Influence of several parameters on the accuracy of a Convolution Quadrature method

T. Betcke[‡] N. Salles[†]

[†] ENSTA-ParisTech Université Paris Saclay
[‡] University College London

21th January 2016

We study a **Z-transform based Convolution Quadrature method** to solve the Wave equation:

$$\begin{cases} \dfrac{\partial^2 u(t;x)}{\partial t^2} - c^2 \Delta u(t;x) = 0, & (t,x) \in [0,T] \times \Omega_e \\ u(0;x) = \dfrac{\partial u(0;x)}{\partial t} = 0 & \\ u(t;x) = g(t;x), & (t,x) \in [0,T] \times \Gamma \end{cases}$$

1. We will show that this CQ method has **two inherent possible errors** in addition to the usual errors (time-discretisation scheme, spatial discretisation . . .)
2. One comes from the boundary conditions of the frequency problems that are not well approximated using a truncated series instead of the infinite sum (bad approximation of the Z-transform)
3. Another comes from the bad approximation of the inverse Z-transform (contour integral) when coming back in time.
4. By introducing N_f, and N_z two new parameters, it is possible to achieve better accuracy.

We study a **Z-transform based Convolution Quadrature method** to solve the Wave equation:

$$\begin{cases} \dfrac{\partial^2 u(t;x)}{\partial t^2} - c^2 \Delta u(t;x) = 0, \quad (t,x) \in [0,T] \times \Omega_e \\ u(0;x) = \dfrac{\partial u(0;x)}{\partial t} = 0 \\ u(t;x) = g(t;x), \quad (t,x) \in [0,T] \times \Gamma \end{cases}$$

1. We will show that this CQ method has **two inherent possible errors** in addition to the usual errors (time-discretisation scheme, spatial discretisation . . .)
2. One comes from the boundary conditions of the frequency problems that are not well approximated using a truncated series instead of the infinite sum (bad approximation of the Z-transform)
3. Another comes from the bad approximation of the inverse Z-transform (contour integral) when coming back in time.
4. By introducing N_f, and N_z two new parameters, it is possible to achieve better accuracy.

We study a **Z-transform based Convolution Quadrature method** to solve the Wave equation:

$$\begin{cases} \dfrac{\partial^2 u(t;x)}{\partial t^2} - c^2 \Delta u(t;x) = 0, \quad (t,x) \in [0,T] \times \Omega_e \\ u(0;x) = \dfrac{\partial u(0;x)}{\partial t} = 0 \\ u(t;x) = g(t;x), \quad (t,x) \in [0,T] \times \Gamma \end{cases}$$

1. We will show that this CQ method has **two inherent possible errors** in addition to the usual errors (time-discretisation scheme, spatial discretisation . . .)
2. One comes from the boundary conditions of the frequency problems that are not well approximated using a truncated series instead of the infinite sum (bad approximation of the Z-transform)
3. Another comes from the bad approximation of the inverse Z-transform (contour integral) when coming back in time.
4. By introducing N_f, and N_z two new parameters, it is possible to achieve better accuracy.

We study a **Z-transform based Convolution Quadrature method** to solve the Wave equation:

$$\begin{cases} \dfrac{\partial^2 u(t;x)}{\partial t^2} - c^2 \Delta u(t;x) = 0, \quad (t,x) \in [0,T] \times \Omega_e \\ u(0;x) = \dfrac{\partial u(0;x)}{\partial t} = 0 \\ u(t;x) = g(t;x), \quad (t,x) \in [0,T] \times \Gamma \end{cases}$$

1. We will show that this CQ method has **two inherent possible errors** in addition to the usual errors (time-discretisation scheme, spatial discretisation . . .)
2. One comes from the boundary conditions of the frequency problems that are not well approximated using a truncated series instead of the infinite sum (bad approximation of the Z-transform)
3. Another comes from the bad approximation of the inverse Z-transform (contour integral) when coming back in time.
4. By introducing N_f, and N_z two new parameters, it is possible to achieve better accuracy.

Outline

1. The Z-Transform based CQ
2. Errors from the Z-transform based CQ
3. Approximation error of the inverse Z-transform
4. Approximation error of the Z-transform

Outline

1. The Z-Transform based CQ
2. Errors from the Z-transform based CQ
3. Approximation error of the inverse Z-transform
4. Approximation error of the Z-transform

- You want to solve a time-domain problem for yesterday?
- You have an existing frequency code?
- You have a lot of computer nodes so you would like to get a "fast" method easily?

$\Longrightarrow$ This Z-transform based CQ method can be really interesting!

- You want to solve a time-domain problem for yesterday?
- You have an existing frequency code?
- You have a lot of computer nodes so you would like to get a "fast" method easily?

$\Longrightarrow$ This Z-transform based CQ method can be really interesting!

In 1 or 2 days you can solve your first time-domain problem with your frequency code.

1) Rewrite problem as a first order system

$$\begin{cases} \frac{1}{c}\frac{\partial}{\partial t} v(t;x) & = Mv(t;x),\ (t;x) \in [0,T] \times \Omega_e \\ v(0;x) & = 0,\ \forall x \in \Omega_e \\ Bv(t;x) & = F(t;x),\ (t;x) \in [0,T] \times \Gamma \end{cases}$$

with $v = \left(u, \frac{1}{c}\partial_t u\right)^T$, $M = \begin{pmatrix} 0 & I \\ \Delta_x & 0 \end{pmatrix}$, $B = \begin{pmatrix} I & 0 \\ 0 & 0 \end{pmatrix}$ and $F(x,t) = (g(x,t), 0)^T$.

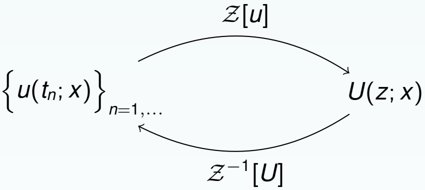
2) Apply a multistep scheme

$$\frac{1}{c\Delta t} \sum_{j \leq n} \gamma_{n-j} v_d(t_j;x) = Mv_d(t_n;x), \text{ for } n = 1, 2, \ldots$$

with $\gamma_0 = 1, \gamma_1 = -1$(Backward Euler)

or $\gamma_0 = 3/2, \gamma_1 = -2, \gamma_1 = 1/2$(BDF-2), and $t_j = j\Delta t$

The Z-transform maps a sequence given at time steps $u(t_n; x), n = 1, \dots$ and $t_n = n\Delta t$, to a function in the frequency domain $U(z; x)$



Z-transform and its inverse

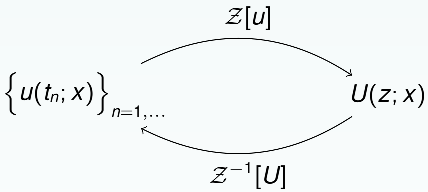
Definitions

$$\mathcal{Z}\left[g\right](z; x) = \sum_{n=0}^{\infty} g(t_n; x) z^n$$

$$\mathcal{Z}^{-1}\left[G\right](t_n; x) = \frac{1}{2\pi i} \int_{|z|=\lambda} \frac{G(z; x)}{z^{n+1}} dz$$

The Z-transform

The Z-transform maps a sequence given at time steps $u(t_n; x)$, $n = 1, \ldots$ and $t_n = n\Delta t$, to a function in the frequency domain $U(z; x)$



Z-transform and its inverse

Definitions

$$\mathcal{Z}\left[g\right](z; x) = \sum_{n=0}^{\infty} g(t_n; x) z^n$$

$$\mathcal{Z}^{-1}\left[G\right](t_n; x) = \frac{1}{2\pi i} \int_{|z|=\lambda} \frac{G(z; x)}{z^{n+1}} dz$$

In practice

$$\widetilde{\mathcal{Z}}_{N_z}\left[g\right](z; x) = \sum_{n=0}^{N_z} g(t_n; x) z^n$$

$$\widetilde{\mathcal{Z}}_{N_f}^{-1}\left[G\right](t_n; x) = \frac{\lambda^{-n}}{N_f} \sum_{\ell=1}^{N_f} G(\lambda z_\ell; x) z_\ell^{-n}$$

3) Apply the Z-transform

$$\sum_{n=0}^{\infty} \frac{1}{c\Delta t} \sum_{j\leq n} \gamma_{n-j} v_d(t_j; x) z^n = M \sum_{n=0}^{\infty} v_d(t_n; x) z^n$$

3) Apply the Z-transform

$$\sum_{n=0}^{\infty} \frac{1}{c\Delta t} \sum_{j \leq n} \gamma_{n-j} v_d(t_j; x) z^n = M \sum_{n=0}^{\infty} v_d(t_n; x) z^n$$

$$\Rightarrow \gamma(\mathbf{z}) \mathbf{V_d} (\mathbf{z}; x) = M \mathbf{V_d}(\mathbf{z}; x)$$

with $\gamma(\mathbf{z}) = \sum_{n \geq 0} \gamma_n \mathbf{z}^n$ and $\mathbf{V_d}(\mathbf{z}; x) = \sum_{n \geq 0} v_d(t_n; x) \mathbf{z}^n$.

3) Apply the Z-transform

$$\sum_{n=0}^{\infty} \frac{1}{c\Delta t} \sum_{j\leq n} \gamma_{n-j} v_d(t_j;x) z^n = M \sum_{n=0}^{\infty} v_d(t_n;x) z^n$$

$$\Rightarrow \gamma(\boldsymbol{z}) \boldsymbol{V_d}\,(\boldsymbol{z};x) = M \boldsymbol{V_d}(\boldsymbol{z};x)$$

with $\gamma(\boldsymbol{z}) = \sum_{n\geq 0} \gamma_n \boldsymbol{z}^n$ and $\boldsymbol{V_d}(\boldsymbol{z};x) = \sum_{n\geq 0} v_d(t_n;x)\boldsymbol{z}^n$.

4) We get the Laplace-domain problem (modified Helmholtz equation)

$$\begin{cases} \left(\frac{\gamma(\boldsymbol{z})}{c\Delta t}\right)^2 U_d(\boldsymbol{z};x) - \Delta U_d(\boldsymbol{z};x) = 0, \quad x \in \Omega_e, \\ U_d(\boldsymbol{z};x) = G(\boldsymbol{z};x), \quad x \in \Gamma, \\ + \text{Outgoing Boundary Condition} \end{cases}$$

Multistep and multistage convolution quadrature for the wave equation: Algorithms and experiments, Banjai L., 2010

5) The discrete time-domain solution is given by the inverse Z-transform

$$u(t_n; x) = \frac{1}{2\pi i}\int_{\mathcal{C}} \frac{U(z;x)}{z^{n+1}}\,dz = \frac{1}{2\pi i}\int_{|z|=\boldsymbol{\lambda}} \frac{U(x,z)}{z^{n+1}}\,dz$$

6) We define the approximation using trapezoidal rule

$$\widetilde{u}_{N_t}(t_n; x) = \frac{\boxed{\lambda^{-n}}}{N_t}\sum_{\ell=1}^{N_t} U(\lambda z_\ell; x) z_\ell^{-n},$$

- Each quadrature point requires solution of the modified Helmholtz problem for a given complex frequency.
- Previous methods use $N_f = N_t$ (number of time steps).
- Actually, $U(x,\overline{z}) = \overline{U}(x,z) \Rightarrow$ we divide by 2 the number of problems to solve.

5) The discrete time-domain solution is given by the inverse Z-transform

$$u(t_n; x) = \frac{1}{2\pi i} \int_{\mathcal{C}} \frac{U(z; x)}{z^{n+1}} dz = \frac{1}{2\pi i} \int_{|z|=\boldsymbol{\lambda}} \frac{U(x, z)}{z^{n+1}} dz$$

6) We define the approximation using trapezoidal rule

$$\widetilde{u}_{N_f}(t_n; x) = \frac{\boxed{\lambda^{-n}}}{N_f} \sum_{\ell=1}^{N_f} U(\lambda z_\ell; x) z_\ell^{-n},$$

- Each quadrature point requires solution of the modified Helmholtz problem for a given complex frequency.
- Previous methods use $N_f = N_t$ (number of time steps).
- Actually, $U(x, \overline{z}) = \overline{U}(x, z) \Rightarrow$ we divide by 2 the number of problems to solve.

5) The discrete time-domain solution is given by the inverse Z-transform

$$u(t_n; x) = \frac{1}{2\pi i} \int_{\mathcal{C}} \frac{U(z; x)}{z^{n+1}} dz = \frac{1}{2\pi i} \int_{|z|=\boldsymbol{\lambda}} \frac{U(x, z)}{z^{n+1}} dz$$

6) We define the approximation using trapezoidal rule

$$\widetilde{u}_{N_f}(t_n; x) = \frac{\boxed{\lambda^{-n}}}{N_f} \sum_{\ell=1}^{N_f} U(\lambda z_\ell; x) z_\ell^{-n},$$

- Each quadrature point requires solution of the modified Helmholtz problem for a given complex frequency.
- Previous methods use $N_f = N_t$ (number of time steps).
- Actually, $U(x, \overline{z}) = \overline{U}(x, z) \Rightarrow$ we divide by 2 the number of problems to solve.

The exponentially convergent trapezoidal rule, Trefethen and Weideman, SIAM Review, 2013.

The steps to use "naive" Z-transform Convolution Quadrature:

1. Evaluation of the boundary condition for each time step.
2. Definition of the wave-numbers depending on the parameters (λ, N_f and the multistep rule).
3. Computation of the Z-transforms of the rhs.
4. Solving frequency problems (can be done in parallel easily).
5. Inverse Z-transform to come back in time.

Z-transform based CQ summarized for analysis

$$u(t_n;x) = \mathcal{Z}^{-1}\left[U\right](t_n;x) = \mathcal{Z}^{-1}\left[\mathcal{B}_{k_z}\{\underbrace{\mathcal{Z}\left[\boldsymbol{g}\right]}_{G}\}\right](t_n;x)$$

Z-transform and its inverse

Definitions

$$\mathcal{Z}\left[g\right](z;x) = \sum_{n=0}^{\infty} g(t_n;x) z^n$$

$$\mathcal{Z}^{-1}\left[G\right](t_n;x) = \frac{1}{2\pi i}\int_{|z|=\lambda} \frac{G(z;x)}{z^{n+1}} dz$$

Z-transform based CQ summarized for analysis

$$u(t_n;x) = \mathcal{Z}^{-1}\left[U\right](t_n;x) = \mathcal{Z}^{-1}\left[\ \mathcal{B}_{k_z}\{\underbrace{\mathcal{Z}\left[\boldsymbol{g}\right]}_{G}\}\ \right](t_n;x)$$

$$\widetilde{u}_{\boldsymbol{N_f},\boldsymbol{N_z}}(t_n;x) = \widetilde{\mathcal{Z}}^{-1}_{\boldsymbol{N_f}}\left[\widetilde{U}_{N_z}\right] = \widetilde{\mathcal{Z}}^{-1}_{\boldsymbol{N_f}}\left[\mathcal{B}_{k_z}\underbrace{\left\{\widetilde{\mathcal{Z}}_{\boldsymbol{N_z}}\left[\boldsymbol{g}\right]\right\}}_{\widetilde{G}_{N_z}}\right](t_n;x)$$

Z-transform and its inverse

Definitions

$$\mathcal{Z}\left[g\right](z;x) = \sum_{n=0}^{\infty} g(t_n;x)z^n$$

$$\mathcal{Z}^{-1}\left[G\right](t_n;x) = \frac{1}{2\pi i}\int_{|z|=\boldsymbol{\lambda}} \frac{G(z;x)}{z^{n+1}}dz$$

In practice

$$\widetilde{\mathcal{Z}}_{\boldsymbol{N_z}}\left[g\right](z;x) = \sum_{n=0}^{\boldsymbol{N_z}} g(t_n;x)z^n$$

$$\widetilde{\mathcal{Z}}^{-1}_{\boldsymbol{N_f}}\left[G\right](t_n;x) = \frac{\boldsymbol{\lambda}^{-n}}{\boldsymbol{N_f}}\sum_{\ell=1}^{\boldsymbol{N_f}} G(\lambda z_\ell;x)z_\ell^{-n}$$

Z-transform based CQ summarized for analysis

$$u(t_n;x) = \mathcal{Z}^{-1}\Big[U\Big](t_n;x) = \mathcal{Z}^{-1}\Big[\ \mathcal{B}_{k_z}\underbrace{\{\mathcal{Z}\,[\mathbf{g}]\}}_{G}\ \Big](t_n;x)$$

$$\widetilde{u}_{N_f,N_z}(t_n;x) = \widetilde{\mathcal{Z}}^{-1}_{N_f}\Big[\widetilde{U}_{N_z}\Big] = \widetilde{\mathcal{Z}}^{-1}_{N_f}\Big[\mathcal{B}_{k_z}\underbrace{\Big\{\widetilde{\mathcal{Z}}_{N_z}\,[\mathbf{g}]\Big\}}_{\widetilde{G}_{N_z}}\Big](t_n;x)$$

with $k_z = i\dfrac{\gamma(z)}{c\Delta t}$

Z-transform and its inverse

Definitions

$$\mathcal{Z}\,[g]\,(z;x) = \sum_{n=0}^{\infty} g(t_n;x)z^n$$

$$\mathcal{Z}^{-1}\,[G]\,(t_n;x) = \frac{1}{2\pi i}\int_{|z|=\lambda}\frac{G(z;x)}{z^{n+1}}dz$$

In practice

$$\widetilde{\mathcal{Z}}_{N_z}\,[g]\,(z;x) = \sum_{n=0}^{N_z} g(t_n;x)z^n$$

$$\widetilde{\mathcal{Z}}^{-1}_{N_f}\,[G]\,(t_n;x) = \frac{\lambda^{-n}}{N_f}\sum_{\ell=1}^{N_f} G(\lambda z_\ell;x)z_\ell^{-n}$$

1. The Z-Transform based CQ
2. Errors from the Z-transform based CQ
3. Approximation error of the inverse Z-transform
4. Approximation error of the Z-transform

Let $u_w(t;x)$ be the exact solution of the wave problem, then

$$u_w(t_n;x) - \widetilde{u}_{\mathbf{N_f},\mathbf{N_z}}(t_n;x) =$$

Let $u_w(t;x)$ be the exact solution of the wave problem, then

$$u_w(t_n;x) - \widetilde{u}_{\mathbf{N_f},\mathbf{N_z}}(t_n;x) = \boxed{\left(u_w(t_n;x) - \mathcal{Z}^{-1}\left[U\right](t_n;x)\right)}$$

Let $u_w(t;x)$ be the exact solution of the wave problem, then

$$u_w(t_n;x) - \widetilde{u}_{\boldsymbol{N_f},\boldsymbol{N_z}}(t_n;x) = \boxed{\left(u_w(t_n;x) - \mathcal{Z}^{-1}\left[U\right](t_n;x)\right)} + \boxed{\left(\mathcal{Z}^{-1}\left[U\right](t_n;x) - \widetilde{\mathcal{Z}}_{\boldsymbol{N_f}}^{-1}\left[U\right](t_n;x)\right)}$$

Let $u_w(t;x)$ be the exact solution of the wave problem, then

$$u_w(t_n;x) - \widetilde{u}_{N_f,N_z}(t_n;x) = \boxed{\left(u_w(t_n;x) - \mathcal{Z}^{-1}[U](t_n;x)\right)}$$
$$+ \boxed{\left(\mathcal{Z}^{-1}[U](t_n;x) - \widetilde{\mathcal{Z}}_{N_f}^{-1}[U](t_n;x)\right)}$$
$$+ \boxed{\left(\widetilde{\mathcal{Z}}_{N_f}^{-1}[U](t_n;x) - \widetilde{\mathcal{Z}}_{N_f}^{-1}\left[\widetilde{U}_{N_z}\right](t_n;x)\right)}$$

Let $u_w(t;x)$ be the exact solution of the wave problem, then

$$u_w(t_n;x) - \widetilde{u}_{\boldsymbol{N_f},\boldsymbol{N_z}}(t_n;x) = \boxed{\left(u_w(t_n;x) - \mathcal{Z}^{-1}\left[U\right](t_n;x)\right)}$$

Section 3 → $$+ \boxed{\left(\mathcal{Z}^{-1}\left[U\right](t_n;x) - \widetilde{\mathcal{Z}}^{-1}_{\boldsymbol{N_f}}\left[U\right](t_n;x)\right)}$$

Section 4 → $$+ \boxed{\left(\widetilde{\mathcal{Z}}^{-1}_{\boldsymbol{N_f}}\left[U\right](t_n;x) - \widetilde{\mathcal{Z}}^{-1}_{\boldsymbol{N_f}}\left[\widetilde{U}_{\boldsymbol{N_z}}\right](t_n;x)\right)}$$

- **Error of the the scheme used for the time discretisation**
- **Error to approximate the contour integral : Approximation error of the inverse Z-transform using a trapezoidal rule**
- **Error on the frequency solution $\widetilde{U}_{N_z}$ coming from the fact we truncate the Z-transform**

Outline

1 The Z-Transform based CQ

2 Errors from the Z-transform based CQ

3 Approximation error of the inverse Z-transform

4 Approximation error of the Z-transform

In this section, we study:

$$E_1(\boldsymbol{\lambda}, \boldsymbol{N_f}) = \mathcal{Z}^{-1}[U](t_n; x) - \widetilde{\mathcal{Z}}_{\boldsymbol{N_f}}^{-1}[U](t_n; x)$$

The question is how well can we approximate the contour integral

$$\mathcal{Z}^{-1}[U](t_n; x) = \frac{1}{2\pi i}\int_{|z|=\boldsymbol{\lambda}} \frac{U(z; x)}{z^{n+1}} dz$$

by (a trapezoidal rule)

$$\widetilde{\mathcal{Z}}_{\boldsymbol{N_f}}^{-1}[U](t_n; x) = \frac{\boldsymbol{\lambda}^{-n}}{\boldsymbol{N_f}} \sum_{\ell=1}^{\boldsymbol{N_f}} U(\lambda z_\ell; x) z_\ell^{-n}$$

In this section, we study:

$$E_1(\boldsymbol{\lambda}, \boldsymbol{N_f}) = \mathcal{Z}^{-1}\left[U\right](t_n; x) - \widetilde{\mathcal{Z}}^{-1}_{\boldsymbol{N_f}}\left[U\right](t_n; x)$$

The question is how well can we approximate the contour integral

$$\mathcal{Z}^{-1}\left[U\right](t_n; x) = \frac{1}{2\pi i}\int_{|z|=\boldsymbol{\lambda}} \frac{U(z; x)}{z^{n+1}} dz$$

by (a trapezoidal rule)

$$\widetilde{\mathcal{Z}}^{-1}_{\boldsymbol{N_f}}\left[U\right](t_n; x) = \frac{\boldsymbol{\lambda}^{-n}}{\boldsymbol{N_f}}\sum_{\ell=1}^{\boldsymbol{N_f}} U(\lambda z_\ell; x) z_\ell^{-n}$$

The answer has been studied in a paper submitted soon. The main result is:

$$|E_1(\boldsymbol{\lambda}, \boldsymbol{N_f})| \approx O\left(\left(\frac{\boldsymbol{\lambda}}{\lambda_U}\right)^{\boldsymbol{N_f}}\right)$$

Betcke T., Salles N., Śmigaj W., *Exponentially Accurate Evaluation of Time-Stepping schemes for the Wave Equation via CQ type methods, Report*

In this section, we study:

$$E_1(\boldsymbol{\lambda}, \boldsymbol{N_f}) = \mathcal{Z}^{-1}[U](t_n; x) - \widetilde{\mathcal{Z}}_{\boldsymbol{N_f}}^{-1}[U](t_n; x)$$

The question is how well can we approximate the contour integral

$$\mathcal{Z}^{-1}[U](t_n; x) = \frac{1}{2\pi i} \int_{|z|=\boldsymbol{\lambda}} \frac{U(z; x)}{z^{n+1}} dz$$

by (a trapezoidal rule)

$$\widetilde{\mathcal{Z}}_{\boldsymbol{N_f}}^{-1}[U](t_n; x) = \frac{\boldsymbol{\lambda}^{-n}}{\boldsymbol{N_f}} \sum_{\ell=1} \boldsymbol{N_f} U(\lambda z_\ell; x) z_\ell^{-n}$$

The answer has been studied in a paper submitted soon. The main result is:

$$|E_1(\boldsymbol{\lambda}, \boldsymbol{N_f})| \approx O\left(\left(\frac{\boldsymbol{\lambda}}{\lambda_U}\right)^{\boldsymbol{N_f}}\right)$$

Betcke T., Salles N., Śmigaj W., *Exponentially Accurate Evaluation of Time-Stepping schemes for the Wave Equation via CQ type methods, Report*

Betcke T., Salles N., Śmigaj W., *Spectral estimates of the inverse Z-Transform arising in Convolution Quadrature methods., Sub. soon*

With an Indirect First Kind Formulation

The solution in Ω_e exists and is obtained by:

$$U(z;x) = \mathcal{B}_{k_z} G(z;x) = \mathcal{S}_{k_z} \circ \mathrm{S}_{k_z}^{-1} G(z;x)$$

The related poles

The representation of U is valid for $k_z \neq ip_j$ and $k_z \neq iq_j$ where

- p_j are the **scattering poles** of the Helmholtz solution operator $\mathcal{B}$
- q_j are the **eigenfrequencies** of the interior Laplacian Dirichlet problem.

Eigenfrequencies of the interior Dirichlet problem

Let q_j be eigenfrequencies of the interior Dirichlet problem:

$$-\Delta v(x) = {q_j}^2 v(x), \quad x \in \Omega_i$$
$$v(x) = 0, x \in \Gamma$$

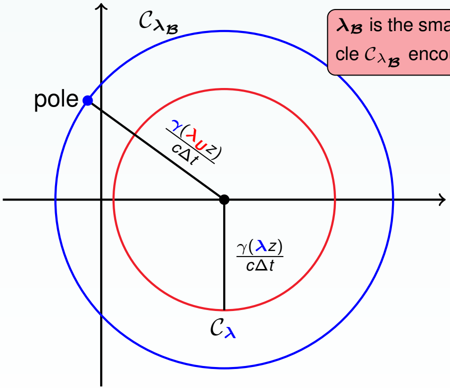
Radius of analyticity

Definition of the radius of analyticity of U

$U(z;x) = \mathcal{B}(k_z)G(z;x) = \mathcal{B}\left(i\frac{\gamma(z)}{c\Delta t}\right)G(z;x)$, then the analyticity of U depends on the ones of $\mathcal{B}(k_z)$ **and** G then: $\boldsymbol{\lambda_U} = \min\{\lambda_G, \lambda_{\mathcal{B}}\}$

Definition of the radius of analyticity of U

$U(z;x) = \mathcal{B}(k_z)G(z;x) = \mathcal{B}\left(i\frac{\gamma(z)}{c\Delta t}\right)G(z;x)$, then the analyticity of U depends on the ones of $\mathcal{B}(k_z)$ **and** G then: $\boldsymbol{\lambda_U} = \min\{\lambda_G, \lambda_{\mathcal{B}}\}$



$\boldsymbol{\lambda_{\mathcal{B}}}$ is the smallest radius such that the circle $\mathcal{C}_{\lambda_{\mathcal{B}}}$ encounters a pole.

Error estimates

Theorem (Error representation)

Let u and $u_{(N_f)}$ the solution using the multistep scheme and using a trapezoidal rule. Let $\boldsymbol{\lambda} < \boldsymbol{\lambda_U}$, where $\boldsymbol{\lambda_U}$ is the radius of analyticity of U. For multistep rules We have the exact error representation

$$\widetilde{u}_{\mathbf{N_f}}(t_n;x) - u(t_n;x) = \widetilde{\mathcal{Z}}_{\mathbf{N_f}}^{-1}\left[U\right](t_n;x) - \mathcal{Z}^{-1}\left[U\right](t_n;x)$$
$$= \sum_{\kappa=1}^{\infty} \boldsymbol{\lambda}^{\kappa N_f} u(t_{n+\kappa \mathbf{N_f}};x).$$

Theorem (Asymptotic error estimate)

Let $0 < \boldsymbol{\lambda} < \boldsymbol{\lambda_U}$. *Then*

$$\left|\mathcal{Z}^{-1}\left[U\right](t_n;x) - \widetilde{\mathcal{Z}}_{\mathbf{N_f}}^{-1}\left[U\right](t_n;x)\right| \mathcal{O}\left(\left(\frac{\boldsymbol{\lambda_U}}{\boldsymbol{\lambda}} - \epsilon\right)^{-N_f}\right)$$

for any $\epsilon > 0$ *as* $N_f \to \infty$.

Theorem (Error representation)

Let u and $u_{(N_f)}$ the solution using the multistep scheme and using a trapezoidal rule. Let $\boldsymbol{\lambda} < \boldsymbol{\lambda_U}$, where $\boldsymbol{\lambda_U}$ is the radius of analyticity of U. For multistep rules We have the exact error representation

$$\widetilde{u}_{\mathbf{N_f}}(t_n;x) - u(t_n;x) = \widetilde{\mathcal{Z}}_{\mathbf{N_f}}^{-1}[U](t_n;x) - \mathcal{Z}^{-1}[U](t_n;x)$$
$$= \sum_{\kappa=1}^{\infty} \boldsymbol{\lambda}^{\kappa N_f} u(t_{n+\kappa \mathbf{N_f}};x).$$

Theorem (Asymptotic error estimate)

Let $0 < \boldsymbol{\lambda} < \boldsymbol{\lambda_U}$. Then

$$\left| \mathcal{Z}^{-1}[U](t_n;x) - \widetilde{\mathcal{Z}}_{\mathbf{N_f}}^{-1}[U](t_n;x) \right| \mathcal{O}\left(\left(\frac{\boldsymbol{\lambda_U}}{\boldsymbol{\lambda}} - \epsilon \right)^{-N_f} \right) \approx \mathcal{O}\left(\left(\frac{\boldsymbol{\lambda}}{\lambda_U} \right)^{\mathbf{N_f}} \right)$$

for any $\epsilon > 0$ as $N_f \to \infty$.

With a *m*-stages Runge-Kutta scheme

The scalar equation becomes a **vector equation**:

$$\left(\frac{\Delta(z)}{c\Delta t}\right)^2 \mathcal{R}(z;x) = \Delta_x \mathcal{R}(z;x), \text{ and } U(z;x) = z^{-1} R_m(z;x),$$

where, $\mathcal{R}(z;x) = (R_1(z;x), R_2(z;x), \cdots, R_m(z;x))$ and (A, b) are part of the Butcher tableau

$$\Delta(z) = \left(A + \frac{z}{1-z}\mathbb{1}b^t\right)^{-1}.$$

In practice, we diagonalize $\Delta(z) = \mathbb{P}(z)\mathbb{D}(z)\mathbb{P}^{-1}(z)$, and $\mathbb{D}(z) = \text{diag}\,(\gamma_1(z), \ldots, \gamma_m(z))$ in order to get ***m* independent scalar problems**. The diagonalisation process is not possible for all frequencies so:

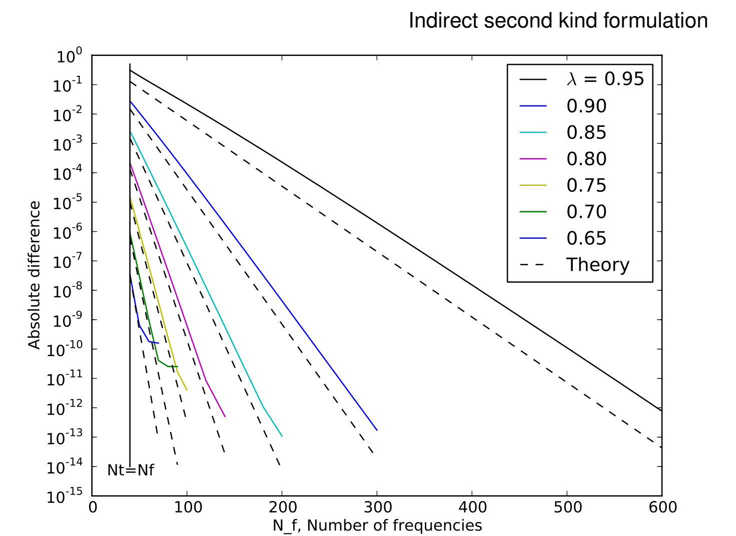
For the analysis of λ_U, we study the vector system directly

The solution $\mathcal{R}(z;x)$ of the vector problem is analytic in z if there is no eigenvalues of $\Delta(z)$, denoted $\lambda_j(z)$, that hits a scattering pole.

Proof: Jordan decomposition ensure unicity and $\mathcal{R}$ is complex differentiable.

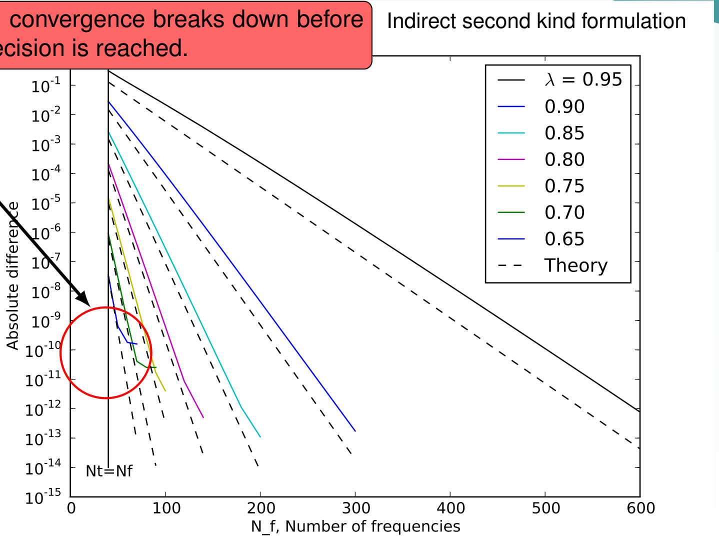
BEM++ 3.0.3

- Core library in C++, complete interface via Python
- Support for Laplace, Helmholtz, Maxwell equations
- Shared-Memory parallelisation (with Intel Threading Building Blocks (TBB))
- Built-In H-Matrix compression
- Support for FEM/BEM coupling with FEniCS
- High-Frequency OSRC preconditioners
- Extensive support for iterative solvers via interfaces to Eigen (C++)
- BSD style open source license
- Currently, Mac and Linux directly supported
- BEM++ now works on any plateform (Windows, Mac, Linux, Solaris) where VirtualBox is available...

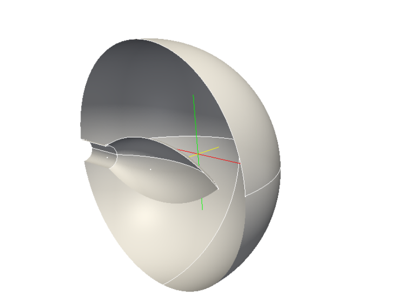


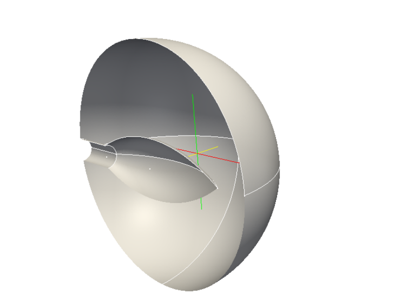
$\lambda_U = 1$ then the theoretical rate of convergence is $O\left(\lambda^{N_f}\right)$

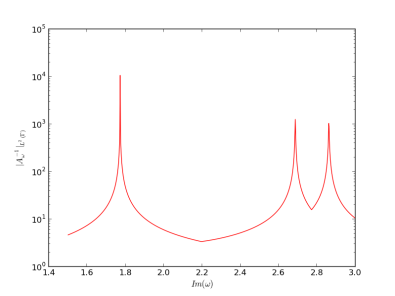
For small $\boldsymbol{\lambda}$, convergence breaks down before machine precision is reached.



$\boldsymbol{\lambda_U} = 1$ then the theoretical rate of convergence is $O\left(\boldsymbol{\lambda}^{N_f}\right)$

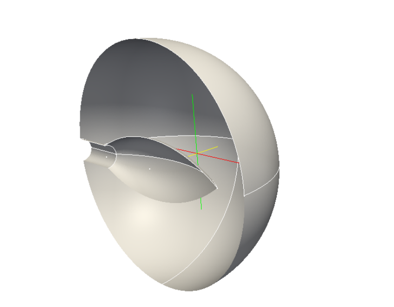


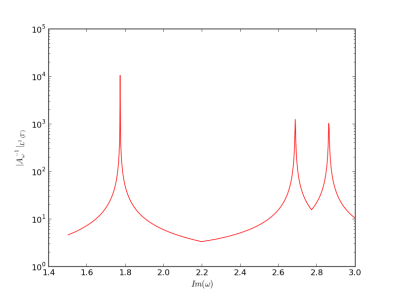




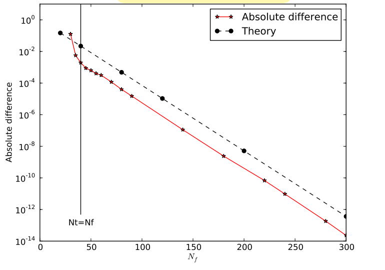
We plot the $L^2(\Gamma)$-norm of the inverse of $\mathbb{A}(\omega)$ (the matrix of the Galerkin discretisation of $\left[\frac{1}{2}I + K_\omega + S_\omega\right]$) when ω is purely imaginary: If z is a pole, then norm of the inverse $\to \infty$ when $\omega \to z$.

Eigenfrequencies are located on the imaginary axis for **the majority** of the intg. formulation.



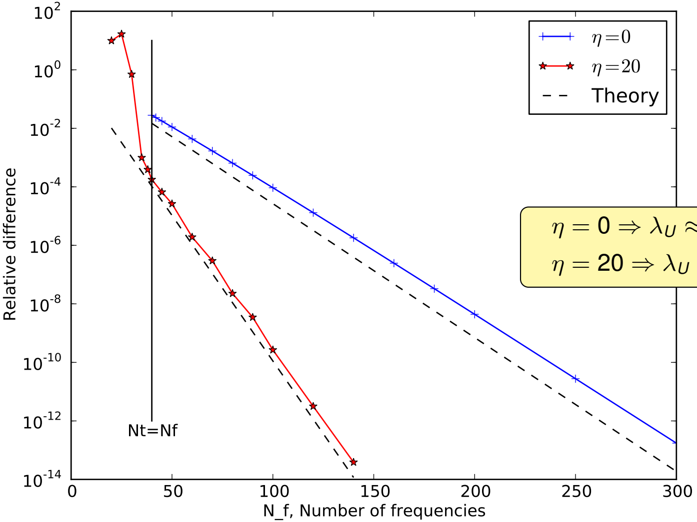


Convergence (0.90896^{N_f})



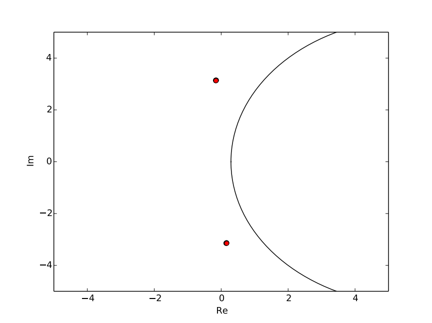
We plot the $L^2(\Gamma)$-norm of the inverse of $\mathbb{A}(\omega)$ (the matrix of the Galerkin discretisation of $\left[\frac{1}{2}I + K_\omega + S_\omega\right]$) when ω is purely imaginary: If z is a pole, then norm of the inverse $\to \infty$ when $\omega \to z$.

Eigenfrequencies are located on the imaginary axis for **the majority** of the intg. formulation.

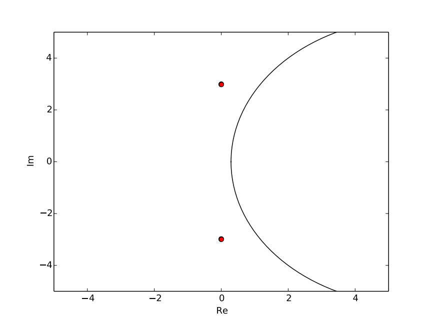


$\eta = 0 \Rightarrow \lambda_U \approx 1.0$

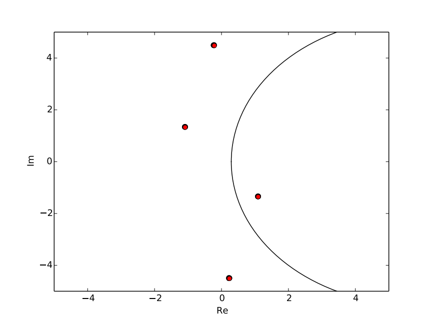
$\eta = 20 \Rightarrow \lambda_U \approx 1.13$



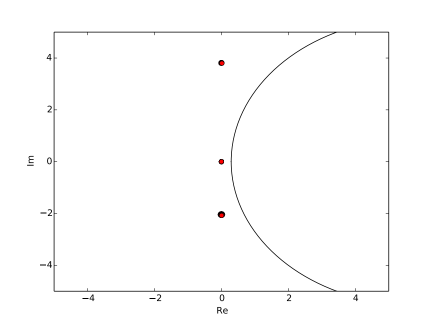
(a) Combined Integral Formulation with $\eta = 20i$.



(b) Combined Integral Formulation with $\eta = 20$.



(c) Combined Integral Formulation with $\eta = i$



(d) Combined Integral Formulation with $\eta = \omega$

Discretization of the time domain CFIE for acoustic scattering problems using convolution quadrature, P. Monk and Q. Chen, 2014.

Pole inside the contour

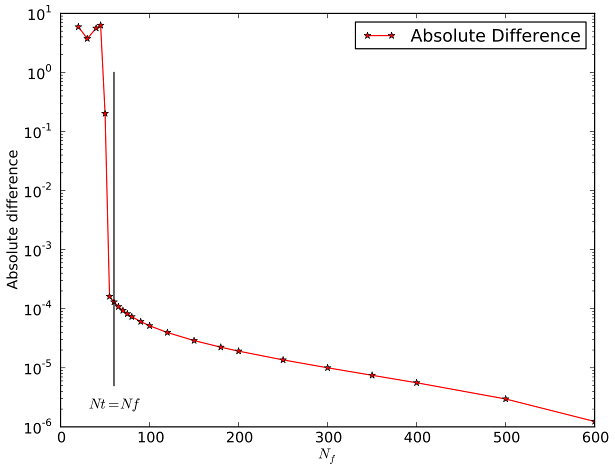
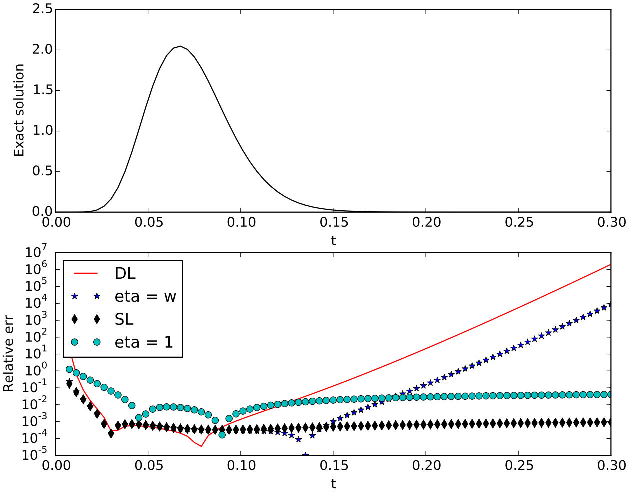


Figure : Convergence when $\eta = i$.

Relative error for four different indirect integral formulations.



1. The Z-Transform based CQ
2. Errors from the Z-transform based CQ
3. Approximation error of the inverse Z-transform
4. Approximation error of the Z-transform

The second error writes as

$$E_2(t_n;x) = \widetilde{\mathcal{Z}}_{N_f}^{-1}\left[U\right](t_n;x) - \widetilde{\mathcal{Z}}_{N_f}^{-1}\left[\widetilde{U}_{N_z}\right](t_n;x)$$
$$= \frac{\lambda^{-n}}{N_f}\sum_{\ell=1}^{N_f}\left(U(\lambda z_\ell;x) - \widetilde{U}_{N_z}(\lambda z_\ell;x)\right) z_\ell^{-n},$$

The second error writes as

$$E_2(t_n;x) = \widetilde{\mathcal{Z}}_{N_f}^{-1}[U](t_n;x) - \widetilde{\mathcal{Z}}_{N_f}^{-1}\left[\widetilde{U}_{N_z}\right](t_n;x)$$
$$= \frac{\boldsymbol{\lambda}^{-n}}{N_f}\sum_{\ell=1}^{N_f}\Big(U(\boldsymbol{\lambda} z_\ell;x) - \widetilde{U}_{N_z}(\boldsymbol{\lambda} z_\ell;x)\Big) z_\ell^{-n},$$

with

$$U(z;x) = \mathcal{B}_{k_z} G(z;x) \quad \text{and} \quad \widetilde{U}_{N_z}(z;x) = \mathcal{B}_{k_z}\widetilde{G}_{N_z}(z;x)$$

and,

$$G(z;x) = \sum_{n\geq 0} g(t_n;x) z^n \quad \text{and} \quad \widetilde{G}_{N_z}(z;x) = \sum_{n=0}^{N_z} g(t_n;x) z^n$$

The second error writes as

$$E_2(t_n;x) = \widetilde{\mathcal{Z}}_{\boldsymbol{N_f}}^{-1}[U](t_n;x) - \widetilde{\mathcal{Z}}_{\boldsymbol{N_f}}^{-1}\left[\widetilde{U}_{\boldsymbol{N_z}}\right](t_n;x)$$

$$= \frac{\boldsymbol{\lambda}^{-n}}{\boldsymbol{N_f}} \sum_{\ell=1}^{\boldsymbol{N_f}} \left(U(\boldsymbol{\lambda} z_\ell;x) - \widetilde{U}_{\boldsymbol{N_z}}(\boldsymbol{\lambda} z_\ell;x) \right) z_\ell^{-n},$$

with

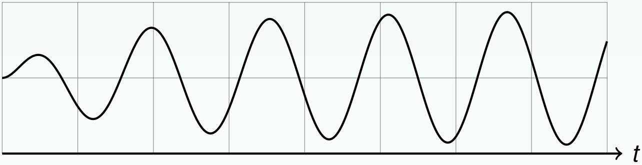
$$U(z;x) = \mathcal{B}_{k_z} G(z;x) \quad \text{and} \quad \widetilde{U}_{\boldsymbol{N_z}}(z;x) = \mathcal{B}_{k_z} \widetilde{G}_{\boldsymbol{N_z}}(z;x)$$

and,

$$G(z;x) = \sum_{n\geq 0} g(t_n;x) z^n \quad \text{and} \quad \widetilde{G}_{\boldsymbol{N_z}}(z;x) = \sum_{n=0}^{\boldsymbol{N_z}} g(t_n;x) z^n$$

We decouple the time steps used to perform the Z-transform to the time steps where we evaluate the solution.
We obtained results with a Gaussian beam as incident wave ; its support is **compact** in time so the computation of the Z-transform is not expensive and the error is very small.

Let's see an incident wave of the form:



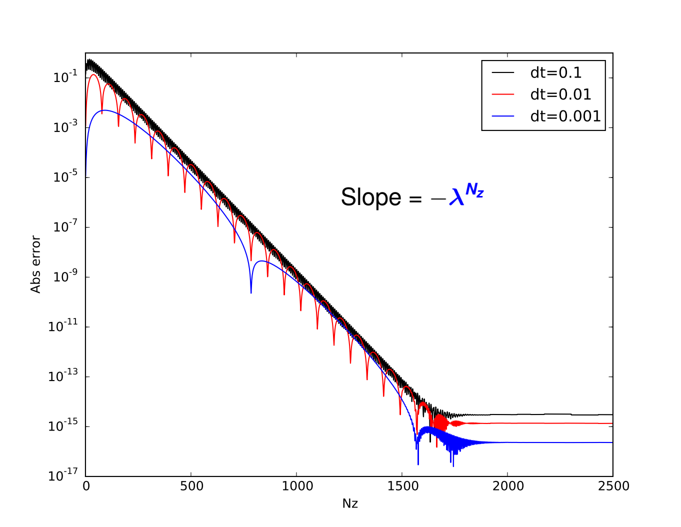
Then the error on the right-hand side is

$$G(\lambda z) - G_{N_z}(\lambda z) = \sum_{n=N_z+1}^{\infty} g(t_n)\lambda^n z^n$$

By increasing the number of time steps N_z to approximate the Z-transform of the time-domain boundary condition, we reduce the error on the right-hand side of our frequency equations.

Incident wave with slow (or no) decay

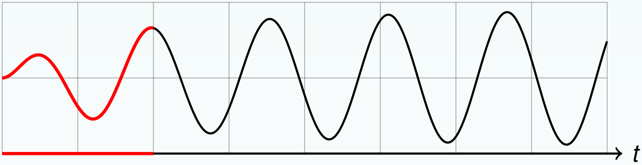
Let's



Then

By in … sform of the
time … nd side of our
frequ

Let's see an incident wave of the form:



Then the error on the right-hand side is

$$G(\lambda z) - G_{N_z}(\lambda z) = \sum_{n=N_z+1}^{\infty} g(t_n)\lambda^n z^n$$

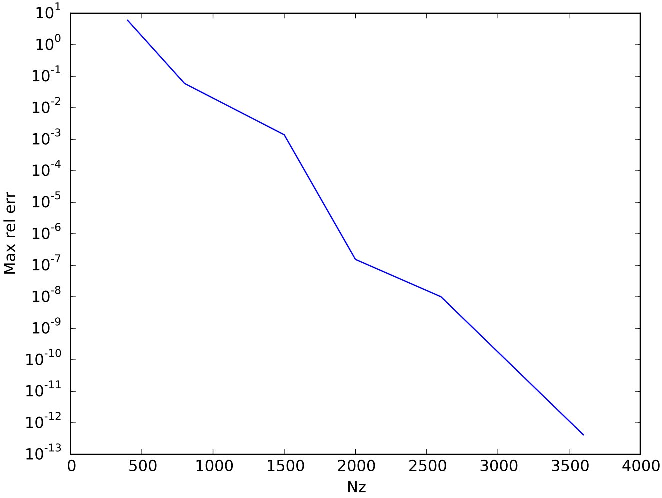
By increasing the number of time steps N_z to approximate the Z-transform of the time-domain boundary condition, we reduce the error on the right-hand side of our frequency equations.

The error related to the approximation error of the Z-Transform of the boundary data:

$$\widetilde{\mathcal{Z}}_{N_f}^{-1}\left[U\right](t_n;x) - \widetilde{\mathcal{Z}}_{N_f}^{-1}\left[\widetilde{U}_{N_z}\right](t_n;x) = \frac{\lambda^{-n}}{N_f}\sum_{\ell=1}^{N_f}\left(U(\lambda z_\ell;x) - \widetilde{U}_{N_z}(\lambda z_\ell;x)\right) z_\ell^{-n}$$

1. Can we bound the error on $\widetilde{\mathcal{Z}}_{N_f}^{-1}\left[U\right](t_n;x) - \widetilde{\mathcal{Z}}_{N_f}^{-1}\left[\widetilde{U}_{N_z}\right](t_n;x)$ by the error $G(z;x) - \widetilde{G}_{N_z}(z;x)$?
2. Is there an incident wave for which this error can be the leading error ?
3. For incident waves with compact support in time, we can reduce the number of evaluation by knowing when we can truncate the sum.

For λ and N_f given, we solve for different N_z and plot the maximal relative error in time.



XLiFE++ (eXtended Library of Finite Elements in C++)

XLiFE++ http://uma.ensta-paristech.fr/soft/XLiFE++/

- Deal with 1D, 2D, 3D scalar/vector transient/stationnary/harmonic pbs
- High order Lagrange FE, edge FE (Hrot, Hdiv), spectral FE
- H1 conform and non conform approximation (DG methods)
- Unassembling FE
- Integral methods (BEM, IR-FE, FEM-BEM)
- Essential condition (periodic, quasi-periodic)
- Absorbing condition, PML, DtN, . . .
- Meshing tools and export tool
- Many solvers (direct solvers, iterative solvers, eigen solvers)
- **In progress: CQ solver (multistep schemes almost done)**
- $\approx$ 120 000 lines
- Multi platform (linux, mac, windows)
- Online and paper documentation

- This CQ method is really easy to implement
- Two errors related to the CQ appeared:
  1. The first error (approximation of the inverse Z-transform) can play an important role on the accuracy of the solution and is analyzed now (See paper)
  2. For the second error (error coming from a "possible" bad approximation of the frequency right-hand sides), it is not clear if it is so important.
- We can tune/optimize the Z-transform based CQ with several parameters: λ, N_f, and N_z.
- The CQ method is "easy" to apply to other problems (Maxwell, elastodynamics), as soon as you have a frequency solver.
- **A Time-Domain solver will be available in the code developed by ENSTA-ParisTech and IRMAR: XLiFE++**
- With Stéphanie Chaillat (ENSTA) we will try some experiments in elastodynamics.

1. Inverse Z-Transform:
   - The error to approximate the contour integral (inverse Z-transform) relies upon the **distance from the contour to the poles** of the frequency solution.
   - The integral formulation used is **important** for the rate of convergence since eigenfrequencies of the interior Laplacian relies upon the integral formulation.
   - Analysis with Runge-Kutta schemes is slightly different because the diagonalisation is not possible for any frequency.

2. Z-transform of the rhs:
   - The $\boldsymbol{\lambda}^n$ appearing in the Z-transform allows to think there is **no special difficulty** with this error provided we take an adapted number of time steps (the error is really smal)
   - One interest is for fine meshes to **reduce the number of terms** to evaluate the Z-transform while keeping a good accuracy.
   - Is there some physical cases for which this error could play an **important role?**
   - Still some work to do to finish the analysis of this error.

Thank you!