Centrum voor Wiskunde en Informatica

REPORT*RAPPORT*

MAS

Modelling, Analysis and Simulation

Modelling, Analysis and Simulation

A nested-grid finite-difference Poisson solver for concentrated source terms

J. Wackers

REPORT MAS-E0312 SEPTEMBER 02, 2003

CWI is the National Research Institute for Mathematics and Computer Science. It is sponsored by the Netherlands Organization for Scientific Research (NWO).
CWI is a founding member of ERCIM, the European Research Consortium for Informatics and Mathematics.

CWI's research has a theme-oriented structure and is grouped into four clusters. Listed below are the names of the clusters and in parentheses their acronyms.

Probability, Networks and Algorithms (PNA)

Software Engineering (SEN)

Modelling, Analysis and Simulation (MAS)

Information Systems (INS)

P.O. Box 94079, 1090 GB Amsterdam (NL)
Kruislaan 413, 1098 SJ Amsterdam (NL)
Telephone +31 20 592 9333
Telefax +31 20 592 4199

ISSN 1386-3703

A nested-grid finite-difference Poisson solver for concentrated source terms

ABSTRACT
For the numerical simulation of electric discharges, electric fields are calculated with source terms that are concentrated on a very small part of the computational domain. Therefore, a grid adaptation procedure is useful here. However, most existing fast Poisson solvers require a uniform grid. In this paper, a solver is described that combines a nested-grid approach with an existing cyclic-reduction Poisson solver. The solution process is started on a coarse grid, on the entire computational domain. New, finer grids are placed on a part of this domain, based on an error estimate. On each grid, the solution is found using the existing solver, the boundary conditions are interpolated from the underlying coarser grid. On the fine grids, even finer grids are placed, etc. The equations that describe the motion of the electric discharge are solved on a different set of nested grids, so an interpolation procedure between these two sets of nested grids is given. This procedure is used to find the input charge and to compute the output electric field. The most important new feature of the method is the error-based refinement criterion, which allows the calculation of an upper limit for the total error. Using this error bound, the accuracy of a computation can be chosen a priori. Results show that the extra error caused by the nested-grid approach can indeed be made arbitrarily small, if desired.

2000 Mathematics Subject Classification: 65N06, 35J05
Keywords and Phrases: Poisson equation, local source-term, nested grids, electric discharge.
Note: This research was carried out under CWI project MAS3 `Nonlinear Dynamics and Complex Systems'.

A nested-grid finite-difference Poisson solver for concentrated source terms

Jeroen Wackers
CWI
P.O. Box 94079, 1090 GB Amsterdam, The Netherlands

ABSTRACT

For the numerical simulation of electric discharges, electric fields are calculated with source terms that are concentrated on a very small part of the computational domain. Therefore, a grid adaptation procedure is useful here. However, most existing fast Poisson solvers require a uniform grid.

In this paper, a solver is described that combines a nested-grid approach with an existing cyclic-reduction Poisson solver. The solution process is started on a coarse grid, on the entire computational domain. New, finer grids are placed on a part of this domain, based on an error estimate. On each grid, the solution is found using the existing solver, the boundary conditions are interpolated from the underlying coarser grid. On the fine grids, even finer grids are placed, etc. The equations that describe the motion of the electric discharge are solved on a different set of nested grids, so an interpolation procedure between these two sets of nested grids is given. This procedure is used to find the input charge and to compute the output electric field.

The most important new feature of the method is the error-based refinement criterion, which allows the calculation of an upper limit for the total error. Using this error bound, the accuracy of a computation can be chosen a priori. Results show that the extra error caused by the nested-grid approach can indeed be made arbitrarily small, if desired.

2000 Mathematics Subject Classification: 65N06, 35J05
Keywords and Phrases: Poisson equation, local source-term, nested grids, electric discharge.
Note: This research was carried out under CWI project MAS3 'Nonlinear Dynamics and Complex Systems'.

1. INTRODUCTION

The numerical solution of Poisson's equation has been studied much and many different solution techniques have been proposed. One of the fastest of these in two dimensions is the cyclic reduction method [3]. This method has one major disadvantage: it works only for uniform rectangular grids.

But in several applications, like the numerical study of electric discharges, Poisson problems appear where the source term, the right hand side of the equation, is only non-zero in a very small area [2]. In the electric discharge problem, the source term is the net space charge, caused by ions and free electrons. This net charge is concentrated near the tip and the end of the discharge streamer. For this type of problems, much calculation can be saved by solving the problem on an adapted grid that has fine cells in the neighbourhood of the non-zero source and coarse cells everywhere else. A disadvantage is that the cyclic reduction algorithm cannot be used directly on these grids.

In previous work at CWI, the cyclic reduction algorithm was combined with a two-grid structure: the equation was first solved on the whole domain on a coarse grid and then it was solved again on a part of the domain, but on a four times finer grid. The boundary conditions for this problem are interpolated from the solution on the coarse grid. The electric field is used as input for a convection-diffusion equation, that describes the motion of the free electrons. This equation is solved only on the area where the charges are non-zero, as this is the only area where something actually happens. The input for the Poisson solver is the charge distribution on the fine grid, its goal is to find the electric field on the same grid.

This solution method for Poisson's equation is extended and improved here. Now, an arbitrary number of nested grids can be chosen, so the grid around the charges can be made much finer than the coarsest grid.

The current method also allows the solution of the convection-diffusion equation on nested grids with different mesh widths.

The most important new aspect of the method is an error-based refinement criterion. The nature of this criterion enables the calculation of an upper bound for the total error. Using this upper bound, solutions can be calculated for which the extra error, caused by the use of the nested grid instead of a uniform fine grid, is arbitrarily small. However, the use of this criterion means that the nested grids used for the convection-diffusion equation do not have the same size as the grids for Poisson's equation. Therefore, interpolation between the grids is needed for the input and output of the Poisson solver.

The new features of the method will be published separately [4]. The current report describes *all* parts of the nested-grid solver: The formulation of the Poisson problem is given in the next section. The third section gives an overview of the method used to solve the Poisson equation and section 4 describes the refinement criterion and the upper bound for the error. The interpolation procedures for the input charge and the output electric field are given in section 5. Finally, section 6 shows numerical results for two test problems and confirms the accuracy of the method.

2. PROBLEM DESCRIPTION

The method described here is developed for an electric discharge problem as in figure 1. Two infinite plates with a voltage of V_0 are placed a distance L apart. A spark on one plate initiates a streamer of charged particles that moves towards the other plate. At each moment, the particles generate an electric field that is superimposed on the initial uniform electric field.

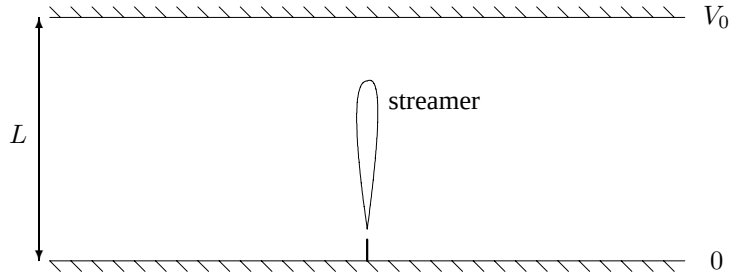


Figure 1: Electric discharge between two infinite plates.

The electric potential is calculated in 3D with radial symmetry, so the actual numerical problem is 2D. The potential Φ generated by the particles is given by Poisson's equation

$$\begin{aligned}
\Delta\phi = \frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial\phi}{\partial r}\right) + \frac{\partial^2\phi}{\partial z^2} = u\left(r,z\right) \qquad & r \in [0, r_{max}]\,, z \in [0, L]\,, \\
\phi = 0 \qquad & z = 0 \text{ and } z = L, \\
\frac{\partial\phi}{\partial r} = 0 \qquad & r = 0, \\
\phi = 0 \qquad & r = r_{max}.
\end{aligned} \tag{2.1}$$

A r_{max} is chosen sufficiently large to make the influence of the boundary negligible. The Neumann condition on $r = 0$ follows from symmetry conditions: a non-zero value of $\frac{\partial\Phi}{\partial r}$ at $r = 0$ implies an infinite value of $\Delta\Phi$ there, i.e. an infinite charge density. The potential on the plates is fixed, so the induced potential is always zero there (see figure 2).

In the actual problem considered, the only charge appears in the head and tail of the streamer. A typical charge distribution is shown in figure 15a.

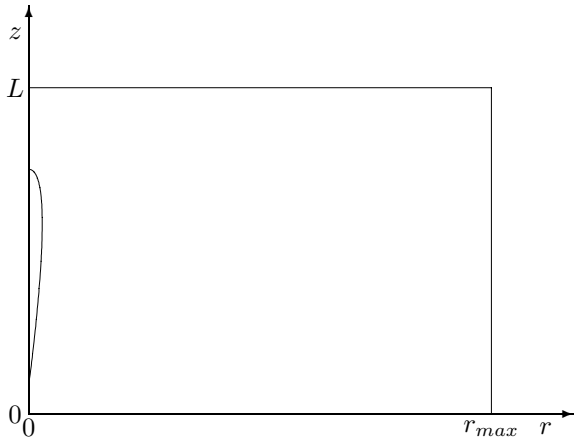


Figure 2: Computational domain.

3. THE METHOD

With the method described here, the Poisson equation is solved on a series of nested grids. It is solved first on a very coarse grid that covers the entire computational domain. The accuracy of this solution is probably good enough far away from the non-zero source term. Closer tot the source, a new fine grid is placed and a new solution is calculated on this grid. The boundary conditions for this fine grid are interpolated from the coarse grid solution. On the fine grid, an even finer grid is placed etc. This section describes the details of the computational method, the refinement criterion is given in the next section.

3.1 The grids

The discretised equation (3.2) is solved on a series of nested grids. The mesh width of the coarsest grid and the maximum level of refinement M are specified. We will call these grids for the solution of the potential Ω^l_ϕ, with the level $l \in [0, M]$. The level of a grid indicates its mesh width, M means that a grid has the smallest mesh sizes in the problem. These mesh sizes are denoted as Δr and Δz. The other grids have larger meshwidths:

$$\begin{aligned} \Delta r^l &= 2^{M-l}\Delta r, \\ \Delta z^l &= 2^{M-l}\Delta z, \end{aligned} \tag{3.1}$$

which means that one cell in Ω^l corresponds to four cells in Ω^{l+1}. There may be more than one grid on each level, therefore the grids are denoted as $\Omega^{l,k}$, $l \in [0, M]$, $k \geq 1$ (see figure 3)[1].

3.2 Calculation of solutions

Equation (2.1) is discretised with a standard second-order accurate central finite-difference approximation.

$$\frac{\Phi_{i+1,j} - 2\Phi_{i,j} + \Phi_{i-1,j}}{\Delta r^2} + \frac{\Phi_{i+1,j} - \Phi_{i-1,j}}{2r_{i,j}\Delta r} + \frac{\Phi_{i,j+1} - 2\Phi_{i,j} + \Phi_{i,j-1}}{\Delta z^2} = f_{i,j}. \tag{3.2}$$

The actual solutions Φ^l are calculated with an existing Poisson solver. Any uniform-grid Poisson solver can be used for this, the actual solution method is not important for the procedure described here.

The only question is, how the boundary conditions for these solutions are chosen. When the grid boundary $\partial\Omega^l_\phi$ lies on the boundary of the entire domain, the boundary conditions from equation (2.1) are used. But when the grid Ω^l_ϕ covers a smaller area than the entire domain, boundary conditions for this grid are set with an interpolation P^l_{l-1} from the solution on the underlying coarse grid Ω^{l-1}_ϕ. We specify that the solution on the

[1] For brevity, the second subscript is not shown from now on, unless it is really needed.

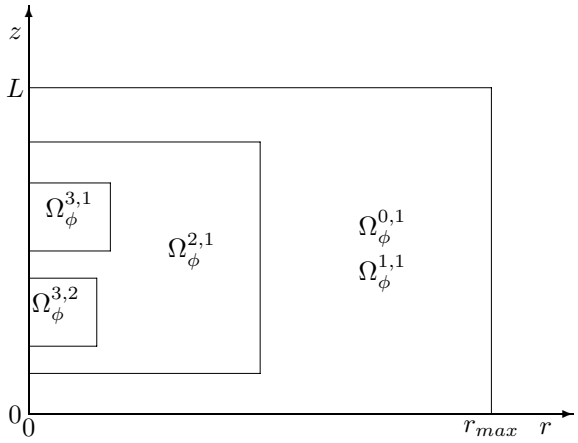


Figure 3: An example of nested grids.

boundary of Ω_ϕ^l is equal to the solution on Ω_ϕ^{l-1}, at the same location. So the boundary conditions are of the inhomogeneous Dirichlet type.

$$\Phi^l = P_{l-1}^l \Phi^{l-1} \qquad \text{on } \partial\Omega_\phi^l. \tag{3.3}$$

For the interpolation, a third-order accurate quadratic formula is used, based on a nine-point stencil (figure 4). A quadratic surface

$$\Phi(r,z) = \Phi_{i,j} + a_1(r-r_i) + a_2(z-z_j) + a_3(r-r_i)^2 + a_4(r-r_i)(z-z_j) + a_5(z-z_j)^2 \tag{3.4}$$

is found by applying a least-squares fit to the Φ in the nine stencil points. This gives for the five coefficients a,

$$\begin{aligned}
a_1 &= \tfrac{1}{6}\left(-\Phi_{i-1,j-1} + \Phi_{i+1,j-1} - \Phi_{i-1,j} + \Phi_{i+1,j} - \Phi_{i-1,j+1} + \Phi_{i+1,j+1}\right), \\
a_2 &= \tfrac{1}{6}\left(-\Phi_{i-1,j-1} - \Phi_{i,j-1} - \Phi_{i+1,j-1} + \Phi_{i-1,j+1} + \Phi_{i,j+1} + \Phi_{i+1,j+1}\right), \\
a_3 &= \tfrac{1}{10}\left(\Phi_{i-1,j-1} - 2\Phi_{i,j-1} + \Phi_{i+1,j-1} + 3\Phi_{i-1,j} - 6\Phi_{i,j} + 3\Phi_{i+1,j}\right. \\
&\qquad \left. + \Phi_{i-1,j+1} - 2\Phi_{i,j+1} + \Phi_{i+1,j+1}\right), \\
a_4 &= \tfrac{1}{4}\left(\Phi_{i-1,j-1} - \Phi_{i+1,j-1} - \Phi_{i-1,j+1} + \Phi_{i+1,j+1}\right), \\
a_5 &= \tfrac{1}{10}\left(\Phi_{i-1,j-1} + 3\Phi_{i,j-1} + \Phi_{i+1,j-1} - 2\Phi_{i-1,j} - 6\Phi_{i,j} - 2\Phi_{i+1,j}\right. \\
&\qquad \left. + \Phi_{i-1,j+1} + 3\Phi_{i,j+1} + \Phi_{i+1,j+1}\right).
\end{aligned} \tag{3.5}$$

The value at the boundary is found by substituting the coordinates of the boundary points (marked as $\times$ in figure 4) in equation (3.4). The stencil, as in figure 4, has some spatial bias towards the smooth (not refined) region.

4. Refinement criterion and error control

The key element of the current method is its choice of refinement criterion. With the criterion presented here, the solution can be calculated to a chosen accuracy in one coarse-to-fine grid cycle, without iteration. This section describes the criterion and derives the upper bound for the error.

4.1 Defining new grids

The two coarsest grids always fill the entire computational domain. But for all finer grids, smaller domains can be chosen. This is essential for the method, as the need to calculate only a small area on the finest grid makes it fast.

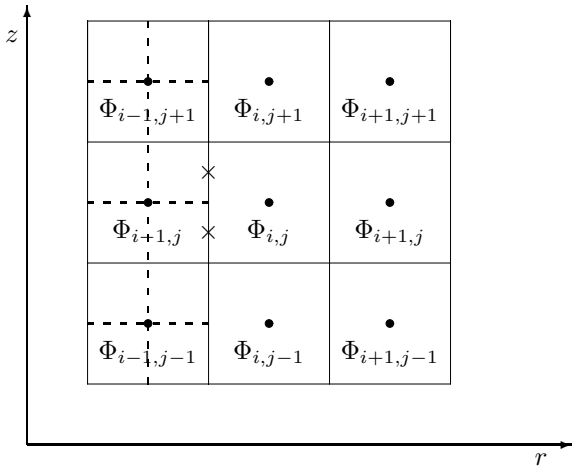


Figure 4: Stencil for the third-order accurate interpolation. The fine grid, for which the boundary values are calculated, is shown dashed. The boundary points are marked $\times$.

The area where a finer grid Ω^{l+1} is placed, is determined from the solution on the next coarsest grid Ω^l. This solution is compared with a potential $\bar{\phi}^l$ that satisfies the exact Poisson equation on Ω^l, but uses the same boundary conditions as Φ^l (equation (3.3)). If these boundary conditions already contain an error, then $\bar{\phi}^l$ is not equal to the exact solution on the entire domain.

$$\begin{aligned} \Delta\bar{\phi}^l &= f, && \text{on } \Omega^l, \\ \bar{\phi}^l &= P_{l-1}\Phi^{l-1}, && \text{on } \partial\Omega^l, \end{aligned} \tag{4.1}$$

where P_{l-1} is the interpolation operator from the discrete solution on Ω^{l-1} to the continuum. The difference between $\bar{\phi}^l$ and Φ^l is the error in the solution on Ω^l only, the error per grid g^l:

$$g^l_{i,j} = \bar{\phi}^l\left(r_{i,j}, z_{i,j}\right) - \Phi^l_{i,j}, \tag{4.2}$$

When the error per grid is known, all cells in Ω^l are determined where $\left|g^l_{i,j}\right| \geq \epsilon$ for a small parameter ϵ and Ω^{l+1} is defined as the smallest rectangle that can be fitted around those cells. When several groups of cells with $|g| \geq \epsilon$ lie far apart, then more than one grid $\Omega^{l,k}$ can be chosen.

To define new grids, an estimate of $g^l_{i,j}$ on the coarse grid is needed. This estimate is found by comparing the solution Φ^l with an interpolation P^l_{l-1} of the solution on the underlying grid Φ^{l-1}, on the domain Ω^l (see figure 5). The boundary values for Φ^{l-1} are the same as for Φ^l, since these were interpolated from Φ^{l-1}.

The discretisation (3.2) is second-order accurate, which means that the error in Φ^l is four times smaller than in Φ^{l-1}. Therefore, if the interpolation errors are negligible,

$$g^l_{i,j} \equiv \bar{\phi}^l\left(r_{i,j}, z_{i,j}\right) - \Phi^l_{i,j} \approx \tfrac{1}{4}\left(\bar{\phi}^l\left(r_{i,j}, z_{i,j}\right) - \left(P^l_{l-1}\Phi^{l-1}\right)_{i,j}\right),$$

and hence

$$g^l_{i,j} \approx \tfrac{1}{3}\left(\Phi^l_{i,j} - \left(P^l_{l-1}\Phi^{l-1}\right)_{i,j}\right). \tag{4.3}$$

For the interpolation, the third-order accurate equation (3.4) is used, to make the interpolation errors small compared to the second-order discretisation errors. In the following error analysis, the interpolation error will be neglected.

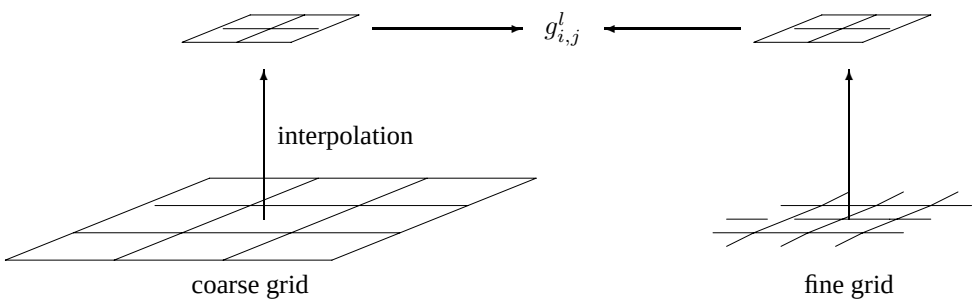


Figure 5: The error is estimated from a comparison between the solutions on two grids.

4.2 Error upper bound

The choice for the size of the fine grids, given in the previous section, allows the calculation of an upper bound for the total error in the solution Φ. We find this upper bound by splitting the error on a grid Ω^l in a component caused by the calculation on the grid itself and a component, caused by the error on the underlying grids.

The total error e is defined as

$$e^l_{i,j} = \phi\left(r^l_i, z^l_j\right) - \Phi^l_{i,j}. \tag{4.4}$$

Let us study this error on one grid Ω^l. Using the definition of the error per grid g^l (equation (4.2)), the total error is

$$e^l_{i,j} = g^l_{i,j} + \phi\left(r^l_i, z^l_j\right) - \bar{\phi}^l\left(r^l_i, z^l_j\right). \tag{4.5}$$

The potentials ϕ and $\bar{\phi}^l$ satisfy Poisson's equation with the *same* source term on Ω^l. Their only difference lies in the boundary conditions, which are interpolated from Ω^{l+1}:

$$\begin{aligned} \Delta\left(\phi - \bar{\phi}^l\right) &= 0, && \text{on } \Omega^l, \\ \phi - \bar{\phi}^l &= P_{l-1} e^{l-1}_{i,j}, && \text{on } \partial\Omega^l. \end{aligned} \tag{4.6}$$

In other words, the difference of ϕ and $\bar{\phi}^l$ satisfies Laplace's equation on Ω^l. Therefore, the maximum principle is valid here:

$$\max_{\Omega^l} \left|\phi - \bar{\phi}^l\right| \leq \max_{\partial\Omega^l} \left|\phi - \bar{\phi}^l\right|. \tag{4.7}$$

The right-hand term is known from equation (4.6):

$$\begin{aligned} \max_{\partial\Omega^l} \left|\phi - \bar{\phi}^l\right| &= \max_{\partial\Omega^l} \left|P_{l-1} e^{l-1}_{i,j}\right|, \\ &= \max_{\partial\Omega^l} \left|P_{l-1}\left(g^{l-1}_{i,j} + \phi\left(r^l_i, z^l_j\right) - \bar{\phi}^{l-1}\left(r^l_i, z^l_j\right)\right)\right|, \\ &\leq \max_{\partial\Omega^l} \left|P_{l-1}\left(g^{l-1}_{i,j} + \max_{\partial\Omega^{l-1}} \left|\phi - \bar{\phi}^{l-1}\right|\right)\right|, \end{aligned} \tag{4.8}$$

so we find, neglecting interpolation errors and the fact that $\max\left|P_{l-1}\left(\cdots\right)\right| \geq \max\left|\left(\cdots\right)\right|$, (the discrete-to-continuous interpolation may overshoot, but the difference is small, compared to the interpolation errors), that

$$\max_{\partial\Omega^l} \left|\phi - \bar{\phi}^l\right| \leq \max_{\partial\Omega^l} \left|g^{l-1}\right| + \max_{\partial\Omega^{l-1}} \left|\phi - \bar{\phi}^{l-1}\right|. \tag{4.9}$$

Until now, the analysis is valid for any choice of location of the grids. But from the definition of the refinement criterion used here, described in section 4.1, it is certain that $\max_{\partial\Omega^l} \left|g^{l-1}\right| \leq \epsilon$. Substituting this in equation (4.9) yields

$$\max_{\partial\Omega^l} \left|\phi - \bar{\phi}^l\right| \leq \epsilon + \max_{\partial\Omega^{l-1}} \left|\phi - \bar{\phi}^{l-1}\right|. \tag{4.10}$$

The grid Ω^1 always covers the entire domain, so it has no error on its boundaries. This allows expansion of the recurrence relation (4.10). Substitution in (4.7) and (4.5) gives an upper bound for the total error e^l.

$$e^l_{i,j} \leq g^l_{i,j} + (l-1)\epsilon. \tag{4.11}$$

This is a very interesting result, as it shows that the extra error, that is passed on from coarser grids, can be made as small as desired by choosing ϵ low enough. As shown, interpolation errors must be small, which is why the third-order accurate interpolation (3.4) is important.

5. Charge mapping and electric field

As stated in the previous section, the size of the potential grids is determined from an error estimate in the potential calculation. This means that the potential grids do not have the same size as the convection-diffusion grids (the input and output grids for the Poisson solver). Usually, the input grids are smaller and the coarsest input grid does not cover the entire domain of the Poisson solution.

This has consequences when variables on the input grids have to be determined on the potential grids or vice versa. That transformation is made twice: the input charges are written to the potential grids and the electric field is calculated on the input grids. This section describes how that is done.

5.1 Grid association

If the potential grids and the input grids do not have the same size or location, it is necessary to use more than one input grid to find the charge on a potential grid. But it usually is not necessary to use all the input grids. We say that two grids Ω^l_Φ and Ω^λ_{in} are *associated* if data from Ω^l_Φ are needed to set data on Ω^λ_{in} or vice versa. Geometrically, this is the case when the areas of Ω^l_Φ and Ω^λ_{in} that are not occupied by their children, overlap (partially).

So, a potential grid Ω^l_Φ and an input grid Ω^λ_{in} are associated if

1. Ω^λ_{in} lies (partially) in Ω^l_Φ and
2. Ω^λ_{in} does not lie entirely in a child of Ω^l_Φ and
3. Ω^l_Φ does not lie entirely in a child of Ω^λ_{in}.

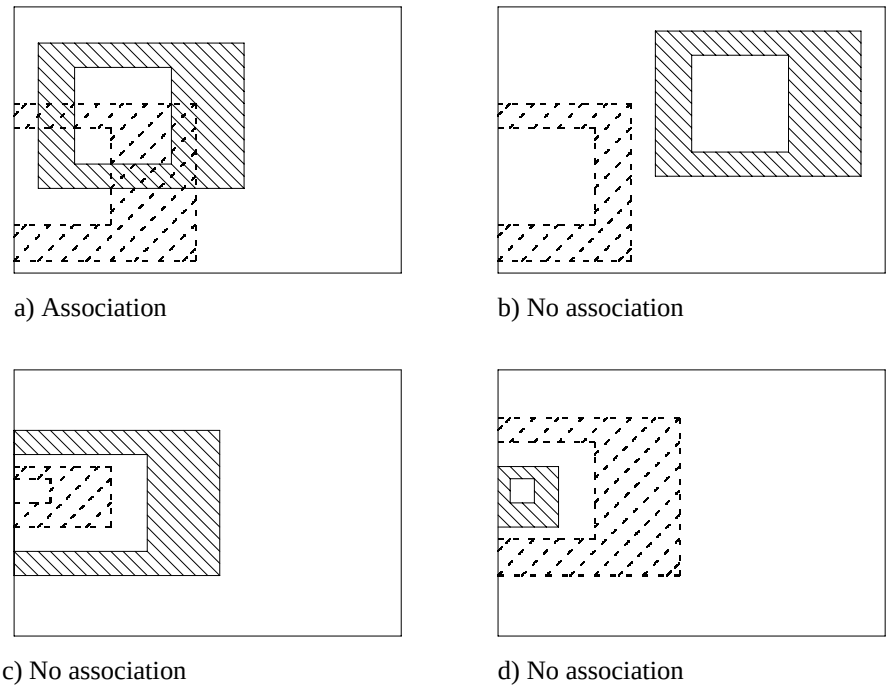
These situations are shown in figure 6.

5.2 Input charges

The input for the Poisson solver is a charge distribution on the input grids. But, in order to calculate the potential, we need to map this charge distribution to the potential grids (see equation (3.2)). This is done in two steps:

1. Calculate the charge on the entire input grids.
2. Map this charge to the potential grids.

Step 1 is needed because the evolution of the charge in time is only calculated on those parts of the grids Ω_{in} that are not covered by child grids. If we want to know the charge on an entire input grid, we have to map the charge on its child grids to the grid itself.



a) Association b) No association

c) No association d) No association

Figure 6: Association of a potential grid (drawn lines) and an input grid (dashed lines). The area not covered by a child grid is hatched.

This mapping is started on the finest grid. The charge in each cell of a parent grid Ω_{in}^{l} is found by averaging the charges in the four corresponding small cells in its child grid.

$$u_{i,j}^{l} = \tfrac{1}{4r_i^l}\left(r_{i-\frac{1}{2}}^{l}\left(u_{2i-1,2j}^{l+1} + u_{2i-1,2j-1}^{l+1}\right) + r_{i+\frac{1}{2}}^{l}\left(u_{2i,2j}^{l+1} + u_{2i,2j-1}^{l+1}\right)\right) / \qquad l = M-1, M-2, \cdots, 0. \tag{5.1}$$

The weighing with r is needed because the charge is not truly 2D, but 3D with radial symmetry.

If the input grids have fewer levels than the potential grids (i.e. the coarsest input grid is finer than the coarsest potential grid), then this procedure is also used to determine the charges on the lowest levels, that do not have any input grids.

The second step is the mapping from the input grids to the potential grids. Consider one potential grid Ω_{ϕ}^{l}. We start by writing the charges in all the input grids on the same level to the corresponding cells in Ω_{ϕ}^{l}. It is not necessary to take into account the input grids Ω_{in}^{λ} with $\lambda > l$, the finer input grids, since their charges are already mapped onto the input grids on level l in step 1.

Then, the charge is set in the cells in Ω_{ϕ}^{l} that have no corresponding cell on the same level in an input grid. These charges are set from the Ω_{in}^{λ} with $\lambda < l$, that are associated with Ω_{ϕ}^{l} (see section 5.1). At this moment, a third-order interpolation as in equation (3.4) is used.

5.3 Output electric field

Once the potential is calculated on the entire domain, the electric field on the input grids Ω_{in}^{λ} is set from this potential. The values of $\Phi_{i,j}$ are known in the middle of the cells, but the electric field is used to calculate fluxes in the cell faces and is therefore calculated in the centres of the cell faces (figure 7). The electric field is

found by taking the derivatives of the potential.

$$\begin{aligned} E_r &= -\frac{\partial \Phi}{\partial r}, \\ E_z &= -\frac{\partial \Phi}{\partial z}. \end{aligned} \tag{5.2}$$

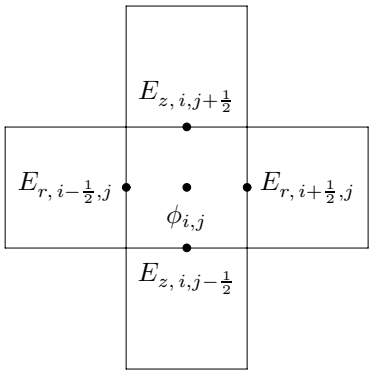


Figure 7: Location of the electric field components.

Since the fluxes are only calculated on those parts of a grid Ω_{in}^{λ} that are not covered by child grids, the electric field needs only be calculated there. We set the electric field on Ω_{in}^{λ} from all the potential grids that are associated with Ω_{in}^{λ}. In every part of Ω_{in}^{λ}, the field is calculated from the finest potential grid available, so each associated grid Ω_{ϕ}^{l} is only used in those areas where it is not covered by its child grids.

For the actual calculation of E, there are three possible situations: the level l of the potential grid can either be the same, higher or lower than λ. In these cases, different methods are used.

$l = \lambda$ When Ω_{ϕ}^{l} has the same mesh width as Ω_{in}^{λ}, the field is set here directly with a second-order accurate discretisation of equation (5.2),

$$\begin{aligned} E_{r,\,i+\frac{1}{2},j} &= \frac{\Phi_{i,j} - \Phi_{i+1,j}}{\Delta r}, \\ E_{z,\,i,j+\frac{1}{2}} &= \frac{\Phi_{i,j} - \Phi_{i,j+1}}{\Delta z}. \end{aligned} \tag{5.3}$$

$l > \lambda$ If the potential is known on coarser grids, two operations are needed to set the electric field in Ω_{in}^{λ}: a differentiation and an interpolation. We could first interpolate the potential to Ω_{in}^{λ} and then take the derivatives with (5.3). However, this procedure causes 'bumps' in the derivatives near the edges of the big cells, because the interpolations in two big neighbour cells do not give the same value on the cell edge between them. This difference may not be big, but it creates a significant change in the derivative.

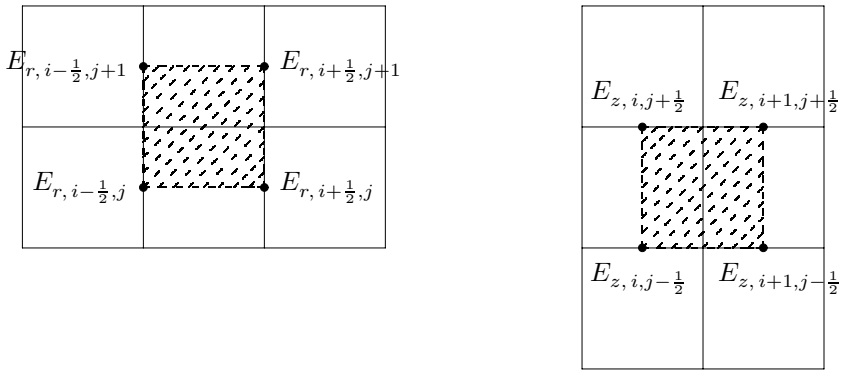
A better approach is to calculate the electric field first on the coarse potential grid, using equation (5.3) and to interpolate the *field* to Ω_{in}^{λ}, instead of the potential. The resulting electric field on Ω_{in}^{λ} is much smoother.

The interpolation is performed with a peicewise bilinear approximation, as shown in figure 8.

$$\begin{aligned}
E_r &= E_{r,\,i-\frac{1}{2},j} + \frac{E_{r,\,i+\frac{1}{2},j} - E_{r,\,i-\frac{1}{2},j}}{\Delta r}(r - r_{i-\frac{1}{2}}) + \frac{E_{r,\,i-\frac{1}{2},j+1} - E_{r,\,i-\frac{1}{2},j}}{\Delta z}(z - z_j) \\
&\quad + \frac{E_{r,\,i-\frac{1}{2},j} - E_{r,\,i+\frac{1}{2},j} - E_{r,\,i-\frac{1}{2},j+1} + E_{r,\,i+\frac{1}{2},j+1}}{\Delta r \Delta z}(r - r_{i-\frac{1}{2}})(z - z_j), \\
E_z &= E_{z,\,i,j-\frac{1}{2}} + \frac{E_{z,\,i+1,j-\frac{1}{2}} - E_{z,\,i,j-\frac{1}{2}}}{\Delta r}(r - r_i) + \frac{E_{z,\,i,j+\frac{1}{2}} - E_{z,\,i,j-\frac{1}{2}}}{\Delta z}(z - z_{j-\frac{1}{2}}) \\
&\quad + \frac{E_{z,\,i,j-\frac{1}{2}} - E_{z,\,i+1,j-\frac{1}{2}} - E_{z,\,i,j+\frac{1}{2}} + E_{z,\,i+1,j+\frac{1}{2}}}{\Delta r \Delta z}(r - r_i)(z - z_{j-\frac{1}{2}}).
\end{aligned} \tag{5.4}$$

This approximation is exact in the four corner points of the staggered cells and continuous over the cell faces. By substituting equation (5.3), it follows that

$$\begin{aligned}
E_r &= \frac{\Phi_{i-1,j} - \Phi_{i,j}}{\Delta r} + \frac{-\Phi_{i-1,j} + 2\Phi_{i,j} - \Phi_{i+1,j}}{\Delta r^2}(r - r_{i-\frac{1}{2}}) \\
&\quad + \frac{-\Phi_{i-1,j} + \Phi_{i,j} + \Phi_{i-1,j+1} - \Phi_{i,j+1}}{\Delta r \Delta z}(z - z_j) \\
&\quad + \frac{\Phi_{i-1,j} - 2\Phi_{i,j} + \Phi_{i+1,j} - \Phi_{i-1,j+1} + 2\Phi_{i,j+1} - \Phi_{i+1,j+1}}{\Delta r^2 \Delta z}(r - r_{i-\frac{1}{2}})(z - z_j), \\
E_z &= \frac{\Phi_{i,j-1} - \Phi_{i,j}}{\Delta z} + \frac{-\Phi_{i,j-1} + \Phi_{i+1,j-1} + \Phi_{i,j} - \Phi_{i+1,j}}{\Delta r \Delta z}(r - r_i) \\
&\quad + \frac{-\Phi_{i,j-1} + 2\Phi_{i,j} - \Phi_{i,j+1}}{\Delta z^2}(z - z_{j-\frac{1}{2}}) \\
&\quad + \frac{\Phi_{i,j-1} - 2\Phi_{i,j} + \Phi_{i,j+1} - \Phi_{i+1,j-1} + 2\Phi_{i+1,j} - \Phi_{i+1,j+1}}{\Delta r \Delta z^2}(r - r_i)(z - z_{j-\frac{1}{2}}).
\end{aligned} \tag{5.5}$$



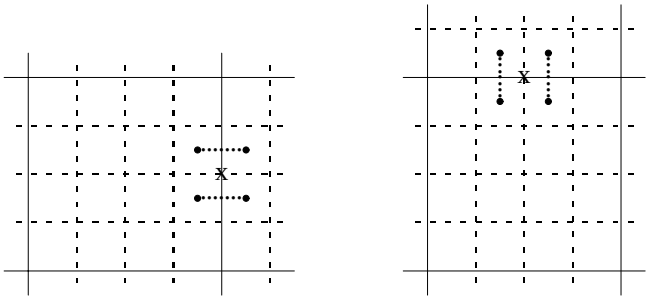
a) Radial component b) Axial component

Figure 8: Bilinear interpolation for the electric field components. The interpolation is valid in the hatched area.

$l > \lambda$ When the input grid has bigger cells than the potential grid, not all cells in the potential grid are needed to find the electric field. Since the field is basically a local variable, it is set from the four small cells

that are closest to the centre of the big cell face (figure 9). The electric field becomes:

$$E^{\lambda}_{r,\,i+\frac{1}{2},j} = \frac{\frac{1}{2}\left(\Phi^{l}_{2^{l-\lambda}i,2^{l-\lambda}\left(j-\frac{1}{2}\right)} + \Phi^{l}_{2^{l-\lambda}i,2^{l-\lambda}\left(j-\frac{1}{2}\right)+1} - \Phi^{l}_{2^{l-\lambda}i+1,2^{l-\lambda}\left(j-\frac{1}{2}\right)} - \Phi^{l}_{2^{l-\lambda}i+1,2^{l-\lambda}\left(j-\frac{1}{2}\right)+1}\right)}{\Delta r^{l}},$$
$$E^{\lambda}_{z,\,i,j+\frac{1}{2}} = \frac{\frac{1}{2}\left(\Phi^{l}_{2^{l-\lambda}\left(i-\frac{1}{2}\right),2^{l-\lambda}j} + \Phi^{l}_{2^{l-\lambda}\left(i-\frac{1}{2}\right)+1,2^{l-\lambda}j} - \Phi^{l}_{2^{l-\lambda}\left(i-\frac{1}{2}\right),2^{l-\lambda}j+1} - \Phi^{l}_{2^{l-\lambda}\left(i-\frac{1}{2}\right)+1,2^{l-\lambda}j+1}\right)}{\Delta z^{l}}. \tag{5.6}$$



a) E_r b) E_z

Figure 9: Electric field if the input grid has a lower level than the potential grid. The potential cells are dashed, the cells used are marked with •, the place of the field component with x.

6. Numerical results

In this section, numerical results are given for the algorithm described in the previous sections. The algorithm is implemented in a Fortran 90 program. For the actual solution of the Poisson equation, the subroutine Hstcyl from the library FISHPAK [5] is used. All calculations are performed on a SUN E250 workstation.

Results are shown for two test problems. The first is a 64 × 64 cell problem with a unit charge in four cells. This problem is used to show the properties of all parts of the algorithm. The second problem is a representative charge distribution in a streamer. CPU times for this problem are measured and compared with the solution on a uniform grid.

6.1 Point charge on square

The first test case is a small problem, with 64 × 64 cells on a fine grid. For simplicity, only one input grid is used. The charge on this grid is zero, except for four cells $\Omega_{i,j}$, $i \in [4,5]$, $j \in [32,33]$, where the charge is 1.

The first step in the solution is the restriction of the charges to coarser grids. For a maximum level $M = 3$, these restrictions are shown in figure 10. We see here how the charge is smeared out over a larger area on the coarse grids. The effect of the weighing with r is also clear.

The solutions are compared here with a solution on a fine, uniform grid. This solution is shown in figure 11, together with an estimation of the total error. The solution shows a smooth peak around the unit charge, the error is alsoo concentrated in the area around the charge. This error has sharp peaks, because the error is proportional to the third and fourth derivative of the solution. The maximum error is of $\mathcal{O}(10^{-5})$, the error away from the charge is of $\mathcal{O}(10^{-6})$.

Solutions on nested grids are computed for three settings of ϵ: 10^{-5}, 10^{-6} and 10^{-7}. Figure 12 shows the grids for these solutions (only the grids for level 1 – 3 can be seen, but the level 0 grid lies in the same location as the level 1 grid) and the difference of the nested-grid solution with the uniform solution. This difference

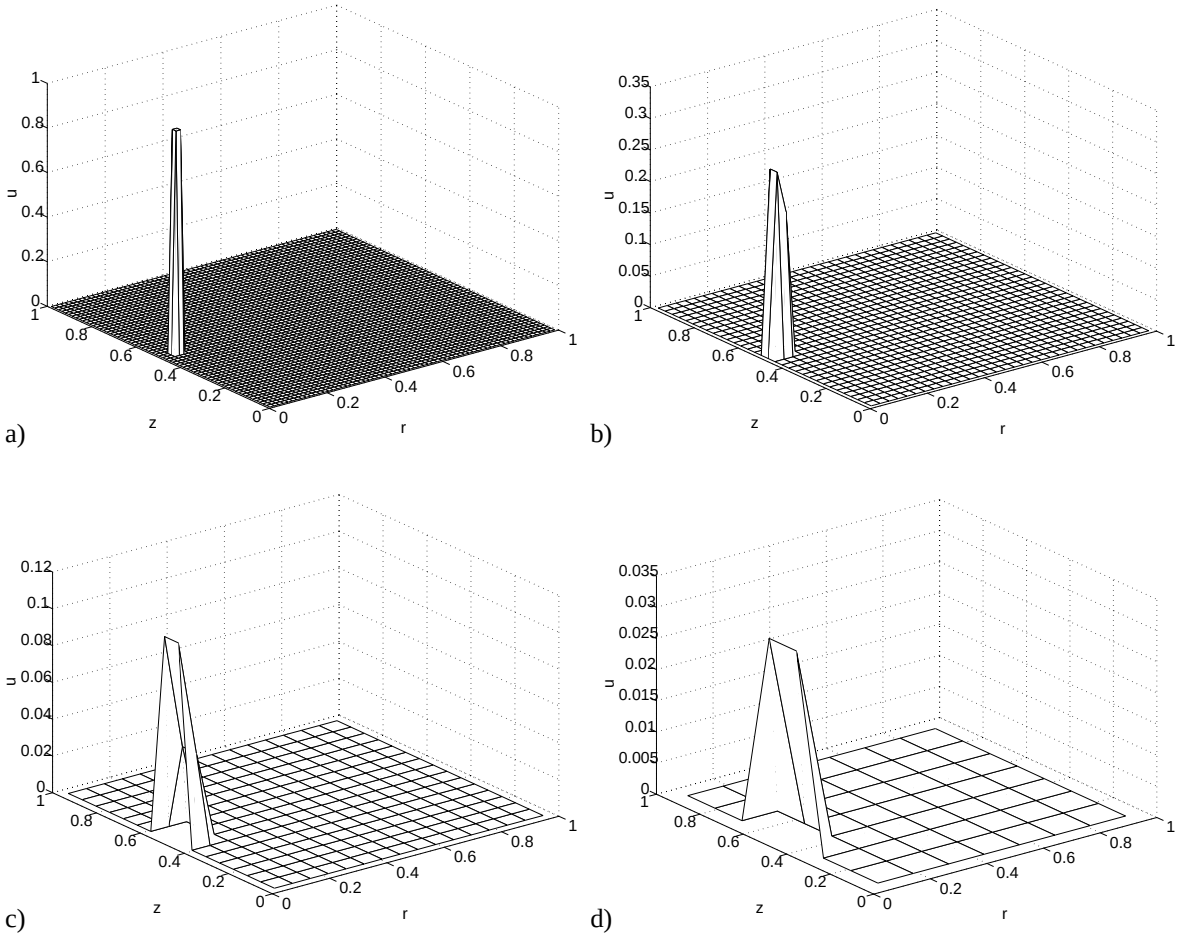
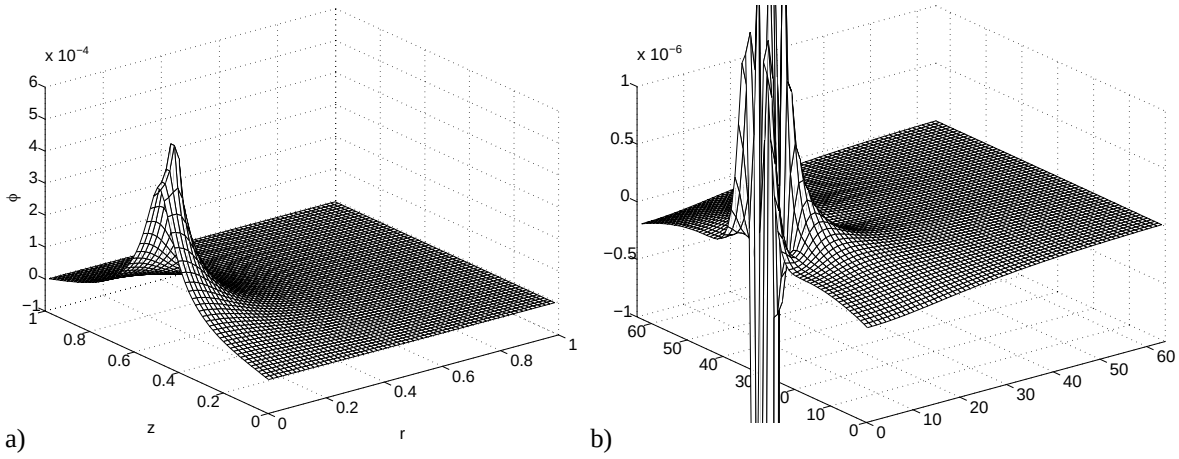


Figure 10: Charge distributions for 64 $\times$ 64 cell problem. (a) level 3, 64 $\times$ 64 cells. (b) level 2, 32 $\times$ 32 cells. (c) level 1, 16 $\times$ 16 cells. (d) level 0, 8 $\times$ 8 cells.



a) b)

Figure 11: Solution (a) and error (b) for 64 $\times$ 64 cell problem, on a uniform 64 $\times$ 64 cell grid. The error is estimated by subtracting the solution from the restriction of a solution on a 512 $\times$ 512 cell grid. The peak extends to $-4 \cdot 10^{-5}$.

is a combination of a global error pattern with local peaks. These peaks are the errors in the interpolation from the coarse Ω_ϕ^l to the fine grid Ω_{in}^M. The total difference must be comparable in size with $(M-1)\epsilon$ in equation (4.11): it decreases linearly with ϵ indeed. Therefore, we can make the extra errors arbitrarily small by choosing ϵ low enough.

The last step in the process is the calculation of the electric field. The uniform electric field components and their errors are shown in figure 13. Like for the potential, these errors are concentrated around the local charge. Figure 14 gives the extra errors in the adapted solution.

These errors are also reduced linearly when ϵ becomes smaller. For high ϵ, they decrease even faster, becasue the highest setting for ϵ gives rather large peaks (figure 14a). We see that the errors are concentrated around the boundaries of the fine grids. Therefore, it is probably advantageous to choose these grids somewhat larger than the error estimate for Φ indicates.

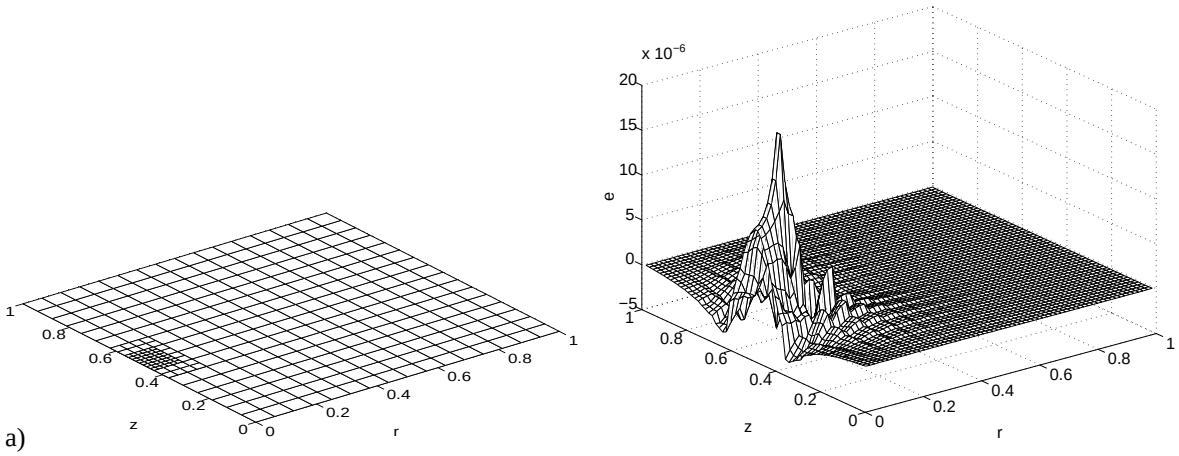
6.2 Realistic streamer

The results from the previous test problem are confirmed by a computation of the potential caused by the charge in an electric discharge, with a negative head and a positive tail (figure 15). The net charge is zero. The basic fine grid has 512 $\times$ 1024 cells and $M = 7$. Again, three settings for ϵ are used: 10^{-1}, 10^{-2} and 10^{-3}.

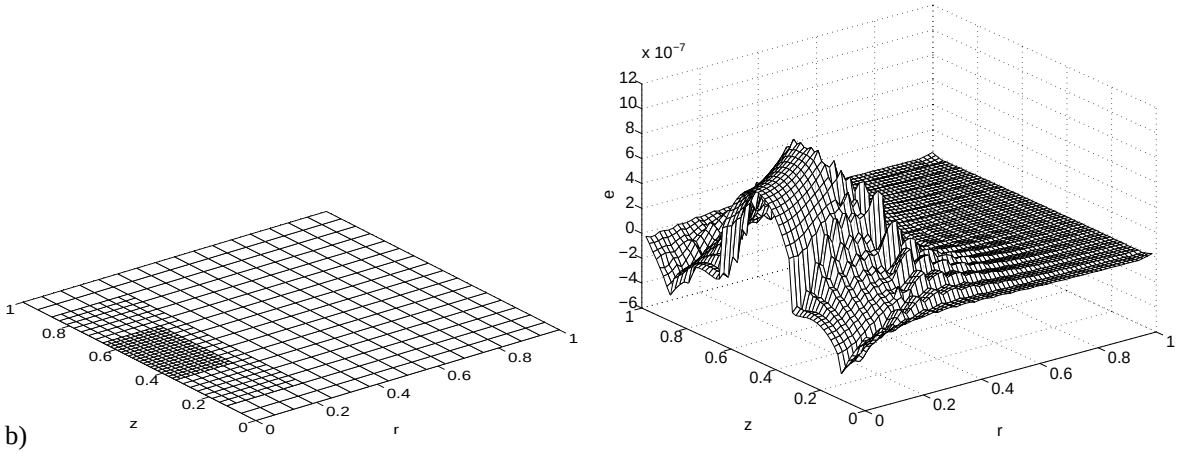
The grids are shown in figure 16a. Note how the finest levels have two grids for the high ϵ, one in the head and one in the tail of the streamer. The difference between the nested-grid solution and the uniform-grid solution is given in figure 16b. Also for this more complex charge distribution, the error scales linearly with ϵ.

The errors in the electric field are calculated, but there is no meaningful way to plot these results in a graph. Small oscillations as in figure 14 make the nature of the global error hard to recognize. But the size of the errors is estimated and given in table **??**. These errors look very much like the error graphs in figure 14, including some local sharp peaks when ϵ is chosen high. For lower ϵ, these peaks disappear almost.

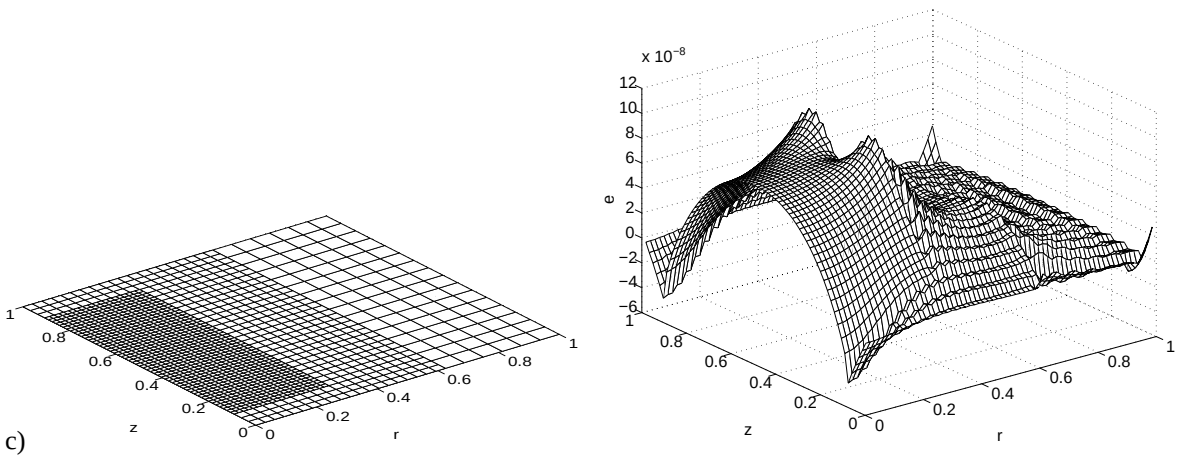
CPU times for the computation of the potential are shown in table 2. As expected from the size of the grids, the computation time changes much with ϵ. The CPU time is reduced between four and 20 times, compared with the uniform-grid computation.



a)

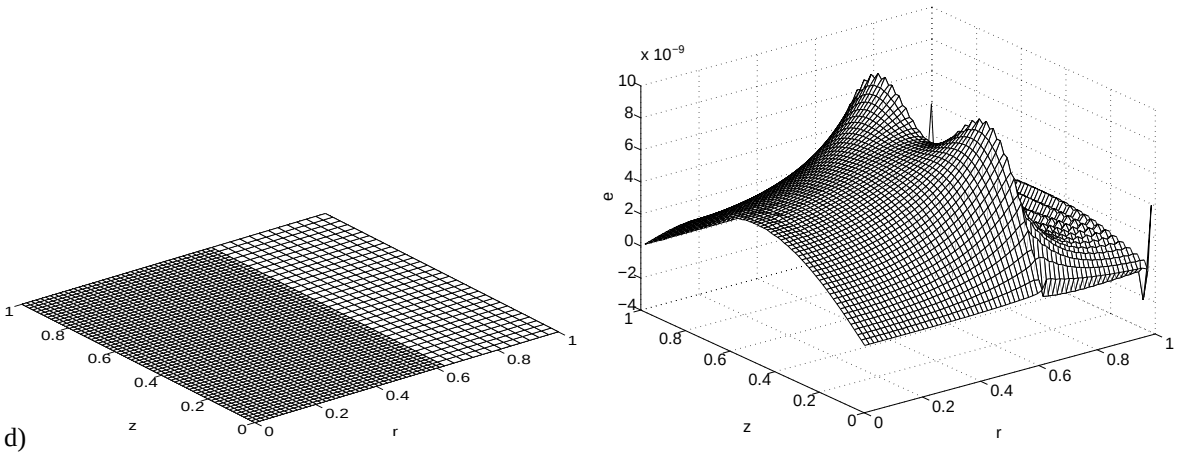


b)



c)

Figure 12: Grids and errors for 64 $\times$ 64 cell problem. The error indicated is the difference with the solution on a uniform 64 $\times$ 64 grid, i.e. the *extra* error caused by the multiple grids. (a) $\epsilon = 10^{-5}$, (b) $\epsilon = 10^{-6}$, (c) $\epsilon = 10^{-7}$.



d)

Figure 12: (cont.) (d) $\epsilon = 10^{-8}$.

ϵ	E_r		E_z	
	global	peak	global	peak
10^{-1}	$2 \cdot 10^{-3}$	$1 \cdot 10^{-1}$	$6 \cdot 10^{-3}$	$1.5 \cdot 10^{-1}$
10^{-2}	$5 \cdot 10^{-4}$	$2 \cdot 10^{-3}$	$4 \cdot 10^{-4}$	$1 \cdot 10^{-3}$
10^{-3}	$5 \cdot 10^{-5}$	$5 \cdot 10^{-5}$	$4 \cdot 10^{-5}$	$2 \cdot 10^{-4}$

Table 1: Realistic streamer, estimate of the extra errors in the electric field. The 'global' error is the magnitude for most of the larger errors, the 'peak' value occurs only very locally.

ϵ	CPU time (s)	% of unif.
uniform grid	0.960	
10^{-1}	0.052	5.4
10^{-2}	0.113	11.8
10^{-3}	0.258	26.9

Table 2: Realistic streamer, CPU times for the solution with three values of ϵ.

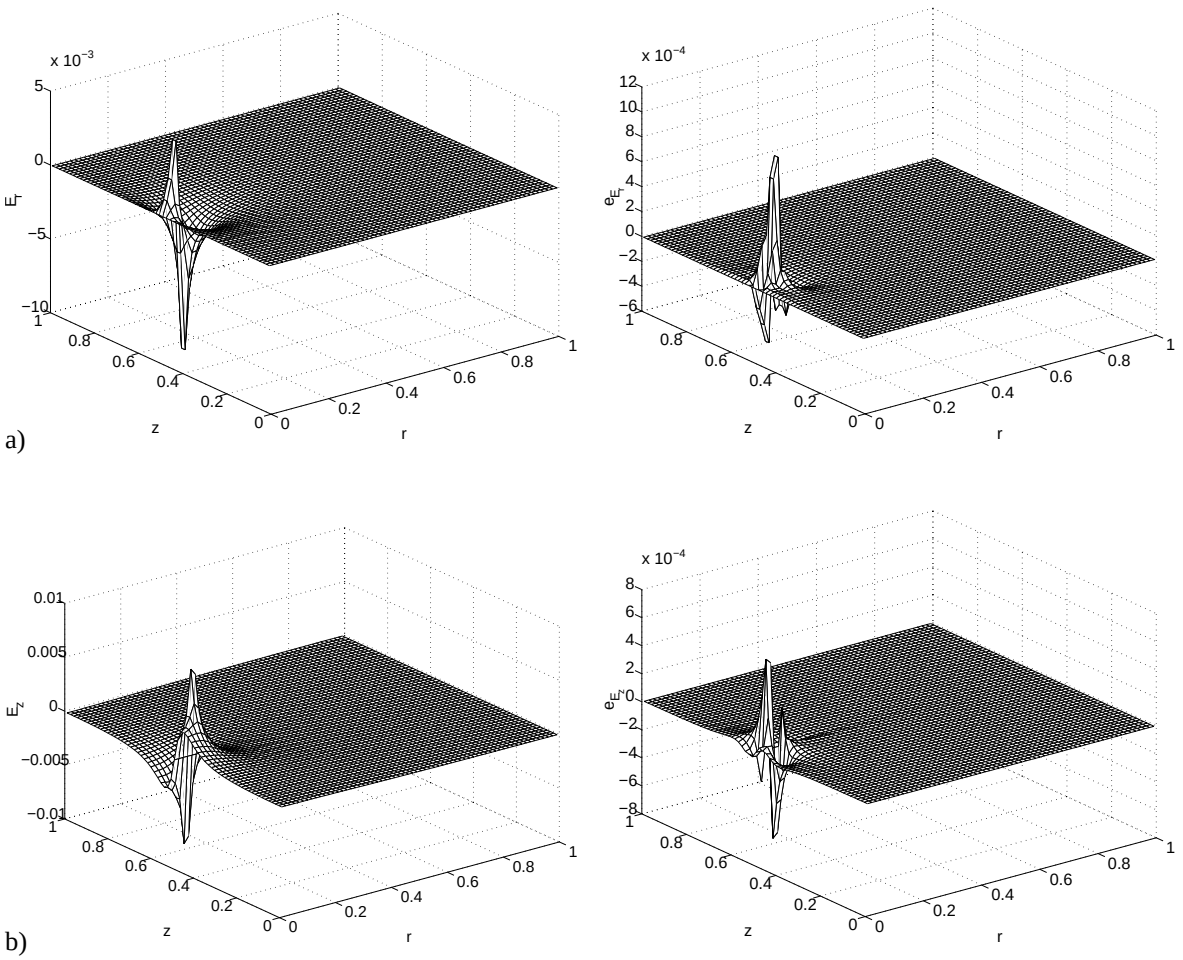


Figure 13: Electric field for 64 $\times$ 64 cell problem, on a uniform 64 $\times$ 64 cell grid. Solution and error for the radial component E_r (a) and the axial component E_z (b).

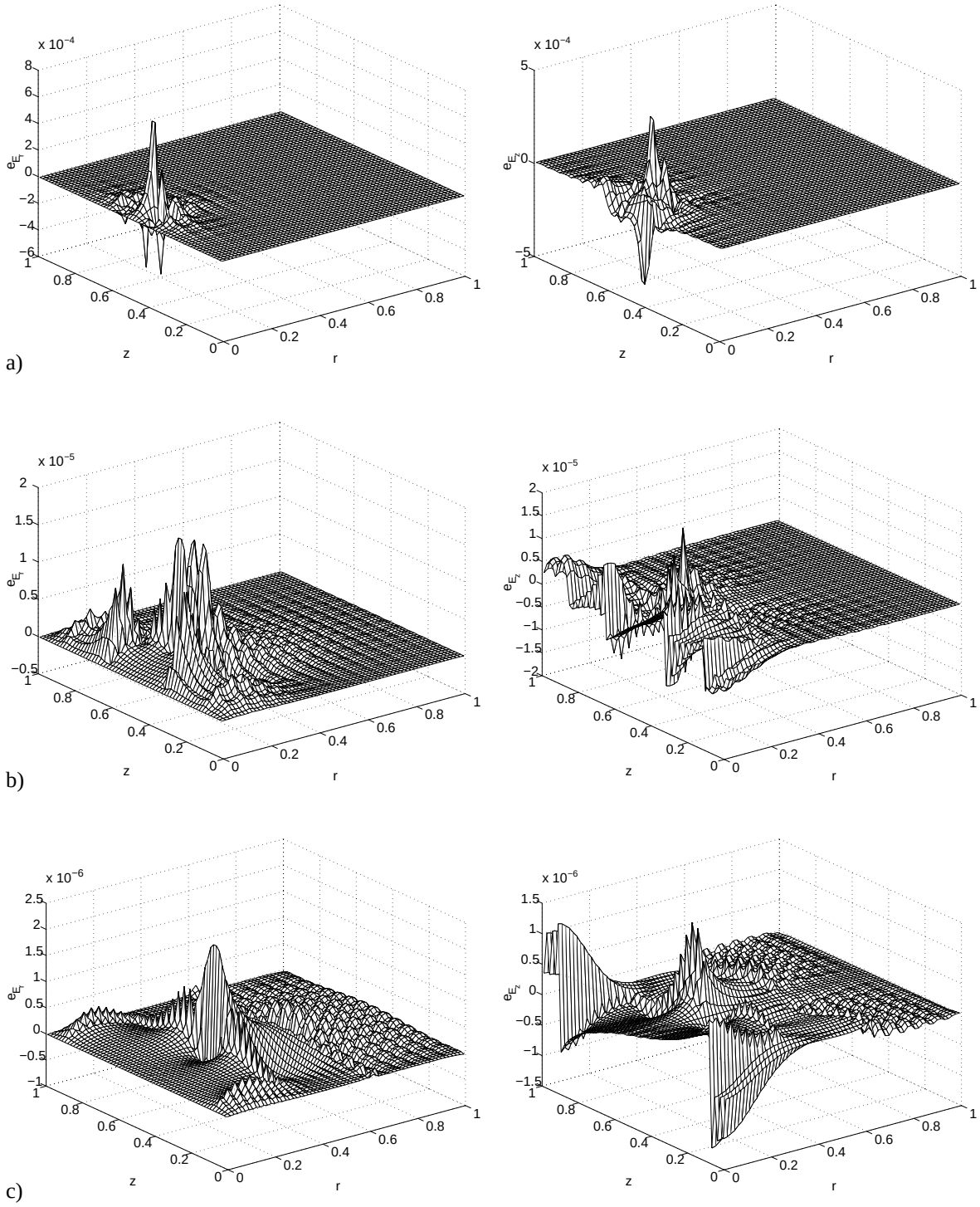
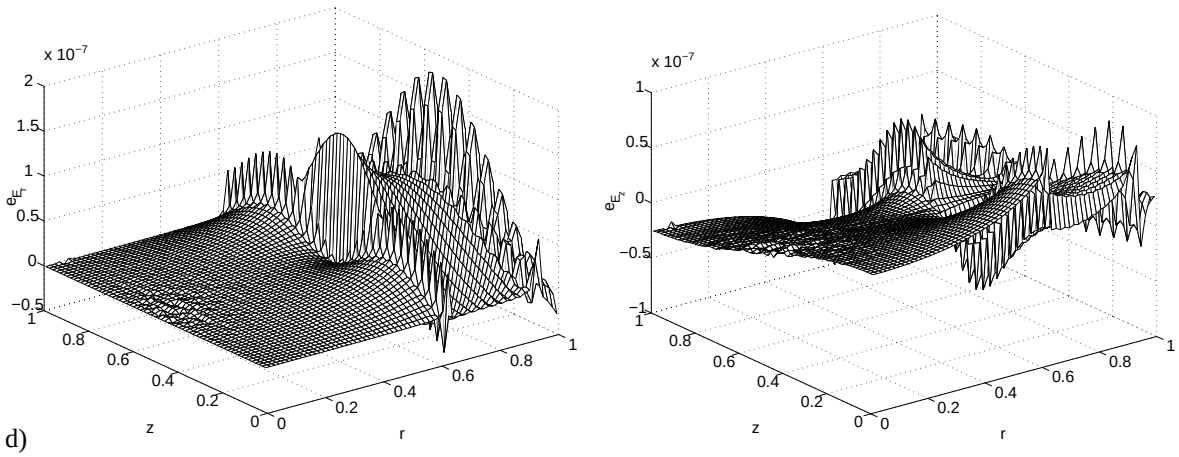


Figure 14: Errors in E_r (left) and E_z (right) for 64 $\times$ 64 cell problem. The error indicated is the difference with the solution on a uniform 64 $\times$ 64 grid. (a) $\epsilon = 10^{-5}$, (b) $\epsilon = 10^{-6}$, (c) $\epsilon = 10^{-7}$.



d)

Figure 14: (cont.) (d) $\epsilon = 10^{-8}$.

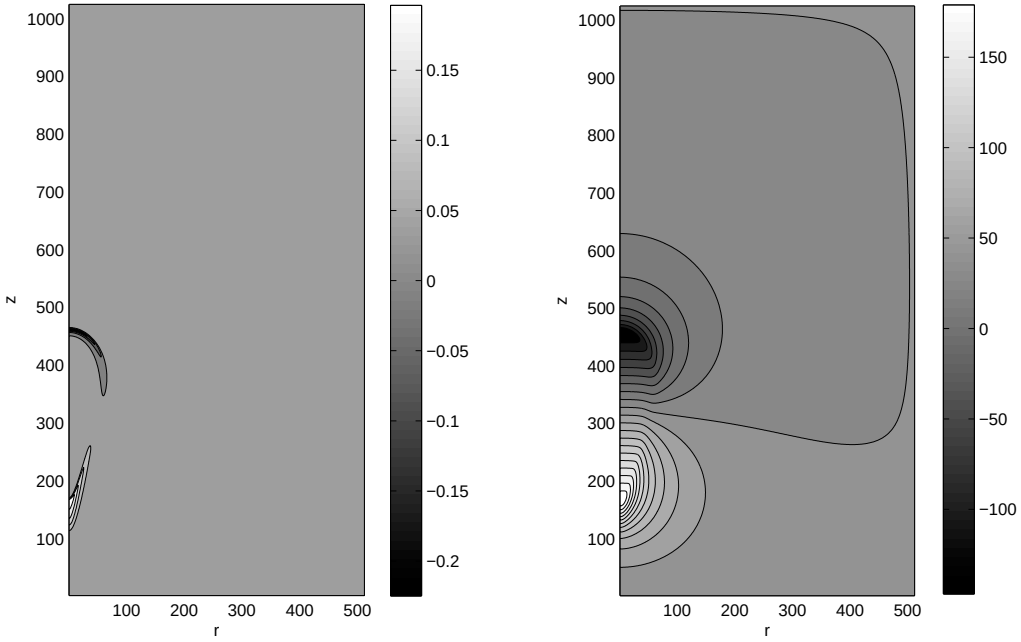


Figure 15: Realistic streamer. The charge f (left) and the potential Φ (right).

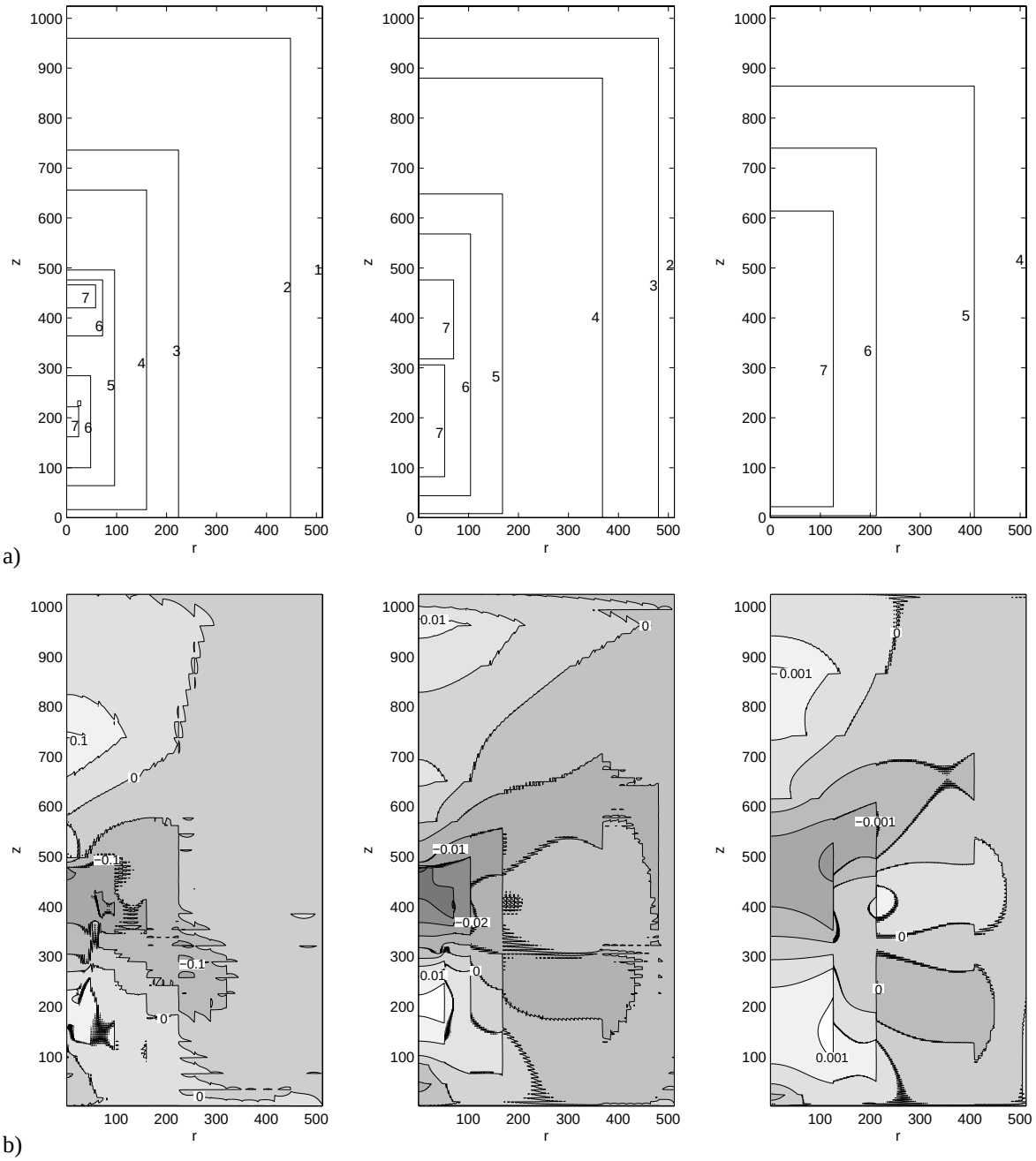


Figure 16: Realistic streamer. Grids and extra error in Φ for $\epsilon = 10^{-1}$, $\epsilon = 10^{-2}$ and $\epsilon = 10^{-3}$.

7. Conclusion

A solver for Poisson's equation is described that combines the speed of a direct solver with nested grids. The refinement criterion chosen, with the accompanying error analysis, keeps the error below a predefined error bound. Results from two test problems confirm this.

When a Poisson problem is solved that has a source term, concentrated on a part of the computational domain, the current solver gives a significant decrease in computation time, compared to a uniform-grid solver. This makes the method interesting for any type of Poisson problem with concentrated source terms.

ReferencesReferenties

1. E.F.F. Botta, K. Dekker, Y. Notay, A. van der Ploeg, C. Vuik, F.W. Wubs, P.M. de Zeeuw. *How fast the Laplace equation was solved in 1995.* Applied Numerical Mathematics **24** (1997), pp. 439 – 455.
2. A. Rocco, U. Ebert and W. Hundsdorfer. *Branching of negative streamers in free flight.* Physical Review E **64** 035102(R) (2002).
3. U. Schumann and R. Sweet. *A direct method for the solution of Poisson's equation with Neumann boundary conditions on a staggered grid of arbitrary size.* J. Comp. Phys. **20** (1976), pp. 171–182.
4. J. Wackers. *A nested-grid direct Poisson solver for concentrated source terms.* Submitted to Journal of Computational and Applied Mathematics.
5. http://www.scd.ucar.edu/softlib/FISHPAK.html