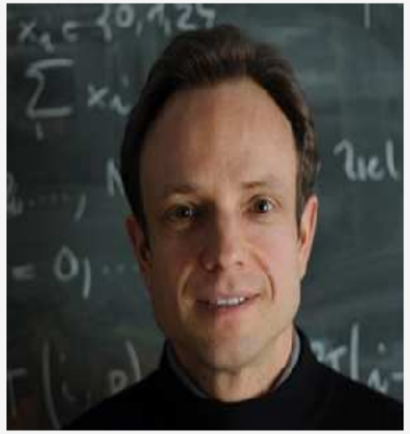
ETH Lecture 401-0674-00L Numerical Methods for Partial Differential Equations

Homework Problem Video Tutorial

Problem 6-5: Symplectic Timestepping for Wave Equations

Prof. R. Hiptmair, SAM, ETH Zurich



Date: April 19, 2019

$\Omega \subset \mathbb{R}^2$, Wave propagation phenomenon

$$\frac{\partial^2 u}{\partial t^2} - \Delta u + c(\boldsymbol{x})u = 0 \quad \text{on} \quad \Omega \times [0,T]\,, \tag{6.5.1a}$$

$$\mathbf{grad}\, u \cdot \boldsymbol{n} = 0 \quad \text{on} \quad \partial\Omega \times [0,T]\,, \quad \text{[hom. Neumann]} \tag{6.5.1b}$$

$$u(0) = u_0\,, \quad \frac{\partial u}{\partial t}(0) = v_0 \quad \text{on} \quad \Omega\,, \tag{6.5.1c}$$

controlled source, $c \geq 0$ initial condition

a) Spatial variational formulation

· Test with $v \in H^1(\Omega)$ & integrate & i.b.p.

$u : [0,T] \to H^1(\Omega)$

$$\begin{cases} \displaystyle\int_\Omega \left(\frac{\partial^2 u}{\partial t^2} v + \mathbf{grad}\, u \cdot \mathbf{grad}\, v + cuv \right) \mathrm{d}\boldsymbol{x} = 0 \quad \forall v \in H^1(\Omega) \quad (\textbf{test space})\,, \\ u(0) = u_0 \in H^1(\Omega)\,. \end{cases} \tag{6.5.2}$$

[Boundary terms vanish due to $\mathbf{grad}\, u \cdot n = 0$]

b) Write down a non-trivial quadratic functional $E : H^1(\Omega) \to \mathbb{R}$, which is conserved during the evolutionm, that is, for which

$$t \mapsto E(\{\boldsymbol{x} \mapsto u(\boldsymbol{x},t)\}) \equiv \text{const} \quad \text{if } u \text{ solves (6.5.1).}$$

(6.5.2) as abstract wave propagation problem

$u : [0,T] \to V_0 : \; m(\ddot{u}, v) + a(u,v) = 0 \quad \forall v \in V_0$

2

$$u : [0,T] \to H_0^1(\Omega): \quad \int_\Omega \rho(x) \cdot \frac{\partial^2 u}{\partial t^2} v \,dx + \int_\Omega \sigma(x) \,\mathbf{grad}\, u \cdot \mathbf{grad}\, v \,dx = 0 \quad \forall v \in H_0^1(\Omega) \tag{6.3.2.14}$$

$\updownarrow$

$$\boxed{u \in V_0: \quad \mathsf{m}(\ddot{u}, v) + \mathsf{a}(u, v) = 0 \quad \forall v \in V_0} \,, \tag{6.3.2.15}$$

$\frac{d}{dt} E(u(t)) = m(\ddot{u}, \dot{u}) + a(u, \dot{u}) = 0$

by with $v := \dot{u}$

Theorem 6.3.2.16. Energy conservation in wave propagation

If $u : \tilde{\Omega} \mapsto \mathbb{R}$ solves (7.2.2.15), *then we have conservation of total energy in the sense that*

$$t \mapsto \tfrac{1}{2}\mathsf{m}\left(\frac{\partial u}{\partial t}, \frac{\partial u}{\partial t}\right) + \tfrac{1}{2}\mathsf{a}(u,u) \equiv \text{const}\,.$$

kinetic energy *elastic (potential) energy, see* (1.2.1.19)

$E(u(t)) = \frac{1}{2} m(\dot{u}, \dot{u}) + \frac{1}{2} a(u,u)$

$$\left.\begin{aligned} \mathsf{m}(u,v) &:= \int_\Omega u(x)v(x)\,dx\,, \\ \mathsf{a}(u,v) &:= \int_\Omega \mathbf{grad}\, u(x) \cdot \mathbf{grad}\, v(x) + c(x)u(x)v(x)\,dx\,. \end{aligned}\right\} \text{symmetric bilinear forms}$$

$$\boxed{E(u) := \tfrac{1}{2}\mathsf{m}\left(\frac{\partial u}{\partial t}, \frac{\partial u}{\partial t}\right) + \tfrac{1}{2}\mathsf{a}(u,u)}\,. \tag{6.5.3}$$

Proof based on product rule for symmetric b.l.f.

$\frac{d}{dt} s(f(t), f(t)) = 2 s(\dot{f}(t), f(t))$

MOL-Approach: $S_1^0(\mathcal{M})$-FEM, basis $\{b_h^\ell\}_{\ell=1}^N$

$$\mathbf{M}\frac{d^2\vec{\mu}}{dt^2}(t) + \mathbf{A}\vec{\mu}(t) = 0\,, \quad \text{and} \quad \vec{\mu}(0) = \vec{\mu}_0\,, \quad \frac{d\vec{\mu}}{dt}(0) = \vec{\eta}_0\,. \tag{6.5.4}$$

c, Matrices?

$$\left.\begin{aligned} M &= [m(b_h^j, b_h^i)]_{i,j=1}^N \\ A &= [a(b_h^j, b_h^i)]_{i,j}^N \end{aligned}\right\} \text{Galerkin matrices!}$$

Time stepping: proposed for ODE in the form

$$\frac{d}{dt}\begin{bmatrix}\mathbf{p}\\ \mathbf{q}\end{bmatrix} = \begin{bmatrix}\mathbf{f}(\mathbf{q},t)\\ \mathbf{g}(\mathbf{p})\end{bmatrix} \quad \text{with continuous functions} \quad \begin{aligned}\mathbf{f} &: \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^n\,,\\ \mathbf{g} &: \mathbb{R}^n \to \mathbb{R}^n\,.\end{aligned} \tag{6.5.5}$$

1st-order ODE on state space $\mathbb{R}^{2n}$

③ SSM starting with $p^{(0)}, q^{(0)} \in \mathbb{R}^n$

$$
\begin{array}{l}
\textbf{for } j := 1 \textbf{ to } M \textbf{ do} \\
\quad \mathbf{p}_{\text{in}} := \mathbf{p}^{(j-1)}, \quad \mathbf{q}_{\text{in}} := \mathbf{q}^{(j-1)} \\
\quad \textbf{for } k := 1 \textbf{ to } s \textbf{ do} \\
\qquad t_k := j\tau + \tau \sum_{\ell=1}^{k-1} a_\ell ; \\
\qquad \mathbf{p}_{\text{out}} := \mathbf{p}_{\text{in}} + \tau b_k \mathbf{f}(\mathbf{q}_{\text{in}}, t_k) ; \\
\qquad \mathbf{q}_{\text{out}} := \mathbf{q}_{\text{in}} + \tau a_k \mathbf{g}(\mathbf{p}_{\text{out}}) ; \\
\qquad \mathbf{p}_{\text{in}} := \mathbf{p}_{\text{out}}, \quad \mathbf{q}_{\text{in}} := \mathbf{q}_{\text{out}} ; \\
\quad \textbf{end} \\
\quad \mathbf{p}^{(j)} := \mathbf{p}_{\text{out}}, \quad \mathbf{q}^{(j)} := \mathbf{q}_{\text{out}} ; \\
\textbf{end}
\end{array}
\tag{6.5.6}
$$

d) Implementation for $\frac{d}{dt}\begin{bmatrix} p \\ q \end{bmatrix} = \begin{bmatrix} q \\ -p \end{bmatrix}$
for $n = 1$

$$\mathbf{f}(\mathbf{q}, t) = q \;,\quad \mathbf{g}(\mathbf{p}) = -p \quad \Rightarrow \quad f(q) := q \;,\quad g(p) := -p \,.$$

e) "Measuring" order of (6.5.6)
↳ [finding an upper bound]
rate of alg. cvg.

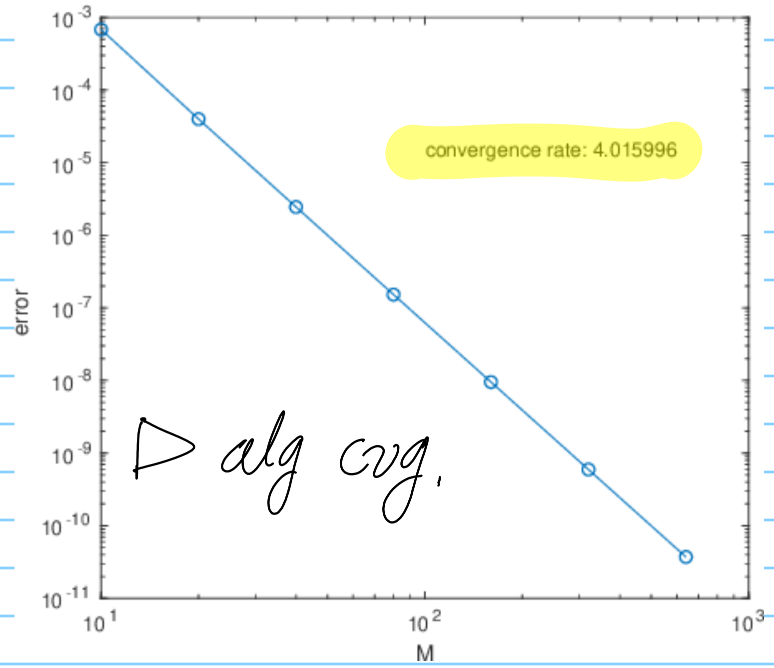
Experimentally determine the order of convergence of the single-step method implemented in Subproblem (6-5.d) in terms of the number of timesteps by

- integrating (6.5.8) over the period $[0, 2\pi]$,
- with initial values $p(0) = 0$, $q(0) = 1$,
- and using $M = 10, 20, 40, 80, 160, 320, 640$ timesteps. [uniform]

Tabulate the following errors at the final time:

$$\operatorname{err}(M) := |p(2\pi) - p^{(M)}| + |q(2\pi) - q^{(M)}| \,, \quad M = 10, 20, 40, 80, 160, 320, 640 \,.$$

To that end complete the function `sympTimesteppingODETest()` in the file `symplectictimesteppingwaves_ode.cc`, which is called from `main()`.



M	$\operatorname{err}(M)$
10	0.000680607
$10 \cdot 2^1$	3.9793e-05
$10 \cdot 2^2$	2.4477e-06
$10 \cdot 2^3$	1.52386e-07
$10 \cdot 2^4$	9.51496e-09
$10 \cdot 2^5$	5.94542e-10
$10 \cdot 2^6$	3.71564e-11

▷ $\tau \to \tau/2 \Rightarrow \text{err} \to \text{err}/16$

rate $\alpha \approx - \log\left(\text{err}(M_1)/\text{err}(M_2)\right) / \log\left(M_1/M_2\right)$

▷ Order ≤ 4

f, MOL-ODE : recast into (6.5.5)

$$\mathbf{M}\frac{d^2\vec{\mu}}{dt^2}(t) + \mathbf{A}\vec{\mu}(t) = 0\,, \quad \text{and} \quad \vec{\mu}(0) = \mu_0\,, \quad \frac{d\vec{\mu}}{dt}(0) = \vec{\eta}_0\,. \tag{6.5.4}$$

$$\Downarrow$$

$$\frac{d}{dt}\begin{bmatrix}\mathbf{p}\\ \mathbf{q}\end{bmatrix} = \begin{bmatrix}\mathbf{f}(\mathbf{q},t)\\ \mathbf{g}(\mathbf{p})\end{bmatrix} \quad \text{with continuous functions} \quad \begin{aligned}&\mathbf{f}: \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^n\,,\\ &\mathbf{g}: \mathbb{R}^n \to \mathbb{R}^n\,.\end{aligned} \tag{6.5.5}$$

Set: $p = \dot{\vec{\mu}}\,, \quad q = \vec{\mu}$

$$\dot{p} = \ddot{\vec{\mu}} = -M^{-1}Aq$$
$$\dot{q} = \dot{\vec{\mu}} = p$$

$$\frac{d}{dt}\begin{bmatrix}\mathbf{p}\\ \mathbf{q}\end{bmatrix} = \begin{bmatrix}\mathbf{f}(\mathbf{q},t)\\ \mathbf{g}(\mathbf{p})\end{bmatrix},$$

where

$$\mathbf{f}(\mathbf{q},t) = -\mathbf{M}^{-1}\mathbf{A}\mathbf{q}\;,\; \mathbf{g}(\mathbf{p}) = \mathbf{p} \quad \Leftrightarrow \quad \frac{d}{dt}\begin{bmatrix}\mathbf{p}\\ \mathbf{q}\end{bmatrix} = \begin{bmatrix}-\mathbf{M}^{-1}\mathbf{A}\mathbf{q}\\ \mathbf{p}\end{bmatrix}. \tag{6.5.11}$$

i, For (6.5.4) rewritten in the form (6.5.5) express the semi-discrete counterpart of the conserved "energy" $E(u)$ found in Sub-problem (6-5.b) in terms of $\mathbf{p}$ and $\mathbf{q}$.

$$E(\vec{\mu})(t) = \frac{1}{2}\cancel{\frac{d\vec{\mu}}{dt}}^{\top}p(t)\mathbf{M}\cancel{\frac{d\vec{\mu}}{dt}}p(t) + \frac{1}{2}\vec{\mu}(\cancel{t}q)^{\top}\mathbf{A}\vec{\mu}(\cancel{t}q)\,. \tag{6.5.16}$$

$$E(\mathbf{p},\mathbf{q}) = \frac{1}{2}\mathbf{p}(t)^{\top}\mathbf{M}\mathbf{p}(t) + \frac{1}{2}\mathbf{q}(t)^{\top}\mathbf{A}\mathbf{q}(t)\,. \tag{6.5.17}$$

g/ In the file `symplectictimesteppingwaves_assemble.h` create a LehrFEM++-based C++ function

```
template<typename FUNC_ALPHA, typename FUNC_GAMMA, typename
  FUNC_BETA>
 Eigen::SparseMatrix<double> assembleGalerkinMatrix(
     std::shared_ptr< UniformScalarFESpace<double>> fe_space_p,
     FUNC_ALPHA&& alpha, FUNC_GAMMA&& gamma, FUNC_BETA&& beta);
```

that, given a mesh $\mathcal{M}$, (approximately, using "safe" local numerical quadrature) computes the $\mathcal{S}_1^0(\mathcal{M})$-based finite element Galerkin matrix for the left-hand side of the variational boundary value problem

$$u \in H^1(\Omega): \int_{\Omega} \alpha(\mathbf{x})\,\mathbf{grad}\,u(\mathbf{x}) \cdot \mathbf{grad}\,v(\mathbf{x}) + \gamma(\mathbf{x})\,u(\mathbf{x})\,v(\mathbf{x})\,\mathrm{d}\mathbf{x} + \int_{\partial\Omega} \beta(\mathbf{x})\,u(\mathbf{x})\,v(\mathbf{x})\,\mathrm{d}S(\mathbf{x}) = \int_{\Omega} f(\mathbf{x})\,v(\mathbf{x})\,\mathrm{d}\mathbf{x} \quad \forall v \in H^1(\Omega)\,, \tag{6.5.12}$$

① functors $x \in \mathbb{R}^2 \to \mathbb{R}$

② Two stage assembly [green] + [yellow]

5

h,

Based on the function `assembleGalerkinMatrix()`, complete the implementation of a C++ class (in the files `symplectictimesteppingwaves.h` and `symplectictimesteppingwaves.cc`)

```
class SympTimestepWaveEq {
  public:
  template <typename FUNCTION>
  SympTimestepWaveEq(std::shared_ptr< UniformScalarFESpace<double>>
      fe_space_p,
                        FUNCTION&& c);
  void compTimestep(double tau, Eigen::VectorXd &p,Eigen::VectorXd
      &q) const;
  private:
  // suitable data members here
};
```

whose method `compTimestep` realizes a single timestep of length $\tau > 0$ of the method (6.5.6) with $s = 3$ and coefficients according to (6.5.7) for the first-order method-of-lines ODE found in Sub-problem (6-5.f). As arguments it takes two vectors `p` and `q` of basis expansion coefficients corresponding to $\mathbf{p}^{(j)}, \mathbf{q}^{(j)}$ from the previous timestep and *replaces* them with the counterparts of $\mathbf{p}^{(j+1)}, \mathbf{q}^{(j+1)}$. The constructor should initialize suitable member variables based on the coefficient function c. Take pains to achieve an efficient implementation by precomputing as much as possible, in particular a factored form of the matrix $\mathbf{M}$ to speed up the solution of linear systems in each timestep.

: assembleGalerkinMatrix() → A, M

Store M in factorized form

$$\frac{d}{dt}\begin{bmatrix}\mathbf{p}\\ \mathbf{q}\end{bmatrix} = \begin{bmatrix}\mathbf{f}(\mathbf{q},t)\\ \mathbf{g}(\mathbf{p})\end{bmatrix},$$

where

$$\mathbf{f}(\mathbf{q},t) = -\mathbf{M}^{-1}\mathbf{A}\mathbf{q} \quad , \quad \mathbf{g}(\mathbf{p}) = \mathbf{p} \quad \Leftrightarrow \quad \frac{d}{dt}\begin{bmatrix}\mathbf{p}\\ \mathbf{q}\end{bmatrix} = \begin{bmatrix}-\mathbf{M}^{-1}\mathbf{A}\mathbf{q}\\ \mathbf{p}\end{bmatrix}. \tag{6.5.11}$$

j,

$$E(\mathbf{p},\mathbf{q}) = \tfrac{1}{2}\mathbf{p}(t)^\top \mathbf{M}\mathbf{p}(t) + \tfrac{1}{2}\mathbf{q}(t)^\top \mathbf{A}\mathbf{q}(t) \ . \tag{6.5.17}$$

Augment the class **SympTimestepWaveEq** by a member function

```
double SympTimestepWaveEq::computeEnergies(
    const Eigen::VectorXd &p,const Eigen::VectorXd &q);
```

that returns the "energy" quantity found in Sub-problem (6-5.i). The arguments `p` and `q` pass the basis expansion coefficient vectors corresponding to $\mathbf{p}$ and $\mathbf{q}$ in the first-order form of the method-of-lines ODE derived in Sub-problem (6-5.f).

Now M to be stored as sparse matrix

k, ...

m, Empirically determine CFL-constraint [Lecture 6.3.5]

if blow-up   if stable

τ_0 $\tau_2 := \frac{1}{2}(\tau_1+\tau_0)$ τ_1

stable   blow-up