An Algorithm for Multi-Resolution Grid Creation Applied to Explicit Finite Difference Scheme

RAKA JOVANOVIC
Institute of Physics
Belgrade
Pregrevica 118, Zemun
SERBIA
rakabog@yahoo.com

MILAN TUBA
Faculty of Mathematics
University of Belgrade
Studentski trg 16
SERBIA
tubamilan@ptt.rs

DANA SIMIAN
Department of Computer Science
Lucian Blaga University of Sibiu
5-7 dr. I. Ratiu str.
ROMANIA
d_simian@yahoo.com

Abstract: - This paper deals with the main shortcoming of finite difference schemes: the use of a discretization grid with the same resolution over the entire problem space. We propose to avoid this problem by using a multi-resolution grid. The algorithm for the grid creation is presented, that is correct for numeric calculations and optimized for the use in program application. The algorithm is illustrated with the numerical simulation of the propagation of a light beam in a photonic lattice. It is implemented by an explicit finite difference method. An explicit method is adopted, due to the multidimensionality of the problem and the presence of nonlinearity. The efficiency of the algorithm is increased by further improving the precision of the explicit method by the use of a multidimensional generalization of the Runge-Kutta scheme.

Key-Words: - Multi-Resolution Grid, Finite Differences

1 Introduction

The finite difference (FD) [11] method for the solution of partial differential equations is commonly used for a wide range of problems in physics and engineering, due to the simplicity of its implementation and parallelization on multiprocessor machines[14,10,8]. One of the biggest drawbacks of the standard FD scheme is that it uses a discretization grid with the same resolution over the entire problem space. Usually, a high resolution is required only in a small fraction of the problem space, and the use of an uniform grid unnecessarily inflates the demand for computer resources and increases the time needed for calculations. One of the possible ways to solve this problem is to combine the finite difference with the finite element methods [12,13]. However, in such case one loses the main advantage of the FD, which is its simplicity of implementation.

This paper presents a simple algorithm for creation of a multi-resolution grid, and its use with an explicit FD method. It is based on the improvement of the usual explicit FD, by using the analog of an one dimensional (1-D) Runge-Kutta procedure which greatly increases the precision. The second improvement is the creation of a multi-resolution grid, which separates the calculation of the optimum grid resolution for the numerical calculations, from the grid corrections for the use in a particular program. As an example, the application of our algorithm is presented for the numerical simulation of the propagation of a light beam in a photonic lattice. This problem was usually treated by the use of the fast Fourier transform (FFT) [5,4]. However, a significant drawback of the FFT is that it requires a grid of the $2^k \times 2^k$ size [9], which needs to be sufficiently large to hold all possible 'interesting' areas. The method represented in this article does not have such a strict rule for the grid size, which can be of any dimension $2^i m \times 2^i n$. While the use of FFT is more efficient in the typical cases, for certain particular cases an FD with adaptive grid can give better results.

The idea of using FD with an adaptive grid is not new[1]. However, in earlier works, the creation of a multi-resolution grid was mostly dedicated to the numerical part of the problem, and not to the implementation as it is done in this paper. The proposed grid correction is aimed for the implementation, and it makes the program implementation and parallelization greatly simpler.

The paper is organized as follows. In Section 2 the basic physical problem is described, and the corresponding analytical equations are presented, which belong to the class of nonlinear Schrödinger (NLS) equations. In Section 3 we apply the FD method to this equation, with a forward difference to get an explicit scheme. In the second part of Section 3, an improvement similar to the one dimensional Runge-Kutta is presented. Section 4 shows the multi-resolution grid creation and its optimization for the use in computer programs. In the second part of Section 4 we present the analyzes of some problems, with the use of this kind of grid, and introduce the necessary conversions of the basic equations from Section 2.

2 Physical Model

It is well known that in linear optical media, the light beams have a tendency to spread as they propagate, due to the diffraction and the dispersion of incoherent light. Conversely, in a carefully designed nonlinear media and under certain conditions, the light waves may propagate without spreading or scattering. Instead, they keep their shape and intensity constant. These dynamically and structurally stable objects are called optical solitons [7]. The stability of these objects results from the interplay of the dispersion and diffraction with the nonlinear effects, which tend to localize the wave. Stable objects, also known as solitons, emerge when effects of diffraction are completely compensated by nonlinear effects. Solitons can be viewed as waves that are restricted to the specific interval of time and region of space.

Photorefractive media are those in which the photorefractive effect takes place, i.e. those whose refractive index is altered in the transverse region that is occupied by the light beam. The interaction of a laser beam with a photorefractive crystal can be described by the paraxial wave equation. We can inspect the beams with copropagating (CO) geometries and with different input beam shapes (Gaussian, dipole, quadrupole and vortex).

Optically induced photonic lattices are the realization of the photonic crystal concept [3]. Photonic crystals are the materials that posses a periodic structure in space, which enables them to control the propagation of light. Photonic crystals can be viewed as the optical analogue of semiconductors, in the sense that they modify the propagation characteristics of light just as an atomic lattice modifies the properties of electrons through a bandgap structure. If, for a certain frequency range, a photonic crystal reflects light of any polarization, incident at any angle, we say that the crystal has a complete photonic band gap (PBG). This can be regarded as the analogue of the bandgap structure in semiconductors.

2.1 Mathematical Model

The behavior of CO beams in photonic lattices is described by a time-independent model for the formation of self-trapped CO optical beams, based on the theory of photorefractive (PR) effect. The mathematical model consists of one wave equation in the paraxial approximation for the propagation of CO beams. The model equation has the standard form of a NLS equation with a nonlinearity that is a rational function of the beam intensity, and in the computational space it has the form [1]:

$$i\partial_z F = -\Delta F - \Gamma F \frac{|F|^2 + I_g}{1 + |F|^2 + I_g} \qquad (1)$$

where F is the forward propagating beam envelope, Δ is the transverse Laplacian, and Γ is the dimensionless coupling constant. The quantity $|F|$ is the laser light intensity, and it is constant over all iterations. The above dimensionless propagation equation is written under the scaling $x/x_0 \longleftarrow x, y/x_0 \longleftarrow y, z/L_D \longleftarrow z$, where x_0 is the typical Full width at half maximum (FWHM) beam waist and L_D is the diffraction length. I_g is the transverse intensity distribution of the optically induced lattice array, formed by positioning Gaussian beams at the sites of the lattice, which is a known function in Equation 1. Different geometries of the lattice can be considered, such as hexagonal, cylindrical and square.

2.1.1 Sub-subsection

3 Application of Finite Differences

For an explicit finite difference method, applied to the 3D function F, defined on a grid $x_i = x_0 + i * h$, $y_i = y_0 + i * h$, and $z_i = z_0 + i * dz$, we use the following approximations:

$$F_{ijk} = F(x_i, y_j, z_k)$$

$$\begin{aligned}
\partial_x F(x_i, y_j, z_k) &= \frac{F_{i+1jk} - F_{ijk}}{h} = F_{ijk}^{x}, O(h) \\
\partial_y F(x_i, y_j, z_k) &= \frac{F_{ij+1k} - F_{ijk}}{h} = F_{ijk}^{y}, O(h) \\
\partial_{x^2}^2 F(x_i, y_j, z_k) &= \frac{F_{i+1jk}^{x} - F_{ijk}^{x}}{h} \\
&= \frac{F_{i+1jk} - 2F_{ijk} + F_{i-1jk}}{h^2} = F_{ijk}^{xx}, O(h^2) \\
\partial_{y^2}^2 F(x_i, y_j, z_k) &= \frac{F_{ij+1k}^{y} - F_{ijk}^{y}}{h} \\
&= \frac{F_{ij+1k} - 2F_{ijk} + F_{ij-1k}}{h^2} = F_{ijk}^{yy}, O(h^2) \\
\Delta F(x_i, y_j, z_k) &= \partial_{x^2}^2 F(x_i, y_j, z_k) + \partial_{y^2}^2 F(x_i, y_j, z_k) \\
&= F_{ijk}^{xx} + F_{ijk}^{yy}, O(h^2)
\end{aligned} \qquad (2)$$

Using the Euler integration for eq. (1), with the step *dz*, and calculating the nonlinear term $|F|^2$ at z_n, we obtain

$$F(z_0, x, y) = F_0 \tag{3}$$

$$\begin{aligned} F_{n+1} &= F(z_{n+1}, x, y) \\ &= F_n + idz(\Delta F_n + \frac{|F_n|^2 + I_g}{1 + |F_n|^2 + I_g}). \end{aligned} \tag{4}$$

These are the same equations that one would obtain by using the forward difference over *z*. In an explicit difference method, we proceed by using the approximations defined by Equations 2, which after the substitution in Equations 3 yield

$$\begin{aligned} i &= 1, nxy \\ j &= 1, nxy \end{aligned} \tag{5}$$

$$\begin{aligned} F_{n+1}((x_i, y_j) = F_n(x_i, y_j) + \\ idz(\frac{F_{i+1j} + F_{ij+1} - 4F_{ij} + F_{ij-1} + F_{i-1j}}{h^2} \\ + \frac{|F_n(x_i, y_j)|^2 + I_{gij}}{1 + |F_n(x_i, y_j)|^2 + I_{gij}}) \end{aligned} \tag{6}$$

When we perform the Euler integration over *z*, the *z* component of the error is proportional to *dz*. This part of the error can be reduced to dz^4 if we use the following analogue of the Runge-Kutta method, viz.

$$\begin{aligned} Diff(F, K) = K \times i \times dz \\ (\frac{F_{i+1j} + F_{ij+1} - 4F_{ij} + F_{ij-1} + F_{i-1j}}{h^2} \\ + \frac{|F_n(x_i, y_j)|^2 + I_{gij}}{1 + |F_n(x_i, y_j)|^2 + I_{gij}}) \end{aligned} \tag{7}$$

$$S_1 = Diff(F_n, 1) \tag{8}$$

$$S_2 = Diff(F_n + \frac{S_1}{2}, \frac{1}{2}) \tag{9}$$

$$S_3 = Diff(F_n + \frac{S_2}{2}, \frac{1}{2}) \tag{10}$$

$$S_4 = Diff(F_n + S_3, 1) \tag{11}$$

$$F_{n+1} = F_n + \frac{1}{6}(S_1 + 2S_2 + 2S_3 \tag{12}$$

The integration of the function *F* over *z* can be done either in the implicit or in the explicit way. We adopt an explicit procedure, because implicit methods, such as the alternative-direction, give a low level of accuracy in the case of multidimensional problems, due to the existence of the nonlinear term $|F|^2$ [6]. The problem of stability of explicit methods is not easy to solve, but the use of an adaptive step for the Runge-Kutta procedure usually gives satisfactory results.

4 Adaptive Grid Implementation

As we have mentioned before, one of the biggest problems with the FD method is that the grid has the same resolution over the entire problem space. However, in the simulation of a CO beam propagation, the most of the beam intensity is usually located in a small part of the problem space, as it can be seen in fig. (1).

We would like to calculate the function *F* more precisely in the high intensity areas, where the changes between neighboring grid points are much bigger than in the low intensity areas. In the latter, the changes are often so small that they can even be neglected. In the application of an adaptive grid to the problem defined by eq. (1), there are two main problems. First, it is necessary to create a grid that has a high resolution at the correct places, and which allows the easy translation of indices from the high to low resolution areas and vice versa, that is required in the program implementation. Second, one needs to implement the necessary corrections to eq. (7) for the grid blocks that have neighbors with different resolutions.

4.1 Grid Creation

For the creation of the grid, first we must define a function that will give an estimate of the expected resolution of the grid at each point in an integer scale. To do this, let us define the function *Mass, Scale*:

$$\begin{aligned} Scale &: \mathbf{R} \longrightarrow \{1, .., ScaleMax\} \\ Mass &: \{1, n\} \times \{1, m\} \longrightarrow Scale \end{aligned} \tag{13}$$

$$\begin{aligned} Mass_{ij} = Mass(i, j) = Scale(|F_{ij} - F_{i+1j}| + |F_{ij} - F_{i-1j}| \\ + |F_{ij} - F_{ij+1}|) + |F_{ij} - F_{ij-1}|) \end{aligned} \tag{14}$$

Here *Mass*, represented as a matrix generated for some function *F*, has the form

$$Mass=\begin{pmatrix}3&3&3&3&2&3&3&1\\3&3&3&3&2&2&3&3\\3&3&3&3&3&2&2&3\\3&3&3&3&3&2&2&3\\3&3&3&3&1&2&2&3\\3&3&3&3&2&2&1&3\\3&3&3&3&2&2&1&3\\3&3&3&3&3&3&3&3\end{pmatrix}\rightarrow\begin{pmatrix}3&3&3&3&2&2&1&1\\3&3&3&3&2&2&1&1\\3&3&3&3&2&2&2&2\\3&3&3&3&2&2&2&2\\2&2&2&2&1&1&1&1\\2&2&2&2&1&1&1&1\\2&2&2&2&2&2&1&1\\2&2&2&2&2&2&1&1\end{pmatrix}=\widehat{Mass}$$

(15)

In this representation of *Mass*, larger values represent the points in which larger grid blocks could be used, and smaller are used for the opposite.

With some corrections, could be used to define a grid with different resolutions, that can be efficiently used in computer programs. For this to be possible, the corrected $\widehat{Mass}$ needs to have the following characteristics:

- grid blocks need to be squares of the size $2^k \times 2^k$,
- for all the blocks of the level k, the values of the indexes i, j in the highest resolution are such that $i, j = 0(mod)2^k$,
- for every i, j, $Mass_{ij} >= \widehat{Mass}_{ij}$
- the level difference between neighboring blocks cannot be bigger than 1.

Equation (15) is the example of a matrix that has been corrected by using these rules.

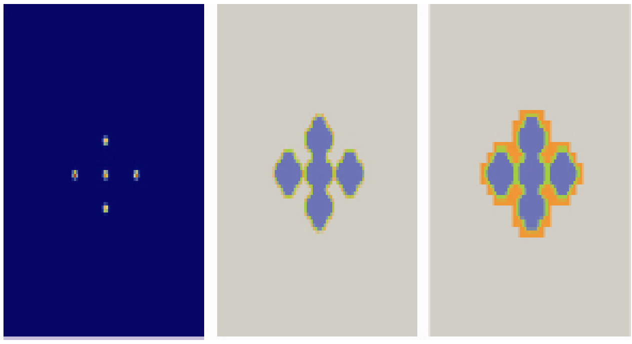
The creation of $\widehat{Mass}$ can be explained the best by the following pseudo code

```
 for Level = 1,ScaleMax
  begin
    Step = 2**Level
      for ix = 0,n :Step
        for iy = 0,m :Step
          CalcMassBlock(ix,iy, Level)
   end
```

We pass through each level, starting from the lowest, and for each block of the size $2^{CurrentLevel} \times 2^{CurrentLevel}$ we call the function *CalcMassBlock* . *CalcMassBlock (GridStartX, GridStartY ,CurrentLevet)* is a function that corrects a matrix block of the size $2^{CurrentLevel} \times 2^{CurrentLevel}$ and the neighboring blocks. It has the following pseudo code

```
function CalcMassBlock
(GridStartX, GridStartY, CurrentLevel)


ActiveBlock =
    block that starts  at GridStartX,
    GridStartY of size
    2**CurrentLevel X 2**CurrentLevel

 if ( (Mass(A) = CurrentLevel))
    begin
       Fill all grid points in
      ActiveBlock with
      min(CurrentLevel,
          CurrentGridPointValue)

      Fill all grid points in blocks
      neighboring   ActiveBlock    of
      level (CurrentLevel+1) with
      min(Level+1,
         CurrentGridPointValue)

    end
```

The creation of this grid is illustrated below by creating the grid that will use for the solution of eq. (1), with an input light ray of the quadrupole type. The input ray is displayed in Fig 1. The figure left shows the intensity distribution, fig center shows the initial mass distribution, or the size of the grid blocks that should be used, and fig. right is the corrected mass.



1. The representation of a quadrupole with central beam with FWHM = 2.124 , in a grid defined by XYMax = 100, NXY = 256. The left image represents intensity distribution. The center image is the initial mass distribution generated using the scale defined by table 19. The right image is the corrected mass distribution. The color scale used for mass distribution is blue for the value 1 (corresponding to the block size 1X1), green for 2(corresponding to block size 2X2), orange for

3(corresponding to block size 4X4), and grey for 4 (corresponding to block size 8X8).

4.2 FDTD Equations for Adaptive Grid

With an adaptive grid the same analogue of the Runge-Kutta method will be used to integrate over the variable *z*. To do this, we must modify eq. (7) in such a way that it can be used with the new grid. There are several parameters that need to be analyzed for each block of level *k*. Let *h* be the grid spacing of the highest resolution. Then h^2 changes to $(2^{k-1}h)^2$. I_g is a precalculated value that represents the transverse intensity distribution of the optically induced lattice array. This parameter should be precalculated with blocks of sizes $(2^{k-1}h)$, in all necessary resolutions and for all possible levels. Let us denote each of these grids with Ig^k . F_{ij} will stay the same, but instead of F_{i-1j}, F_{i+1j}, F_{ij-1} and F_{ij+1}, we will use the representatives for right, left, upper or lower grid block of the size 2^{k-1} . In this problem, an arithmetic average of block members is used as the representative of a block, but for different problems solved by the same method, one may adopt a different procedure to calculate the representatives. Now eq. (7) takes the form

$$DiffR(F,K) = K \times i \times dz \left(\frac{F_{right} + F_{down} - 4F_{ij} + F_{up} + F_{left}}{(2^{Level-1}h)^2} \frac{|F_n(x_i,y_j)|^2 + I_g^{Level}}{1 + |F_n(x_i,y_j)|^2 + I_g^{Level}} \right) \quad (16)$$

The original matrix that represents *F* is not equal at all grid points within a block of size $2^k \times 2^k$ defined in $\widehat{Mass}$, which is required for the use of eq. (16). A simple approach in creating $\widehat{F}$ would be to use the arithmetic average of the corresponding values in *F*. However, this is not correct, because

$$|F|^2 = \int_{x,y} |F(x,y)|^2\, dxdy \simeq \sum_i \sum_j |F_{ij}|^2 \quad (17)$$

is constant in equation (1). Because of this, we calculate the new value for a block of level k, B in $\widehat{F}$ as

$$\widehat{F_B} = \sqrt{\frac{\sum_{(i,j)\in B} |F_{ij}|^2}{2^{k-1}}} \quad (18)$$

4.3 Software Implementation

When creating a software implementation of simulation of the propagation of light beams in photonic lattices, the problem defined be eq. (1), as physical input parameters we have adopted Γ, *xymax* , *nxy* , and from *h* which is calculated; *nz* and *zmax* from which we calculate *dz* , type of lattice from which I_g is precalculated in the required resolutions, and the type of the input ray from which we calculate *F* . A different type of input parameter is *Scale* that defines the creation of *Mass*, which is a table that represents the mass value depending on the sum of differences

Scale	$5*10^{-3}$	10^{-4}	10^{-6}	10^{-7}
Mass	1	2	3	4

(19)

During the simulation tests, the use of eq. (13) for the calculation of *Mass* gave poor results, and it was replaced by

$$DistNorm(i,j,d) = |F_{ij} - F_{i+dj}| + |F_{ij} - F_{i-dj}| + |F_{ij} - F_{ij+d}|) + |F_{ij} - F_{ij-d}| \quad (20)$$

$$\begin{aligned} Mass_{ij} &= Mass(i,j) \\ &= Scale(DistNorm(i,j,1) \\ &\quad + \frac{1}{2} DistNorm(i,j,2) \\ &\quad + \frac{1}{4} DistNorm(i,j,3) \\ &\quad + \frac{1}{8} DistNorm(i,j,4)) \end{aligned} \quad (21)$$

Parameter *Scale* proved to be of great importance for the simulation calculations, and to obtain the best results with respect to the relation between the precision and the performance, it was necessary to perform a series of tests, for each type of input parameters. When using the adaptive grid, *Mass* and $\widehat{F}$ should be calculated once in the beginning of the simulation, and again every *N* (depends of the problem) of iterations to reflect the changes on . In each iteration calculations are done just for the top left point of each block using the Runge-Kutta analog with difference defined be eq. 16, and the *Mass* grid is used to jump to the next point (representing a block) that needs calculation.

5 Conclusion

In this paper an algorithm is shown for the creation and implementation of an adaptive grid, as well as the implementation of the FD method with such algorithm. The basic steps of its implementation to (2+1) dimensional problems are illustrated by the application of FD with an adaptive grid to the simulation of the propagation of light beams in photonic lattices. The first step is the conversion of the starting equation to the approximation of derivatives with finite differences. It is shown that the integration over can be performed much more efficiently by using an analogue of Runge-Kutta, instead of directly using the finite differences. The necessary calculation time for the FD can be considerably reduced by using an adaptive grid. In our simulations, this time was reduced up to 50%, depending on the input parameters. An algorithm for the creation of multi-resolution grids has been presented. It is divided in two parts, i.e. the creation of a mass grid that represents the level of detail that is needed at different grid points, and the conversion of the mass grid to one that could be used in computer programs. This approach is very convenient, because for the creation of grids for different problems, only the first part of the algorithm needs to be modified according to the characteristics of the problem. The use in a computer program of a grid created this way, is simple and allows an easy parallelization on multiprocessor machines. A similar approach could be used also for (3+1) dimensional problems, with an even greater level of reduction of the calculation time, on which a further research will be done.

References:

[1] Belić M., Petrović M., Jović D., Strinić A., Arsenović D., Motzek K., Kaiser F., Ph. Jander, Denz C., Tlidi., and Mandel ., Transverse modulational instabilities of counterpropagating solitons in photorefractive crystals, *Optics Express*, Vol. 12, No. 4, 2004, pp. 708-716.

[2] Chang B.J.; Kim H.-S.; Lee H.Y.; Braunstein J., An Algorithm for Generation of Non-uniform Meshes for Finite Difference Time Domain Simulations,*Electromagnetic Field Computation, 2006 12th Biennial IEEE Conference,* Vol , No , 2006, pp:53 – 53

[3] Joannopoulos J.D., Meade R. D., and Winn J. N., *Photonic Crystals: Molding the flow of light*, Princeton University Press, 1995.

[4] Jović D., Jovanović R., Prvanović S., Petrović M., and Belić M. Counterpropagating beams in rotationally symmetric photonic lattices, *Optical Materials* , Vol. 30 ,2008, pp.1173–1176

[5] Jović D. M., Prvanović S., Jovanović R. D., and Petrović M. S., Gaussian induced rotation in periodic photonic lattices,Gaussian induced rotation in periodic photonic lattices, *Optic Letters*, Vol. 32, No 13, 2007, pp1857-1859

[6] Houffman D. J., *Numerical methods for scientists and engineers*, McGraw-Hill ,1992

[7] Kivshar Y. S. and Agrawal G. P., *Optical Solitons*, Academic Press, San Diego, 2003

[8] Mastorakis N E. , An Extended Crank-Nicholson Method and its Applications in the Solution of Partial Differential Equations: 1-D and 3-D Conduction Equations , *WSEAS Transactions on Mathematics*, Vol. 6, No. 1, 2007, pp 215-225

[9] Press William H., Teukolsky Saul A., Vetterling William T., Flannery Brian P., *Numerical Recepies in C: The Art of Scientific computing*, Cambrige University Press, 1992.

[10] Sun X.H, Joslin R.D., Massively Parallel Algorithm for Compact Finite Difference Schemes, *Parallel Processing,* Vol 3, No , 1994, pp282 – 289

[11] Trefethen L. N., *Finite Difference and Spectral Methods for Ordinary and Partial Differential Equations*, Cornell University ,1996.

[12] Wang B.L, Tian Z.H, Application of finite element–finite difference method to the determination of transient temperature field in functionally graded materials, *Finite elements in analysis and design*, Vol. 41, No 4, 2005, pp. 335-349

[13] Xing J. T., Price W. G., Chen Y. G., A Mixed Finite-Element Finite-Difference Method for Nonlinear Fluid-Structure Interaction Dynamics. I. Fluid-Rigid Structure Interaction, Proceedings: *Mathematical, Physical and Engineering Sciences*, Vol. 459, No. 2038, 2003, pp. 2399-2430.

[14] Zhao J., Davison M., Corless R. M., Compact finite difference method for American option pricing, *Journal of Computational and Applied Mathematics,* Vol. 206 , No. 1, 2007, pp. 306-321