Graz University of Technology
Institute of Applied Mechanics

Generalized Convolution Quadrature for an Elastodynamic BEM

Martin Schanz

BIRS-workshop: Computational and Numerical Analysis of Transient Problems in Acoustics, Elasticity, and Electromagnetism
Banff, Canada
January 19, 2016

Content

1 Generalized Convolution Quadrature Method
- Quadrature formula
- Algorithm

2 Boundary element formulations in dynamics
- Governing equations
- Boundary element formulation

3 Numerical Examples
- Problem setting
- Convergence study
- Wave propagation in a bar

Content

1 Generalized Convolution Quadrature Method
- Quadrature formula
- Algorithm

2 Boundary element formulations in dynamics
- Governing equations
- Boundary element formulation

3 Numerical Examples
- Problem setting
- Convergence study
- Wave propagation in a bar

Content

1 Generalized Convolution Quadrature Method
- Quadrature formula
- Algorithm

2 Boundary element formulations in dynamics
- Governing equations
- Boundary element formulation

3 Numerical Examples
- Problem setting
- Convergence study
- Wave propagation in a bar

Content

1. Generalized Convolution Quadrature Method
   - Quadrature formula
   - Algorithm
2. Boundary element formulations in dynamics
   - Governing equations
   - Boundary element formulation
3. Numerical Examples
   - Problem setting
   - Convergence study
   - Wave propagation in a bar

Convolution integral

- Convolution integral with the Laplace transformed function $\hat{f}(s)$

$$y(t) = (f * g)(t) = \left(\hat{f}(\partial_t) g\right)(t) = \int_0^t f(t-\tau) g(\tau) \, d\tau$$

$$= \frac{1}{2\pi i} \int_C \hat{f}(s) \underbrace{\int_0^t e^{s(t-\tau)} g(\tau) \, d\tau}_{x(t,s)} \, ds$$

- Integral is equivalent to solution of ODE

$$\frac{\partial}{\partial t} x(t,s) = s x(t,s) + g(t) \qquad \text{with} \quad x(t=0,s) = 0$$

- Implicit Euler for ODE, $[0,T] = [0, t_1, t_2, \ldots, t_N]$, variable time steps $\Delta t_i, i = 1, 2, \ldots, N$

$$x_n(s) = \frac{x_{n-1}(s)}{1 - \Delta t_n s} + \frac{\Delta t_n}{1 - \Delta t_n s} g_n = \sum_{j=1}^{n} \Delta t_j g_j \prod_{k=j}^{n} \frac{1}{1 - \Delta t_k s}$$

Convolution integral

- Convolution integral with the Laplace transformed function $\hat{f}(s)$

$$y(t) = (f * g)(t) = \left(\hat{f}(\partial_t) g\right)(t) = \int_0^t f(t-\tau) g(\tau) \, \mathrm{d}\tau$$

$$= \frac{1}{2\pi \mathrm{i}} \int_C \hat{f}(s) \underbrace{\int_0^t e^{s(t-\tau)} g(\tau) \, \mathrm{d}\tau}_{x(t,s)} \, \mathrm{d}s$$

- Integral is equivalent to solution of ODE

$$\frac{\partial}{\partial t} x(t,s) = s x(t,s) + g(t) \qquad \text{with} \quad x(t=0,s) = 0$$

- Implicit Euler for ODE , $[0,T] = [0, t_1, t_2, \ldots, t_N]$, variable time steps $\Delta t_i, i = 1, 2, \ldots, N$

$$x_n(s) = \frac{x_{n-1}(s)}{1 - \Delta t_n s} + \frac{\Delta t_n}{1 - \Delta t_n s} g_n = \sum_{j=1}^{n} \Delta t_j g_j \prod_{k=j}^{n} \frac{1}{1 - \Delta t_k s}$$

Time stepping formula

- Solution at the discrete time t_n

$$
\begin{aligned}
y(t_n) &= \frac{1}{2\pi \mathrm{i}} \int_C \hat{f}(s)\, x_n(s)\, \mathrm{d}s \\
&= \frac{1}{2\pi \mathrm{i}} \int_C \frac{\hat{f}(s)\,\Delta t_n}{1-\Delta t_n s} g_n \,\mathrm{d}s + \frac{1}{2\pi \mathrm{i}} \int_C \hat{f}(s) \frac{x_{n-1}(s)}{1-\Delta t_n s}\,\mathrm{d}s \\
&= \hat{f}\left(\frac{1}{\Delta t_n}\right) g_n + \frac{1}{2\pi \mathrm{i}} \int_C \hat{f}(s) \frac{x_{n-1}(s)}{1-\Delta t_n s}\,\mathrm{d}s .
\end{aligned}
$$

- Recursion formula for the implicit Euler

$$
\begin{aligned}
y(t_n) &= \frac{1}{2\pi \mathrm{i}} \int_C \hat{f}(s) \sum_{j=1}^{n} \Delta t_j g_j \prod_{k=j}^{n} \frac{1}{1-\Delta t_k s}\,\mathrm{d}s \\
&= \hat{f}\left(\frac{1}{\Delta t_n}\right) g_n + \sum_{j=1}^{n-1} \Delta t_j g_j \frac{1}{2\pi \mathrm{i}} \int_C \hat{f}(s) \prod_{k=j}^{n} \frac{1}{1-\Delta t_k s}\,\mathrm{d}s
\end{aligned}
$$

- Complex integral is solved with a quadrature formula

Time stepping formula

- Solution at the discrete time t_n

$$
\begin{aligned}
y(t_n) &= \frac{1}{2\pi i}\int_C \hat{f}(s)\, x_n(s)\,\mathrm{d}s \\
&= \frac{1}{2\pi i}\int_C \frac{\hat{f}(s)\,\Delta t_n}{1-\Delta t_n s} g_n\,\mathrm{d}s + \frac{1}{2\pi i}\int_C \hat{f}(s)\,\frac{x_{n-1}(s)}{1-\Delta t_n s}\,\mathrm{d}s \\
&= \hat{f}\left(\frac{1}{\Delta t_n}\right) g_n + \frac{1}{2\pi i}\int_C \hat{f}(s)\,\frac{x_{n-1}(s)}{1-\Delta t_n s}\,\mathrm{d}s\,.
\end{aligned}
$$

- Recursion formula for the implicit Euler

$$
\begin{aligned}
y(t_n) &= \frac{1}{2\pi i}\int_C \hat{f}(s) \sum_{j=1}^{n} \Delta t_j g_j \prod_{k=j}^{n} \frac{1}{1-\Delta t_k s}\,\mathrm{d}s \\
&= \hat{f}\left(\frac{1}{\Delta t_n}\right) g_n + \sum_{j=1}^{n-1} \Delta t_j g_j \frac{1}{2\pi i}\int_C \hat{f}(s) \prod_{k=j}^{n} \frac{1}{1-\Delta t_k s}\,\mathrm{d}s
\end{aligned}
$$

- Complex integral is solved with a quadrature formula

Time stepping formula

- Solution at the discrete time t_n

$$
\begin{aligned}
y(t_n) &= \frac{1}{2\pi\mathrm{i}} \int_C \hat{f}(s)\, x_n(s)\,\mathrm{d}s \\
&= \frac{1}{2\pi\mathrm{i}} \int_C \frac{\hat{f}(s)\,\Delta t_n}{1-\Delta t_n s} g_n \,\mathrm{d}s + \frac{1}{2\pi\mathrm{i}} \int_C \hat{f}(s)\,\frac{x_{n-1}(s)}{1-\Delta t_n s}\,\mathrm{d}s \\
&= \hat{f}\left(\frac{1}{\Delta t_n}\right) g_n + \frac{1}{2\pi\mathrm{i}} \int_C \hat{f}(s)\,\frac{x_{n-1}(s)}{1-\Delta t_n s}\,\mathrm{d}s .
\end{aligned}
$$

- Recursion formula for the implicit Euler

$$
\begin{aligned}
y(t_n) &= \frac{1}{2\pi\mathrm{i}} \int_C \hat{f}(s) \sum_{j=1}^{n} \Delta t_j g_j \prod_{k=j}^{n} \frac{1}{1-\Delta t_k s}\,\mathrm{d}s \\
&= \hat{f}\left(\frac{1}{\Delta t_n}\right) g_n + \sum_{j=1}^{n-1} \Delta t_j g_j \frac{1}{2\pi\mathrm{i}} \int_C \hat{f}(s) \prod_{k=j}^{n} \frac{1}{1-\Delta t_k s}\,\mathrm{d}s
\end{aligned}
$$

- Complex integral is solved with a quadrature formula

Algorithm

- First Euler step

$$y(t_1) = \hat{f}\left(\frac{1}{\Delta t_1}\right) g_1$$

with implicit assumption of zero initial condition

- For all steps $n = 2, \ldots, N$ the algorithm has two steps
  1. Update the solution vector x_{n-1} at all integration points s_ℓ with an implicit Euler step

$$x_{n-1}(s_\ell) = \frac{x_{n-2}(s_\ell)}{1 - \Delta t_{n-1} s_\ell} + \frac{\Delta t_{n-1}}{1 - \Delta t_{n-1} s_\ell} g_{n-1}$$

  for $\ell = 1, \ldots, N_Q$ with the number of integration points N_Q.

  2. Compute the solution of the integral at the actual time step t_n

$$y(t_n) = \hat{f}\left(\frac{1}{\Delta t_n}\right) g_n + \sum_{\ell=1}^{N_Q} \omega_\ell \frac{\hat{f}(s_\ell)}{1 - \Delta t_n s_\ell} x_{n-1}(s_\ell)$$

- Essential parameter: $N_Q = N \log(N)$, integration is dependent on $q = \frac{\Delta t_{max}}{\Delta t_{min}}$

Algorithm

- First Euler step

$$y(t_1) = \hat{f}\left(\frac{1}{\Delta t_1}\right) g_1$$

with implicit assumption of zero initial condition

- For all steps $n = 2, \ldots, N$ the algorithm has two steps

1. Update the solution vector x_{n-1} at all integration points s_ℓ with an implicit Euler step

$$x_{n-1}(s_\ell) = \frac{x_{n-2}(s_\ell)}{1 - \Delta t_{n-1} s_\ell} + \frac{\Delta t_{n-1}}{1 - \Delta t_{n-1} s_\ell} g_{n-1}$$

for $\ell = 1, \ldots, N_Q$ with the number of integration points N_Q.

2. Compute the solution of the integral at the actual time step t_n

$$y(t_n) = \hat{f}\left(\frac{1}{\Delta t_n}\right) g_n + \sum_{\ell=1}^{N_Q} \omega_\ell \frac{\hat{f}(s_\ell)}{1 - \Delta t_n s_\ell} x_{n-1}(s_\ell)$$

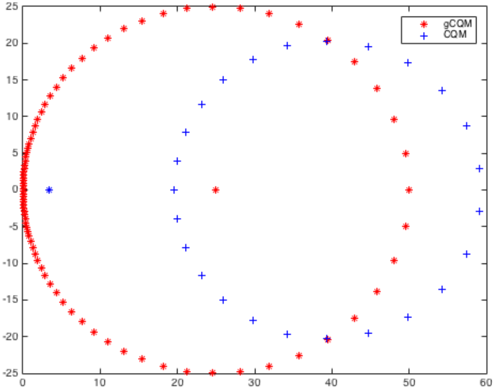
- Essential parameter: $N_Q = N \log(N)$, integration is dependent on $q = \frac{\Delta t_{max}}{\Delta t_{min}}$

Numerical integration

Integration weights and points

$$s_\ell = \gamma(\sigma_\ell) \qquad \omega_\ell = \frac{4K\left(k^2\right)}{2\pi i}\gamma'\left(\sigma_\ell\right)$$

for $N = 25, T = 5, t_n = \left(\frac{n}{N}\right)^\alpha T, \alpha = 1.5$



Equivalence CQM with constant step size

- gCQM with implicit Euler for constant time step sizes

$$
\begin{aligned}
y(t_n) &= \hat{f}\left(\frac{1}{\Delta t}\right) g_n + \sum_{j=1}^{n-1} \Delta t g_j \frac{1}{2\pi \mathrm{i}} \int_C \hat{f}(s) \prod_{k=j}^{n} \frac{1}{1-\Delta t s} \, \mathrm{d}s \\
&= \hat{f}\left(\frac{1}{\Delta t}\right) g_n + \sum_{j=1}^{n-1} g_j \frac{1}{2\pi \mathrm{i}} \int_C \hat{f}(s) \frac{\Delta t}{(1-\Delta t s)^{n-j+1}} \, \mathrm{d}s \, .
\end{aligned}
$$

- Substitution $z = 1 - \Delta t s$ $\Rightarrow$ original algorithm with BDF 1 $\gamma(z) = 1 - z$

$$
\begin{aligned}
y(t_n) &= \hat{f}\left(\frac{1}{\Delta t}\right) g_n + \sum_{j=1}^{n-1} g_j \frac{1}{2\pi \mathrm{i}} \int_C \hat{f}\left(\frac{1-z}{\Delta t}\right) z^{-(n-j)-1} \, \mathrm{d}z \\
&= \sum_{j=1}^{n} \frac{1}{2\pi \mathrm{i}} \int_C \hat{f}\left(\frac{\gamma(z)}{\Delta t}\right) z^{-(n-j)-1} \, \mathrm{d}z \, g_j = \sum_{j=1}^{n} \omega_{n-j}^{*}\left(\hat{f}, \Delta t\right) g_j
\end{aligned}
$$

Equivalence CQM with constant step size

- gCQM with implicit Euler for constant time step sizes

$$y(t_n) = \hat{f}\left(\frac{1}{\Delta t}\right) g_n + \sum_{j=1}^{n-1} \Delta t g_j \frac{1}{2\pi \mathrm{i}} \int_C \hat{f}(s) \prod_{k=j}^{n} \frac{1}{1-\Delta t s} \,\mathrm{d}s$$

$$= \hat{f}\left(\frac{1}{\Delta t}\right) g_n + \sum_{j=1}^{n-1} g_j \frac{1}{2\pi \mathrm{i}} \int_C \hat{f}(s) \frac{\Delta t}{(1-\Delta t s)^{n-j+1}} \,\mathrm{d}s\,.$$

- Substitution $z = 1 - \Delta t s$ $\Rightarrow$ original algorithm with BDF 1 $\gamma(z) = 1 - z$

$$y(t_n) = \hat{f}\left(\frac{1}{\Delta t}\right) g_n + \sum_{j=1}^{n-1} g_j \frac{1}{2\pi \mathrm{i}} \int_C \hat{f}\left(\frac{1-z}{\Delta t}\right) z^{-(n-j)-1} \,\mathrm{d}z$$

$$= \sum_{j=1}^{n} \frac{1}{2\pi \mathrm{i}} \int_C \hat{f}\left(\frac{\gamma(z)}{\Delta t}\right) z^{-(n-j)-1} \,\mathrm{d}z\, g_j = \sum_{j=1}^{n} \omega^{*}_{n-j}\left(\hat{f}, \Delta t\right) g_j$$

Content

1 Generalized Convolution Quadrature Method
- Quadrature formula
- Algorithm

2 Boundary element formulations in dynamics
- Governing equations
- Boundary element formulation

3 Numerical Examples
- Problem setting
- Convergence study
- Wave propagation in a bar

Elastodynamics

- Governing equation for elastodynamics

$$\begin{aligned} c_1^2\nabla(\nabla\cdot\mathbf{u}(\mathbf{x},t)) - c_2^2\nabla\times(\nabla\times\mathbf{u}(\mathbf{x},t)) &= \frac{\partial^2\mathbf{u}}{\partial t^2}(\mathbf{x},t) \quad && (\mathbf{x},t)\in\Omega\times(0,T) \\ \mathbf{u}(\mathbf{y},t) &= \mathbf{g}_D(\mathbf{y},t) && (\mathbf{y},t)\in\Gamma_D\times(0,T) \\ \mathbf{t}(\mathbf{y},t) &= \mathbf{g}_N(\mathbf{y},t) && (\mathbf{y},t)\in\Gamma_N\times(0,T) \\ \mathbf{u}(\mathbf{x},0) = \frac{\partial\mathbf{u}}{\partial t}(\mathbf{x},0) &= \mathbf{0} && (\mathbf{x},t)\in\Omega\times(0) \end{aligned}$$

  in domain Ω with boundary $\Gamma=\Gamma_D\cup\Gamma_N$

- Wave speeds

$$c_1=\sqrt{\frac{E\,(1-\nu)}{\rho\,(1-2\nu)\,(1+\nu)}} \qquad c_2=\sqrt{\frac{E}{\rho 2\,(1+\nu)}}\,,$$

  with Young's modulus E and Poisson's ratio ν

- Traction operator (Hooke's law)

$$\mathbf{t}(\mathbf{y},t)=(\mathcal{T}\mathbf{u})(\mathbf{y},t)=\lim_{\Omega\ni\mathbf{x}\to\mathbf{y}\in\Gamma}[\sigma(\mathbf{x},t)\cdot\mathbf{n}(\mathbf{y})]$$

Boundary integral equation

- First integral equation

$$(\mathcal{V}*\mathbf{t})(\mathbf{x},t)=\mathcal{C}(\mathbf{x})\mathbf{u}(\mathbf{x},t)+(\mathcal{K}*\mathbf{u})(\mathbf{x},t)\qquad(\mathbf{x},t)\in\Gamma\times(0,\infty)$$

- Integral operators

$$(\mathcal{V}*\mathbf{t})(\mathbf{x},t)=\int\limits_0^t\int\limits_\Gamma\mathbf{U}(\mathbf{x}-\mathbf{y},t-\tau)\mathbf{t}(\mathbf{y},\tau)\,\mathrm{d}s_{\mathbf{y}}\,\mathrm{d}\tau$$

$$\mathcal{C}(\mathbf{x})=\mathcal{I}+\lim_{\varepsilon\to0}\int\limits_{\partial B_\varepsilon(\mathbf{x})\cap\Omega}(\mathcal{T}_{\mathbf{y}}\mathbf{U})^\top(\mathbf{x}-\mathbf{y},0)\,\mathrm{d}s_{\mathbf{y}}$$

$$(\mathcal{K}*\mathbf{u})(\mathbf{x},t)=\lim_{\varepsilon\to0}\int\limits_0^t\int\limits_{\Gamma\setminus B_\varepsilon(\mathbf{x})}(\mathcal{T}_{\mathbf{y}}\mathbf{U})^\top(\mathbf{x}-\mathbf{y},t-\tau)\mathbf{u}(\mathbf{y},\tau)\,\mathrm{d}s_{\mathbf{y}}\,\mathrm{d}\tau$$

Spatial discretization

- Introduction of boundary elements τ_e

$$\Gamma_h = \bigcup_{e=1}^{N_e} \tau_e$$

- Introduction of shape functions

$$u_k(\mathbf{y},t) = \sum_{i=1}^{N} u_k^i(t)\varphi_i(\mathbf{y}) \qquad \text{and} \qquad t_k(\mathbf{y},t) = \sum_{j=1}^{M} t_k^j(t)\psi_j(\mathbf{y})$$

- Partitioning of the boundary and **collocation**

$$\left(\begin{bmatrix} \hat{\mathsf{V}}_{DD}(\partial_t) & -\hat{\mathsf{K}}_{DN}(\partial_t) \\ \hat{\mathsf{V}}_{ND}(\partial_t) & -\left(\mathsf{C}_{NN} + \hat{\mathsf{K}}_{NN}(\partial_t)\right) \end{bmatrix} \begin{bmatrix} \mathsf{t}_D \\ \mathsf{u}_N \end{bmatrix} \right)(t) = \left(\begin{bmatrix} \mathsf{C}_{DD} + \hat{\mathsf{K}}_{DD}(\partial_t) & -\hat{\mathsf{V}}_{DN}(\partial_t) \\ \hat{\mathsf{K}}_{ND}(\partial_t) & -\hat{\mathsf{V}}_{NN}(\partial_t) \end{bmatrix} \begin{bmatrix} \mathsf{g}_D \\ \mathsf{g}_N \end{bmatrix} \right)(t)$$

Temporal discretization

- Application of the gCQM to the convolution in time

$$\begin{bmatrix} \hat{\mathsf{V}}_{DD} & -\hat{\mathsf{K}}_{DN} \\ \hat{\mathsf{V}}_{ND} & -\left(\mathsf{C}_{NN}+\hat{\mathsf{K}}_{NN}\right) \end{bmatrix} \left(\Delta t_n^{-1}\right) \begin{bmatrix} \mathsf{t}_D^n \\ \mathsf{u}_N^n \end{bmatrix} = \begin{bmatrix} \mathsf{f}_D^n \\ \mathsf{f}_N^n \end{bmatrix} + \sum_{\ell=1}^{N_Q} \frac{\omega_\ell}{1-\Delta t_n s_\ell} \left\{ \begin{bmatrix} \hat{\mathsf{K}}_{DN}\mathsf{u}_N^{n-1} \\ \hat{\mathsf{K}}_{NN}\mathsf{u}_N^{n-1} \end{bmatrix}(s_\ell) - \begin{bmatrix} \hat{\mathsf{V}}_{DD}\mathsf{t}_D^{n-1} \\ \hat{\mathsf{V}}_{ND}\mathsf{t}_D^{n-1} \end{bmatrix}(s_\ell) \right\}$$

- Known right hand side

$$\begin{bmatrix} \mathsf{f}_D^n \\ \mathsf{f}_N^n \end{bmatrix} = \begin{bmatrix} \mathsf{C}_{DD}+\hat{\mathsf{K}}_{DD} & -\hat{\mathsf{V}}_{DN} \\ \hat{\mathsf{K}}_{ND} & -\hat{\mathsf{V}}_{NN} \end{bmatrix} \left(\Delta t_n^{-1}\right) \begin{bmatrix} \mathsf{g}_D^n \\ \mathsf{g}_N^n \end{bmatrix} + \sum_{\ell=1}^{N_Q} \frac{\omega_\ell}{1-\Delta t_n s_\ell} \left\{ \begin{bmatrix} \hat{\mathsf{K}}_{DD}\mathsf{g}_D^{n-1} \\ \hat{\mathsf{K}}_{ND}\mathsf{g}_D^{n-1} \end{bmatrix}(s_\ell) - \begin{bmatrix} \hat{\mathsf{V}}_{DN}\mathsf{g}_N^{n-1} \\ \hat{\mathsf{V}}_{NN}\mathsf{g}_N^{n-1} \end{bmatrix}(s_\ell) \right\}$$

- Complexity

$$\mathcal{O}(M^2N)+\mathcal{O}(M^2N_Q)) = \mathcal{O}(M^2N)+\mathcal{O}(M^2N\log N))$$

Temporal discretization

- Application of the gCQM to the convolution in time

$$\begin{bmatrix} \hat{\mathsf{V}}_{DD} & -\hat{\mathsf{K}}_{DN} \\ \hat{\mathsf{V}}_{ND} & -\left(\mathsf{C}_{NN}+\hat{\mathsf{K}}_{NN}\right) \end{bmatrix}\left(\Delta t_n^{-1}\right)\begin{bmatrix} \mathsf{t}_D^n \\ \mathsf{u}_N^n \end{bmatrix} = \begin{bmatrix} \mathsf{f}_D^n \\ \mathsf{f}_N^n \end{bmatrix} + \sum_{\ell=1}^{N_Q} \frac{\omega_\ell}{1-\Delta t_n s_\ell}\left\{ \begin{bmatrix} \hat{\mathsf{K}}_{DN}\mathsf{u}_N^{n-1} \\ \hat{\mathsf{K}}_{NN}\mathsf{u}_N^{n-1} \end{bmatrix}(s_\ell) - \begin{bmatrix} \hat{\mathsf{V}}_{DD}\mathsf{t}_D^{n-1} \\ \hat{\mathsf{V}}_{ND}\mathsf{t}_D^{n-1} \end{bmatrix}(s_\ell) \right\}$$

- Known right hand side

$$\begin{bmatrix} \mathsf{f}_D^n \\ \mathsf{f}_N^n \end{bmatrix} = \begin{bmatrix} \mathsf{C}_{DD}+\hat{\mathsf{K}}_{DD} & -\hat{\mathsf{V}}_{DN} \\ \hat{\mathsf{K}}_{ND} & -\hat{\mathsf{V}}_{NN} \end{bmatrix}\left(\Delta t_n^{-1}\right)\begin{bmatrix} \mathsf{g}_D^n \\ \mathsf{g}_N^n \end{bmatrix} + \sum_{\ell=1}^{N_Q} \frac{\omega_\ell}{1-\Delta t_n s_\ell}\left\{ \begin{bmatrix} \hat{\mathsf{K}}_{DD}\mathsf{g}_D^{n-1} \\ \hat{\mathsf{K}}_{ND}\mathsf{g}_D^{n-1} \end{bmatrix}(s_\ell) - \begin{bmatrix} \hat{\mathsf{V}}_{DN}\mathsf{g}_N^{n-1} \\ \hat{\mathsf{V}}_{NN}\mathsf{g}_N^{n-1} \end{bmatrix}(s_\ell) \right\}$$

- Complexity

$$\mathcal{O}(M^2 N) + \mathcal{O}(M^2 N_Q)) = \mathcal{O}(M^2 N) + \mathcal{O}(M^2 N \log N))$$

Content

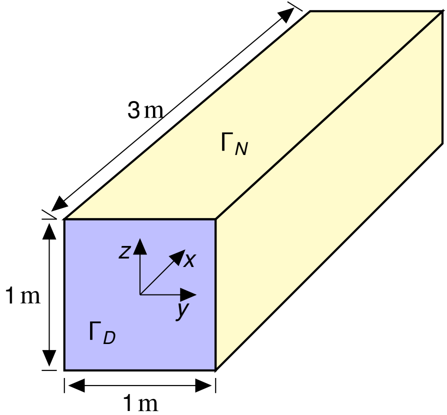
1 Generalized Convolution Quadrature Method
- Quadrature formula
- Algorithm

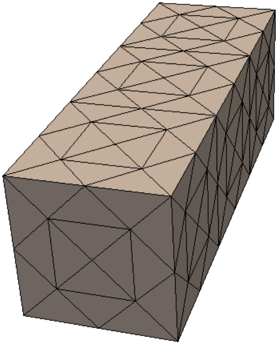
2 Boundary element formulations in dynamics
- Governing equations
- Boundary element formulation

3 Numerical Examples
- Problem setting
- Convergence study
- Wave propagation in a bar

Data for the tests I

- Geometry and mesh





- Material data: Young's modulus $E = 1\,{}^{\mathrm{N}}\!/\!_{\mathrm{m}^2}$, Poisson's ratio $\nu = 0$, and density $\rho = 1\,{}^{\mathrm{kg}}\!/\!_{\mathrm{m}^3}$
  $\Rightarrow$ wave speeds

$$c_1 = 1\,{}^{\mathrm{m}}\!/\!_{\mathrm{s}} \qquad c_2 = \sqrt{0.5}\,{}^{\mathrm{m}}\!/\!_{\mathrm{s}}\,.$$

Data for the tests II

- Displacements: linear shape functions, Tractions: constant shape functions
- Properties of the used meshes $\beta = c_1 \Delta t_{const} / h$

	Number		h	$\beta = 1$		$\beta = 0.25$		$\beta = 0.0625$	
	elements	nodes		Δt	N	Δt	N	Δt	N
1	56	30	1 m	1	5	0.25	20	0.0625	80
2	224	114	0.5 m	0.5	10	0.125	40	0.03125	160
3	896	450	0.25 m	0.25	20	0.0625	80	0.015625	320
4	3584	1794	0.125 m	0.125	40	0.03125	160	0.0078125	640

- Variable time steps

$$\Delta t_n = \left(n + \frac{(n-1)^{\alpha}}{N} \right) \Delta t_{const} \qquad \text{with} \quad T = N \Delta t_{const}$$

Convergence study

- Boundary condition from smooth solution of the PDE with
  - source at point $\mathbf{P} = (1.5, 2, 2)^T$
  - direction of source $\mathbf{d} = (1, 1, 1)^T$
  - temporal behavior of source

$$f(t) = e^{-2\left(t - \frac{r}{c\alpha} - \frac{2}{5}\right)^2}$$

- Error definition

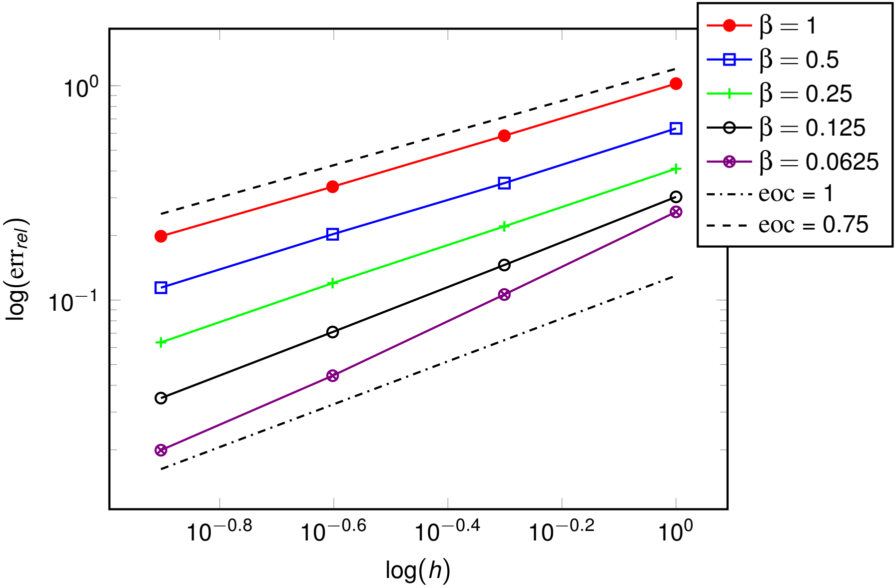
$$\mathrm{err}_{abs} = \sqrt{\sum_{n=0}^{N} \Delta t_n \|\mathbf{u}(\mathbf{x}, t_n) - \mathbf{u}_h(\mathbf{x}, t_n)\|^2_{L_2(\Gamma)}}$$

$$\mathrm{err}_{rel} = \mathrm{err}_{abs} \left(\sum_{n=0}^{N} \Delta t_n \|\mathbf{u}(\mathbf{x}, t_n)\|^2_{L_2(\Gamma)} \right)^{-\frac{1}{2}} \quad \forall \mathbf{x} \in \Gamma$$

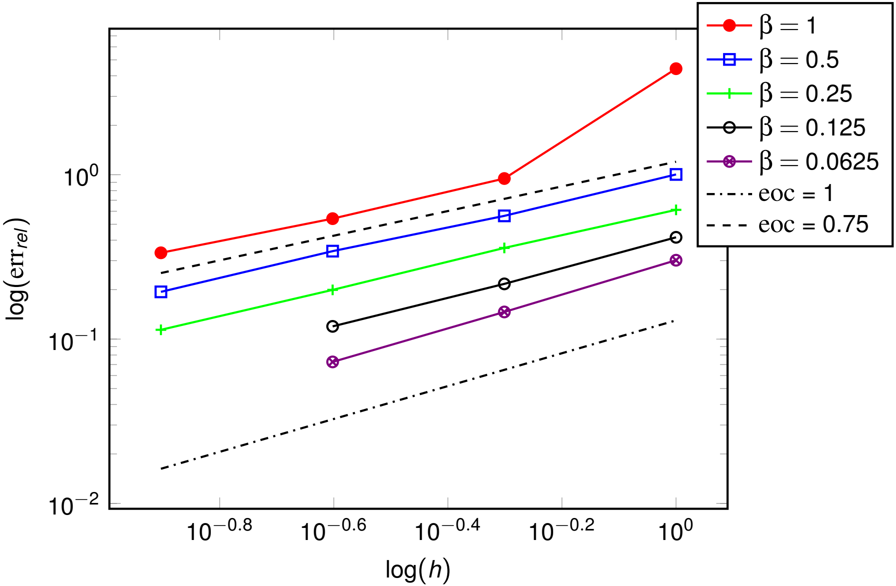
- Order of convergence

$$\mathrm{eoc} = \log_2 \left(\frac{\mathrm{err}_h}{\mathrm{err}_{h+1}} \right)$$

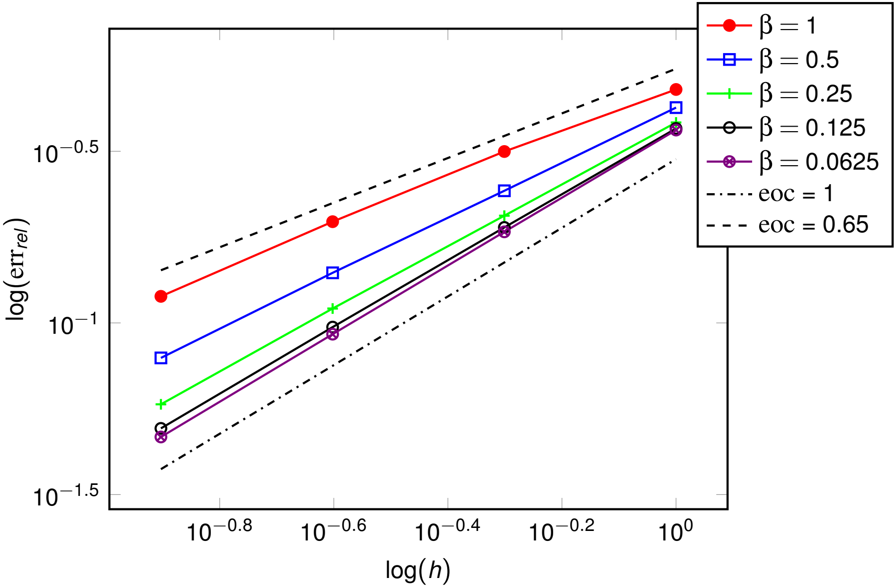
Dirichlet error: constant time step size



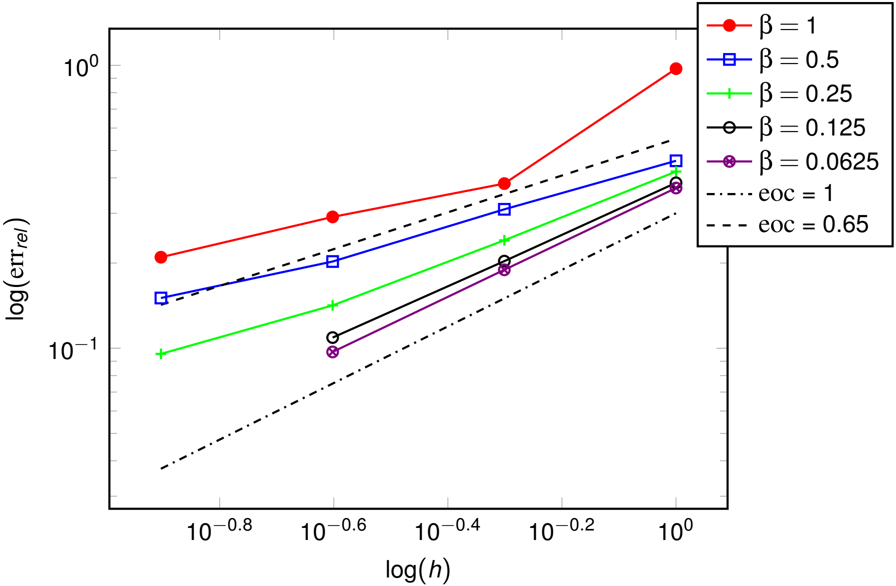
Dirichlet error: variable time step size



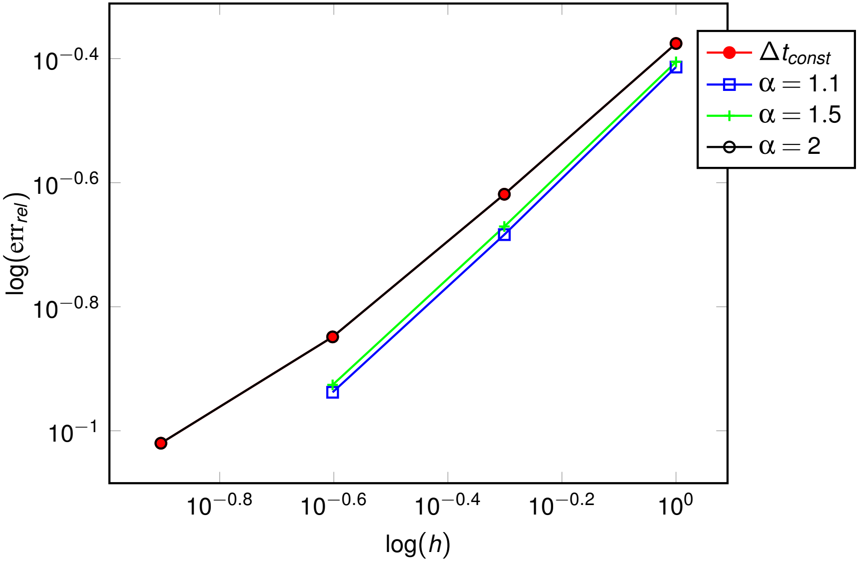
Neumann error: constant time step size



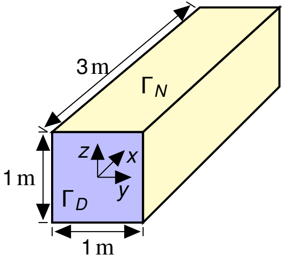
Neumann error: variable time step size



Neumann error: different time gradings



Wave propagation



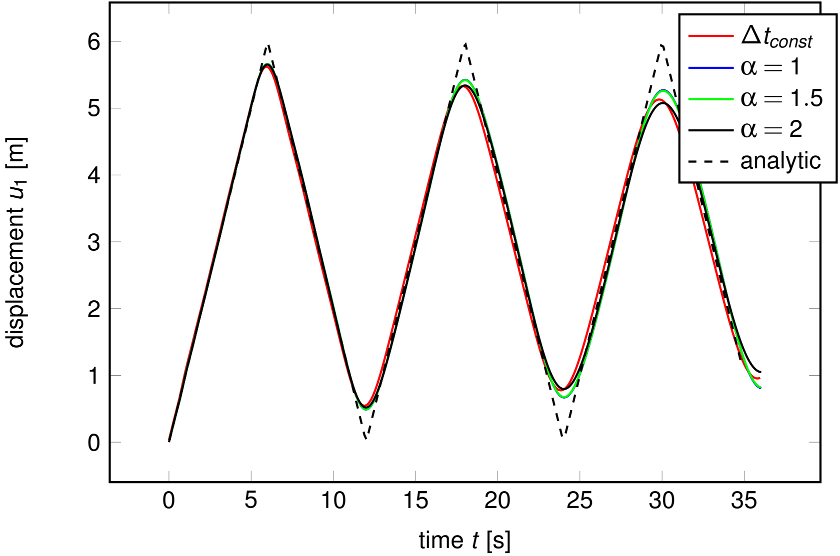
- 3-d bar with geometry from above
- Dirichlet BC set to zero: bar is fixed
- Neumann BC on the longitudinal surfaces set to zero
- Neumann BC on back side $\{\mathbf{x}_{load} \in \Gamma | x_1 = 3\,\mathrm{m}, -0.5\,\mathrm{m} \leq x_2, x_3 \leq 0.5\,\mathrm{m}\}$: load $t_1(\mathbf{x}_{load}, t) = 1\,\mathrm{N}/\mathrm{m}^2 f(t)$

$$f(t) = H(t) \qquad H(t) = 1 \;\; \forall\, t > 0$$

$$f(t) = te^{-t} \quad \forall\, t \geq 0$$

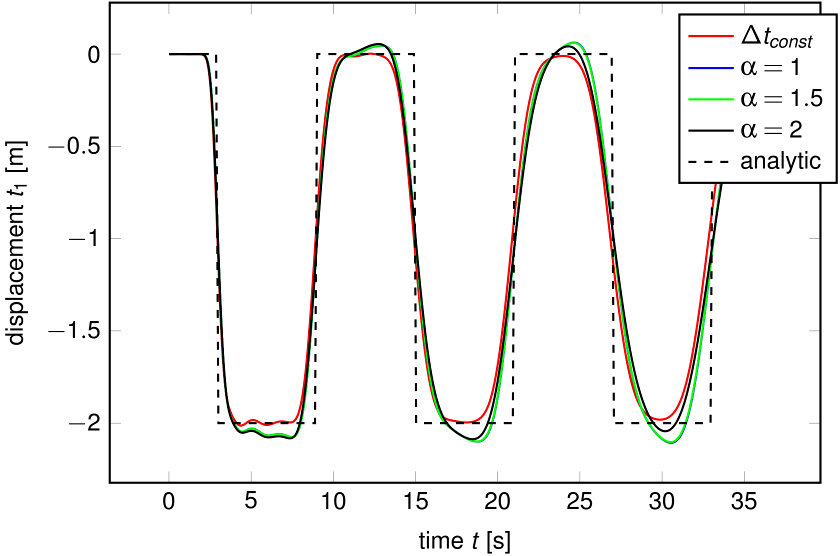
Displacement solution: Time discretisation

$\beta = 0.0625$, mesh 2



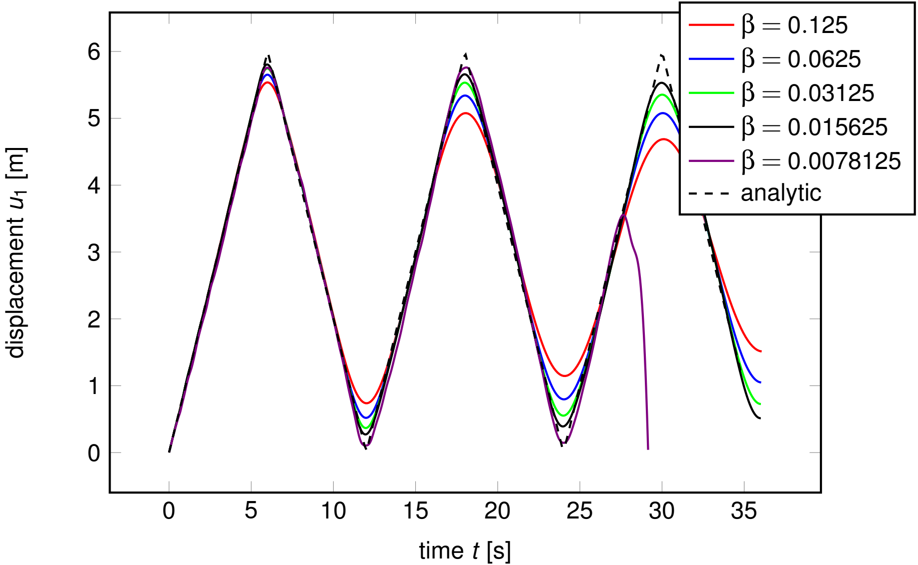
Traction solution: Time discretisation

$\beta = 0.0625$, mesh 2



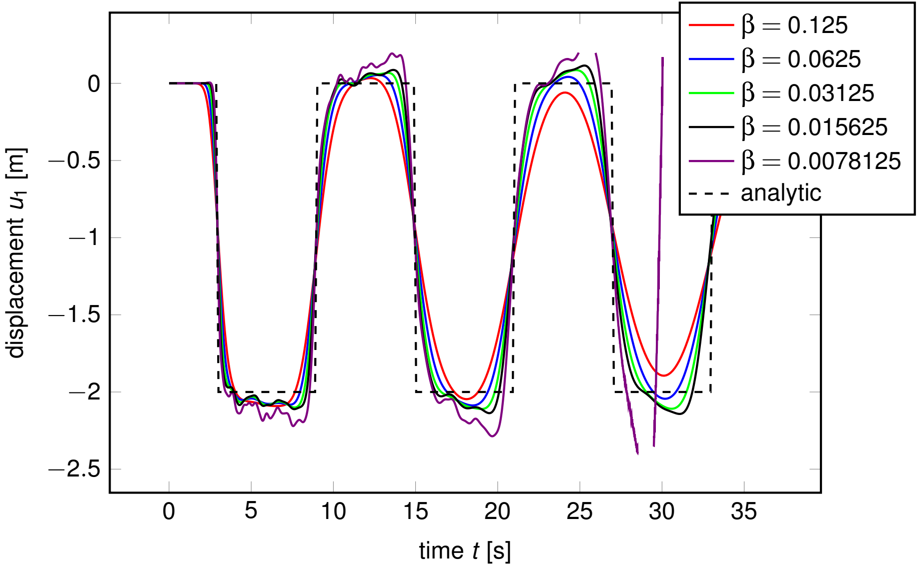
Displacement solution: Time discretisation

$\alpha = 2$, mesh 2



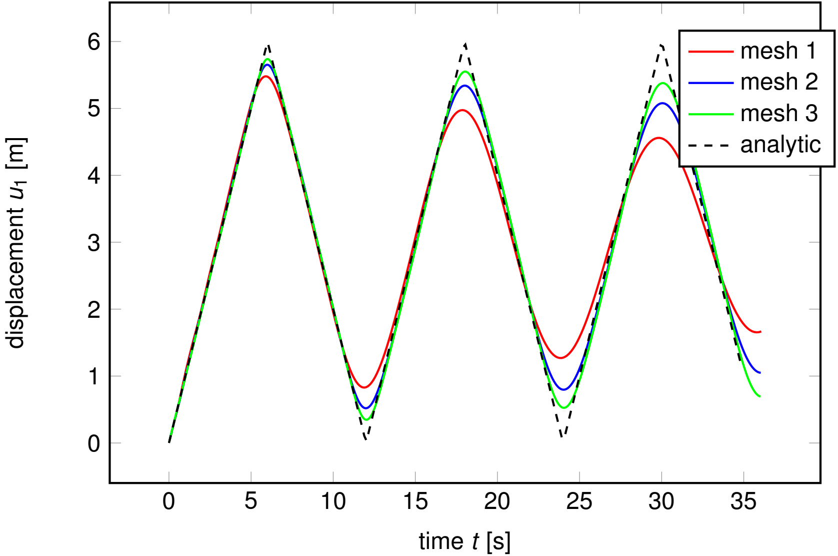
Traction solution: Time discretisation

$\alpha = 2$, mesh 2



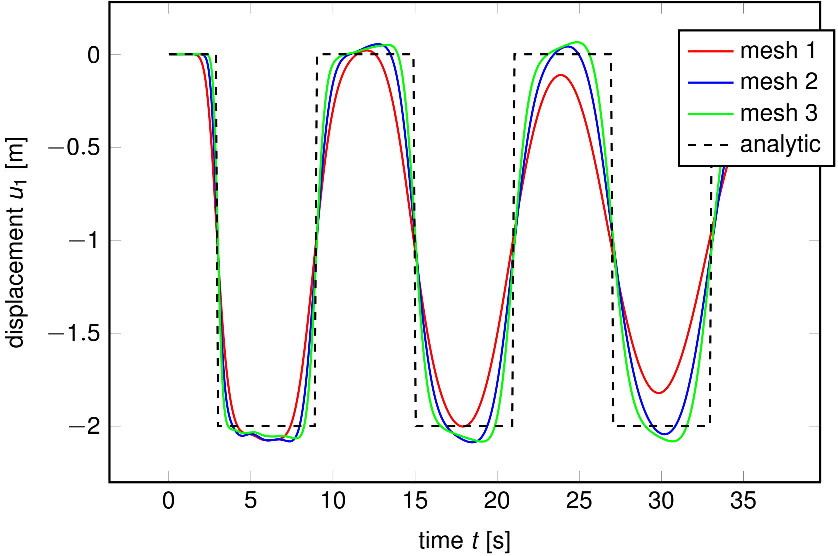
Displacement solution: Spatial discretisation

$\alpha = 2, \beta = 0.0625$



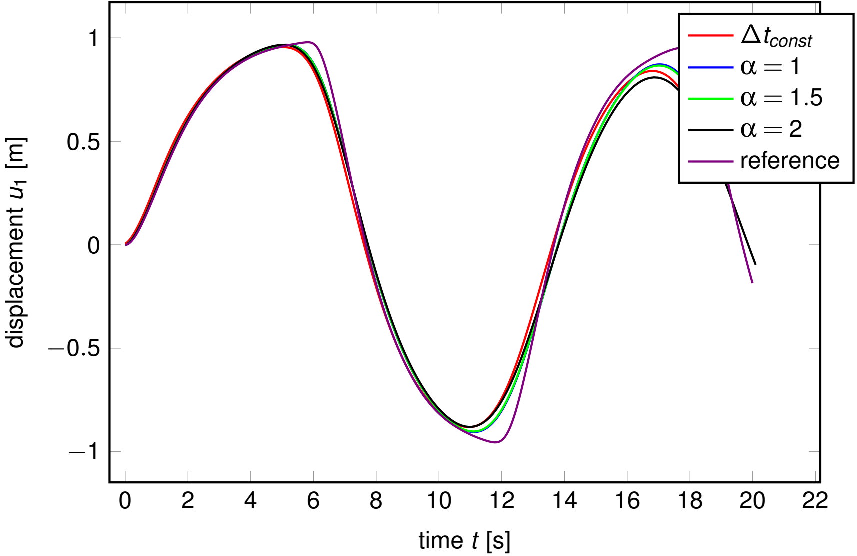
Traction solution: Spatial discretisation

$\alpha = 2, \beta = 0.0625$



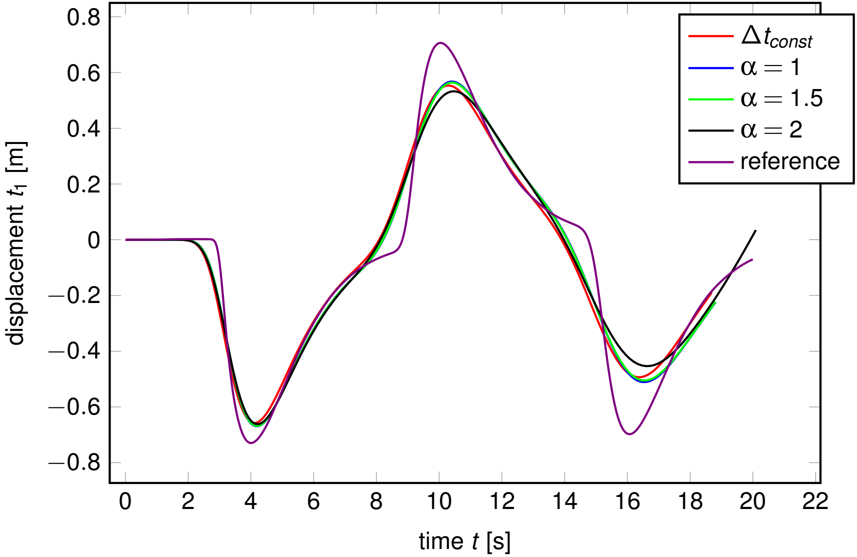
Displacement solution: Time discretisation

$\beta = 0.0625$, mesh 2



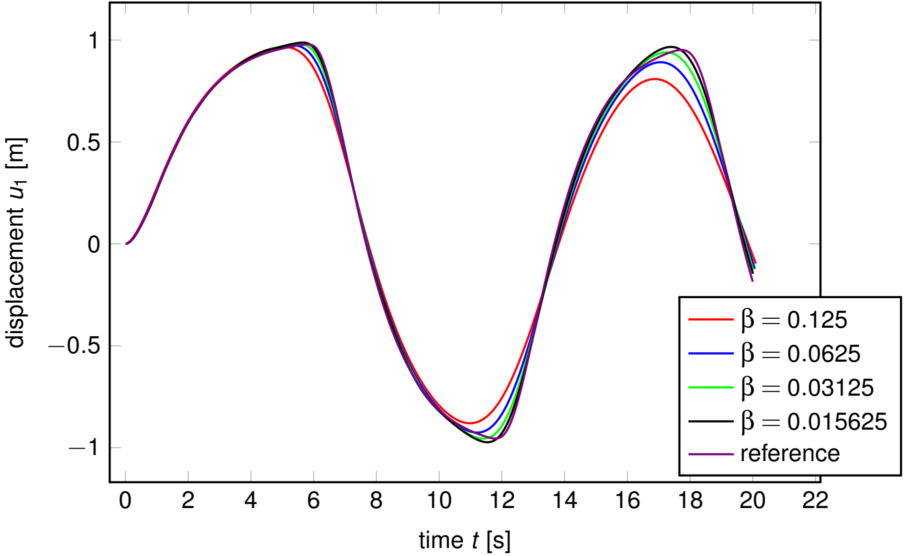
Traction solution: Time discretisation

$\beta = 0.0625$, mesh 2



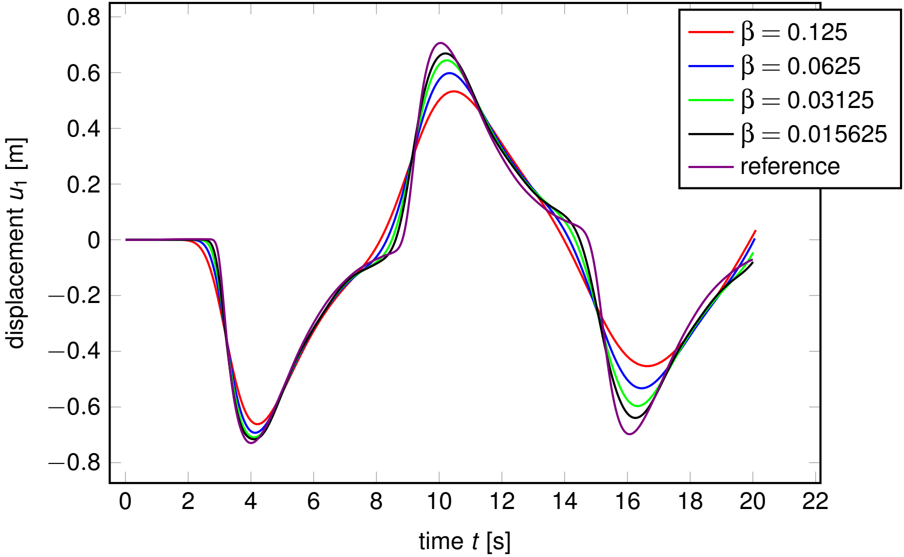
Displacement solution: Time discretisation

$\alpha = 2$, mesh 2



Traction solution: Time discretisation

$\alpha = 2$, mesh 2



Conclusions

- Generalized Convolution Quadrature Method realized for
  - direct collocation BEM
  - mixed problem in elastodynamics

  following "M. Lopez-Fernandez and S. Sauter. Generalized convolution quadrature with variable time stepping. part II: Algorithm and numerical results. *Preprint Universität Zürich*, 09-2012, 2012"
- BDF 1 with variable time stepping used — BDF 2 is in preparation (testing is missing)
- Method has similar behavior compared to original CQM
- Storage requirements are huge due to $N_Q = N \log N$
- Algorithm can easily parallelized (in principle, Cache size??)
- Fast methods from elliptic frequency dependent problems can directly be used for matrix-vector product – No solve in Laplace domain!

Conclusions

- Generalized Convolution Quadrature Method realized for
  - direct collocation BEM
  - mixed problem in elastodynamics

  following "M. Lopez-Fernandez and S. Sauter. Generalized convolution quadrature with variable time stepping. part II: Algorithm and numerical results. *Preprint Universität Zürich*, 09-2012, 2012"
- BDF 1 with variable time stepping used — BDF 2 is in preparation (testing is missing)
- Method has similar behavior compared to original CQM
- Storage requirements are huge due to $N_Q = N \log N$
- Algorithm can easily parallelized (in principle, Cache size??)
- Fast methods from elliptic frequency dependent problems can directly be used for matrix-vector product – No solve in Laplace domain!

Conclusions

- Generalized Convolution Quadrature Method realized for
  - direct collocation BEM
  - mixed problem in elastodynamics

  following “M. Lopez-Fernandez and S. Sauter. Generalized convolution quadrature with variable time stepping. part II: Algorithm and numerical results. *Preprint Universität Zürich*, 09-2012, 2012”
- BDF 1 with variable time stepping used — BDF 2 is in preparation (testing is missing)
- Method has similar behavior compared to original CQM
- Storage requirements are huge due to $N_Q = N \log N$
- Algorithm can easily parallelized (in principle, Cache size??)
- Fast methods from elliptic frequency dependent problems can directly be used for matrix-vector product – No solve in Laplace domain!

Conclusions

- Generalized Convolution Quadrature Method realized for
  - direct collocation BEM
  - mixed problem in elastodynamics

  following "M. Lopez-Fernandez and S. Sauter. Generalized convolution quadrature with variable time stepping. part II: Algorithm and numerical results. *Preprint Universität Zürich*, 09-2012, 2012"
- BDF 1 with variable time stepping used — BDF 2 is in preparation (testing is missing)
- Method has similar behavior compared to original CQM
- Storage requirements are huge due to $N_Q = N \log N$
- Algorithm can easily parallelized (in principle, Cache size??)
- Fast methods from elliptic frequency dependent problems can directly be used for matrix-vector product – No solve in Laplace domain!

Conclusions

- Generalized Convolution Quadrature Method realized for
  - direct collocation BEM
  - mixed problem in elastodynamics

  following “M. Lopez-Fernandez and S. Sauter. Generalized convolution quadrature with variable time stepping. part II: Algorithm and numerical results. *Preprint Universität Zürich*, 09-2012, 2012”
- BDF 1 with variable time stepping used — BDF 2 is in preparation (testing is missing)
- Method has similar behavior compared to original CQM
- Storage requirements are huge due to $N_Q = N \log N$
- Algorithm can easily parallelized (in principle, Cache size??)
- Fast methods from elliptic frequency dependent problems can directly be used for matrix-vector product – No solve in Laplace domain!

Graz University of Technology
Institute of Applied Mechanics

Generalized Convolution Quadrature for an Elastodynamic BEM

Martin Schanz

BIRS-workshop: Computational and Numerical Analysis of Transient Problems in Acoustics, Elasticity, and Electromagnetism
Banff, Canada
January 19, 2016