

Parallel Algorithm for Filamentation of High-Power Super-Short Laser Pulses

Svyatoslav Shlenov and Alexei Bezborodov
M.V. Lomonosov Moscow State University
Physics Faculty and International Laser Center
Leninskie Gori, MSU, 119992 Moscow GSP-2, Russia
shlenov@phys.msu.ru

Andrei Smirnov
West Virginia University
Morgantown, WV
andrei.smirnov@mail.wvu.edu

Abstract

An efficient parallel algorithm is proposed for solving the filamentation problem of high-power femtosecond laser pulses in gases. It was demonstrated that the algorithm possesses the property of coarse-grained parallelism and has a good scale-up characteristics when used on workstation clusters.

Keywords: Parallel Algorithms, Nonlinear Optics, Ultra-Short Light Pulses, Laser Physics

1 Introduction

Self-focusing of tera- and sub-terawatt femtosecond pulses in transparent media, such as the atmosphere, can commonly occur due to the effect of Kerr nonlinearity [1]. When the nonlinear focusing intensity reaches 10^{13} W/cm^2 the competing effect of de-focusing due to multiphoton ionization becomes evident. Combined action of these effects leads to the appearance of long filament with a relatively high energy concentration [2, 3, 4]. A many-fold excess of the critical self-focusing power, P_{cr} , leads to the appearance of multiple filaments in a single light pulse (Fig.1), which happens spontaneously in the turbulent atmosphere.

A characteristic feature of the problem is a large range of space-scales, where the pulse filamentation takes place. For instance, the cross-sectional size of a typical beam at the exit of the laser system can exceed 1 cm [5], whereas the transversal size of the filament is usually less than 100μ . Another peculiarity is that the process of plasma generation is induced by the field of the femtosecond pulse itself, i.e. the appearance of long fila-

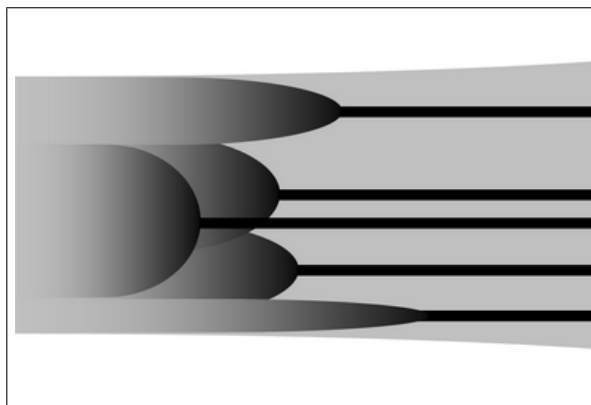


Figure 1: Illustration to *filamentation* phenomenon. In a wide laser pulse very narrow ($d_0 \sim 20 - 100 \mu$) and very long (several meters and greater) filaments are formed.

ments in the pulse and the generation of free electrons happen coherently. The concentration of plasma is growing during the pulse which makes the problem inherently time-dependent. Numerical modelling of the filamentation process under these conditions requires either using of a rough quasi-steady model of pulse propagation, or employing high-performance computing facilities [6, 7].

In the current study a parallel algorithm for modeling plasma filamentation is proposed. The method was implemented in a numerical algorithm and tested on several computer clusters.

2 Method

A physical process of the light pulse propagation can be described as follows. At the front of a powerful femtosecond pulse the critical power is reached, and subsequent "layers" of the pulse experience self-focusing (Fig.1). The threshold intensity of multiphoton ionization is reached in the region of nonlinear focusing and causes the appearance of free electrons. These electrons accumulate during the pulse action and lead to the formation of a dynamic plasma channel of a complex shape.

A mathematical model for the analysis of the filamentation of powerful femtosecond laser pulses is based on a nonlinear Schrödinger-like equation for the slowly varying envelope $E(x, y, z, t)$ of the linearly polarized laser pulse [8]:

$$2ik \frac{\partial E}{\partial z} = \frac{\partial^2 E}{\partial x^2} + \frac{\partial^2 E}{\partial y^2} + \frac{2k^2}{n_0} (n_2 |E|^2 + \Delta n_p + \Delta \tilde{n}) E \quad (1)$$

where k is the wave number, n_0 is the refraction index of an unperturbed medium, n_2 is the Kerr nonlinearity coefficient, which for air is about $n_2 = 1.75 \times 10^{-19} \text{cm}^2/W$. The addition to the refraction index Δn_p , i.e. the contribution of the self-induced plasma, can be estimated as

$$\Delta n_p = -\frac{n_0 \omega_p^2}{2\omega^2}$$

where $\omega_p = \sqrt{4\pi e^2 N_e / m_e}$ is the plasma frequency, ω is the light frequency, e , m_e are the electron charge and mass respectively. The concentration of the free electrons in plasma, N_e , obeys the relation:

$$\frac{\partial N_e}{\partial t} = R (|E|^2) (N_0 - N_e) \quad (2)$$

where drift of the electrons is ignored on the femtosecond time scale of the pulse, R is the multi-photon ionization rate of air molecules (mainly oxygen and nitrogen), which is a function of instantaneous intensity, $|E|^2$, and N_0 is the concentration of the neutral molecules.

To complete the picture we included into (1) the linear addition $\Delta \tilde{n}$ to the refraction index n accounting for the atmospheric turbulence [9]. It should be noted that for the femtosecond pulse propagation through a long atmospheric path the dispersion becomes an important factor

and equation (1) takes a more complex form, which then should include the second time derivative [10, 11].

Equations (1), and (2) should be complemented with the initial conditions:

$$E(x, y, z = 0) = E_0(x, y, t)$$

$$N_e(x, y, z, t = -\infty) = 0$$

which correspond to the given pulse $E_0(x, y, t)$ at the entry to the medium ($z=0$) and the absence of free electrons in front of the pulse.

Numerical solution is done by discretizing the equation on a grid in a 4D space-time. There are two first order partial derivatives of unknown functions E and N_e over z and t coordinates in the equation system (1, 2). In finite differencing approximation one can say that the values of these functions on the z -planes z_i , and temporal slices t_j depend only on the values in the neighboring planes z_{i-1} , and temporal slices t_{j-1} .

The size of the grid in the xy cross-section, L_x , should several times exceed the radius of the beam, $a_0 = 1 \text{ cm}$, and the grid-cell size, Δx , should be much smaller than the diameter of the filament, $d_0 = 50 \mu$. Thus, the estimate of the required number of grid nodes, $N_x = L_x / \Delta x$, along the transversal coordinate is on the order of $N_x = 2^{13} = 8192$.

The upper estimate for the grid cell size, Δz , along the axial coordinate, z , can be taken as the diffraction length on the transversal scale of the filament. This estimate for the fundamental wavelength of Ti:sapphire laser $\lambda = 800 \text{ nm}$ gives the value $\Delta z = \pi d_0^2 / 2\lambda = 0.5 \text{ cm}$. As a result the number of the grid-nodes in the axial direction, N_z , over the light path of $L = 10 \text{ m}$ should be at least $N_z = 2000$, giving total number of space nodes of about $N = N_x N_y N_z = 1.3 \times 10^{11}$ (we consider $N_x = N_y$), and about $N_t = 10^3$ time steps.

These estimates show that to store the concentration of electrons N_e at the grid nodes about 500 GB of RAM will be needed. This will require the usage of distributed computing systems and parallel algorithms [12].

In our numerical algorithm we used weak nonlinearity approximation and split-step method to solve equation (1). According to this method on each integration step along z the following set of equations was solved:

$$2ik \frac{\partial E}{\partial z} = \frac{\partial^2 E}{\partial x^2} + \frac{\partial^2 E}{\partial y^2} E \quad (3)$$

and

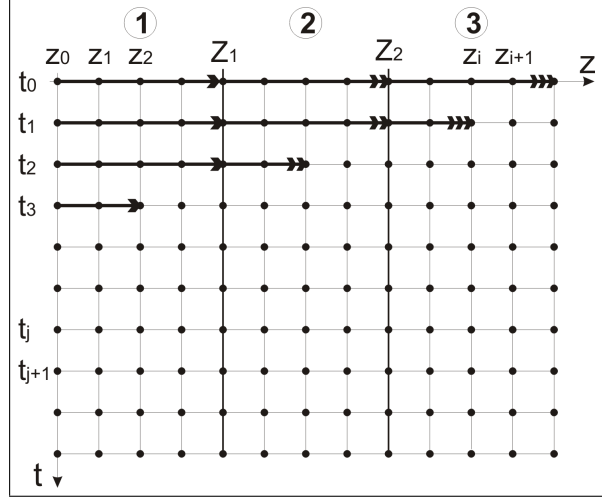
$$2ik \frac{\partial E}{\partial z} = \frac{2k^2}{n_0} (n_2 |E^2| + \Delta n_p + \Delta \tilde{n}) E \quad (4)$$

coupled with equation (2).

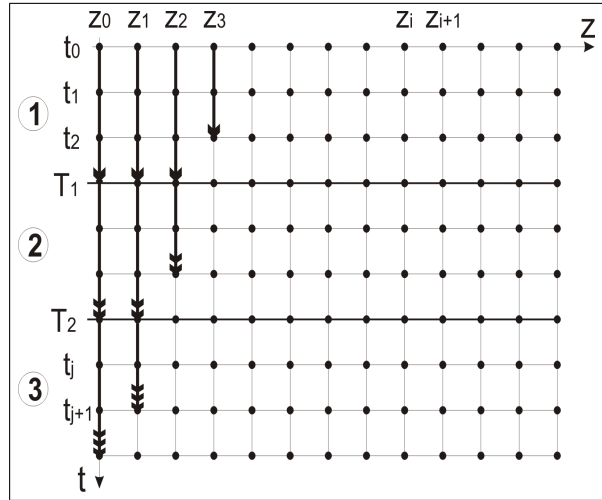
Figure 2 shows the computational grid in $z - t$ coordinates for the solution of (1, 2). Every point on the figure represents in reality a set of nodes in $x - y$ cross-section. The initial conditions for the field E are given on the plane $z = 0$. There is no plasma before the passage of the pulse front: $N_e(t_0) = 0$. The explicit iterative scheme loops over all the grid-nodes along z and t . Looping can be done both by rows and columns of the grid. This reflects the existence of two primary progress coordinates in this problem: z and t . In the first case (Fig.2(a)), the external is the time cycle over t , and the looping over the nodes in $x - y$ cross-section is done by rows. In this case, to compute the plasma concentration $N_e(z_i, t_j)$, one has to store its values $N_e(z_i, t_{j-1})$ from the preceding time level, t_{j-1} . In the second case (Fig.2(b)) the outer loop is over z and the inside looping is done by columns. In this approach one has to keep the values of a complex field $E(z_i, t_j)$ from the previous plane: $E(z_{i-1}, t_j)$.

Possible methods of parallelization are also shown in Fig.2. Let's consider one of them in Fig.2(b). A cluster-node is schematically represented by a number in circle (there are total of $K = 3$ nodes). Every node is assigned its *zone of influence*, i.e. the set of subsequent planes z_i , where the solution is obtained. The direction of the looping is shown by arrows. At the boundaries of the zones of influence at the points where the arrows are broken the exchange of information with other cluster nodes takes place. The set of all these points forms the exchange planes, Z_k .

The first node is assigned a zone between z_0 and the first exchange plane Z_1 . The algorithm starts with the zero-time level, t_0 , and the first cluster-node performs calculations of the zero temporal pulse slice propagation along the z -coordinate. The obtained plasma concentration values are stored in memory. After reaching the exchange plane, Z_1 , the first computing node passes the values of the complex field, E , to the second one, which in turn uses these data as initial conditions to proceed with the computation of the propagation of the same temporal slice in its own zone of influence until the next exchange plane, Z_2 . Meanwhile the first node continues the solution on the next temporal slice, using the information on plasma concentration, N_e , obtained from the compu-



(a) Each node processes a sub-set of spatial planes, z_i . Z_1, Z_2 are the inter-node exchange boundaries.



(b) Each node processes a sub-set of time-slices, t_j . T_1, T_2 are the inter-node exchange boundaries.

Figure 2: Two computational grids for the parallel execution on the cluster with 3 computing nodes. Arrows show the seep direction of the grid nodes as performed by the algorithm. Different cluster nodes are marked by numbers in circles.

tations on the previous level.

Similarly, after reaching the exchange plane Z_2 , the second computing node transfers the values of the field E to the third node, while the first two nodes continue the calculations on the next time levels, etc. After the data have been passed to the last computing node through the exchange plane Z_{K-1} , all cluster nodes are fully engaged in the parallel calculations in a manner shown in Fig.2(a). The needed memory for the task is uniformly distributed among the cluster nodes. Since the number of nodes, K , in a cluster is as a rule less than the number of z -steps, N_z , the parallelism of the problem is expected to be of a "coarse-grained" type, and the data exchange will not significantly impact the efficiency of the computations.

3 Results

To verify the proposed approach a series of tests were conducted on the solution of the problem formulated in system (1, 2) with $\Delta\tilde{n} = 0$ on several clusters and different networks. Table 1 shows the characteristics of the computational facilities on various clusters.

On cluster "D" the logical node in terms of MPI coincides with the physical two-processor node. On all other clusters the MPI communication library is configured in a way, that every processor is a logical node, occupying half of the memory of the physical node. The most powerful cluster, *A*, consists of 160 logical nodes with 2GB RAM each.

Let's consider the first of the parallel algorithms described above (Fig.2(a)). In the cluster tests up to 8 nodes were used, with each node processing the domain of 9 z -planes. The degree of coarseness of the problem was determined by the communication overhead. To determine the latter two time intervals were collected: τ_c - the computational time spent by each node between two communication events, and τ_e - the time spent on data exchange between the nodes. The time interval τ_e includes both time spent on data transfer and idling time spent on waiting for a communication event, i.e. this is a total time spent by a node between the end of a calculation on one time slice and beginning of a calculation on another time slice. The time intervals were determined using the `times()` function of standard library. The results were averaged over all nodes, and for each node over all time levels t_j , totaling M realizations.

For comparison we performed experiments on grids of

different size, ranging from $N_x N_y = 1024 \times 1024$ to $N_x N_y = 8192 \times 8192$. Each inter-node communication required the exchange of $8N_x N_y$ bytes.

The results of the numerical tests – the averages of times τ_c and τ_e and their standard errors, Err , are summarized in Table 2. The number of realizations, M , was varying from 700 to 800 for clusters *A-C* and from 63 to 225 for cluster *D*.

Analysis of the results shows that the time of data transfer, τ_e , on each cluster constitutes only a small fraction of the computational time, τ_c . The worst result percent-wise was demonstrated by cluster *B*, where the ratio τ_e/τ_c exceeded 10% for all grids considered. It can be attributed to the slowest network realized on that cluster (Fast Ethernet). The best result was shown on cluster *C*, where the communication overhead was not greater than 1.5% for all N_x . This cluster, as well as cluster *A* had obviously the fastest communication network. Since the most powerful cluster, *A*, had better value of interval τ_c , the ratio τ_e/τ_c for this cluster turned out a little bit worse – 4%-6% for the same set of N_x and it reached as much as 8% for the largest grid with $N_x = 8192$.

We should note that the greatest contribution to the computing time, τ_c , is made by the solution of the diffraction equation (3), which was done in the spectral space. The number of fast Fourier transform (FFT) operations realized in the solver is proportional to $N_x N_y \log(N_x N_y)$. It can be seen from the data of Table 2 that with the doubling of the grid sizes N_x and N_y the computational time τ_c indeed increases somewhat more than four times. It should also be noted that despite good characteristics of cluster *D*, the computational time τ_c happened to be the longest. It can possibly be the result of non-optimal cluster tuning.

4 Conclusions

The set of tests conducted on different clusters has shown that the proposed algorithm for the solution of femtosecond pulse filamentation problem is indeed of a coarse grained nature. The time of inter-node data exchange takes only a small fraction of the computational time. The exceptions can be related to the slow networks. Thus, with the grows of the number of cluster nodes used a near linear scaleup can be achieved. This makes a full-scale numerical simulation of femtosecond pulse multifilamentation on a cluster with a few hundred processors feasible.

It can be argued on the latter that with the increase in the number of nodes a communication "saturation" effect can be observed, since the parallelization algorithm involves the simultaneous data exchange between all neighboring nodes. However, the obtained results indicate that in this case the waiting time for communication completion becomes much less than the computational time. An additional argument in favor of this is the fact, that the results on the A cluster with total 160 nodes were obtained under the realistic conditions, when the resources of the network were engaged by other tasks. The results also showed a good repeatability at different times.

It should be noted that the current parallelization strategy can be easily adapted to the heterogeneous clusters, or computational grid environments with the nodes of varying performance. Such nodes should be assigned proportionally smaller zone of influence, so as to preserve the same execution time per grid-node, and to avoid idling of high performance nodes.

References

- [1] Govind Agrawal. *Nonlinear fiber optics*. 3e. Academic Press, 2001.
- [2] A. Braun, G. Korn, X. Liu, D. Du, J. Squier, and G. Mourou. Self-channeling of high-peak-power femtosecond laser pulses in air. *Optics Letters*, 20:73–75, 1995.
- [3] E.T.J. Nibbering, P.F. Curley, G. Grillon, B.S. Prade, M.A. Franco, F. Salin, and A. Mysyrowicz. Conical emission from self-guided femtosecond pulses in air. *Optics Letters*, 21(1):62–64, 1996.
- [4] A. Brodeur, C.Y. Chien, F.A. Ilkov, S.L. Chin, O.G. Kosareva, and V.P. Kandidov. Moving focus in the propagation of ultrashort laser pulses in air. *Optics Letters*, 22(5):304–306, 1997.
- [5] S.L. Chin, S. Petit, W. Liu, A. Iwasaki, M.-C. Nadeu, V.P. Kandidov, O.G. Kosareva, and K.Yu. Andrianov. Interference of transverse rings in multifilamentation of powerful femtosecond laser pulses in air. *Optics Communications*, 210:329–341, 2002.
- [6] L. Berge, S. Skupin, F. Lederer, G. Mejean, J. Yu, J. Kasparian, E. Salmon, J.P. Wolf, M. Rodriguez, L. Woste, R. Bourayou, and R. Sauerbrey. Multiple filamentation of terawatt laser pulses in air. *Physics Review Letters*, pages 225002–4, 2004.
- [7] S. Skupin, L. Berge, U. Peschel, F. Lederer, G. Mejean, J. Yu, J. Kasparian, E. Salmon, J.P. Wolf, M. Rodriguez, L. Woste, R. Bourayou, and R. Sauerbrey. Filamentation of femtosecond light pulses in the air: Turbulent cells versus long-range clusters. *Physics Review E*, 70:046602–15, 2004.
- [8] O.G. Kosareva and S.A. Shlenov. Scattering of the ultrashort ionizing laser pulse in gas. *Izvestiya Rossiiskoi Akademii Nauk. Seriya Fizicheskaya*, 56(9):56–62, 1992.
- [9] S.A. Shlenov and V.P. Kandidov. Filament bunch formation upon femtosecond laser pulse propagation through the turbulent atmosphere. part 1. method. *Atmos. Oceanic Opt.*, 17(8):565–570, 2004.
- [10] M. Mlejnek, E.M. Wright, and J.V. Moloney. Dynamic spatial replenishment of femtosecond pulses propagating in air. *Optics Letters*, 23:382–384, 1998.
- [11] J.R. Penano, P. Sprangle, B. Hafizi, A. Ting, D.F. Gordon, and C.A. Kapetanacos. Propagation of ultra-short, intense laser pulses in air. *Physics of Plasmas*, 11(5):2865–2874, 2004.
- [12] P.M. Lushnikov. Fully parallel algorithm for simulating dispersion-managed wavelength-division-multiplexed optical fiber systems. *Optics Letters*, 27(11):939–941, 2002.

Cluster	<i>RSSMSU</i> (<i>ANT</i>) "A"	<i>RSSMSU</i> (<i>AQUA</i>) "B"	<i>PSIRAS</i> (<i>Pervenez - M</i>) "C"	<i>ILCMSU</i> "D"
Nodes	80	41	16	7
Hardware	<i>Opteron248</i> 2.2GHz, 4GB	<i>PentiumIII</i> 1GHz, 1GB	<i>AthlonMP1800</i> 1.5GHz, 1GB	<i>Xenon</i> 2.6GHz, 1.5GB
Network	Infiniband	<i>Fast Ethernet</i>	SCI	<i>Gigabit Ethernet</i>
OS	<i>SUSELinux9.1</i> <i>MVAPICH0.9.4</i>	<i>RedHatLinux7.3</i> <i>MPICH1.2.5</i>	<i>RHEL3.2.2</i> <i>ScaMPI1.13.15</i>	<i>FedoraCore2</i> <i>MPICH1.2.7</i>
Compiler	INTEL 8.1	INTEL 8.1	GCC 3.2.2	INTEL 8.0
Performance <i>LINPACKtest</i> (<i>Gflops</i>)	512	31	57	18

Table 1: Cluster specifications. All clusters have 2 processor nodes.

N_x, N_y	$\tau_{c,c}$	$Err_{,c}$	$\tau_{e,c}$	$Err_{,c}$
Cluster A (8 processors)				
1024	6.77	0.01	0.4	0.1
2048	29.91	0.06	1.5	0.1
4096	143.8	0.3	5.3	0.3
8192	595.4	1.4	46	1.4
Cluster B (8 processors)				
1024	17.01	0.03	1.91	0.05
2048	67.5	0.15	10.0	0.2
4096	309.0	0.6	55.3	0.9
Cluster C (8 processors)				
1024	25.26	0.02	0.42	0.03
2048	108.57	0.03	1.59	0.09
4096	510.6	0.3	6.1	0.5
Cluster D (6 processors)				
1024	32.9	0.3	1.87	0.15
2048	132.5	1.9	4.9	0.5
4096	783	46	53	8

Table 2: Test results on different clusters.