Simulation of Laser Propagation in a Plasma with a Frequency Helmholtz Equation

S. Desroziers**, **F. Nataf***, R. Sentis**

*Laboratoire J.L. Lions, CNRS UMR7598. Paris, FRANCE

Commissariat à l'Énergie Atomique, CEA-DAM, Bruyères le Chatel, FRANCE

Outline of the talk

1. Motivation

   Laser-Plasma interaction

   The Equations

   Difficulties

2. Numerical strategy

   Non matching Grid

   Domain Decomposition

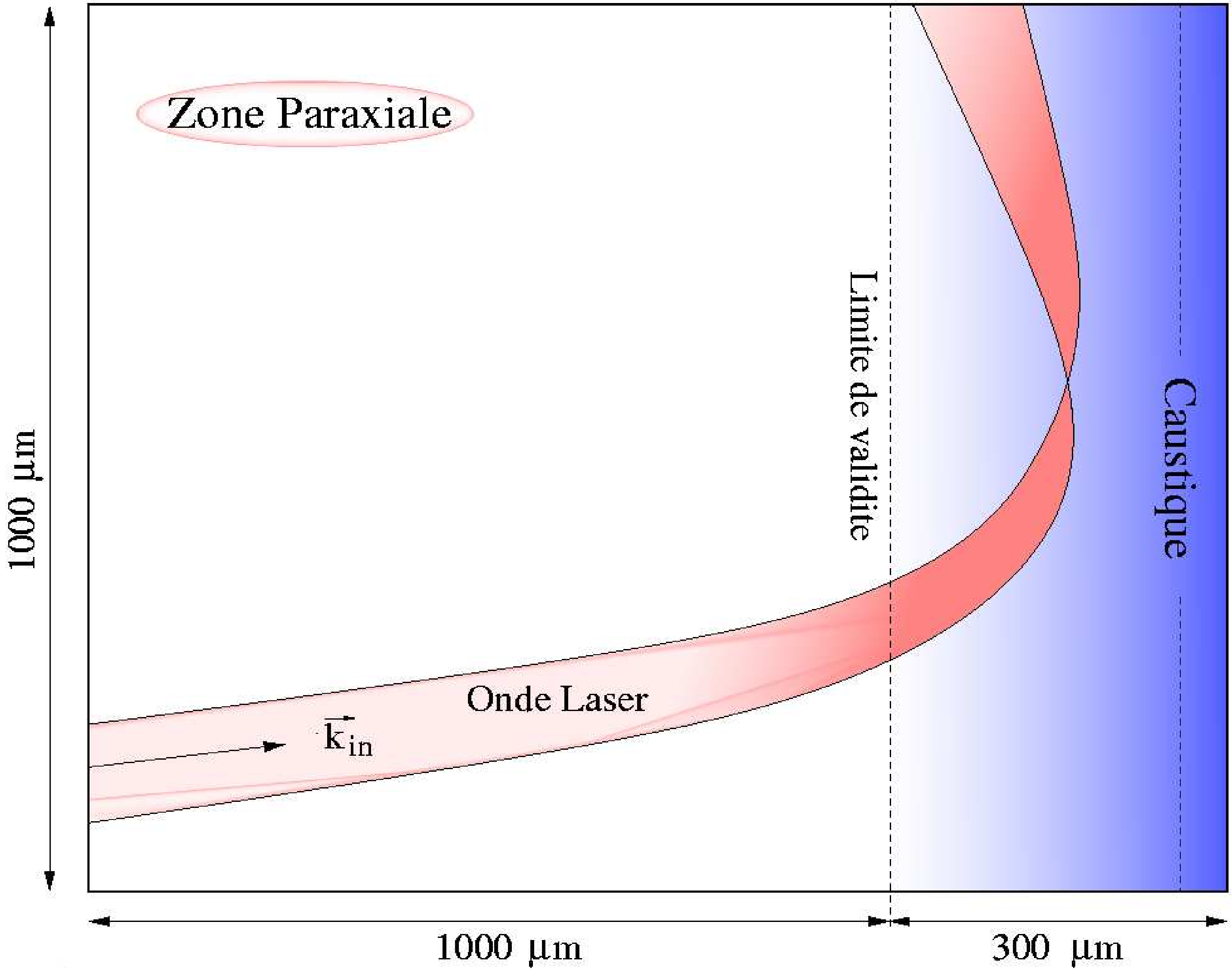
   Cyclic Reduction

   Parallelism

3. Numerical Results

4. Conclusion and Prospects

Physical problem



Deflection of a laser beam by a plasma

Equations (1/2)

- Plasma: Euler equations

$$\begin{cases} \dfrac{\partial N_I}{\partial t} + \nabla(N_I \vec{U}) = 0 \\ m_I \left(\dfrac{\partial}{\partial t}(N_I \vec{U}) + \nabla(N_I \vec{U} \cdot \vec{U}) \right) + \nabla P = -N_I \gamma \nabla |\psi|^2 \end{cases}$$

with: plasma velocity: $\vec{U}$, Pressure: P, Electronic density: $N_e = Z N_I$, laser energy: $|\psi|^2$

coupled with propagation models for the laser:

Equations for the laser (2/2)

- Time harmonic wave equation (Helmholtz) :

$$\boxed{\left[\epsilon^2 \Delta + \mathrm{i}\nu + (1 - N_e)\right] \psi = 0}$$

- Assumptions on the density $N_e(x, y) = N_0(x) + \delta_N(x, y)$ with

$$\delta_N(x, y) << N_0(x)$$

(propagative equation) $\quad 0 < N_0(x) < 1 \quad$ (elliptic equation)

- and where valid: Paraxial approximation (Schroedinger type) :
Let $\psi = \Psi e^{\mathrm{i}\frac{\vec{k}\cdot\vec{x}}{\epsilon}}$ where the vector $\vec{k}$ satisfies the eikonal equation $|\vec{k}|^2 = 1 - N_0$

$$\boxed{\epsilon^2 \Delta_\perp \Psi + \epsilon \mathrm{i} \Psi \nabla \cdot \vec{k} + 2\epsilon \mathrm{i} \vec{k} \cdot \nabla \Psi + \mathrm{i}\nu_0 \psi - \delta_N \Psi = 0}$$

Difficulties

- Multiscale problem in time and space
- Coupling the Euler equations with the propagative ones
- Coupling the Paraxial zone ($h \simeq \lambda_0$) with the Helmholtz zone ($h \simeq \lambda_0/10$)
- Solving a very large variable coefficient Helmholtz problem in a non symmetric form (due to the use of Perfectly Matched Layers)
- Realistic computation $\Rightarrow$ some hundreds of millions of unknowns mostly in the Helmholtz zone.

We shall use a combination of

- Grid interpolation between the various grids (hydrodynamic, Paraxial and Helmholtz)
- Specific solver that takes advantage of $N_0(x) >> \delta N(x, y)$.

Non-matching Grids

- The mesh for the fluid is much coarser than the mesh for the Helmholtz equations: linear interpolation gives good results
- Coupling between the paraxial and the Helmholtz zones where equations and grids are not the same.
  It is achieved via a discretized absorbing boundary condition

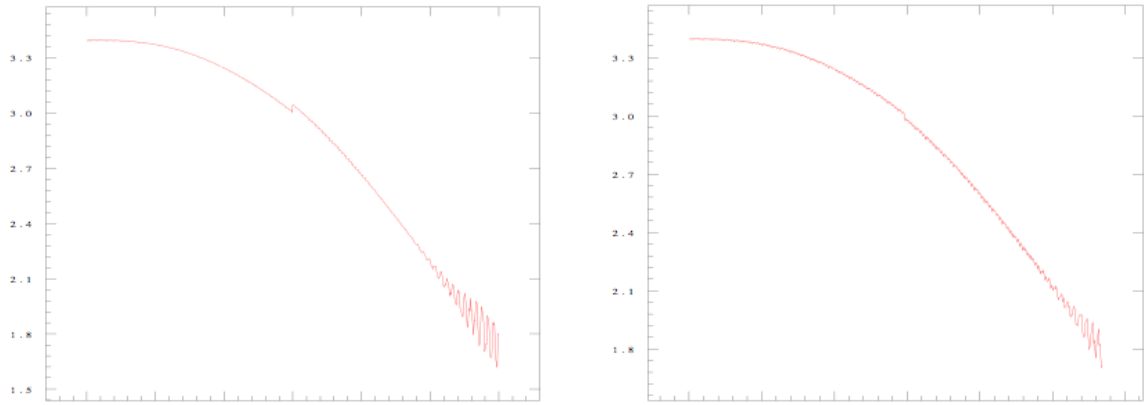


Figure 1: Laser intensity vs. x for two couplings between the Paraxial and Helmhltz zones

Global strategy for solving the Helmholtz problem

The most CPU and storage demanding part is the solve of the Helmholtz problem at each time step.

In a Krylov based method, we precondition the Helmholtz operator

$$\epsilon^2 \Delta \psi_c + \mathrm{i}\nu\psi_c + (1 - N_0(x))\psi_c - \delta_N(x,y)\psi_c$$

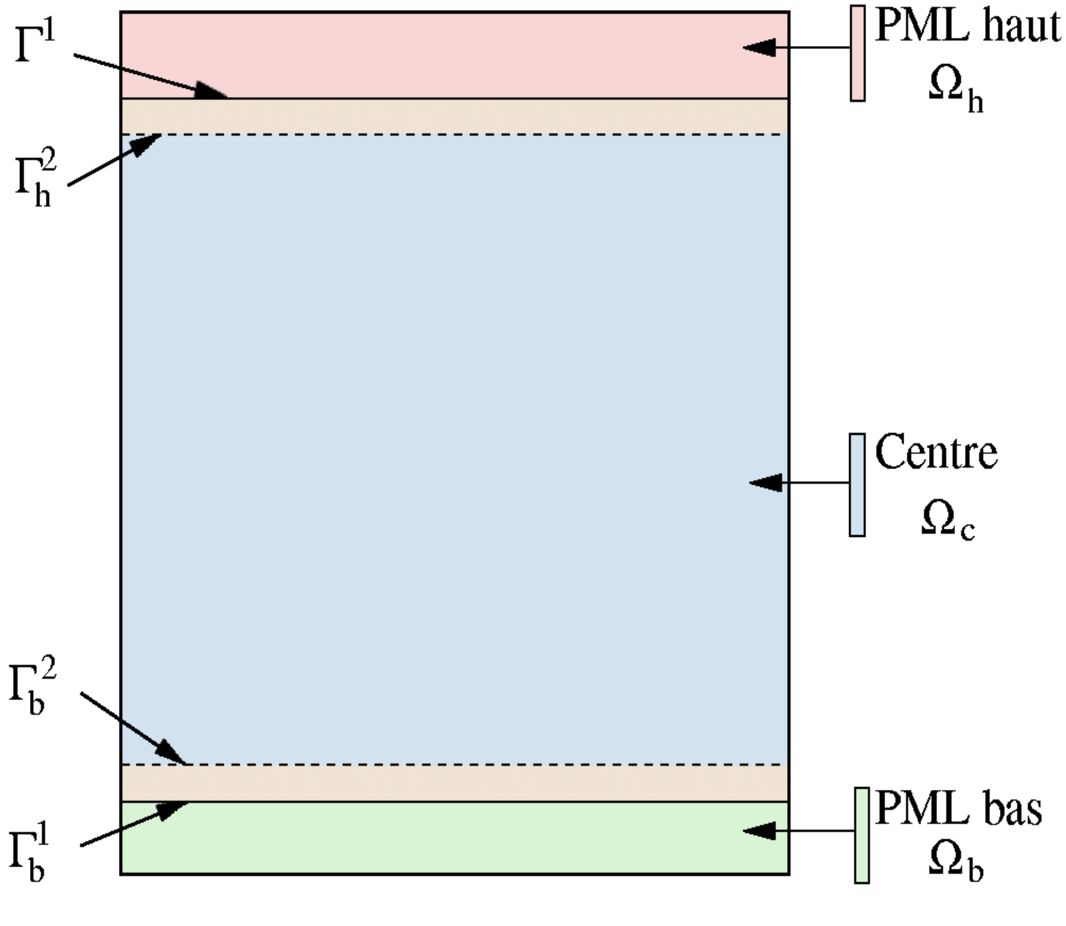
by

$$\epsilon^2 \Delta \psi_c + \mathrm{i}\nu\psi_c + (1 - N_0(x))\psi_c$$

which is solved by a cyclic reduction method.

In order to take care of boundary conditions, we use a domain decomposition method.

Overlapping Domain Decomposition (1/3)



The "Helmholtz" computational domain is decomposed into three subdomains: two long PMLs and a large Helmholtz central zone.

Overlapping domain decomposition method (2/3)

Robin interface conditions between the PMLs and Helmholtz zones

$$\begin{cases} \epsilon^2 \left[\eta(y)\dfrac{\partial}{\partial y}\left(\eta(y)\dfrac{\partial}{\partial y}\right) + \dfrac{\partial^2}{\partial x^2}\right]\psi_h + \mathrm{i}\nu\psi_h + (1-N_0)\psi_h = 0 & \text{in} \quad \Omega_h \\ \dfrac{\partial \psi_h}{\partial y} + \alpha\psi_h = \dfrac{\partial \psi_c}{\partial y} + \alpha\psi_c & \text{on} \quad \Gamma_h^2 \end{cases}$$

$$\begin{cases} \epsilon^2 \Delta\psi_c + \mathrm{i}\nu\psi_c + (1-N_0)\psi_c - \delta_N\psi_c = 0 & \text{in} \quad \Omega_c \\ \dfrac{\partial \psi_c}{\partial y} + \alpha\psi_c = \dfrac{\partial \psi_h}{\partial y} + \alpha\psi_h & \text{on} \quad \Gamma_h^1 \\ \dfrac{\partial \psi_c}{\partial y} + \alpha\psi_c = \dfrac{\partial \psi_b}{\partial y} + \alpha\psi_b & \text{on} \quad \Gamma_b^1 \end{cases}$$

$$\begin{cases} \epsilon^2 \left[\eta(y)\dfrac{\partial}{\partial y}\left(\eta(y)\dfrac{\partial}{\partial y}\right) + \dfrac{\partial^2}{\partial x^2}\right]\psi_b + \mathrm{i}\nu\psi_b + (1-N_0)\psi_b = 0 & \text{in} \quad \Omega_b \\ \dfrac{\partial \psi_b}{\partial y} + \alpha\psi_b = \dfrac{\partial \psi_c}{\partial y} + \alpha\psi_c & \text{on} \quad \Gamma_b^2 \end{cases}$$

Overlapping Domain Decomposition (3/3)

- Algebraic formulation :

$$\text{Let} \quad A = \begin{bmatrix} A_{P1} & C_1 & 0 \\ C_2 & A_H & C_3 \\ 0 & C_4 & A_{P2} \end{bmatrix}, \quad \text{Solve:} \quad A \begin{pmatrix} X_h \\ X_c \\ X_b \end{pmatrix} = b\,.$$

- Algebraic decomposition :

$$A_D = \begin{bmatrix} A_{P1} & 0 & 0 \\ 0 & A_G & 0 \\ 0 & 0 & A_{P2} \end{bmatrix} \quad \text{and} \quad A_E = \begin{bmatrix} 0 & C_1 & 0 \\ C_2 & A_{\delta N} & C_3 \\ 0 & C_4 & 0 \end{bmatrix}$$

- Remarks
  - GMRES algorithm preconditioned by A_D (the fluctuations δ_N are treated iteratively)
  - A_G is a very large matrix but with a simple structure
  - The matrices A_{P1}, A_{P2} are factorized by a direct method

Cyclic Reduction for solving $A_D u = f$ (1/2)

$$A_D u = \begin{bmatrix} A & -T & & & \\ -T & A & -T & & \\ & \ddots & \ddots & \ddots & \\ & & & -T & A \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{bmatrix}$$

where $T = cI$. Recursively and in parallel:

- Elimination

$$\left\{ \begin{array}{cccccccccccc} -Tu_{i-2} & + & Au_{i-1} & - & Tu_i & & & & & = & f_{i-1} \\ & - & Tu_{i-1} & + & Au_i & - & Tu_{i+1} & & & = & f_i \\ & & & - & Tu_i & + & Au_{i+1} & - & Tu_{i+2} & = & f_{i+1} \end{array} \right.$$

- Reduced system

$$-TA^{-1}Tu_{i-2} + (A - 2TA^{-1}T)u_i - TA^{-1}Tu_{i+2} = f_i + TA^{-1}(f_{i-1} + f_{i+1})$$

- Redistribution

$$Au_{i-1} = f_{i-1} + T(u_{i-2} + u_i)$$

Cyclic Reduction for solving $A_D u = f$ (2/2)

- Via a LR (Parlett) diagonalization process (A is a tridiagonal matrix so that the LR method is much cheaper than the QR method):

$$A = Q\Lambda^{(0)}Q^T, \quad T = Q\Gamma^{(0)}Q^T \quad \text{et} \quad QQ^T = I$$

- Induction formulas

$$\left\{ \begin{array}{lcl} T^{(r)} & = & \left(T^{(r-1)}\right)^2 \left(A^{(r-1)}\right)^{-1} \\ A^{(r)} & = & \left(A^{(r-1)}\right)^{-1} - 2T^{(r)} \end{array} \right. \Longrightarrow \left\{ \begin{array}{lcl} \Gamma^{(r)} & = & \left(\Lambda^{(r-1)}\right)^2 \left(\Lambda^{(r-1)}\right)^{-1} \\ \Lambda^{(r)} & = & \left(\Lambda^{(r-1)}\right)^{-1} - 2\Gamma^{(r)} \end{array} \right.$$

- Elimination

$$x = (T^{(r-1)})^2 (A^{(r-1)})^{-1} y \quad \Longrightarrow \quad x = Q(T^{(r-1)})^2 (\Lambda^{(r-1)})^{-1} Q^T y$$

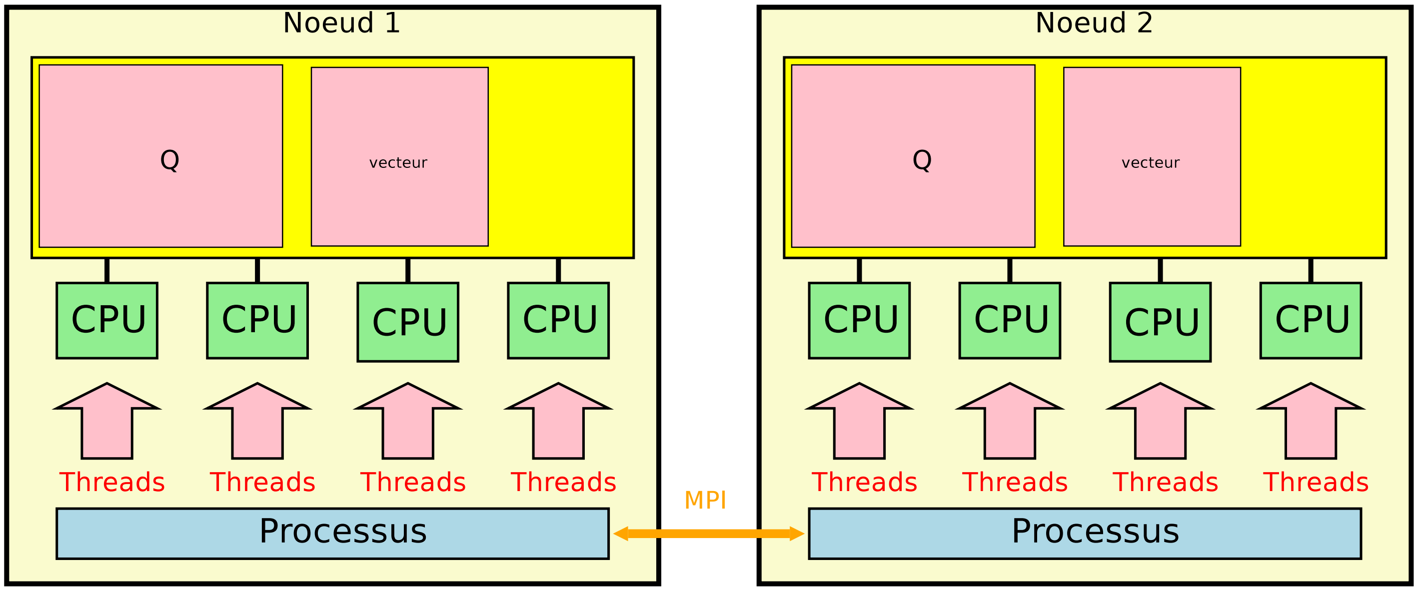
- Redistribution

$$x = (A^{(r-1)})^{-1}(y + T^{(r)} z) \quad \Longrightarrow \quad x = (\Lambda^{(r-1)})^{-1}(y + \Gamma^{(r)} z)$$

- Constraints
  - Storage of the full $nx \times nx$ complex matrix Q
  - Efficient matrix-vector products

Computer implementation

- *HERA* software (C++)
- BLAS routines
- complex *LR* subroutine
- Hybrid MPI (internode) / Multithreading pthread (intranode)



Numerical simulations (I)

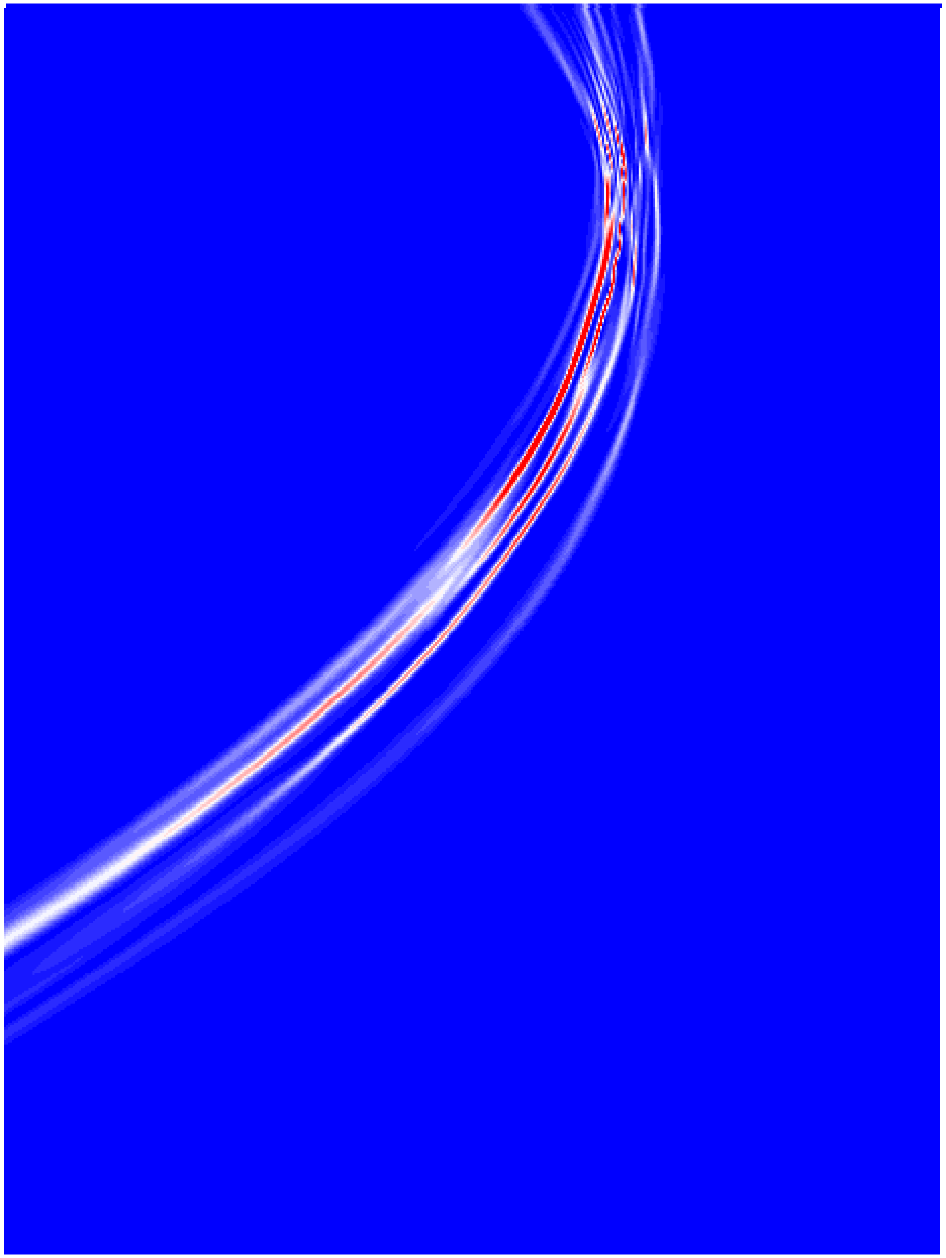
Discretization

- $L_x = 700\ \lambda_0$, $L_y = 1000\ \lambda_0$
- 10 points per wavelength in the Helmholtz zone
- 40 millions unknowns in the Helmholtz zone, 2.8 millions fluid unknowns.
- Density N_0 linear from 0.1 to 1 (critical density)

Solvers

- 128 processors
- 18.4s per GMRES iteration
- Elapsed time for the full simulation: 8 hours

Deflection of the laser beams (I)



@

Numerical simulations with a vertical plasma velocity (II)

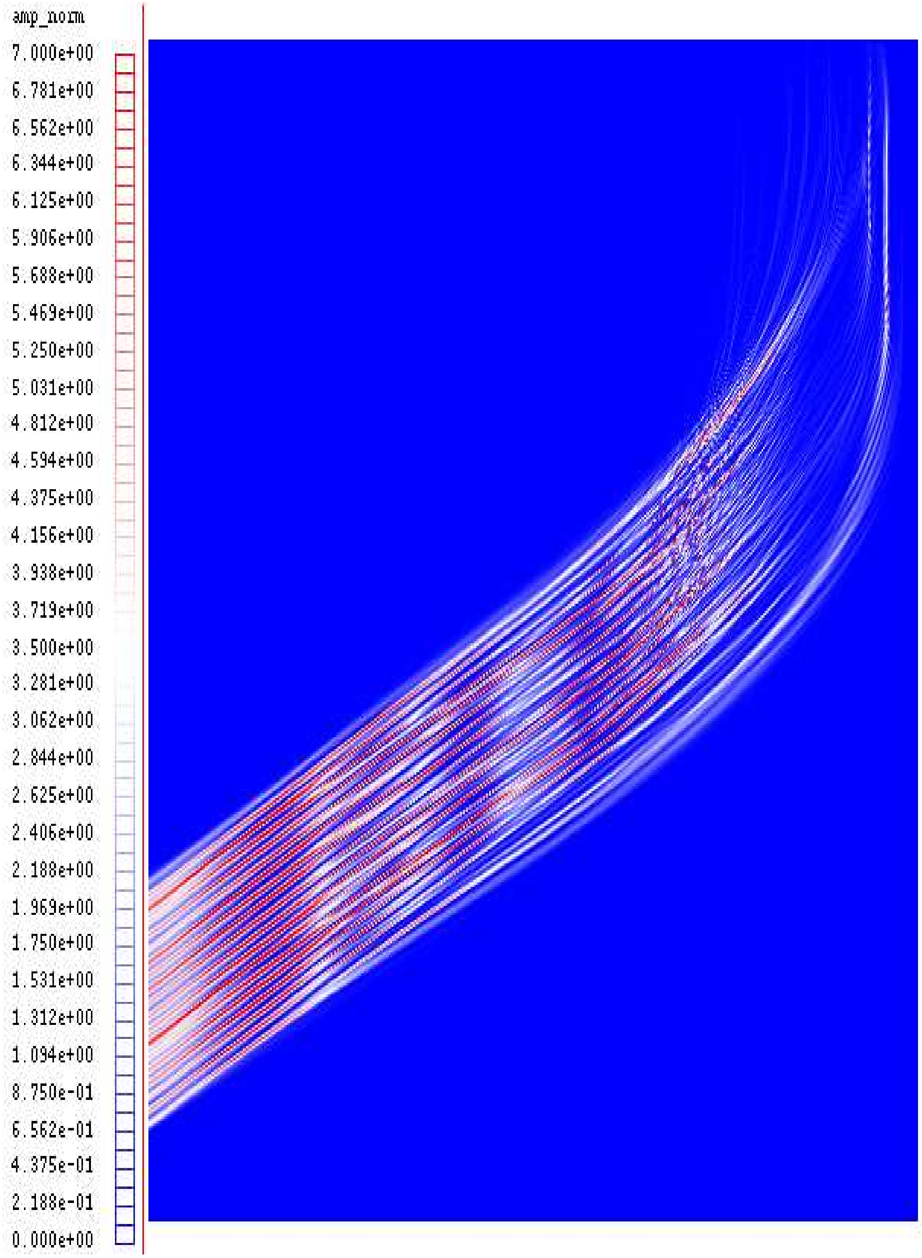
Discretization

- $L_x = 2000\ \lambda_0$, $L_y = 2000\ \lambda_0$
- 10 points per wavelength in the Helmholtz zone
- 200 millions unknowns in the Helmholtz zone, 16 millions fluid unknowns.
- Density N_0 linear from 0.1 to 1 (critical density)

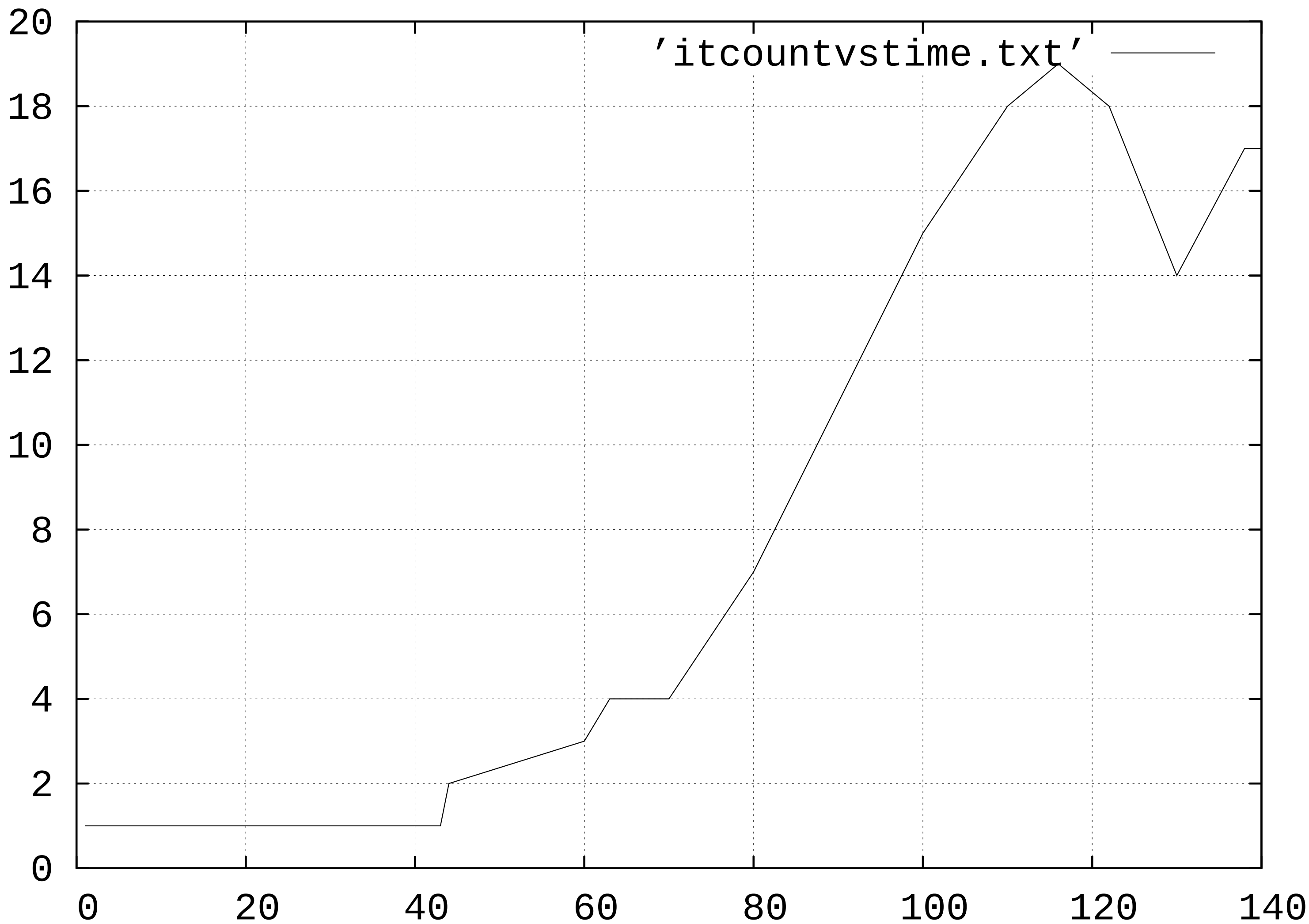
Solvers

- 256 processors
- 348s per GMRES iteration
- Laser simulated during a physical time of 11ps
- Elapsed time for the full simulation: 8 hours

Deflection of the laser beams (II)



Number of GMRES iterations vs. time



As time increases, δ_N increases and so the number of GMRES iterations.

CPU time per GMRES iteration

Fixed size problem: 40 millions unknowns

Nb Procs	16	32	64	128
CPU per GMRES iteration	$492s$	$249s$	$126s$	$64s$
Efficiency GMRES	1	0.987	0.976	0.96

Increased size problem: the number of points in both directions are doubled

Nb Procs	1	4	16	64	256
# d.o.f. $\times 10^6$	0.4	1.6	6.3	25.4	101.6
CPU LR	$1s$	$3s$	$12s$	$48s$	$189s$
CPU per GMRES iteration	$4.8s$	$11.6s$	$24s$	$47s$	$93s$

Conclusion and prospects

++ The goal is achieved: laser-plasma interaction with hundreds of millions of unknowns.

++ Paraxial/Helmholtz coupling

+ − Scalability in y but not in x

− − Number of GMRES iterations increases as time increases

Prospects

- More subdomains in order to

  be scalable in x (smaller matrices Q)

  use local averages for the density in the cyclic reduction (break the increase in the number of iterations as the time increases)

  take advantage of laser free zones which are dead zones (reduced CPU)

Thanks !