Smoothed Particle Electromagnetics Modelling on HPC-GRID Environment

Guido Ala[1] and Elisa Francomano[2]

[1] Dipartimento di Ingegneria Elettrica, Elettronica,Telecomunicazioni, di Tecnologie Chimiche,
Automazione e Modelli Matematici
Università degli Studi di Palermo, Palermo, Italy Viale delle Scienze, Palermo
guido.ala@unipa.it

[2] Dipartimento di Ingegneria Chimica, Gestionale, Informatica e Meccanica
Università degli Studi di Palermo, Palermo, Italy Viale delle Scienze, Palermo
elisa.francomano@unipa.it

Abstract **─**In this paper a meshless approach on a high performance grid computing environment to run fast onerous electromagnetic numerical simulations, is presented. The grid computing and the message passing interface standard have been employed to improve the computational efficiency of the Smoothed Particle Electromagnetics meshless solver adopted. Applications involving an high number of particles can run on a grid computational environment simulating complex domains not accessible before and offer a promising approach for the coupling of particle models to continuous models. The used meshless solver is straightforward to program and fully parallelizable. The results of the parallel numerical scheme are reported and tested on a transverse electric propagation case study taken into account to assess the computational performance.

Index Terms **─** Electromagnetic Simulation, Grid Computing, Meshless method, Smoothed Particle Electromagnetics.

I. INTRODUCTION

In recent years the numerical treatment of partial differential equations with meshfree discretization has become a very active research area. Meshfree methods have undergone substantial development since the mid 1990s. The growing interest in these methods is due to the fact that they are very flexible numerical tools where node configurations have no fixed connectivity and have some advantageous features especially attractive when dealing with multiscale phenomena: a-priori knowledge about particular local behaviour of the solution can be introduced easily in the meshfree approximation space, and an enrichment of a coarse scale approximation with fine scale information is possible in a seamless way. Due to their independence of a mesh, meshfree methods can deal with diffuse in-homogeneity and complex geometry of the domain in a more easy way than standard discretization techniques [1-4]. For a large number of standard numerical methods that solve differential partial equations, the process needs the construction of a linear system and its numerical solution. This is true for finite element method (FEM), element free Galerkin (EFG) method and meshless local Petrov-Galerkin (MLPG) method, for instance. Different approaches are used in technical literature to parallelize numerical methods [5-6].

On the other hand, the approximation of the field variables in meshfree methods is usually based on processing information belonging to a local domain in the neighboring of the observation point, and the procedure has showed to be a simple way to speedup, so justifying the research for. The implementation of meshfree methods and their parallelization, requires special attention enabling to work with a wide range of complex and cumbersome applications. Smoothed Particle Hydrodynamic (SPH) was one of the first mesh-

Submitted On: Dec. 6, 2011
Accepted On: Mar. 4, 2012

free methods to be proposed [7-15] and recently it has been reformulated to solve onerous time domain electromagnetic problems [16-17]. In time evolutionary simulations, the resulting meshless method, addressed as Smoothed Particles Electromagnetics (SPEM), computes the field variables by means of an integral formulation performed over a set of particles identified by a kernel function. The SPEM method, does not deal with a grid at all and the solution is computed directly for each field point using the neighboring information by avoiding the generation of a linear system. The spatial derivatives are computed by transferring the differentiation from the field variables into the kernel by employing two set of staggered particle distributions keeping information on magnetic and electric field components respectively. The method is fully parallelizable and characteristics of the collections of set and points, with implications for the performance of the algorithm, are considered and also exhibited through numerical examples. In SPEM the processing is totally independent on each field point at a given time step and, in this paper, it has been parallelized by using the Message Passing Interface (MPI) library in a grid environment. The resulting processing times are compared with the ones from sequential version. Moreover, the numerical solutions have been compared with the experimental results obtained in a sequential way obtaining a very satisfactory agreement, as confirmed by the performance analysis reported in the next of the paper. The paper is organized as follows. In section II the fundamentals of SPEM meshless solver are briefly summarized. In section III the features of the parallel approach are reported: the pre-processed computational step and the temporal step are addressed, respectively. Section IV validates the computational scheme by referring to a transverse electric (TE) simulation at different time steps.

II. THE MESHLESS SOLVER

SPH method is based on a set of points scattered in the domain involving a kernel function in order to discretize partial differential equations without any underlying mesh. In the absence of mesh, the spatial derivatives for each point of the domain have to be computed in order to proceed to the time integration. The spatial derivatives are determined into a finite domain surrounding each point of interest by means of a kernel function, and each of these points carry the discrete electric E and magnetic H field's quantities. The term kernel refers to a weighting function and defines how much each field variable contributes to the field variable at a point r.

By considering a given function $A(r)$ it is possible to convolute it by using its values and the chosen kernel function within a compact support D proportional to the so-called smoothing length, noted h, standing for the meshless equivalent of a space step used in classical mesh based methods:

$$< A(r) > = \int_D A(r')W(r - r' ,h)dr' . \tag{1}$$

The convolution (1) is usually referred as *kernel approximation*. The kernel function has the following properties:

$$\int_D W(r - r' ,h)dr' = 1 , \tag{2}$$

$$\underbrace{lim}_{h\to 0} \int_D W(r - r' ,h)dr' = \delta(r - r') . \tag{3}$$

The kernel function depends on the distance. In this study, D is defined with a radius equal to $2h$ and the simulations have been performed by employing as kernel the standard cubic B-spline:

$$W(r - r' ,h) = \frac{\alpha}{h^d} \begin{cases} \frac{2}{3} - (\frac{\|r - r'\|}{h})^2 + \frac{1}{2}(\frac{\|r - r'\|}{h})^3 , & 0 \le \frac{\|r - r'\|}{h} < 1 \\ \frac{1}{6}(2 - \frac{\|r - r'\|}{h})^3 , & 1 \le \frac{\|r - r'\|}{h} < 2 \\ 0 & otherwise \end{cases} \tag{4}$$

where d is the dimension of the problem domain and $\alpha = 1, \frac{15}{7\pi}, \frac{3}{2\pi}$ for $d=1,2,3$, respectively. In Fig. 1 the 2-d cubic B-spline is reported. One of the major advantages in using this kernel function is that it has compact support: particles interaction are zero at distances major than $2h$ (Fig. 2).

If $A(r')$ is known only at N discrete points $r_1, r_2, ..., r_N$ the equation (1) is discretized as follows:

$$A(r_i) = \sum_{j=1}^{N} W(r_i - r_j, h)A(r_j)dV_j . \tag{5}$$

In a similar way, as an example, the gradient of any field function can be approximated by means of the following expression:

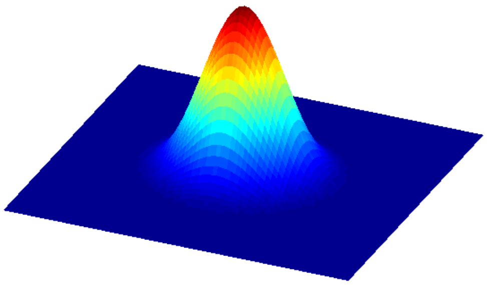


Fig. 1. 2-d cubic B-spline.

$$< \nabla_r A(r) >= \int_D \nabla_{r'} A(r') W(r-r',h) dr' , \qquad (6)$$

where $\nabla_{r'}$ indicates the derivative with respect to the primed coordinates and, in the discrete domain:

$$\nabla A(r_i) = \sum_{j=1}^{N} A(r_j) \nabla_i W(r_i - r_j ,h) dV_j \qquad (7)$$

where ∇_i indicates the spatial derivative with respect to particle *i*'s coordinates.

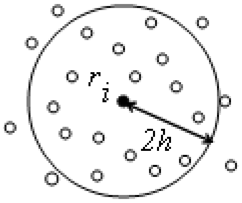


Fig. 2. Support domain of the kernel function.

Equation (7) is one of the main reason for which SPH method is so popular. It removes the need for a mesh to compute spatial derivatives. As well known, electromagnetic transients phenomena are described by Maxwell's curl equations in time domain, which in a non-dissipative medium can be written as follows:

$$\partial_t S \begin{pmatrix} E \\ H \end{pmatrix} = L \begin{pmatrix} E \\ H \end{pmatrix} \qquad (8)$$

where $S = \begin{pmatrix} \varepsilon & 0 \\ 0 & \mu \end{pmatrix}$, ε is the medium permittivity, μ the medium permeability and:

$$L = \begin{pmatrix} 0 & \nabla\times \\ \nabla\times & 0 \end{pmatrix} \qquad (9)$$

is the curl operator matrix. In contrast to general second-order problems (e.g., wave equations) where only one field, electric or magnetic, needs to be calculated, special arrangements of the particles need when the Maxwell's equations in first-order form are solved [18]. In the present paper, by following the FDTD method [19], in which the staggered Yee scheme yields second-order accuracy, two set of staggered particle distributions for *E* and *H* fields are considered. All fields components are stored in two set of nodes, *E*-particles and *H*-particles, respectively (Fig. 3) [16-17]. This position leads to two separate set of shape functions that approximate the *E* and *H* field component values, respectively (Fig. 4). The spatial derivatives of the electric field *E* are approximated by means of the derivatives of the kernel function centred in an *E*-particle and considering the *H*-particles as neighbouring, i.e. $\partial_q E \approx \partial_q W_E(r_i^E, r_j^H)$, $q = x,y,z$. In a similar way the spatial derivatives of the magnetic field *H* are approximated by means of the derivatives of the kernel function centred in an *H*-particle and by considering the *E*-particles as neighbouring, i.e. $\partial_q H \approx \partial_q W_H(r_i^H, r_j^E)$, $q = x,y,z$.

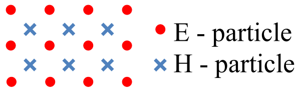


Fig. 3. *E*-particles and *H*-particles distribution.

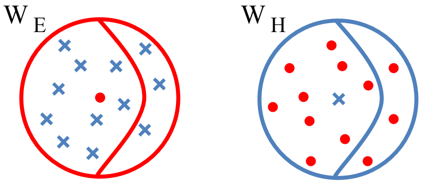


Fig. 4. Kernel functions with staggered *E*- and *H*-particles distribution.

In a transverse electric (TE) case with the E_z, H_x, H_y field components propagating in the *x*-*y*- directions, the problem is reduced to the following form:

$$\partial_t S \begin{pmatrix} E_z \\ H_x \\ H_y \end{pmatrix} = L^h \begin{pmatrix} E_z \\ H_x \\ H_y \end{pmatrix} \qquad (10)$$

E_z, H_x, H_y are vectors of length equal to the electric and magnetic field particles respectively. The matrix for *TE* waves contains the spatial derivatives in the *x*- and *y*-directions:

$$S = \begin{pmatrix} \varepsilon & 0 & 0 \\ 0 & \mu & 0 \\ 0 & 0 & \mu \end{pmatrix},$$

$$L^h = \begin{pmatrix} 0 & -\partial_y W_E & \partial_x W_E \\ -\partial_y W_H & 0 & 0 \\ \partial_x W_H & 0 & 0 \end{pmatrix} \tag{11}$$

In the matrix L^h, the spatial derivatives of the shape functions W_E and W_H in the *x*- and *y*-directions centred in the electric and magnetic field particles are employed, respectively:

$$\begin{aligned} \partial_q W_E &\equiv \partial_q W_E(r_i^E - r_j^H, h) \\ \partial_q W_H &= \partial_q W_H(r_i^H - r_j^E, h) \end{aligned} \qquad q{=}x{,}y \tag{12}$$

The temporal derivatives are discretized by a staggered march in time scheme and, by retaining the nomenclature of the previous section, the explicit time-domain update equations can be expressed as:

$$\begin{aligned} E^{(n+1/2)} &= E^{(n-1/2)} + \frac{\Delta t}{\varepsilon_0}\left[\partial_x W_E H_y^{(n)} - \partial_y W_E H_x^{(n)}\right] \\ H_x^{(n+1)} &= H_x^{(n)} - \frac{\Delta t}{\mu_0}\partial_y W_H E_z^{(n+1/2)} \\ H_y^{(n+1)} &= H_y^{(n)} + \frac{\Delta t}{\mu_0}\partial_x W_H E_z^{(n+1/2)} \end{aligned} \tag{13}$$

where the superscripts indicate the index of the time step, and the condition:

$$\Delta t \le min \frac{d_{min_i}}{c_0} \tag{14}$$

for the time stepping has been employed. This estimate is based on the distance to the closest neighbour of node and the speed of light c_0.

In order to simulate unbounded propagation, in SPEM the domain is truncated by introducing the well-known perfectly matched layer (PML) [20]. In this way, magnetic and electric field are progressively forced to zero within the external layer. As a consequence, the PML boundary conditions considerably reduce the numerical results corruption deriving from particles lacking outside the boundaries [8]. The particle belonging to PML are treated in the same way as for the particle in the interior domain.

III. THE PARALLEL APPROACH

In SPEM, the approximation process can be divided into two fundamentals steps. A pre-processing step for the selection of the neighbouring particles and the computation of the kernel derivatives, and a time dependent step for updating the electric and the magnetic field components values. In Fig. 4 a briefly description of the computation scheme is reported.

PRE-PROCESSING STEP
1. Computation of the distances 2. Computation of the electric field derivatives ∂W_E 3. Computation of the magnetic field derivatives ∂W_H
TEMPORAL STEP
4. Initial Condition 5. Temporal Loop 5.1 Updating of the electric field 5.2 Updating of the magnetic field

Fig. 4. Sketch of the computational process.

A. Pre-processing step

In order to avoid inefficient implementations, the particles must be carefully distributed in the problem domain. With the aim of generating a good load balancing and reducing the data transfer, the problem domain is broken up into strips among the processors. Each processor contains its own strip and it is in charge of calculating the interaction of each fixed particle with the neighboring ones. For the computation of the fields derivatives the interactions of each particle with its neighbors need, and it is performed out of the temporal step. A large part of the computational burden depends on the search procedure; it is therefore essential that efficient methods in a sequential way have been adopted for such a search. The determination of which particles are inside the interaction range, requires the computation of all pair-wise distances, whose computational time would be unpractical for large problems. This was a huge waste of CPU time since each particle gets non-zero values from only a small fraction of the total *N* particles. Simulations with a large number of particles are possible only if an efficient neighborhood search algorithm is employed. Working in a parallel way this bottle-neck is really reduced. Anyway, a framework of fixed cells is carried out to improve the neighboring search. For each particle, the interactions with its neighbours only within the kernel support of radius equal to *2h*, have to be computed.

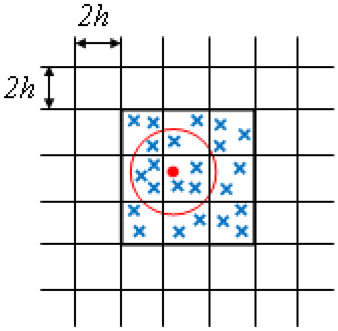


Fig. 5. Underlying grid.

Therefore firstly, a tessellation of the physical space through a cartesian grid composed of square cells with size equal to the kernel radius is carried out (Fig. 5). For a given particle a list of the particles contained in its cell, as well as in eight cells in 2-d around it have to be generated in order to find its neighbouring. Thus, only particles in the approximating area are tested, and any chance of missing an interaction is avoided. In the context of variable smoothing length, since the cell sizes may differ from the kernel radius, the number of cells to be explored is greater than nine. In this case the cell size is set as the lowest value of the smoothing length. It is important to address that the underlying grid is only used for the neighbouring search procedure.

In order to increase the speed up of the code a neighbour list for each particle is generated. A 2-d array X of cells is used to keep track of neighbours. Each cell in the grid contains two linked lists, one for the free particles in that cell and one for the boundary particles. Each cell in X was a square with side equal to *2h*, or two times the smoothing length. The reason the cells are *2h* in size is because the particles only interact up to distance minor than *2h* and thus only need to check the current cell and neighbouring cells to find all the possible particles that interact with the current ones.

B. Temporal step

In SPEM the processing is totally independent on each particle at a given time step, so the parallelization is done mainly based on particles distribution to processors. As reported in section II the algorithm uses a leap-frog integration time step: first the values must be predicted at a half step forward and then they are used to compute the changes in all variables due to interacting with neighbours.

Then all these information are used to compute the values at full time step ahead. After each time step, the buffer row is reloaded. These buffer rows include the particles that border but belonging to another processor. Only the fields computed on the particles lying on the boundaries, as shown in Fig. 6 in grey, have to be transferred to the adjacent processor. Each processor is interested in this data transfer. After the fields computation, all processors are synchronized in sending the updated fields. So working a good load balancing and a good data transfer is performed. In Fig. 7 the computational scheme at each time step *t* is reported by considering the P_N processors used in the computation. As reported in Fig. 7 each processor is computing the same operations: first of all the distances and the kernel derivatives are performed. The full line address the synchronization statements until all the processors end their work; at the end of the elaboration each processor produces the data regarding its region domain.

IV. VALIDATION RESULTS

In this section a *TE* case study has been addressed. A Gaussian pulse propagating in a 2-d domain of 6400 randomly placed particles is considered (Fig. 8). The source is placed in the domain in the central position and the wave propagation is to the boundaries of the domain. In Fig. 8 the simulations for computing the electric field E_z at different time steps are reported: the propagating wave crosses over the boundary and the propagation go over the space. As already underlined, in the simulation the PML [19] have been used by avoiding the wave reflections. In Table 1 the computational time for $N_P = 1$ and $N_P = 4$ processors has been reported. The code has been written by employing the MPI paradigm. For the sake of completeness, the obtained results have been compared with classic FDTD simulation. The obtained $\|\cdot\|_2$ relative error is equal to $5.61 \cdot 10^{-5}$, by using about 50 neighbors for a fixed particle [16].

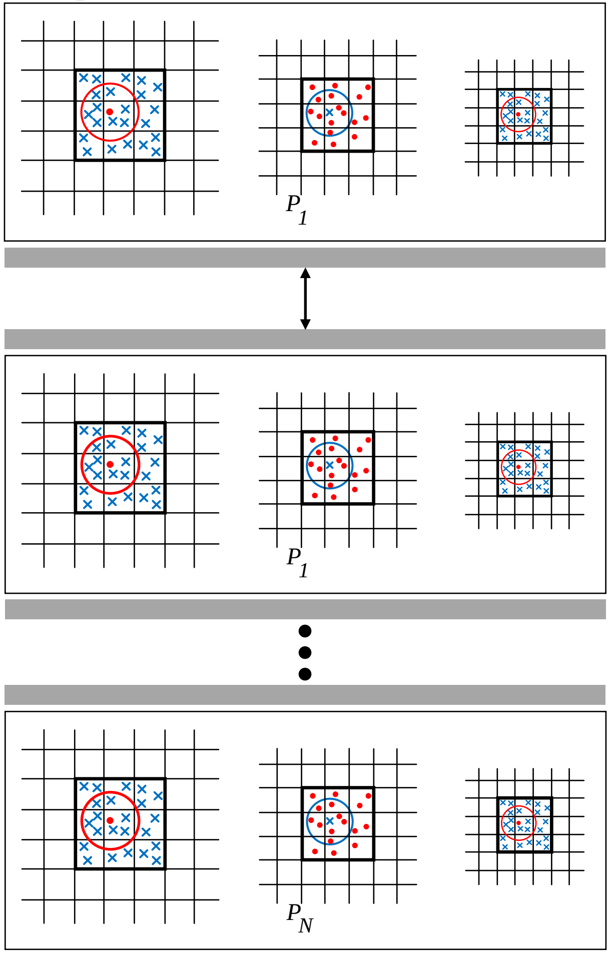


Fig. 6. Non-overlapping data regions.

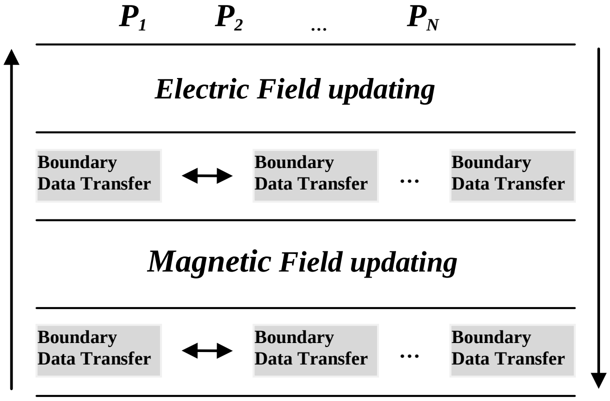


Fig. 7 . Computational scheme at each time step *t* .

The computing infrastructure is based on IBM Blade Centre H chassis each containing up to 14 IBM LS21 "blades" interconnected both with a double Gigabit Ethernet network, for normal communications with redundancy and load balancing, and a CISCO Topspin Infiniband-4X network, required to provide the Grid with high performance computing (HPC) functionalities.

Table 1: Elaboration Time [ms]

Computational stage	**N$_P$ =1**	**N$_P$ =4**
Pre-processing step	19217	4930
Temporal steps (*n*=4)	200	58
Total	19417	4988

The infrastructure is built with identical hardware and software at all sites. This choice was made on purpose to allow for the maximum interoperability and realizes a homogeneous environment which is a fundamental condition for an HPC Grid environment be able to run distributed parallel jobs of applications adopting the MPI paradigm. Each "blade" is equipped with 2 AMD Opteron 2218 rev. F dual-core processors with a clock rate of 2,6 GHz able to natively execute x86 32 and 64 bits binary code. Each processor has 2 GB of DDR2 RAM at 667 MHz (8 GB in total per "blade") and it is equipped with a direct communication channel to the other processor on the same motherboard. The memory controller is integrated on board. The storage infrastructure is based on IBM DS 4200 Storage Systems that provide high features of redundancy, management and reliability. In fact, a DS 4200 Storage System supports several types of RAID and has an intrinsic redundancy of all critical components (fan, power, controller, etc.) to assure maximum reliability. It allows expansions up to 56 TB each with SATA disks. Each Storage System is managed by two IBM x3655 servers that "export" the IBM GPFS parallel file system to all computing nodes. Overall, about 2000 CPU cores and more than 200 TB of disk storage space are currently available on the infrastructure. A reasonable time estimation [21] for the parallel implementation is:

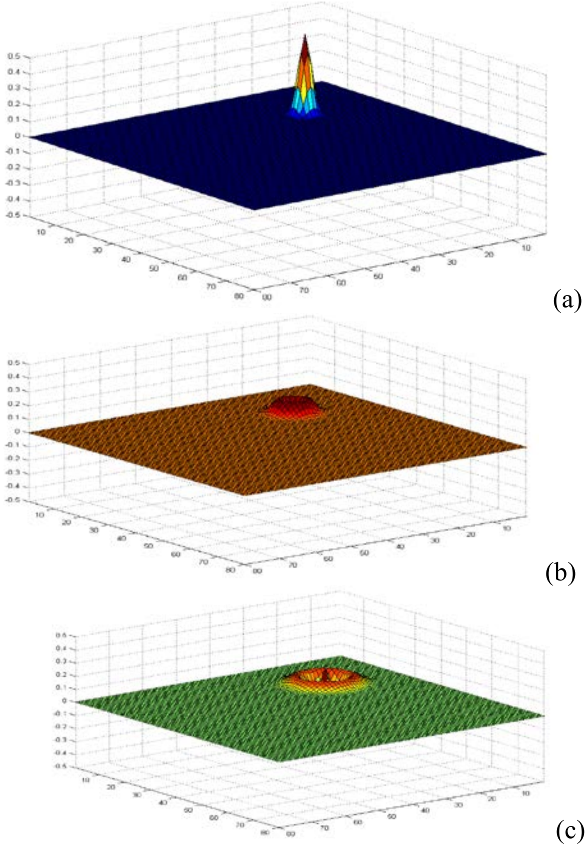
$$T_P = T_S / N_P + O \qquad (15)$$

where T_s is the serial computational time, N_P is the number of processors and O is the overhead. The overhead O in the case of the proposed non-overlap paradigm is constant and is determined only by small data exchange between two adjacent processors P_i and P_{i+1}. Due to these considerations, by using the non-overlap data paradigm, a real speed-up close to the theoretical one and a highly scalable algorithm are obtained.

V. CONCLUSIONS

In this paper a parallel approach of the meshless solver SPEM to investigate transient electromagnetic propagation, is presented.

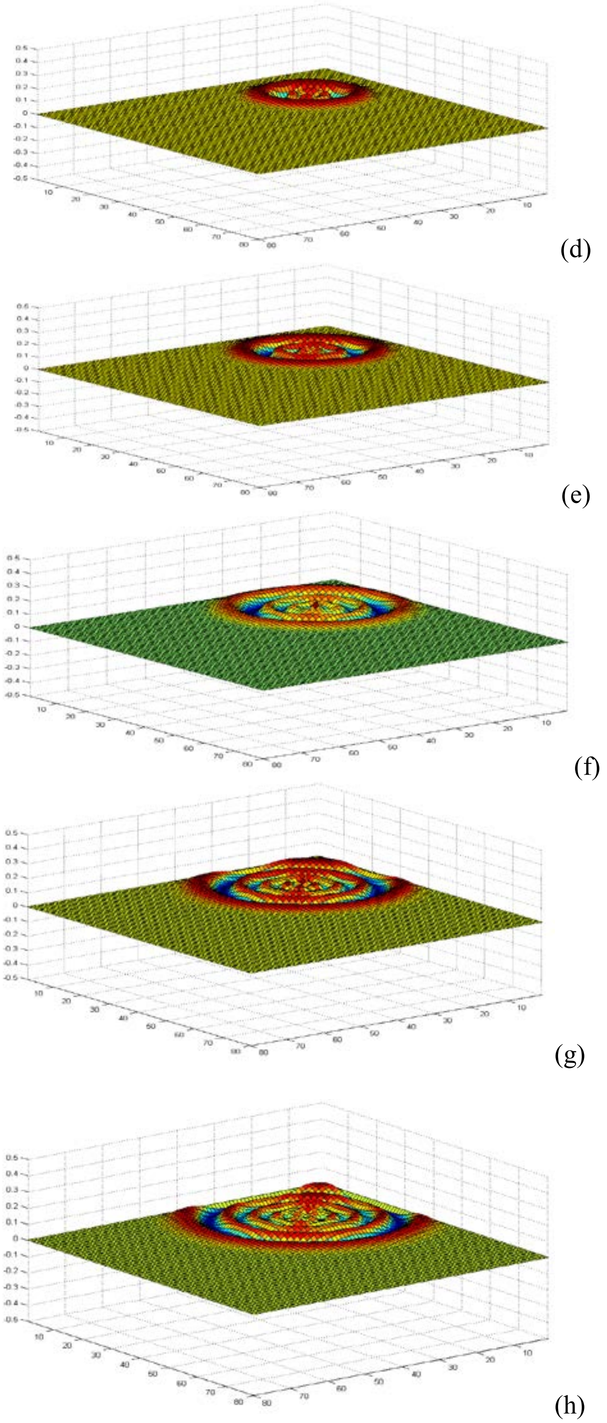
An insight is given into the relative computational burden, and some suggestions are provided on the computational and data structure of the neighborhood search by working with a GRID computational environment with HPC functionalities. Results are provided for a *TE* case study. The process involves local operators which use neighboring values to generate partial results; data strip partitioning with no data overlapping has shown to be suitable by working on distributed multiprocessor systems. Each data strip can be handled as a stand-alone problem and an exiguous amount of data transfer needs at each time step and a good task balancing is provided.



(a)

(b)

(c)



(d)

(e)

(f)

(g)

(h)

Fig. 8. Electric field E_z [V/m] propagation at different time steps, for the *TE* case study.

REFERENCES

[1] Y. Yu, Z. Chen, A 3-D Radial Point Interpolation Method for Meshless Time-Domain Modeling," *IEEE Transactions on Microwave Theory and Techniques,* vol. 57, no. 8, pp. 2015-2020, 2009.

[2] Y. Yu, Z. Chen, "Towards the development of an unconditionally stable time-domain meshless method," *IEEE Transactions on Microwave Theory and Techniques,* vol. 58, no. 3, pp. 578-586, 2010.

[3] X. Chen; Z. Chen; Y. Yu; D. Su, An Unconditionally Stable Radial Point Interpolation Meshless Method With Laguerre Polynomials," *IEEE Transactions on Microwave Theory and Techniques,* vol. 59, no. 10, pp. 3756-3763, 2011.

[4] Richard K. Gordon, W. Elliott Hutchcraft, "The Use of Multiquadric Radial Basis Functions in Open Region Problems," *Applied Computational Electromagnetics Society (ACES) Journal*, vol. 21, no. 2, pp. 127 – 134, July 2006.

[5] T. Iwashita, M. Shimasaki, J. Lu, "Parallel ICCG Solvers for a Finite-Element Eddy-Current Analysis on Heterogeneous Parallel Computation Environment," *Applied Computational Electromagnetics Society (ACES) Journal,* vol. 22, no. 2, pp. 195 – 200, July 2007.

[6] I. González, E. García, F. Saez de Adana, M. F. Cátedra, "MONURBS: A Parallelized Fast Multipole Multilevel Code Analyzing Complex Bodies Modeled by NURBS Surfaces," *Applied Computational Electromagnetics Society (ACES) Journal,* vol. 23, no. 2, pp. 134 – 142, June 2008.

[7] R. A. Gingold, J. J. Monaghan, "Smoothed Particle Hydrodynamics: Theory and Application to Non spherical Stars", *Mon. Not. Roy. Astron. Soc.*, vol.181, pp. 375-389, 1977.

[8] G. R. Liu, M. B. Liu, *Smoothed Particle Hydrodynamics - a Mesh-Free Particle Method*, World Scientific Publishing, Singapore, 2003.

[9] L. B. Lucy, "A numerical approach to the testing of fusion process", *The Astronomical Journal,* vol. 82, pp. 1013-1024, 1977.

[10] J. J. Monaghan, J. C. Lattanzio, "A refined particle method for astrophysical problems", *Astron.Astrophys,* vol. 149, pp.135–143, 1985.

[11] J. J. Monaghan, "An introduction to SPH", *Comput. Phys. Commun,* vol. 48, pp. 89-96, 1988.

[12] J. J. Monaghan, "Smoothed particle hydrodynamics", *Annu. Rev. Astron. Astrophys,* vol. 30, pp. 543–574, 1992.

[13] T. Belytschko, Y. Krongauz, D. Organ, M. Fleming, P. Krysl, "Meshless methods: an overview and recent developments", *Comput. Meth. Appl. Mech. Eng.,* vol. 139, pp. 3–47, 1996.

[14] G. R. Liu, *Mesh Free Methods—Moving beyond the Finite Element Method*, CRC Press, BocaRaton, 2003.

[15] P. Laguna, "Smoothed particle interpolation", *Astrophys. J.,* vol. 439 , pp. 814–821, 1995.

[16] G. Ala, E. Francomano, "An improved Smoothed Particle Electromagnetics method in 3D time domain simulations", *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields,* DOI: 10.1002 /jnm.834, 2011.

[17] G. Ala, G. Di Blasi, E. Francomano, "A numerical meshless particle method in solving the magnetoencephalography forward problem", *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields,* DOI: 10.1002 /jnm.1828, 2012.

[18] M. N. O. Sadiku, *Numerical Techniques in Electromagnetics,* CRC Press, Boca Raton, FL, 1992.

[19] D. M. Sullivan, *Electromagnetic Simulation Using the FDTD Method*, IEEE Press, New York, 2000.

[20] J. P. Berenger, "A Perfectly Matched Layer for the Absorption of Electromagnetic Waves", *Journal of Computational Physics,* vol. 114, pp. 185-200, 1994.

[21] D. P. Bertesekas, J. N. Tsitsiklis, *Parallel and Distributed Computation Numerical Methods,* Prentice Hall, 1989.

Dr. Prof. Guido Ala was born in 1964, he received the Laurea degree (Master degree) in Electrotechnical Engineering (cum laude and with honours) and the PhD degree in Electrotechnical Engineering (Electrical Sciences) from the University of Palermo, Italy, in 1989 and in 1994, respectively. After some industrial experiences he joined the Department of Electrical, Electronic and Telecommunication Engineering of the University of Palermo where he was appointed as an university researcher from 1996 to 2004. Since 2005 he is an associate professor at the same Department (now DIEETCAM). He teaches Fundamental of Circuit Theory, Principles of Electrical Engineering, Electromagnetic Compatibility, Electrical Engineering Numerical Modelling also for PhD students, and, from 2008 to 2011 he was the director of the master degree course in Electrical Engineering. His main research interests are in the fields of electromagnetic numerical methods, electromagnetic transient analysis, lightning, electromagnetic compatibility. He is also the tutor of students working with MIUR projects fellowships.

Dr. Prof. Elisa Francomano is an associate professor of Numerical Analysis, full time, from 2002 at the University of Palermo, Italy. She received the master degree in mathematics (cum laude) from the University of Palermo, in 1988. In 1989 she joined the Istituto di CAlcolo e Reti ad alte prestazioni (ICAR-National Research Council) on a fellowship, working on numerical methods and parallel computing. From 1992 to 2002, she was Senior University Researcher. Since 1989 she has been in the activities of Numerical Analysis group of the University of Palermo and she has given lectures in many courses for students in engineering, in chemistry, in mathematics and in statistics. She thought Numerical Analysis, Fundamentals of Computer Science, Parallel Algorithms, for PhD students too. She is a tutor for students working with MIUR and National Research Council projects fellowships and is in charge of national and international scientific projects. Since 2005 to 2010 she was vice-director of the Centro Interdipartimentale di Tecnologie della Conoscenza (CITC) of the University of Palermo, Italy. Her main research interests are in the fields of computational science, with major applications in electromagnetic transient analysis and in computer vision dealing with numerical linear algebra, integral and partial and differential equations, approximation theory, high-performance scientific computing.