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

R. Sentis[†], S. Desroziers[†], F.Nataf [‡]

[†] CEA/Bruyeres, Service SEL, 91680 Bruyeres, France
[‡] Labo J-L-Lions, Université Paris VI, 75013 Paris, France

The aim of this work is to perform numerical simulations of the propagation of a laser beam in a plasma. At each time step, one has to solve a Helmholtz equation with variable coefficients in a domain which may contain more than hundred millions of cells.

One uses an iterative method of Krylov type to deal with this system. At each inner iteration, the preconditioning amounts essentially to solve a linear system which corresponds to the same five-diagonal symmetric non-hermitian matrix. If n_x and n_y denote the number of discretization points in each spatial direction, this matrix is block tri-diagonal and the diagonal blocks are equal to a square matrix A of dimension n_x which corresponds to the discretization form of a one-dimension wave operator. The corresponding linear system is solved by a block cyclic reduction method.

The crucial point is the product of a full square matrix Q of dimension n_x by a set of n_y vectors where Q corresponds to the basis of the n_x eigenvectors of the tri-diagonal symmetric matrix A. We show some results which are obtained on a parallel architecture. Simulations with 200 millions of cells have run on 200 processors and the results are presented.

1   Introduction

The numerical simulation of propagation of high power intensity lasers in a plasma is of importance for the "NIF project" in USA and "LMJ Facility project" in France. It is a very challenging area for scientific computing indeed the wave length $2\pi/k_0$ is equal to a fraction of one micron and the simulation domain has to be much larger than 500 microns. One knows that in a plasma the index of refraction is equal to $\sqrt{1 - N_e/N_c}$, where N_e is the electron plasma density and the critical density N_c is a constant depending only on the wave length. In macroscopic simulations (where the simulation lengths are in the order of some millimeters), geometrical optics models are used and numerical solutions are based on ray tracing methods. To into account more specific phenomena such as diffraction, autofocusing and filamentation, one generally uses models based on a paraxial approximation of the full wave equation (see for example [5], [1]

or, for a new approach in a tilted frame, [6]). But this approximation is valid only if the macroscopic index of refraction is a very smooth function, in such a way that the wave vector is quite constant in the simulation domain.

There are situations where the macroscopic variations of the plasma density N_e are not small. Particularly if one considers a laser beam propagating in a region near the critical density, it undergoes a total change of direction near a surface called *caustic surface* and the wave vector is strongly varying near this surface. So, the paraxial approximation is no more valid and one has to deal with a model based on a frequency wave equation (obtained by time envelope of the solution of the full Maxwell system).

Whatever propagation model is used, it is necessary to perform a coupling with the fluid dynamics system for modelling the plasma behavior. For a derivation of the models and a physical exposition of the phenomena under interest, see e.g. [13] or [7]. This paper is aiming at describing the numerical methods for solving the frequency wave equation. Notice that our simulation have been performed to take into account diffraction, refraction and auto-focusing phenomena but the Brillouin parametric instabilities which create laser backscattering are not taken into account up to now.

In the section 2, we describe the model based on the frequency wave equation. In this paper, only 2D problems are considered but the method may be extended to 3D computations. Denote by $\mathbf{x}$ the space variable and set $\mathbf{x} = (x, y)$ the two spatial coordinates. After time discretization, to find the laser field ψ at each time step, one has to solve a Helmholtz equation of the following form

$$\Delta\psi + \left(k_0^2(1 - N) + ik_0\mu_0\right)\psi = f \tag{1}$$

where f is a given complex function and μ_0 a real coefficient. We assume that the gradient of the macrocsopic non-dimension density $N(\mathbf{x}) = N_e/N_c$ is parallel to the x-axis, then we set

$$N(x, y) = N_0(x) + \delta N(x, y) \tag{2}$$

where N_0 depends on the x variable only and δN is small compared to 1. To solve accurately equation (1), one considers a spatial discretization of finite difference type with a spatial step equal to a fraction of the wave length. If n_x and n_y denote the number of discretization points in each direction, it leads to solve a the linear system with $n_x n_y$ degrees of freedom (which may be equal to 10^8 for a typical 2D spatial domain). One chooses an iterative method of Krylov type with a preconditioning which amounts to solve a linear system corresponding to a five-diagonal symmetric non-hermitian matrix A_I

$$A_I = \begin{pmatrix} \alpha + A/2 & -T & & & \\ -T & A & -T & & \\ & -T & A & -T & \dots \\ & & -T & A & \dots \end{pmatrix}$$

where T is equal to a constant times the identity matrix of dimension n_x, the matrix A of dimension n_x is equal to $2T$ plus the matrix of discretization of one-dimension Helmholtz operator and α is a constant.

Since the matrix A_I is separable, the corresponding linear system may be solved by the block cyclic reduction method. This method which is derived from the classical cyclic reduction method, has been used for instance in [12] for the numerical solution of Helmholtz problems, but the problem here is a more complicated, indeed one has to deal with Perfectly Matched Layers on two sides of the simulation domain.

The crucial point is the multiplication of a full square matrix Q of dimension n_x by the set of n_y vectors which are of length n_x, where Q corresponds to the basis of the n_x eigenvectors of the tri-diagonal matrix A.

In section 3, we describe the key points of the numerical method for solving (1). In section 4 we give some details on the parallel implementation ; for that purpose the processors are shared out according to horizontal slabs. In the last section we present numerical results in a small simulation domain with only 3 millions of cells and another case of 200 millions of cells which has run on 200 PEs.

2 The model and the boundary conditions.

The laser beam is characterized by an electromagnetic wave with a fixed pulsation ck_0 where c is the light speed and the wave length in vacuum is equal to $2\pi/k_0$. For modelling the laser, one considers the time envelope $\psi = \psi(t, \mathbf{x})$ of the transverse electric field. It is a slowly time varying complex function. On the other hand, for modelling the plasma behavior one introduces the non-dimension electron density $N = N(t, \mathbf{x})$ and the plasma velocity $\mathbf{U} = \mathbf{U}(t, \mathbf{x})$.

Modelling of the plasma. For the plasma, the simplest model is the following one. Let $P = P(N, \mathbf{x})$ a smooth function of the density N which may depend also of the position $\mathbf{x}$, according to the variation of the electron temperature. Then one has to solve the following barotropic Euler system :

$$\frac{\partial}{\partial t}N + \nabla(N\mathbf{U}) = 0, \tag{3}$$

$$\frac{\partial}{\partial t}(N\mathbf{U}) + \nabla(N\mathbf{U}\mathbf{U}) + \nabla(P(N)) = -N\gamma_p\nabla|\psi|^2. \tag{4}$$

The term $\gamma_p\nabla|\psi|^2$ corresponds to a ponderomotive force due to a laser pressure (the coefficient γ_p is a constant depending only on the ion species).

Modelling of the laser beam. The laser field $\psi = \psi(t, \mathbf{x})$ is a solution to the following frequency wave equation (which is of Schrödinger type)

$$2i\frac{1}{c}\frac{\partial}{\partial t}\psi + \frac{1}{k_0}\Delta\psi + k_0(1 - N)\psi + i\nu\psi = 0, \tag{5}$$

where the absorption coefficient ν depends on space and the density $N = N(t, \mathbf{x})$ is solution to the fluid system stated above. Of course, the problem is interesting only in the region where $N(t, \mathbf{x}) \leq 1$.

General framework. For the numerical solution of the fluid system, we use the method described in [8] or [1] which has been implemented in a parallel

platform called HERA. For solving (5), the spatial mesh has to be very fine, at least 10 cells per wave length in each direction. Generally the modulus $|\psi|$ of the electric field is slowly varying with respect to the spatial variable, one can use a crude mesh for the simulation of the Euler system (the mesh size has to be of order of the $2\pi/k_0$). If the modulus $|\psi|$ was not slowly varying in a region, one would have to solve in this region the Euler system with a fine mesh also.

So we handle a two-level mesh of finite difference type : in a 2D simulation, each cell of the fluid system is divided into $p_0 \times p_0$ cells for the Helmholtz level, with $p_0 = 10$ or 5. We assume in the whole paper that the hypothesis (2) holds, so it allows to perform a preconditioning of the global linear system by another system which is simpler since it does not take into account the perturbation $\delta N(x,y)$; this last system corresponds to a separable matrix and therefore a block cyclic reduction method may be used for its numerical solution.

Boundary conditions. The laser beam is assumed to enter in $x = 0$. Since the density N depends mainly on the $x-$variable, we may denote by N^{in} the mean value of the incoming density on the boundary and by N^{out} the mean value of the density on the outgoing boundary . Let $\mathbf{e}_b$ be the unit vector related to the direction of the incoming laser beam and set $\mathbf{K}^{in} = \mathbf{e}_b\sqrt{1 - N^{in}}$. The boundary condition on the part of the boundary $(x = 0)$ reads (with $\mathbf{n} = (-1, 0)$ the outwards normal to the boundary)

$$(k_0^{-1}\mathbf{n}.\nabla + i\mathbf{K}^{in}.\mathbf{n})(\psi - \alpha^{in}e^{ik_0\mathbf{K}^{in}\mathbf{x}}) = 0. \tag{6}$$

where $\alpha^{in} = \alpha^{in}(y)$ is a smooth function which is, roughly speaking, independent of the time. On the part of the boundary $x = x_{\max}$, there are two cases according to the value N^{out} :

i) If $N^{out} > 1$ the wave does not propagate up to the boundary and the boundary condition may read as $\partial\psi/\partial x = 0$.

ii) If $N^{out} \leq 1$ it is necessary to consider a transparent boundary condition. Here we take the simplest one, that is to say

$$(k_0^{-1}\mathbf{n}.\nabla + i\sqrt{1 - N^{out}})(\psi) = 0.$$

On the other hand, on the part of the boundary corresponding to $y = 0$ and $y = y_{\max}$, it is crucial to have a good transparent boundary condition, so we introduce perfectly matched layers (the P.M.L. of [2]). For the simple equation $-\Delta\psi - \omega^2\psi = f$, this technique amounts to replace in the neighborhood of the boundary, the operator $\frac{\partial}{\partial y}$ by $\left(1 + \frac{\sigma}{i\omega}\right)^{-1}\frac{\partial}{\partial y}$, where σ is a damping function which is not zero only on two or three wave lengths and which increases very fast up to the boundary. Notice that the feature of this method is that it is necessary to modify the discretization of the Laplace operator on a small zone near the boundary.

At each time step δt determined by the CFL criterion for the Euler system, one solves first the Euler system with the ponderomotive force and afterwards the frequency wave equation (5). For the time discretization of this equation, an implicit scheme is used. The length $c\delta t$ is very large compared to the spatial step

therefore the time derivative term may be considered as a perturbation and one has to solve the following equation of the Helmholtz type

$$\Delta\psi + \left(k_0^2(1 - N) + ik_0(\mu_0 + \nu)\right)\psi = i\mu_0\psi^{ini} \tag{7}$$

where $\mu_0 = 2k_0/(c\delta t)$. The boundary conditions are the same as above.

3   Principle of the Numerical methods for the Helmholtz equation

The spatial discretization (7) is the classical one of finite difference type. Denote by n_x and n_y the number of discretization points in each direction.

Beside the interior domain, there are two zones corresponding to the two PMLs near the boundary $y = 0$ and $y = y_{max}$, the width of these layers corresponds to $2p_0$ points. Then the linear system to be solved has the following form

$$\begin{pmatrix} P_1 & C_1 & 0 \\ E_1 & A_I + D & E_2 \\ 0 & C_2 & P_2 \end{pmatrix} \Psi = F \tag{8}$$

where P_1 and P_2 are square matrices whose dimension is $2p_0 n_x$, it corresponds to the discretization of the equation in the P.M.L. On the other hand C_i, E_i are coupling matrices (whose dimensions are $n_x n_y$, times $2p_0 n_x$). The square matrix A_I whose dimension is $n_x n_y$, corresponds to the discretization of

$$(\Delta + k_0^2(1 - N_0) + ik_0\mu_0)\bullet$$

in the interior domain. Moreover, D is a diagonal matrix corresponding to the terms $\delta N(x, y) + ik_0\nu(x, y)$. Notice that the domain decomposition method is used with Robin interface conditions (see [10], [3]) which corresponds to a discretization of the condition on the interfaces between subdomains

$$\frac{\partial}{\partial n}\psi + \alpha\psi = \frac{\partial}{\partial n}\psi^{neib} + \alpha\psi^{neib}$$

(α is a complex parameter and ψ^{neib} is the value in the other subdomain).

3.1   Solution of the linear system

An iterative Krylov method is used to solve (8) : the GMRES method without restarting seems to be the best choice ; the preconditioning is performed by solving the linear system based on A_I in the interior domain and on P_1 and P_2 in the P.M.L. domains, that is to say the main point is to solve as fast as possible a system of the following form

$$\mathcal{P}U = f, \quad \text{where} \quad \mathcal{P} = \begin{pmatrix} P_1 & 0 & 0 \\ 0 & A_I & 0 \\ 0 & 0 & P_2 \end{pmatrix} \tag{9}$$

Matrices P_1 and P_2 may be easily factorized in the standard LU product. The 5-diagonal matrix A_I is very large but separable and symmetric non-hermitian. It amounts to the folowing system

$$
\begin{pmatrix}
B & -T & & & \\
-T & A & -T & & \\
& \ddots & \ddots & \ddots & \\
& & -T & A & -T \\
& & & -T & B
\end{pmatrix}
\begin{pmatrix}
u_1 \\
u_2 \\
\vdots \\
u_{n_y-1} \\
u_{n_y}
\end{pmatrix}
=
\begin{pmatrix}
f_1 \\
f_2 \\
\vdots \\
f_{n_y-1} \\
f_{n_y}
\end{pmatrix}
\tag{10}
$$

where the elements u_m and f_m are n_x-vectors, T is equal to a constant times the identity matrix, $B = A/2 + \alpha I$ and A is a tri-diagonal matrix of dimension n_x equal to $2T$ plus the matrix of discretization of the operator

$$
(\frac{\partial^2}{\partial x^2} + k_0^2(1 - N_0) + ik_0\mu_0) \bullet .
$$

3.2 The cyclic reduction method

To solve the system (10) in the central domain, we use the block cyclic reduction method. Let us recall the principle of this method. For the sake of simplicity, assume $n_y = 2^k - 1$. We know that A and T are commutative. Consider 3 successive lines of (10) for $i = 2, 4, ..., n_y - 1$:

$$
\begin{cases}
-Tu_{i-2} + Au_{i-1} - Tu_i & = f_{i-1} \\
\quad - Tu_{i-1} + Au_i - Tu_{i+1} & = f_i \\
\quad\quad - Tu_i + Au_{i+1} - Tu_{i+2} & = f_{i+1}.
\end{cases}
\tag{11}
$$

After a linear combination of these lines, we get :

$$
-T^2A^{-1}u_{i-2} + \left(A - 2T^2A^{-1}\right)u_i - T^2A^{-1}u_{i+2} = f_i + TA^{-1}\left(f_{i-1} + f_{i+1}\right) \tag{12}
$$

After this first step, the elimination procedure may be performed again by induction. That is to say, denote $A^{(0)} = A$, $B^{(0)} = B$, $T^{(0)} = T$ and $f^{(0)} = f$; after r elimination steps, the reduced system for $0 \le r \le k - 1$ owns $2^{k-r} - 1$ blocs and reads as:

$$
\begin{pmatrix}
B^{(r)} & -T^{(r)} & & & \\
-T^{(r)} & A^{(r)} & -T^{(r)} & & \\
& \ddots & \ddots & \ddots & \\
& & -T^{(r)} & A^{(r)} & -T^{(r)} \\
& & & -T^{(r)} & B^{(r)}
\end{pmatrix}
\begin{pmatrix}
u_{2^r} \\
u_{2.2^r} \\
\vdots \\
u_{(n_y-1)-2^r+1} \\
u_{n_y-2^r+1}
\end{pmatrix}
=
\begin{pmatrix}
f_{2^r}^{(r)} \\
f_{2.2^r}^{(r)} \\
\vdots \\
f_{(n_y-1)-2^r+1}^{(r)} \\
f_{n_y-2^r+1}^{(r)}
\end{pmatrix}
$$

where for $r = 1, ..., k - 2$:

$$A^{(r)} = A^{(r-1)} - 2 \left(T^{(r-1)} \right)^2 \left(A^{(r-1)} \right)^{-1}$$

$$B^{(r)} = A^{(r-1)} - \left(T^{(r-1)} \right)^2 \left(\left(A^{(r-1)} \right)^{-1} + \left(B^{(r-1)} \right)^{-1} \right) \qquad (13)$$

$$T^{(r)} = \left(T^{(r-1)} \right)^2 \left(A^{(r-1)} \right)^{-1}$$

For the right hand side, we get the induction formula :

$$f_{i.2^r}^{(r)} = f_{i.2^r}^{(r-1)} + T^{(r-1)} \left(A^{(r-1)} \right)^{-1} \left(f_{i.2^r - 2^{r-1}}^{(r-1)} + f_{i.2^r + 2^{r-1}}^{(r-1)} \right) \qquad (14)$$

After all the elimination steps, it remains only one equation for finding $u_{2^{k-1}}$. Once this value is obtained, one deduces all the other values step by step by induction.

4   Parallel implementation

Notice first that $A = ik_0\mu_0 + A^0$ where A^0 is a symmetric tri-diagonal matrix whose coefficients are real except the one in the first line and the first column (due to the boundary condition (6)). We have checked that it is possible to find a basis of eigenvectors of A^0 which are orthogonal for the pseudo scalar product $< u, v >= u^T.v$. They are computed by using the algorithm of Parlett (cf. [11]) although it was designed for Hermitian matrices. So denote Q the matrix whose columns are the eigenvectors of A^0 , the matrix Q is orthonormal for the pseudo scalar product, that is to say

$$QQ^T = Q^TQ = I$$

Since T is the identity matrix up to a multiplicative constant, one can introduce the diagonal matrices $\Lambda^{(0)}$ and $\Gamma^{(0)}$

$$A = Q\Lambda^{(0)}Q^T, \quad T = Q\Gamma^{(0)}Q^T. \qquad (15)$$

So we get
$$A^{(r)} = Q\Lambda^{(r)}Q^T, \quad T^{(r)} = Q\Gamma^{(r)}Q^T \qquad (16)$$

with the following induction formulas

$$\Lambda^{(r)} = \Lambda^{(r-1)} - 2 \left(\Gamma^{(r-1)} \right)^2 \left(\Lambda^{(r-1)} \right)^{-1}, \quad \Gamma^{(r)} = \left(\Gamma^{(r-1)} \right)^2 \left(\Lambda^{(r-1)} \right)^{-1} \qquad (17)$$

Let us summarize the algorithm
• Introduce the vectors $\tilde{f}_i$ transformed of f_i in the eigenvector basis

$$\tilde{f}_i = Q^T f_i \quad \text{for} \quad i = 1, \ldots, n_y.$$

• At each step r, the vector $\tilde{f}^r{}_i$ transformed of $f^r{}_i$ of the right hand side, reads

$$\tilde{f}_{i.2^r}^{(r)} = \tilde{f}_{i.2^r}^{(r-1)} + \Gamma^{(r-1)} \left(\Lambda^{(r-1)}\right)^{-1} \left(\tilde{f}_{i.2^r-2^{r-1}}^{(r-1)} + \tilde{f}_{i.2^r+2^{r-1}}^{(r-1)}\right)$$

• One computes the vectors $\tilde{u}_{2^{k-1}}$ by solving

$$\Lambda^{(k-1)}\tilde{u}_{2^{k-1}} = \tilde{f}_{2^{k-1}}^{(k-1)}$$

• One recursively distributes the solutions by solving sub-systems of the following type

$$\Lambda^{(r)}\tilde{u}_{j.2^{r+1}-2^r} = \tilde{g}_{j.2^{r+1}-2^r}^{(r)}$$

$$\text{where} \quad \tilde{g}_{j.2^{r+1}-2^r}^{(r)} = \tilde{f}_{j.2^{r+1}-2^r}^{(r)} + \Gamma^{(r)} \left(\tilde{u}_{(j-1).2^{r+1}} + \tilde{u}_{(j).2^{r+1}}\right)$$

• Lastly, the solution u is given by

$$u_i = Q\tilde{u}_i \text{ pour } i = 1, \ldots, n_y.$$

For the parallel implementation, the processors are shared out according to horizontal slabs. The crucial point is the product of a full matrix Q of dimension $n_x \times n_x$ by the set of n_y vectors. Multi-thread techniques are used to deal with this point. On each node of the architecture, there are 4 processors. So each node is devoted to 4 subdomains, the matrix Q is stored on the local memory of the node and the products of the matrix by the vectors are performed simultaneously by the 4 processors.

Scalability. The code has run on a massively parallel architecture with HP-Compaq processors of the EV67 type. For a typical problem, with $40\ 10^6$ complex unknowns, when the CPU time for one Krilov iteration is equal to 1 with 16 processors, it is equal to 0.98 with 32 processors and 0.96 with 64 processors, so the efficiency of the parallelism is very good.

On the other hand, consider now problems whose size is multiplied by 2 in each direction. When the number of degrees of freedom is $n_x n_y = 1.6\ 10^6$, the CPU time with 4 processors is equal to 1 for one Krilov iteration, it is equal to 2.1 with 16 processors for $n_x n_y = 6.4\ 10^6$, and it is equal to 4.2 with 32 processors for $n_x n_y = 25.6\ 10^6$. That is to say the CPU is about two time larger when the number of processors and the number of degrees of freedom are 4 times larger ; this is coherent with the fact that the number of operations for the cyclic reduction method grows like $n_x^2 n_y$.

5   Numerical results.

The incoming boundary condition is roughly speaking equal to a sum of narrow Gaussian functions depending of the y variable ; the half height width of each Gaussian function is equal to 8 wave lengths and is assumed to describe a speckle (a speckle is a light spot of high intensity). One considers first a simulation domain of 100×300 wave lengths ; the initial profile of density is a linear function

increasing from 0.1 at $x = 0$ to 1.1 at $x = x_{max}$. The incoming boundary condition consists in three speckles with the same incidence angle. At the Helmholtz level, one handles only 3 millions of cells. With 32 PEs, the CPU time is about 20 seconds per time step for approximately 10 Krylov iterations at each time step. Without the coupling with the plasma, it is well known that the solution is very close to the one given by the geometrical optics ; the speckles propagate in a parallel way, undergo macroscopic refraction when the electron density increases and are tangent to a caustic line (here it is the line corresponding to $x = x_\star$ such that $N_0(x_\star) = \cos^2(\theta)$, where θ is the incidence angle of the speckles). With our model, if the laser intensity is low (which corresponds to a weak coupling with the plasma), one notices that a small digging of the plasma density occurs. This digging is more significant when the laser intensity is larger, then an autofocusing phenomenon takes place. On figure 1, one sees the map of the laser intensity that is to say the quantity $|\psi|^2$, which corresponds to this situation after some picoseconds, knowing that the time step in about 0.05 picosecond. We notice here that the speckles undergo autofocusing phenomena and some filamentation may be observed.
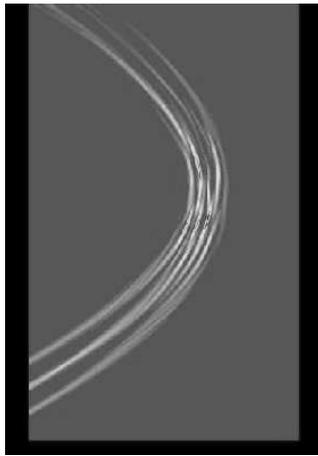
Fig. 1. Laser intensity at time 3 ps. Incoming boundary condition with 3 speckles

Another case will be presented, corresponding to simulation domain of 2000×2000 wave lengths. In the left half of the domain, the electron density is constant and equal to 0.15 and one uses the paraxial model ; in the right half of the domain the density increases from 0.15 to 0.95 and the Helmholtz model is used. The incoming boundary condition consists in 20 speckles with various intensity. In this domain, one handles 200 millions of cells (whose size is 1/10 of the wave length) and the simulation have run on 200 PEs. The map of the laser intensity is shown on figure 2 after 22 ps (the time step is roughly equal to 0.02 ps). We have chosen a small absorption coefficient $\nu = 2.10^{-5} N^2$ so the problem is

quite sharp. Here the digging of the plasma is locally very important since the variation of density δN reaches 0.05 in a region where $N(x) = 0.8$, see a map of the non-dimension density on the figure 3. About 13 iterations of the Krilov method are enough to converge. The CPU time is equal to 240 s for solving the full linear system and 270 s for each time step (including the solution of the LR method and the hydrodynamics of the plasma).
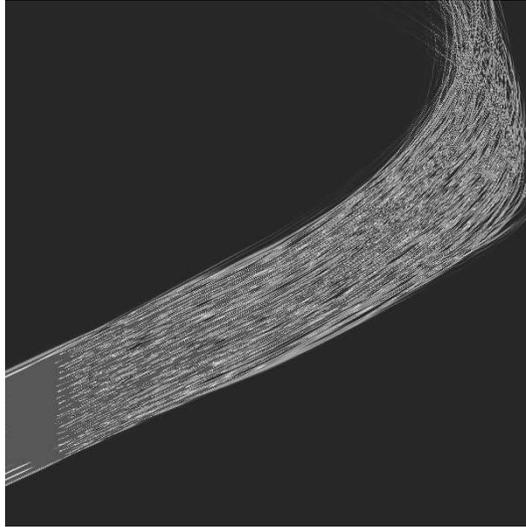
Fig. 2. Laser intensity at time 22 ps. Incoming condition with 20 speckles.

Conclusion.

In the framework of the hydrodynamics parallel platform HERA, we have developed a solver for the laser propagation based on the frequency wave equation. The assumption that the density N depends mainly on the $x-$variable only allows to perform a preconditioning by a domain decomposition method (two PMLs and a large Helmholtz zone) where the linear system corresponding to the Helmholtz zone is solved by the block cyclic reduction method. This kind of simulation is new. Up to our knowledge, the solution of this kind of model in a wide two-dimension domain has been published only in [9], but the framework is different : the gradient of the electron density is 20 times more larger and the simulation domain is 100 times smaller than in our problem.

Most of the computer time is spent by the product of the full matrix Q by a set a n_y vectors. In the future some CPU time may be saved if inside the inner iteration loop of the Krylov method, we do not consider all the spatial domain that is to say all the n_y vectors but only the vectors which does not belong to some subinterval $[n_y^1, n_y^2]$ for instance the ones where the solution varies very few from an iteration to the other.
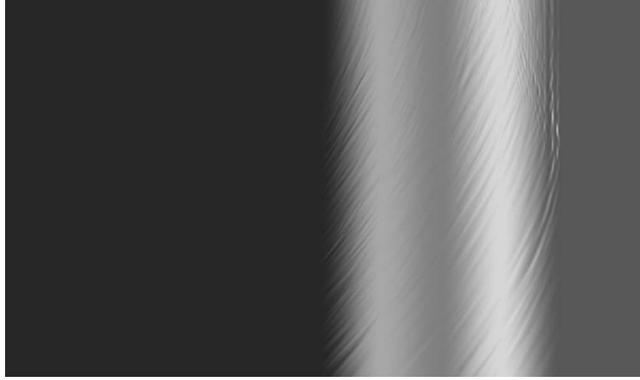
Fig. 3. Plasma density at time 22 ps (zoom). Incoming condition with 20 speckles.

References

1. Ph. Ballereau, M.Casanova, F.Duboc, D. Dureau, H.Jourdren, P.Loiseau, J.Metral, O.Morice, R. Sentis. *Coupling Hydrodynamics with a Paraxial solver,* to appear.
2. J.-P. Berenger. A Perfectly Matched Layer for the Absorption of Electromagnetic Waves. *J. Comp. Physics* 114, p185-200 (1994).
3. B. Després, Domain decomposition method and the Helmholtz problem. II, in: *Second International Conference on Mathematical and Numerical Aspects of Wave Propagation (Newark, DE, 1993)*, SIAM, (1993).
4. S. Desroziers *Modelisation de la propagation laser par résolution de l'équation d'Helmholtz*, Ph. D. dissertation, University Paris VI, (2006).
5. M.R. Dorr , F.X. Garaizar, J.A. Hittinger, *Simuation of laser-Plasma filamentation.* J. Comp. Phys. **17**, p233-263 (2002).
6. M. Doumic, F. Golse, R. Sentis. *Propagation laser paraxiale en coordonnées obliques,* Note C. R. Ac. Sciences, Paris, série I, t.**336**, p.23-28 (2003).
7. S. Hüller, Ph. Mounaix, V.T. Tikhonchuk, D. Pesme. *Interaction of two neighboring laser beams,* Phys. Plasmas, **4**, p.2670-2680, (1997).
8. H. Jourdren. HERA hydrodynamics AMR Plateform for multiphysics simulation, *Proceedings of Chicago workshop on AMR methods,* Springer Verlag, Berlin (2004).
9. Maximov A.V. and al Modeling of stimulated Brillouin .. Phys Plasmaa **11**, p.2994-3000, (2004).
10. P.-L. Lions, On the Schwarz alternating method, III. in: *Third International Symposium on Domain Decomposition Methods for PDE*, Chan and al., ed., SIAM (1990).
11. B. N.Parlett *: Acta Numerica,* p 459-491, (1995).
12. T. Rossi, J. Toivanen. *A parallel fast direct solver for block tridiagonal systems with separable matrices*, SIAM J. Sci. Comput. **20** (1999), pp. 1778-1796.
13. R. Sentis, *Mathematical Models for Laser-Plasma Interaction,* ESAIM: Math. Modelling and Num. Analysis, **39**, p275-318 (2005).