$

1 Introduction

2 The wave equation
- An algorithm for the ODE
- Error analysis
- Diagonalization of B
- Optimization of the algorithm
- Application to an industrial case

3 Conclusion and Perspectives

4 Bibliography

The Crank-Nicolson method

$$\begin{cases} d_t u = \dot{u} \\ d_t \dot{u} = \ddot{u} \\ \ddot{u} + a^2 u = 0 \end{cases} \qquad \begin{cases} \frac{1}{k_n}(u^n - u^{n-1}) = \frac{1}{2}(\dot{u}^n + \dot{u}^{n-1}), \\ \frac{1}{k_n}(\dot{u}^n - \dot{u}^{n-1}) = \frac{1}{2}(\ddot{u}^n + \ddot{u}^{n-1}), \\ \ddot{u}^n + a^2 u^n = 0. \end{cases}$$

$$U = \begin{pmatrix} u \\ a\dot{u} \end{pmatrix}, \;\; U_n = \begin{pmatrix} u_n \\ a\dot{u}_n \end{pmatrix}$$

The Crank-Nicolson method

$$\begin{cases} d_t u = \dot{u} \\ d_t \dot{u} = \ddot{u} \\ \ddot{u} + a^2 u = 0 \end{cases} \qquad \begin{cases} \frac{1}{k_n}(u^n - u^{n-1}) = \frac{1}{2}(\dot{u}^n + \dot{u}^{n-1}), \\ \frac{1}{k_n}(\dot{u}^n - \dot{u}^{n-1}) = \frac{1}{2}(\ddot{u}^n + \ddot{u}^{n-1}), \\ \ddot{u}^n + a^2 u^n = 0. \end{cases}$$

$$U = \begin{pmatrix} u \\ a\dot{u} \end{pmatrix}, \; U_n = \begin{pmatrix} u_n \\ a\dot{u}_n \end{pmatrix}$$

$$\boxed{k_n(\rho) := \rho^{n-1} k_1, \; \mathcal{T}_\rho := (k_1 \cdots, k_N) = k_1(1, \cdots, \rho^{N-1}), \; \sum_{n=1}^{N} k_n = T}$$

THEOREM Given a, T and N, for ε small,

$$\|U_N(\mathcal{T}_{1+\varepsilon}) - U_N(\mathcal{T}_1)\| = \boxed{\phi(\tfrac{aT}{2N}, N)\; \varepsilon^2} \|U_0\| + \mathcal{O}(\varepsilon^3),$$

$$\text{where } \phi(y, N) := \frac{N(N^2-1)}{6} \frac{y^3}{(1+y^2)^2}.$$

Matrix formulation (J. Rannou, T. Tran's Thesis)

$$\begin{cases} d_t u = \dot{u} \\ d_t \dot{u} = \ddot{u} \\ \ddot{u} - a^2 u = 0 \end{cases} \qquad \begin{cases} \frac{1}{k_n}(u^n - u^{n-1}) = \frac{1}{2}(\dot{u}^n + \dot{u}^{n-1}) \\ \frac{1}{k_n}(\dot{u}^n - \dot{u}^{n-1}) = \frac{1}{2}(\ddot{u}^n + \ddot{u}^{n-1}) \\ \ddot{u}^n - a^2 u^n = 0 \end{cases}$$

Matrix formulation (J. Rannou, T. Tran's Thesis)

$$\begin{cases} d_t u = \dot{u} \\ d_t \dot{u} = \ddot{u} \\ \ddot{u} - a^2 u = 0 \end{cases} \qquad \begin{cases} \frac{1}{k_n}(u^n - u^{n-1}) = \frac{1}{2}(\dot{u}^n + \dot{u}^{n-1}) \\ \frac{1}{k_n}(\dot{u}^n - \dot{u}^{n-1}) = \frac{1}{2}(\ddot{u}^n + \ddot{u}^{n-1}) \\ \ddot{u}^n - a^2 u^n = 0 \end{cases}$$

$$\boxed{\mathbf{u} = (u^1, \cdots, u^N) \qquad (B + aI)\mathbf{u} = f}$$

Matrix formulation (J. Rannou, T. Tran's Thesis)

$$\begin{cases} d_t u = \dot{u} \\ d_t \dot{u} = \ddot{u} \\ \ddot{u} - a^2 u = 0 \end{cases} \qquad \begin{cases} \frac{1}{k_n}(u^n - u^{n-1}) = \frac{1}{2}(\dot{u}^n + \dot{u}^{n-1}) \\ \frac{1}{k_n}(\dot{u}^n - \dot{u}^{n-1}) = \frac{1}{2}(\ddot{u}^n + \ddot{u}^{n-1}) \\ \ddot{u}^n - a^2 u^n = 0 \end{cases}$$

$$\boxed{\mathbf{u} = (u^1, \cdots, u^N) \qquad (B + aI)\mathbf{u} = f}$$

$$B = (C^{-1} B_1)^2$$

$$C = \frac{1}{2}\begin{pmatrix} 1 & & & \\ 1 & 1 & & \\ & \ddots & \ddots & \\ & & 1 & 1 \end{pmatrix} \quad B_1 = \begin{pmatrix} 1/k_1 & & & \\ -1/k_2 & 1/k_2 & & \\ & \ddots & \ddots & \\ & & -1/k_N & 1/k_N \end{pmatrix}$$

Matrix formulation (J. Rannou, T. Tran's Thesis)

$$\begin{cases} d_t u = \dot{u} \\ d_t \dot{u} = \ddot{u} \\ \ddot{u} - a^2 u = 0 \end{cases} \qquad \begin{cases} \frac{1}{k_n}(u^n - u^{n-1}) = \frac{1}{2}(\dot{u}^n + \dot{u}^{n-1}) \\ \frac{1}{k_n}(\dot{u}^n - \dot{u}^{n-1}) = \frac{1}{2}(\ddot{u}^n + \ddot{u}^{n-1}) \\ \ddot{u}^n - a^2 u^n = 0 \end{cases}$$

$$\boxed{\mathbf{u} = (u^1, \cdots, u^N) \qquad (B + aI)\mathbf{u} = f}$$

$$B = (C^{-1} B_1)^2$$

$$C = \frac{1}{2}\begin{pmatrix} 1 & & & \\ 1 & 1 & & \\ & \ddots & \ddots & \\ & & 1 & 1 \end{pmatrix} \qquad B_1 = \frac{1}{k_1}\begin{pmatrix} 1 & & & \\ -\frac{1}{\rho} & \frac{1}{\rho} & & \\ & \ddots & \ddots & \\ & & -\frac{1}{\rho^{N-1}} & \frac{1}{\rho^{N-1}} \end{pmatrix}$$

Computation of the eigenvectors

Special family : triangular unipotent Toeplitz matrices

$$T(X_1, \dots, X_{M-1}) = \begin{pmatrix} 1 & & & & \\ X_1 & \ddots & & 0 & \\ X_2 & \ddots & 1 & & \\ \vdots & \ddots & \ddots & \ddots & \\ X_{N-1} & & X_2 & X_1 & 1 \end{pmatrix}$$

THEOREM $k_n = \rho^{n-1} k_1 \implies B = VDV^{-1}$, with

$$V = T(P_1, \dots, P_{N-1}), \quad \text{with} \quad P_n := \prod_{j=1}^{n} \frac{1+\rho^j}{1-\rho^j},$$

$$V^{-1} = T(Q_1, \dots, Q_{N-1}), \quad \text{with} \quad Q_n := \rho^{-n} \prod_{j=1}^{n} \frac{1+\rho^{-j+2}}{1-\rho^{-j}}$$

$$D = diag(\frac{4}{{k_1}^2}, \cdots, \frac{4}{{k_N}^2})$$

Sketch of proof

THEOREM

$$(1) \quad V = T(P_1, \ldots, P_{N-1}) \quad P_n = \prod_{i=1}^{n} \frac{1+\rho^j}{1-\rho^j} \quad \text{easy}$$

Sketch of proof

THEOREM

$$(1) \quad V = T(P_1, \ldots, P_{N-1}) \quad P_n = \prod_{i=1}^{n} \frac{1+\rho^j}{1-\rho^j} \quad \text{easy}$$

$$(2) \quad V^{-1} = T(Q_1, \ldots, Q_{N-1}) \quad Q_n = \prod_{i=1}^{n} \frac{1+\rho^{-j+2}}{1-\rho^{-j}} \quad ??$$

Sketch of proof

THEOREM

$$(1) \quad V = T(P_1, \ldots, P_{N-1}) \quad P_n = \prod_{i=1}^{n} \frac{1+\rho^j}{1-\rho^j} \quad \text{easy}$$

$$(2) \quad V^{-1} = T(Q_1, \ldots, Q_{N-1}) \quad Q_n = \prod_{i=1}^{n} \frac{1+\rho^{-j+2}}{1-\rho^{-j}} \quad ??$$

Equivalent to proving that

$$P_n + P_{n-1}Q_1 + \ldots + P_1 Q_{n-1} + Q_n = 0 \text{ for } 1 \leq n \leq N-1$$

Convention: $P_0 = Q_0 = 1$.

Sketch of proof

$$\boxed{\sum_{k=0}^{n} P_k \, Q_{n-k} = 0, \quad P_n = \prod_{i=1}^{n} \frac{1+\rho^j}{1-\rho^j}}$$

Sketch of proof

$$\boxed{\sum_{k=0}^{n} P_k\, Q_{n-k} = 0, \quad P_n = \prod_{i=1}^{n} \frac{1+\rho^j}{1-\rho^j}}$$

Gauss' hypergeometric series (1812)

$${}_2F_1(a_1, a_2; b; x) := \sum_{n=0}^{\infty} \frac{[a_1]^n [a_2]^n}{[b]^n\, n!} x^n,$$

$$[a]^n := a(a+1)\cdots(a+n-1) = \frac{\Gamma(a+n)}{\Gamma(a)}$$

Sketch of proof

$$\boxed{\sum_{k=0}^{n} P_k\, Q_{n-k} = 0, \quad P_n = \prod_{i=1}^{n} \frac{1+\rho^j}{1-\rho^j}}$$

Gauss' hypergeometric series (1812)

$$ {}_2F_1(a_1, a_2; b; x) := \sum_{n=0}^{\infty} \frac{[a_1]^n [a_2]^n}{[b]^n\, n!} x^n, $$

$$ [a]^n := a(a+1)\cdots(a+n-1) = \frac{\Gamma(a+n)}{\Gamma(a)} $$

Heine's q-hypergeometric series (1847)

$$ {}_2\varphi_1(a_1, a_2; b; \rho; x) := \sum_{n=0}^{\infty} \frac{(a_1;\rho)_n (a_2;\rho)_n}{(b;\rho)_n (\rho;\rho)_n} x^n, $$

$$ (a;\rho)_n := (1-a)(1-\rho a)\cdots(1-\rho^{n-1}a) $$

Sketch of proof

$$\sum_{k=0}^{n} P_k\, Q_{n-k} = 0, \quad P_n = \prod_{i=1}^{n} \frac{1+\rho^j}{1-\rho^j}$$

Gauss' hypergeometric series (1812)

$${}_2F_1(a_1, a_2; b; x) := \sum_{n=0}^{\infty} \frac{[a_1]^n [a_2]^n}{[b]^n\, n!} x^n,$$

$$[a]^n := a(a+1)\cdots(a+n-1) = \frac{\Gamma(a+n)}{\Gamma(a)}$$

Summation formula

$${}_2F_1(a_1, a_2; b; 1) = \frac{\Gamma(b)\Gamma(b-a_1-a_2)}{\Gamma(b-a_1)\Gamma(b-a_2)}$$

Heine's q-hypergeometric series (1847)

$${}_2\varphi_1(a_1, a_2; b; \rho; x) := \sum_{n=0}^{\infty} \frac{(a_1;\rho)_n (a_2;\rho)_n}{(b;\rho)_n (\rho;\rho)_n} x^n,$$

$$(a;\rho)_n := (1-a)(1-\rho a)\cdots(1-\rho^{n-1} a)$$

Summation formula

$${}_2\varphi_1(a_1, a_2; b; \rho; \frac{b}{a_1 a_2}) = \frac{(\frac{b}{a_1};\rho)_\infty (\frac{b}{a_2};\rho)_\infty}{(b;\rho)_\infty (\frac{b}{a_1 a_2};\rho)_\infty}$$

Sketch of proof

$$\sum_{k=0}^{n} P_k\, Q_{n-k} = 0, \quad P_n = \prod_{i=1}^{n} \frac{1+\rho^j}{1-\rho^j}$$

Gauss' hypergeometric series (1812)

$${}_2F_1(a_1, a_2; b; x) := \sum_{n=0}^{\infty} \frac{[a_1]^n [a_2]^n}{[b]^n\, n!} x^n,$$

$$[a]^n := a(a+1)\cdots(a+n-1) = \frac{\Gamma(a+n)}{\Gamma(a)}$$

Summation formula

$${}_2F_1(a_1, a_2; b; 1) = \frac{\Gamma(b)\Gamma(b-a_1-a_2)}{\Gamma(b-a_1)\Gamma(b-a_2)}$$

Heine's q-hypergeometric series (1847)

$${}_2\varphi_1(a_1, a_2; b; \rho; x) := \sum_{n=0}^{\infty} \frac{(a_1;\rho)_n (a_2;\rho)_n}{(b;\rho)_n (\rho;\rho)_n} x^n,$$

$$(a;\rho)_n := (1-a)(1-\rho a)\cdots(1-\rho^{n-1}a)$$

Summation formula

$${}_2\varphi_1(a_1, a_2; b; \rho; \frac{b}{a_1 a_2}) = \frac{(\frac{b}{a_1};\rho)_\infty (\frac{b}{a_2};\rho)_\infty}{(b;\rho)_\infty (\frac{b}{a_1 a_2};\rho)_\infty}$$

$$P_n = \frac{(-\rho;\rho)_n}{(\rho;\rho)_n}$$

Sketch of proof, continue

$$\boxed{\sum_{k=0}^{n} P_k\, Q_{n-k} = 0, \quad P_n = \prod_{i=1}^{n} \frac{1+\rho^j}{1-\rho^j} = \frac{(-\rho;\rho)_n}{(\rho;\rho)_n}}$$

$${}_2\varphi_1(a_1, a_2; b; \rho; \frac{b}{a_1 a_2}) := \sum_{k=0}^{\infty} \frac{(a_1;\rho)_k (a_2;\rho)_k}{(b;\rho)_k (\rho;\rho)_k} \left(\frac{b}{a_1 a_2}\right)^k = \frac{(\frac{b}{a_1};\rho)_\infty (\frac{b}{a_2};\rho)_\infty}{(b;\rho)_\infty (\frac{b}{a_1 a_2};\rho)_\infty}$$

$$(a;\rho)_k := \prod_{i=0}^{k-1} (1-\rho^i a)$$

Sketch of proof, continue

$$\sum_{k=0}^{n} P_k\, Q_{n-k} = 0, \quad P_n = \prod_{i=1}^{n} \frac{1+\rho^j}{1-\rho^j} = \frac{(-\rho;\rho)_n}{(\rho;\rho)_n}$$

$$ {}_2\varphi_1(a_1, a_2; b; \rho; \frac{b}{a_1 a_2}) := \sum_{k=0}^{\infty} \frac{(a_1;\rho)_k (a_2;\rho)_k}{(b;\rho)_k (\rho;\rho)_k} \left(\frac{b}{a_1 a_2}\right)^k = \frac{(\frac{b}{a_1};\rho)_\infty (\frac{b}{a_2};\rho)_\infty}{(b;\rho)_\infty (\frac{b}{a_1 a_2};\rho)_\infty}$$

$$(a;\rho)_k := \prod_{i=0}^{k-1} (1-\rho^i a)$$

q-Zhu-Vandermonde formula

Sketch of proof, continue

$$\boxed{\sum_{k=0}^{n} P_k\, Q_{n-k} = 0, \quad P_n = \prod_{j=1}^{n} \frac{1+\rho^j}{1-\rho^j} = \frac{(-\rho;\rho)_n}{(\rho;\rho)_n}}$$

$$ {}_2\varphi_1(a_1, a_2; b; \rho; \frac{b}{a_1 a_2}) := \sum_{k=0}^{\infty} \frac{(a_1;\rho)_k (a_2;\rho)_k}{(b;\rho)_k (\rho;\rho)_k} \left(\frac{b}{a_1 a_2}\right)^k = \frac{(\frac{b}{a_1};\rho)_\infty (\frac{b}{a_2};\rho)_\infty}{(b;\rho)_\infty (\frac{b}{a_1 a_2};\rho)_\infty}$$

$$(a;\rho)_k := \prod_{i=0}^{k-1} (1-\rho^i a)$$

q-Zhu-Vandermonde formula

$$a_1 = \rho^{-k},\ a_2 = -\rho,\ b = -\rho^{-k+2}, \quad \sum_{k=0}^{n} \frac{(-\rho;\rho)_k (\rho^{-n};\rho)_k}{(\rho;\rho)_k (-\rho^{-n+2};\rho)_k} \rho^k = 0.$$

Matrix S, properties

Normalize the eigenvectors with respect to the ℓ^2 norm: $S = V\tilde{D}$, $\tilde{d}_i = 1/\|V^{(i)}\|_2$.

$$B = VDV^{-1} = SDS^{-1}.$$

Roundoff estimate

$$(1)\ (B + aI)\mathbf{u} = F, \qquad (2)\ \hat{S}(D + aI)\widehat{S^{-1}}\hat{\mathbf{u}} = F$$

Backward error analysis (Higham, Golub):$\underline{u}$ denotes the machine precision

$$(2) \iff (B+\delta B)\hat{\mathbf{u}} = F, \quad \|\delta B\| \leq (2N+1)\underline{u}\| \, |S||S^{-1}| \, \| \|D+aI\| + \mathcal{O}(\underline{u}^2).$$

$$\frac{\|\mathbf{u} - \hat{\mathbf{u}}\|}{\|\mathbf{u}\|} \leq \mathrm{cond}(B)\frac{\|\delta B\|}{\|B\|} \leq (2N+1)\underline{u} \, \|B^{-1}\| \, \| \, |S||S^{-1}| \, \| \, \|D + aI\|$$

Roundoff estimate

$$(1)\ (B+aI)\mathbf{u}=F, \qquad (2)\ \hat{S}(D+aI)\widehat{S^{-1}}\hat{\mathbf{u}}=F$$

Backward error analysis (Higham, Golub): $\underline{u}$ denotes the machine precision

$$(2) \iff (B+\delta B)\hat{\mathbf{u}}=F, \quad \|\delta B\|\le(2N+1)\underline{u}\|\,|S||S^{-1}|\,\|\|D+aI\|+\mathcal{O}(\underline{u}^2).$$

$$\frac{\|\mathbf{u}-\hat{\mathbf{u}}\|}{\|\mathbf{u}\|}\le \mathrm{cond}(B)\frac{\|\delta B\|}{\|B\|}\le(2N+1)\underline{u}\ \|B^{-1}\|\ \|\,|S||S^{-1}|\,\|\ \|D+aI\|$$

THEOREM.

$$\frac{\|\mathbf{u}-\hat{\mathbf{u}}\|_\infty}{\|\mathbf{u}\|_\infty}\lesssim \boxed{\underline{u}\ \psi_1(\tfrac{aT}{2N},N)\varepsilon^{-(N-1)}},$$

where $\psi_1(y,N):=\dfrac{2^{2(N+1)}}{(N-1)!}(1+2N(N-1))(1+y^2)$.

Sharper estimate: $\psi_3(y,N):=\frac{2^{2N-\frac{1}{2}}N}{(N-1)!}\frac{1}{y^2+1}$.

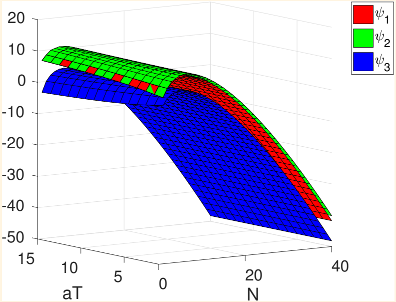


Figure: Comparison of the logarithm of the functions ψ_j, $j = 1, 2, 3$

Total error

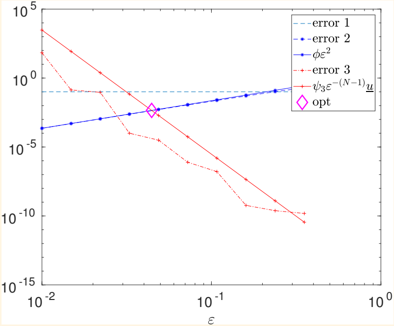
$$\underbrace{\frac{|u(t_N) - \widehat{u}_N|}{|u_0|}}_{\text{TOTAL ERROR}} \lesssim \underbrace{\frac{|u(t_N) - \overline{u}_N|}{|u_0|}}_{\text{ERROR 1: APPROXIMATION WITH EQUAL TIME STEPS}}$$

$$+ \underbrace{\frac{|\overline{u}_N - u_N|}{|u_0|}}_{\text{ERROR 2: DUE TO HETEROGENEOUS TIME STEPS}} \qquad \phi(y, N)\varepsilon^2$$

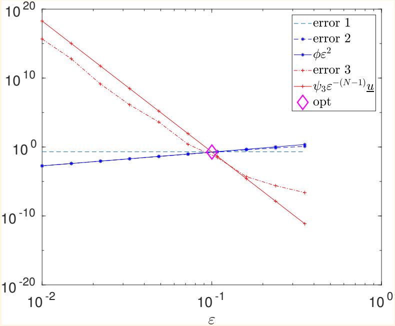
$$+ \underbrace{\frac{|u_N - \widehat{u}_N|}{|u_0|}}_{\text{ERROR 3: DUE TO DIAGONALIZATION}} \qquad \psi_3(y, N)\, \underline{u}\, \varepsilon^{-(N-1)}$$

Total error

$$\textsc{Total error} \lesssim \quad \textsc{error 1}\text{: approximation with equal time steps}$$
$$+ \quad \textsc{error 2}\text{: due to heterogeneous time steps}$$
$$+ \quad \textsc{error 3}\text{: due to diagonalization}$$



$T = 5, \; a = 1, \; N = 10$



$T = 10, \; a = 1, \; N = 20$

Optimization of ε

THEOREM For $\varepsilon = \varepsilon^*(aT, N)$ with

$$\varepsilon^*(aT,N) = \left(\frac{3\ 2^{2N}}{(N^2-1)(N-1)!} \, \frac{1+y^2}{y^3} \, \underline{u} \right)^{\frac{1}{N+1}}, \quad \text{with } y = \frac{aT}{2N},$$

the error due to time parallelization is asymptotically comparable to the one produced by the geometric time partition.

Optimization of ε

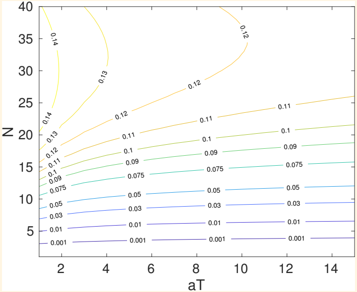
THEOREM For $\varepsilon = \varepsilon^*(aT, N)$ with

$$\varepsilon^*(aT, N) = \left(\frac{3\ 2^{2N}}{(N^2-1)(N-1)!} \, \frac{1+y^2}{y^3} \, \underline{u} \right)^{\frac{1}{N+1}}, \quad \text{with } y = \frac{aT}{2N},$$

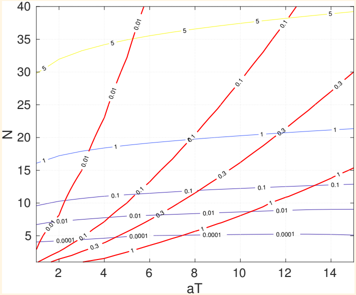
the error due to time parallelization is asymptotically comparable to the one produced by the geometric time partition.

$$\underbrace{\phi(y, N)\varepsilon^2}_{\text{Discretization error}} = \underbrace{\psi_3(y, N)\, \underline{u}\, \varepsilon^{-(N-1)}}_{\text{Parallelization error}}$$

Limiting value $\varepsilon^*(aT, N)$



$\varepsilon^*(aT, N)$



error 2/error 1(blue), and error 1 (red).

One dimensional wave equation

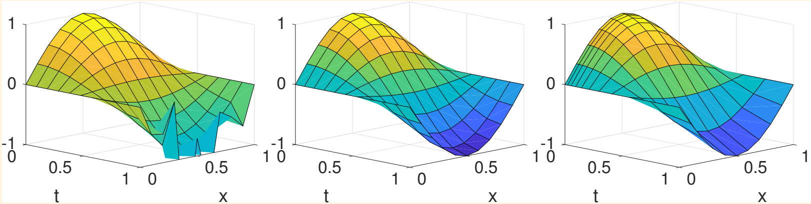


Figure: Approximate solutions obtained by the time parallel algorithm using diagonalization. Left: $\varepsilon = 0.015$. Middle: $\varepsilon = \varepsilon^* = 0.05$. Right: $\varepsilon = 0.3$.

Two dimensional wave equation

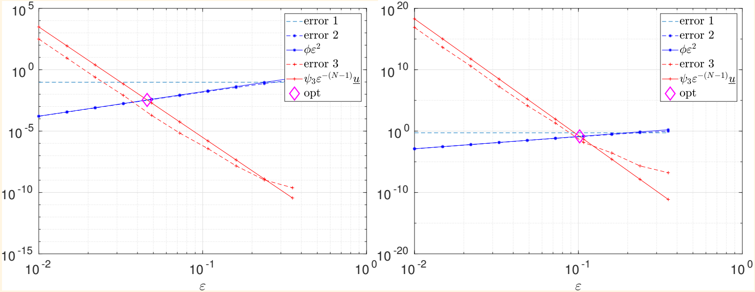
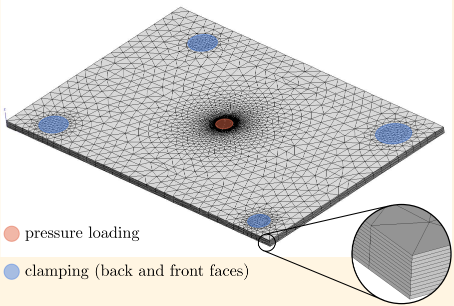


Figure: Discretization and parallelization errors in 1d, together with our theoretical bounds for the PDE. Left: $T = 1$, $N = 10$. Right: $T = 2$, $N = 20$.

Description

Response of a carbon/epoxy laminated composite panel (used in aeronautical industry) to an impact-like loading (transverse isotropic Hooke law).



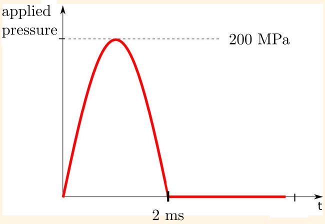


Figure: Mesh configuration and loading for the elasticity problem.

2000 time steps over the 10ms simulation range. 152607 degrees of freedom, 2000 time steps over the 10ms simulation range (time windows).

Results, MPI

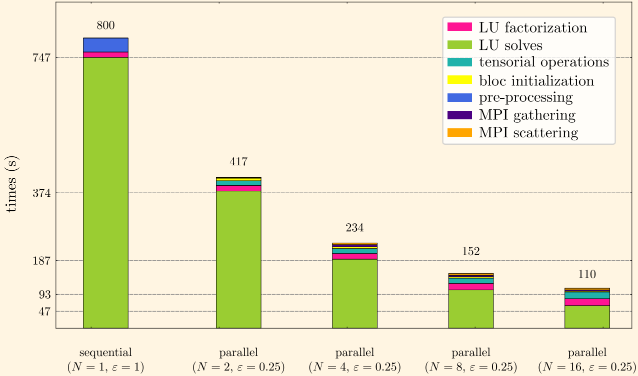


Figure: Computing times for the industrial elasticity problem.

Results

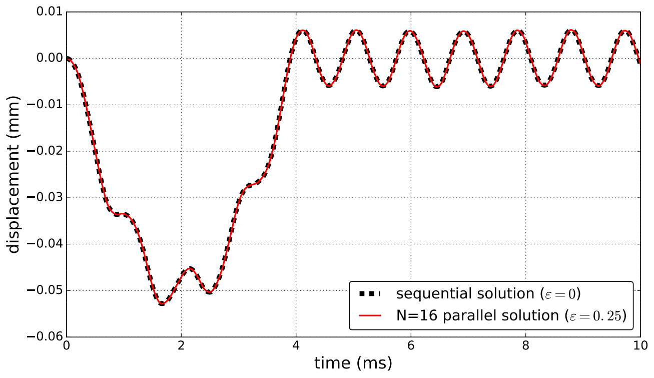


Figure: Deflection of the central node on the back face of the plate for the sequential and the parallel solution with $N = 16$.

N	2	4	8	16
$\text{Eff} := \dfrac{Time(1proc)}{N \times Time(Nproc)}$	0.96	0.86	0.66	0.45

Improving the efficiency: asynchronous computations

N	Numwin	Time	Error	Eff
1	128	0.497E+01	5.77E-007	
2	64	0.254E+01	6.13E-007	97.83 %
4	32	0.132E+01	7.71E-007	94.13 %
8	16	0.709E+00	1.71E-006	88.75 %
16	8	0.407E+00	5.15E-005	77.65 %

Improving the efficiency: asynchronous computations

N	Numwin	Time	Error	Eff
1	128	0.497E+01	5.77E-007	
2	64	0.254E+01	6.13E-007	97.83 %
4	32	0.132E+01	7.71E-007	94.13 %
8	16	0.709E+00	1.71E-006	88.75 %
16	8	0.407E+00	5.15E-005	77.65 %

N	2	4	8	16
CG	0.243E+01	0.125E+01	0.636E+00	0.319E+00
Total	0.254E+01	0.132E+01	0.709E+00	0.407E+00

Outline

Conclusion

1. Robust strategy for parallelization in time. Independent of the space-discretization.

Conclusion

1. Robust strategy for parallelization in time. Independent of the space-discretization.
2. The gain in optimal number of processors is significative: one could solve the problem using 30 processors, and would obtain an error which is within a factor two of the sequential computation.

Conclusion

1. Robust strategy for parallelization in time. Independent of the space-discretization.
2. The gain in optimal number of processors is significative: one could solve the problem using 30 processors, and would obtain an error which is within a factor two of the sequential computation.
3. Extension to nonlinear problems, coupled with Newton (DD23).

Perspectives

1. Parallelization in space in combination with the time-parallel method to solve the PDE thus adding another dimension to the parallelization process through a completely parallel time-space subdomains.

Perspectives

1. Parallelization in space in combination with the time-parallel method to solve the PDE thus adding another dimension to the parallelization process through a completely parallel time-space subdomains.
2. Application to control problems.

Outline

A few references for iterative methods

J.-L. Lions, Y. Maday, and G. Turinici.
Résolution d'EDP par un schéma en temps "pararéel".
C. R. Acad. Sci. Paris Sér. I Math., 332(7):661–668, 2001.

Amodio, Pierluigi, and Luigi Brugnano.
Parallel solution in time of ODEs: some achievements and perspectives.
Applied Numerical Mathematics ,59 (3): 424–435, 2009.

M. J. Gander and S. Güttel.
PARAEXP: A parallel integrator for linear initial-value problems.
SIAM Journal on Scientific Computing, 35(2):C123–C142, 2013.

References for the direct method

Y. Maday and E. M. Rønquist.
Parallelization in time through tensor-product space–time solvers.
Comptes Rendus Mathematiques, 346(1):113–118, 2008.

J. Rannou, J. Ryan,
Time parallelization of linear transient dynamic problems through the Newmark tensor-product form.
ECCOMAS, Wien, september 2012

M. Gander, L. Halpern, J. Ryan, and T. T. B. Tran.
A direct solver for time parallelization.
DD22, Lugano, september 2014, proceeding to appear

M. Gander, L. Halpern, J. Rannou, and J. Ryan
A Direct Time Parallel Solver by Diagonalization for the Wave Equation.
to be submitted soon

Nonlinear problems

$$u_t = f(u), \qquad \frac{u_n - u_{n-1}}{k_n} = f(u_n), \qquad \mathbf{F}(\mathbf{u}) := B\mathbf{u} - f(\mathbf{u}) = 0$$

Newton's method, $D(\mathbf{u}) := \mathrm{diag}(f'(u_1), f'(u_2), \ldots, f'(u_n))$

$$(B - D(\mathbf{u}^{m-1}))\mathbf{u}^m = \mathbf{f}(\mathbf{u}^{m-1}) - D(\mathbf{u}^{m-1})\mathbf{u}^{m-1},$$

Quasi-Newton $\qquad D(\mathbf{u}) \approx \frac{1}{N}\sum_{j=1}^{n} f'(u_j)\, I.$

Nonlinear problems

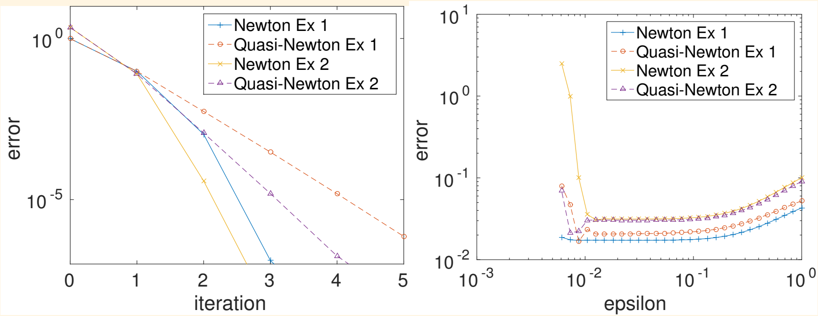


Figure: Left: linear convergence of the time parallel Quasi-Newton method for two model problems($-u^2$ and $\sqrt{u}$) . Right: accuracy for different choices of the time grid stretching ε.