High order time stepping and local time stepping for first order wave problems

Marc Duruflé[1] Sébastien Imperiale[2]

[1] Project Bacchus
INRIA Bordeaux Sud-Ouest

[2] Project POems
INRIA Rocquencourt

September 13, 2011

Bibliography and motivation

- Case of second-order hyperbolic problems treated by :
  Jean-Charles Gilbert and Patrick Joly *Higher order time stepping for second order hyperbolic problems and optimal CFL conditions*,
  Julien Diaz and Marcus Grote, *Energy Conserving Explicit Local Time-Stepping for Second-Order Wave Equations*

- For first-order hyperbolic problems, second-order time scheme :
  Serge Piperno *Symplectic local time-stepping in non-dissipative DGTD methods applied to wave propagation problems*

Model problem

First-order hyperbolic problem :

$$\frac{\partial U}{\partial t} + \sum_{i=1}^{d} A_i(x)\,\frac{\partial U}{\partial x_i} = f(x,t)$$

with $A_i(x)$ symmetric matrices.
Use of Local Discontinuous formulation with centered fluxes :

$$\int_K \frac{\partial U}{\partial t}\varphi\,dx - \int_K \sum_{i=1}^{d} A_i(x)\,U\,\frac{\partial \varphi}{\partial x_i}\,dx + \int_{\partial K} (\sum_{i=1}^{d} A_i(x) n_i)\{U\}\varphi\,dx = \int_K f(x,t)\varphi\,dx$$

$U, \varphi \in V_h = \{u \in (L^2(\Omega))^s$ such that $u \circ F_e \in \mathbb{P}_{r_e}$ or $\mathbb{Q}_{r_e}\}$
Order of approximation r_e is different for each element e of the mesh

Model problem

First-order hyperbolic problem :

$$\frac{\partial U}{\partial t} + \sum_{i=1}^{d} A_i(x) \frac{\partial U}{\partial x_i} = f(x,t)$$

with $A_i(x)$ symmetric matrices.
Associated evolution problem :

$$\frac{dU}{dt} + K_h U = F_h$$

With conservative boundary conditions, K_h is skew-symmetric.

Modified equation approach

Second-order leap-frog scheme :

$$\frac{U^{n+1} - U^{n-1}}{2\Delta t} + K_h U^n = \mathcal{F}_h^n,$$

Stability condition of this scheme :

$$\Delta t ||K_h||_2 \leq 1$$

Small elements in the mesh $\Rightarrow$ restrictive CFL

In absence of source, the exact solution is given by

$$\frac{U^{n+1} - U^{n-1}}{2} = i \sin(i \Delta t K_h) U^n$$

Taylor expansion of the sinus provide the following scheme of order $2m + 2$:

$$\frac{U^{n+1} - U^{n-1}}{2} + \left[\Delta t\, K_h + \sum_{q=1}^{m} \frac{(\Delta t\, K_h)^{2q+1}}{(2q+1)!} \right] U^n = 0.$$

Cost of this scheme : $2m + 1$ matrix-vector products with K_h

Stability condition of modified equation

Let us denote the polynomial :

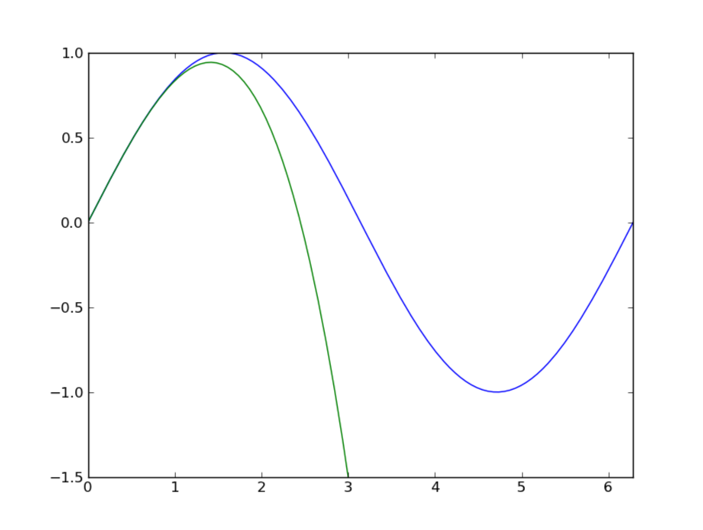
$$\tau_m(x) = x + \sum_{q=1}^{m} (-1)^q \frac{x^{2q+1}}{(2q+1)!}$$

Stability is obtained if

$$|\tau_m(x)| \leq 1 \Leftrightarrow x \in [0, \alpha_m]$$

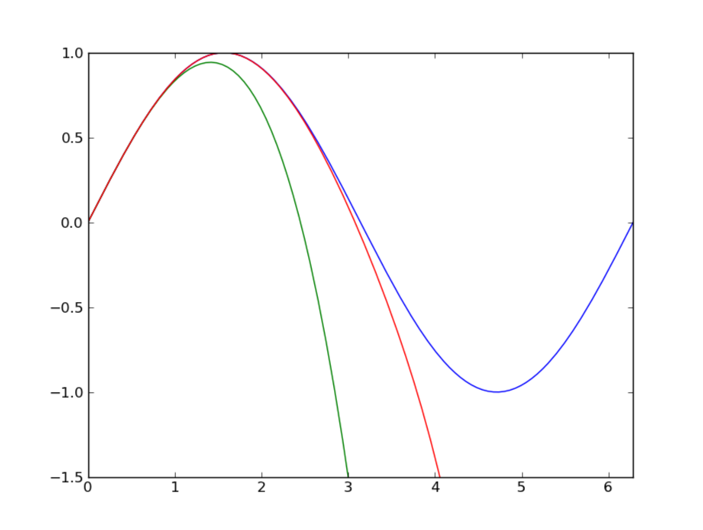
Stability condition of modified equation

For *m* even, $\alpha_m \leq \frac{3\pi}{2}$



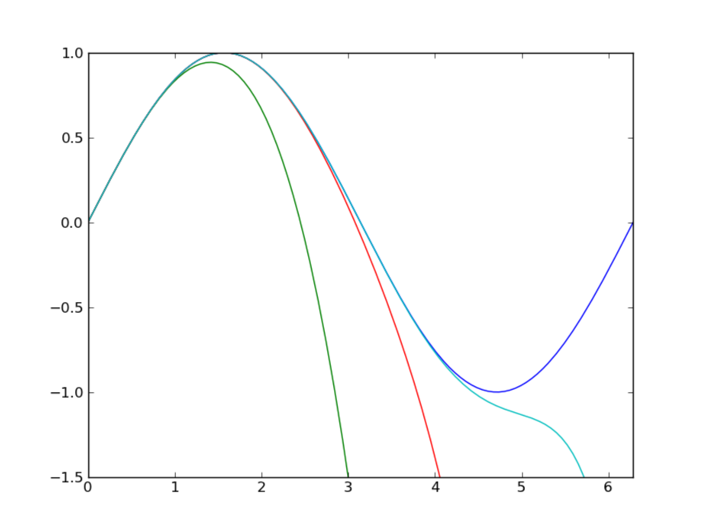
Stability condition of modified equation

For m even, $$\alpha_m \leq \frac{3\pi}{2}$$



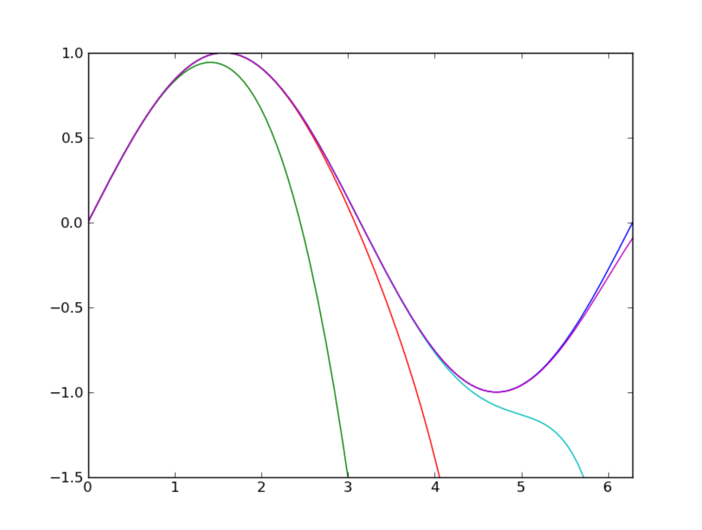
Stability condition of modified equation

For m even, $$\alpha_m \leq \frac{3\pi}{2}$$



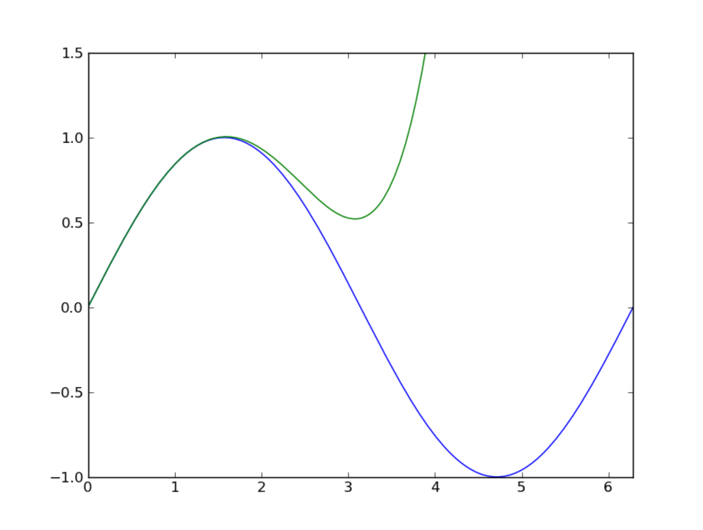
Stability condition of modified equation

For m even, $\qquad \alpha_m \leq \dfrac{3\pi}{2}$



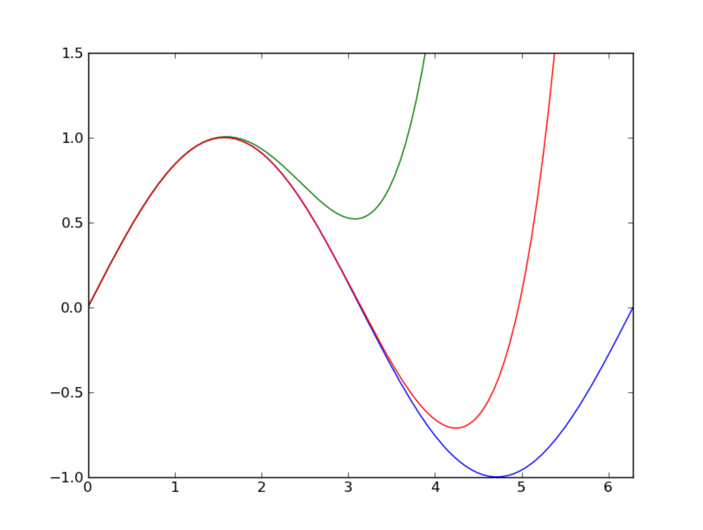
Stability condition of modified equation

For *m* odd, $\alpha_m \leq \frac{\pi}{2}$



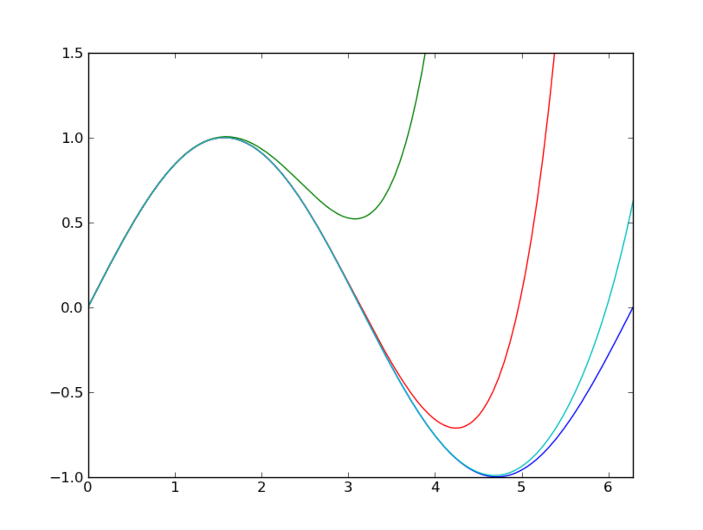
Stability condition of modified equation

For *m* odd, $\alpha_m \leq \frac{\pi}{2}$



Stability condition of modified equation

For *m* odd, $\alpha_m \leq \frac{\pi}{2}$



Improvement of modified equation

Higher-order terms are added to increase CFL number

$$\frac{U^{n+1} - U^{n-1}}{2} + \left[\sum_{q=0}^{m} \frac{(\Delta t \, K_h)^{2q+1}}{(2q+1)!} \right] U^n + \left[\sum_{q=m+1}^{r} \alpha_q (\Delta t K_h)^{2q+1} \right] U^n = 0$$

This scheme is written under the form

$$U^{n+1} - U^{n-1} + 2i \, \mathcal{T}_{2r+1}(i \Delta t K_h) U^n = 0$$

Improvement of modified equation

Higher-order terms are added to increase CFL number

$$\frac{U^{n+1} - U^{n-1}}{2} + \left[\sum_{q=0}^{m} \frac{(\Delta t\, K_h)^{2q+1}}{(2q+1)!} \right] U^n + \left[\sum_{q=m+1}^{r} \alpha_q (\Delta t K_h)^{2q+1} \right] U^n = 0$$

This scheme is written under the form

$$U^{n+1} - U^{n-1} + 2i\,\mathcal{T}_{2r+1}(i\Delta t K_h) U^n = 0$$

Optimal polynomial for $m = 0$ (second-order), and nearly optimal for $m = 1$ (fourth-order)

$$\mathcal{T}^m_{2r+1}(x) \;=\; \frac{1}{\xi_r} \mathcal{T}^{Cheb}_{2r+1}\left(\frac{(-1)^r \xi_r^m\, x}{(2r+1)} \right)$$

where $\mathcal{T}^{Cheb}_{2r+1}$ are Chebyshev polynomials of the first kind and

$$\xi_r^0 \;=\; 1, \quad \xi_r^1 \;=\; \frac{2r+1}{2\sqrt{r(r+1)}},$$

Improvement of modified equation

Optimal polynomial for $m = 0$ (second-order), and nearly optimal for $m = 1$ (fourth-order)

$$\mathcal{T}^m_{2r+1}(x) \;=\; \frac{1}{\xi_r}\mathcal{T}^{Cheb}_{2r+1}\Big(\frac{(-1)^r \xi^m_r\, x}{(2r+1)}\Big)$$

where $\mathcal{T}^{Cheb}_{2r+1}$ are Chebyshev polynomials of the first kind and

$$\xi^0_r \;=\; 1, \quad \xi^1_r \;=\; \frac{2r+1}{2\sqrt{r(r+1)}},$$

Stability condition :

$$\Delta t||K_h||_2 \;\leq\; \frac{2r+1}{\xi^m_r}$$

with

$$\xi^1_r \;=\; 1 + O(\frac{1}{r^2})$$

Optimal polynomials

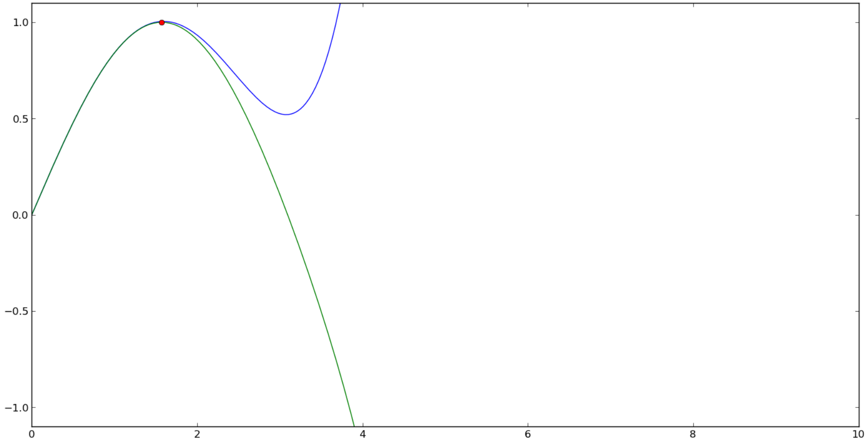
Optimal polynomial is sought, by searching tangent points $\tau_1, \tau_2, ..., \tau_k$ such that

$$\begin{cases} \mathcal{T}_{opt}(\tau_i) = -1 \text{ or } 1 \\ \mathcal{T}'_{opt}(\tau_i) = 0 \end{cases}$$

The associated non-linear system with unknowns $\tau_1, ..., \tau_k$ is solved numerically with Newton's method.

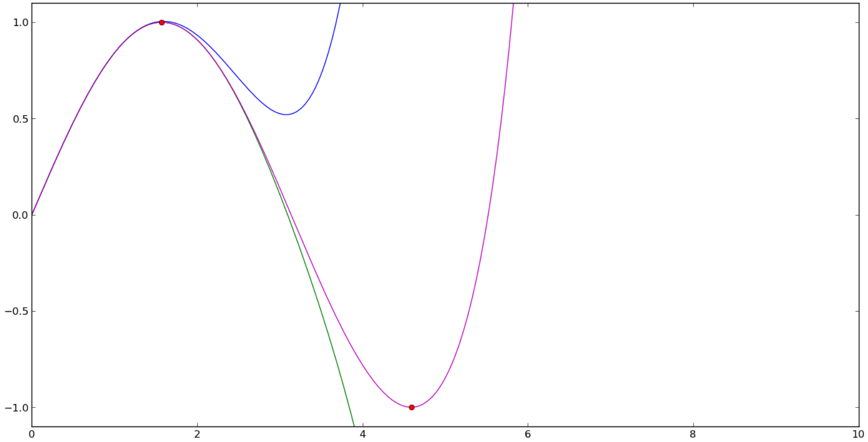
Optimal polynomials

Tangents points in red :



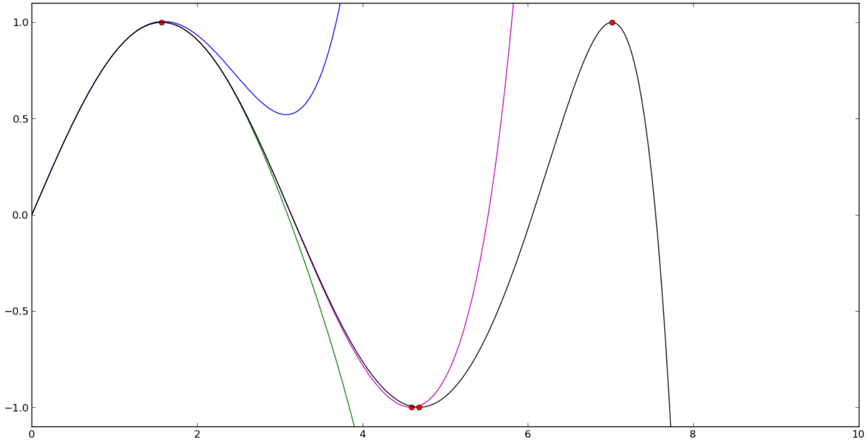
Optimal polynomials

Tangents points in red :



Optimal polynomials

Tangents points in red :



Optimal polynomials

CFL α_m obtained with this procedure :

m / r	0	1	2	3	4	5	6
0	1	3	5	7	9	11	13
1	-	2.85	4.91	6.94	8.95	10.96	12.97
2	-	-	1.49	3.84	5.80	7.71	9.61
3	-	-	-	3.79	5.77	7.69	9.59

Optimal polynomials

Efficiency compared to leap-frog scheme :

m / r	0	1	2	3	4	5	6
0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
1	-	0.95	0.982	0.991	0.994	0.996	0.998
2	-	-	0.298	0.549	0.644	0.701	0.739
3	-	-	-	0.541	0.641	0.699	0.738

Chebyshev reccurence

- Use of Horner algorithm leads to numerical instabilities for large values of r
- Use of Chebyshev reccurence leads to stable algorithms :

$$\mathcal{Q}_0 = U^n$$

$$\mathcal{Q}_1 = \frac{\xi_r}{2r+1} \Delta t K_h U^n$$

$$\mathcal{Q}_n = \frac{2\xi_r}{2r+1} \Delta t K_h \mathcal{Q}_{n-1} + \mathcal{Q}_{n-2}$$

$$\ldots$$

$$U^{n+1} = U^{n-1} - \frac{2}{\xi_r} \mathcal{Q}_{2r+1}$$

Two-level time stepping

Computational domain split into a "fine region" and a "coarse region"
P_h : projector onto the fine region

$$\frac{U^{n+1} - U^{n-1}}{2} + \Big[\sum_{q=0}^{m} \frac{(\Delta t\, K_h)^{2q+1}}{(2q+1)!}\Big] U^n$$
$$+ \Big[\sum_{q=m+1}^{r} \alpha_q\, (\Delta t K_h P_h)^{2q}\Big]\, \Delta t K_h U^n \; = \; 0,$$

Presence of P_h $\Rightarrow$ terms of second sum are computed only on the "fine region"
Skew-symmetry of the matrices $K_h P_h K_h \cdots K_h P_h K_h$ $\Rightarrow$ stability of this scheme (CFL not controlled)
α_q are the coefficients defined previously so that CFL is increased.

Stable two-level algorithm

For $m = 0$, it is equivalent to the following scheme (obtained by reproducing the strategy of Diaz and Grote) :

$$
\left\{
\begin{array}{l}
w_h = K_h(I - P_h)U^n \\
Q_0 = U^n \\
Q_1 = -\dfrac{\Delta t}{2r+1}(w_h + K_h P_h Q_0) \\
\text{For } k = 1, 2r \\
\qquad Q_{k+1} = Q_{k-1} - \dfrac{2\Delta t}{2r+1}(K_h P_h Q_k + w \delta_k \text{ even}) \\
\text{End For} \\
U^{n+1} = U^{n-1} + 2 Q_{2r+1}
\end{array}
\right.
$$

Stable algorithm even for large values of r

Multilevel algorithm

Domain split into hierarchical subdomains

$$\Omega = \bigcup \Omega_i = \bigcup K_e$$

with

$$\Omega \supset \Omega_1 \supset \Omega_2 \cdots \supset \Omega_r$$

Multilevel algorithm

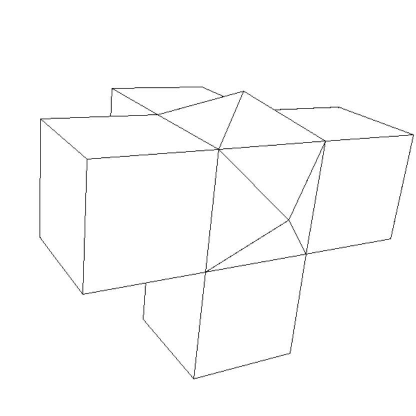
For each element, a nominal time step is computed

$$\Delta t_e = \frac{(2r+1)c}{\xi_r^m ||\mathcal{P}_e K_h \mathcal{P}_e||_2}$$

where c is a safety coefficient depending on the element.

Multilevel algorithm

by considering only direct neighbors of each element



Multilevel algorithm

Global time step Δt is chosen by the user, then a level i is affected to each element with respect to the rule :

$$\text{if} \quad \Delta t_e \leq \frac{\xi_i^m \Delta t}{2i+1}, \quad \text{then} \quad K_e \in \Omega_i.$$

Multilevel algorithm

We consider the following time scheme

$$\frac{U^{n+1} - U^{n-1}}{2\Delta t} \quad + K_h U^n + \Delta t^2 K_h P_1 K_h P_1 K_h U^n$$
$$+\Delta t^4 K_h P_1 K_h P_2 K_h P_2 K_h P_1 K_h U^n$$
$$+\Delta t^6 K_h P_1 K_h P_2 K_h P_3 K_h P_3 K_h P_2 K_h P_1 K_h U^n + \cdots = 0$$

where P_k are diagonal matrices :

$$P_k = \begin{pmatrix} \beta_k^0 & \ldots & & \\ \ldots & \beta_k^1 & \ldots & \\ & \ldots & \ldots & \ldots \\ & & \ldots & \beta_k^r \end{pmatrix}$$

with

$$\beta_k^m = 0, \ \forall m < k$$

Multilevel algorithm

If we write the expansion of optimal polynomial $\tau_{opt}^k(X)$ as :

$$\tau_{opt}^k(X) \;=\; X + \gamma_1^k X^3 \;+\; \gamma_2^k X^5 + \cdots + \gamma_k^k X^{2k+1}$$

Coefficients β_k^m are chosen to coincide with these polynomials for each level

```
For k = 1, r
    For m = 1, k-1
        β_k^m = 0
    End For
    For m = k, r
        β_k^m = sqrt(γ_k^m)
        For n = 1, k-1
            β_k^m = β_k^m / β_n^m
        End For
    End For
End For
```

Multilevel algorithm

Use of Horner algorithm :

$$
\begin{aligned}
Q_0 &= \Delta t K_h U^n \\
Q_1 &= \Delta t K_h P_1 Q_0 \\
Q_2 &= \Delta t K_h P_2 Q_1 \\
&\dots \\
Q_r &= \Delta t K_h P_r Q_{r-1} \\
Q_{r-1} &= Q_{r-1} + \Delta t K_h P_r Q_r \\
Q_{r-2} &= Q_{r-2} + \Delta t K_h P_{r-1} Q_{r-1} \\
&\dots \\
Q_0 &= Q_0 + \Delta t K_h P_1 Q_1 \\
U^{n+1} &= U^{n-1} - 2 Q_0
\end{aligned}
$$

unstable due to round-off errors when $r \geq 14$.

2-D numerical results

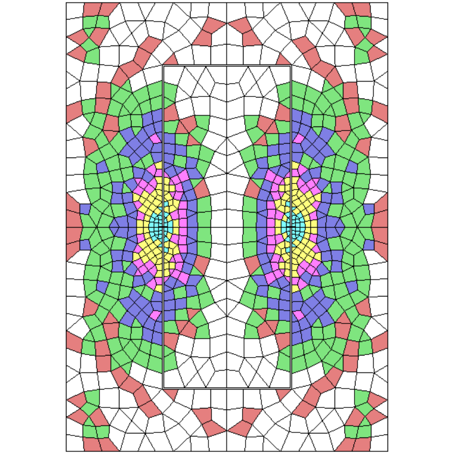
We consider wave equation

$$A_i = \begin{pmatrix} 0 & e_i^* \\ e_i & 0 \end{pmatrix}$$

and Neumann boundary conditions so that K_h is skew-symmetric

2-D numerical results

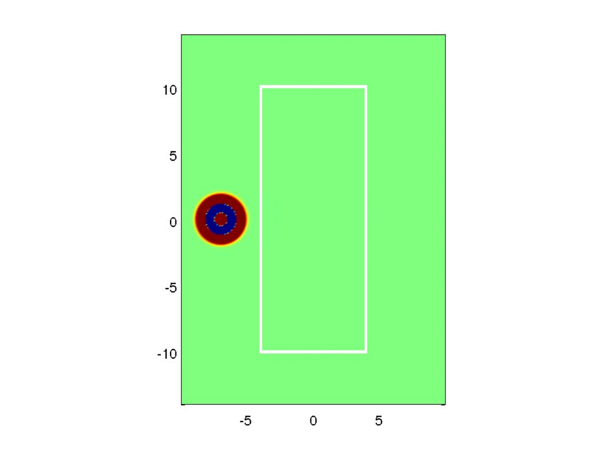
Box pierced with two small holes



each color corresponds to a different order of approximation in space

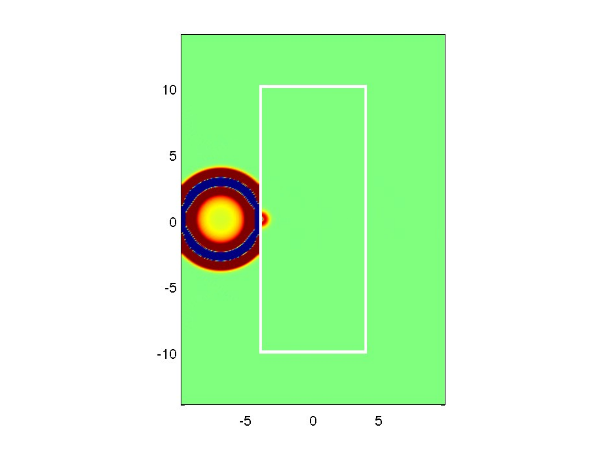
2-D numerical results

Solution obtained for $t = 2$



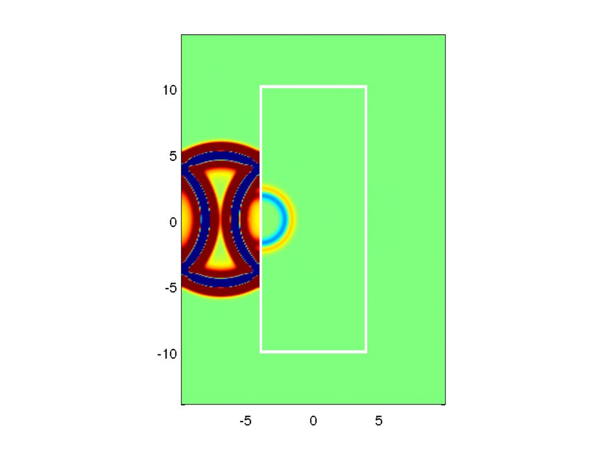
2-D numerical results

Solution obtained for $t = 4$



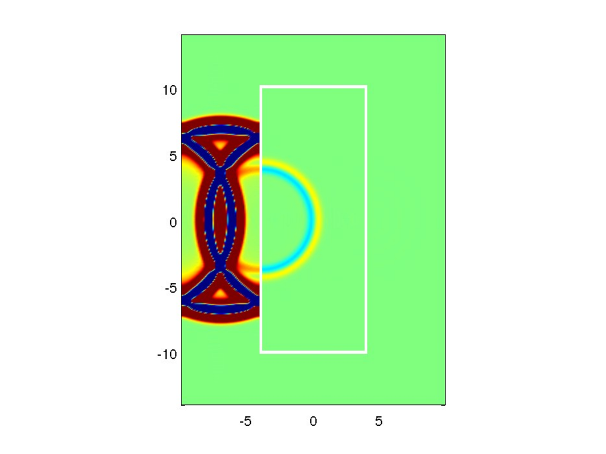
2-D numerical results

Solution obtained for $t = 6$



2-D numerical results

Solution obtained for $t = 8$



2-D numerical results

$\Delta t_{max} = 0.01036, \quad \Delta t_{min} = 0.000737$

Ratio $\dfrac{\Delta t_{max}}{\Delta t_{min}} = 14.1$

Computational time with optimized fourth order ($\Delta t = 0.005$): 767s

2-D numerical results

$\Delta t_{max} = 0.01036, \quad \Delta t_{min} = 0.000737$

Ratio $\frac{\Delta t_{max}}{\Delta t_{min}}$ = 14.1

Computational time with optimized fourth order ($\Delta t = 0.005$): 767s

Fourth-order local time stepping with the following repartition :

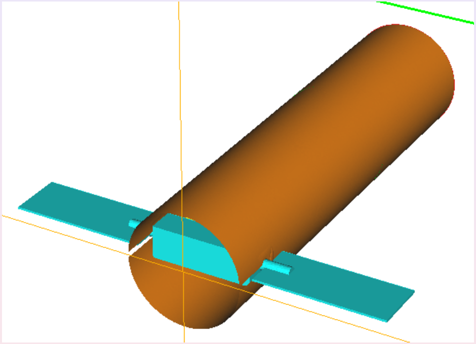
Level	1	2	3	4	5	6	7	8
Number of elements	1024	0	0	0	0	0	16	4

L^2 error for $t = 10$: 7.78e-6

Computational time ($\Delta t = 0.01$): 177s

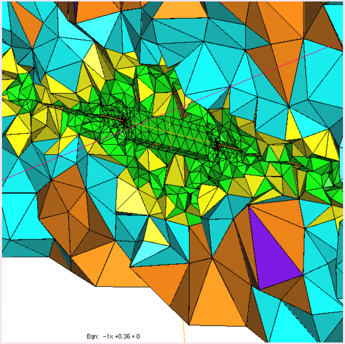
3-D numerical results

Scattering by a satellite



3-D numerical results

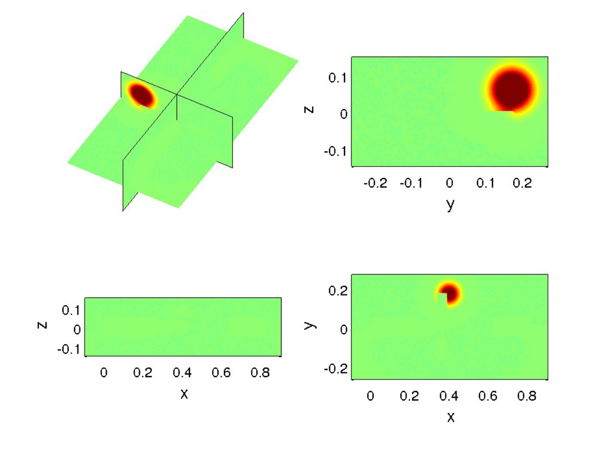
Mesh used for the simulations



each color corresponds to a different order of approximation in space

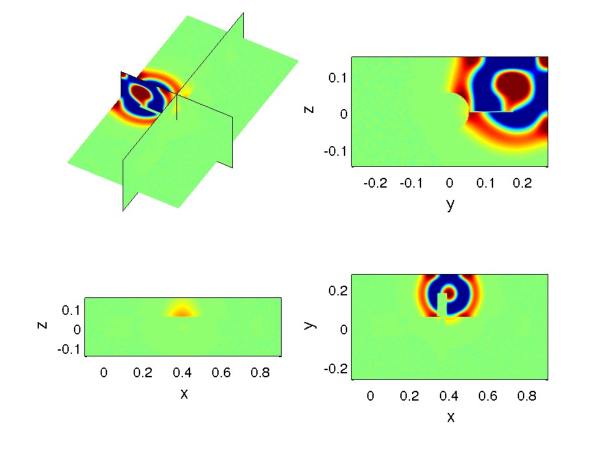
3-D numerical results

Solution obtained for $t = 0.1$



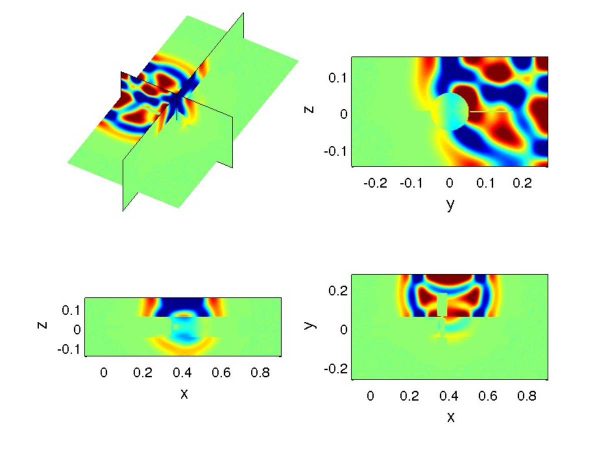
3-D numerical results

Solution obtained for $t = 0.2$

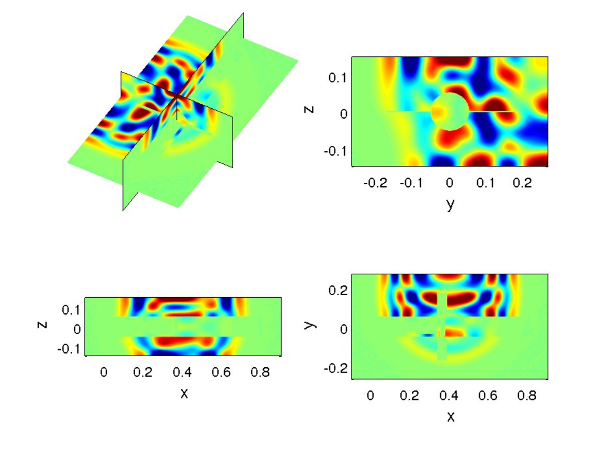


3-D numerical results

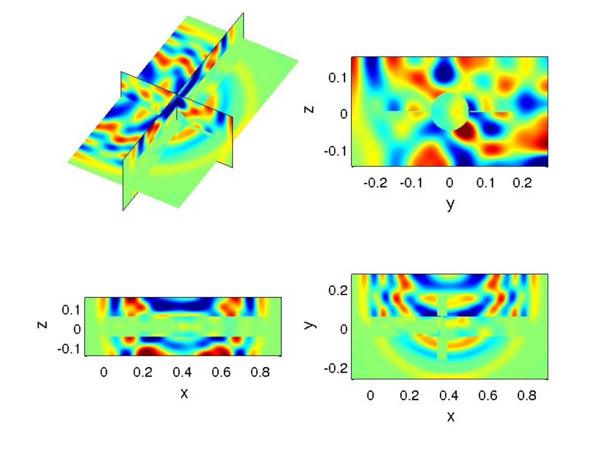
Solution obtained for $t = 0.3$



Solution obtained for $t = 0.4$



Solution obtained for $t = 0.5$



3-D numerical results

$\Delta t_{max} = 1.177e-3, \quad \Delta t_{min} = 1.442e-5$

Ratio $\dfrac{\Delta t_{max}}{\Delta t_{min}}$ = 81.6

Computational time with standard leap frog ($\Delta t = 1e-5$): 63.4h

3-D numerical results

$\Delta t_{max} = 1.177e-3, \quad \Delta t_{min} = 1.442e-5$

Ratio $\dfrac{\Delta t_{max}}{\Delta t_{min}}$ = 81.6

Computational time with standard leap frog ($\Delta t = 1e-5$): 63.4h

Second-order local time stepping with the following repartition :

Level	1	2	3	4	5	6	7	8	9	10
Number of elements	64468	7629	867	35	3	0	0	3	2	1

L^2 error for $t = 0.5$: 2.31e-3

Computational time ($\Delta t = 2.5e-4$) : 9.48h

Runge-Kutta schemes

Another interesting set of schemes is obtained by considering Taylor expansion of the exponential. In absence of source :

$$U^{n+1} = \sum_{q=0}^{m} \frac{(\Delta t K_h)^q}{q!} U^n$$

Such schemes coincide with Runge-Kutta schemes for $m \leq 4$
Advantage : K_h can be any matrix (not only skew-symmetric matrices)

Runge-Kutta schemes

Cost of this scheme : m matrix-vector products
Stability condition for a skew-symmetric matrix :

$$\Delta t ||K_h||_2 = \alpha_m$$

with :

- $m = 4k + 1$ or $m = 4k + 2$, $\alpha_m = 0 \Rightarrow$ scheme always unstable
- $m = 4k + 3$, $\alpha_m < \pi$ and tends to $\frac{\pi}{2}$
- $m = 4k$, $\alpha_m < \frac{3\pi}{2}$ and tends to π

Optimal Runge-Kutta schemes

For second-order schemes, the following polynomials provide a CFL of m along the imaginary axis :

$$R_m(z) = \frac{1}{2} V_{m-1}(z) + V_m(z) + \frac{1}{2} V_{m+1}(z)$$

with $V_m(z) = i^m T_m(\frac{z}{im})$
associated scheme :

$$U^{n+1} = R_m(\Delta t K_h) U^n$$

Optimal Runge-Kutta schemes

For second-order schemes, the following polynomials provide a CFL of m along the imaginary axis :

$$R_m(z) = \frac{1}{2} V_{m-1}(z) + V_m(z) + \frac{1}{2} V_{m+1}(z)$$

with $V_m(z) = i^m T_m(\frac{z}{im})$
associated scheme :

$$U^{n+1} = R_m(\Delta t K_h) U^n$$

For fourth-order schemes, Kinmark-Gray polynomials :

$$K_m(z) = \frac{1}{\sqrt{\beta^2 + 1}} \left(i^{m+1} \beta \, T_{m-1}(\frac{iz}{\beta}) + \frac{i^m}{2} \left[(m-2) \, T_m(\frac{iz}{\beta}) - m T_{m-2}(\frac{iz}{\beta}) \right] \right)$$

$$CFL = \beta = \sqrt{(m-1)^2 - 1}$$

Local time-stepping with Runge-Kutta approach

Multilevel algorithm :

$$U^{n+1} = U^n + (\Delta t K_h + \frac{\Delta t^2}{2} K_h^2 + \frac{\Delta t^3}{6} K_h^3 + \frac{\Delta t^4}{24} K_h^4) U^n$$
$$+ \Delta t^5 K_h P_1 K_h^4 U^n + \Delta t^6 K_h P_2 K_h P_1 K_h^4 U^n$$

coefficients of diagonal matrices P_1, P_2, etc, deduced from coefficients of Kinmark polynomials

Local time-stepping with Runge-Kutta approach

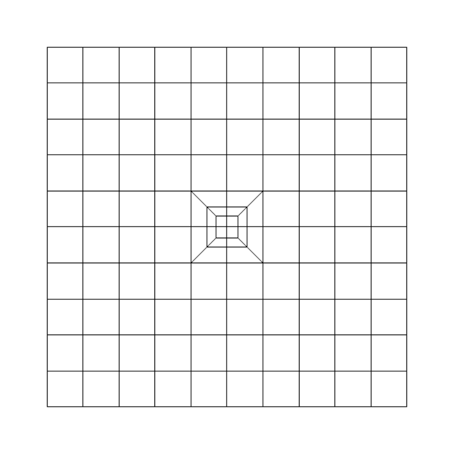
Multilevel algorithm :

$$U^{n+1} = U^n + (\Delta t K_h + \frac{\Delta t^2}{2} K_h^2 + \frac{\Delta t^3}{6} K_h^3 + \frac{\Delta t^4}{24} K_h^4) U^n$$
$$+\Delta t^5 K_h P_1 K_h^4 U^n + \Delta t^6 K_h P_2 K_h P_1 K_h^4 U^n$$

coefficients of diagonal matrices P_1, P_2, etc, deduced from coefficients of Kinmark polynomials

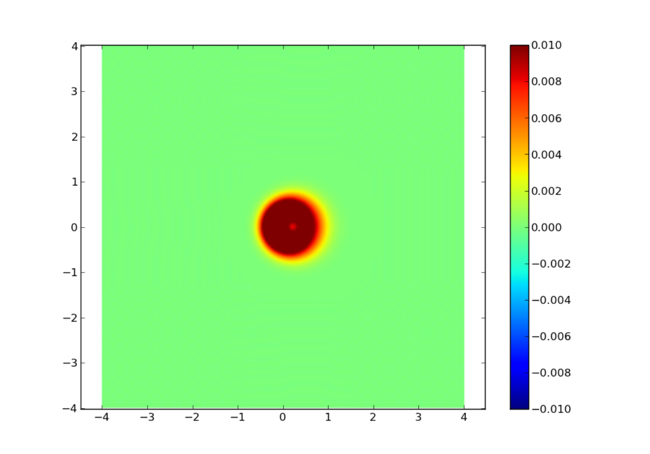
Local time-stepping with Runge-Kutta approach

Numerical experiments with a mesh refined at the origin



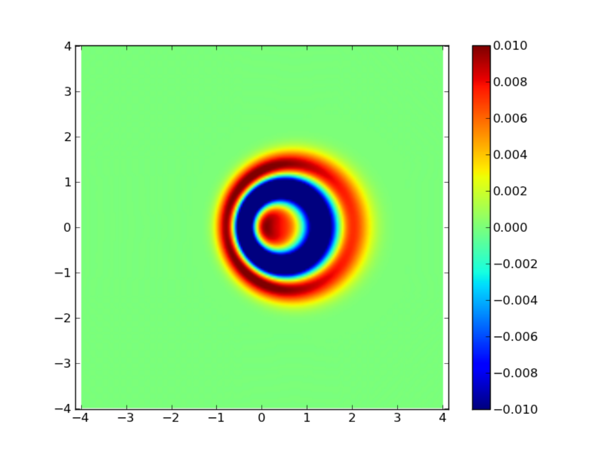
Local time-stepping with Runge-Kutta approach

Aeroacoustics with an uniform flow and absorbing boundary condition



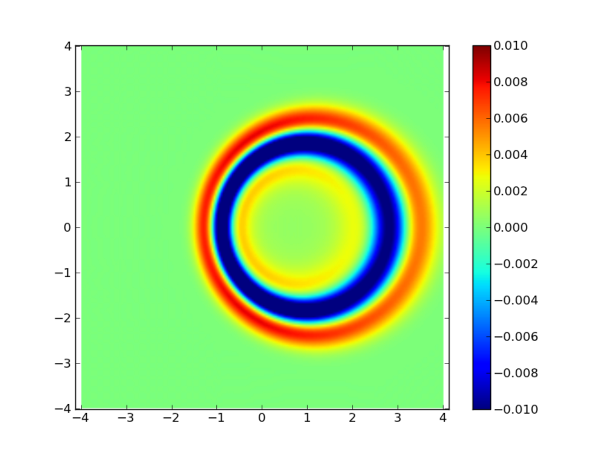
Local time-stepping with Runge-Kutta approach

Aeroacoustics with an uniform flow and absorbing boundary condition



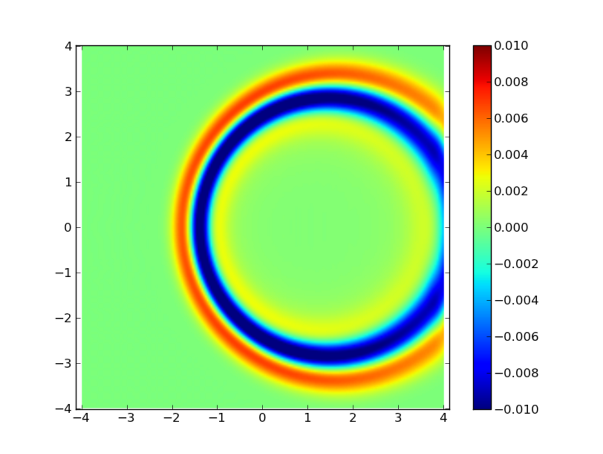
Local time-stepping with Runge-Kutta approach

Aeroacoustics with an uniform flow and absorbing boundary condition



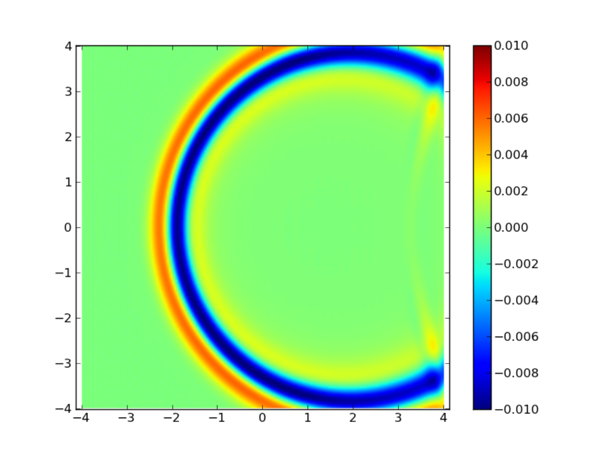
Local time-stepping with Runge-Kutta approach

Aeroacoustics with an uniform flow and absorbing boundary condition



Local time-stepping with Runge-Kutta approach

Aeroacoustics with an uniform flow and absorbing boundary condition



L^2 error : 1.8e-3

Prospects

- Stable multi-level algorithm for large values of r
- Better knowledge of the global CFL from the local time steps Δt_e
- Handle dissipative terms (due to upwind fluxes) locally with uncentered approximations