Local defect correction with different grid types

Nefedov, V.; Mattheij, R.M.M.

Published: 01/01/2001

Document Version
Publisher’s PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

• A submitted manuscript is the author's version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

Citation for published version (APA):

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

• Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain
• You may freely distribute the URL identifying the publication in the public portal

Take down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Download date: 22. Oct. 2018
Local defect correction with different grid types

by

V. Nefedov and R.M.M. Mattheij
Local Defect Correction with Different Grid Types

V.Nefedov* and R.M.M.Mattheij*

Abstract

For a Poisson problem with a solution having large gradients in (nearly) circular subregions a Local Defect method is considered. The problem on the global domain is discretised on a cartesian grid while the restriction of the problem to a circular subdomain is discretised on a polar grid. The two discretizations are then combined in an iterative way. We show that LDC can be viewed as an iterative method for the Poisson equation on a single composite cartesian-polar grid. The efficiency of methods is illustrated by numerical examples.

1 Introduction

A frequently occurring phenomenon in BVP's is a solution that has large gradients over a relatively small region see Figure 1. When a finite element method [3] is used, this situation can be resolved by discretising the problem using a mesh, which is finer in the region of the higher activity of the solution. However, since the finite difference or the finite volume methods make use of structured meshes, a locally refined mesh in this case is not attractive. A way out is to use a non-uniform tensor grid, but this is still not the best choice due to a considerable overload. A far better alternative would be a method which combines the solution on a global coarse mesh with the solution on a local finer mesh. Local Defect Correction (LDC), first introduced in [5] does exactly that. It is an iterative procedure which not just combines the solutions on the two grids but also improves the quality of the composite solution by providing information exchange between the grids. LDC was extensively analysed for the Poisson equation when cartesian coordinates are used for both global and local problems, cf [2], [4], see Figure 2a.

In this paper we study a more complicated case. If the problem is such that it contains a subregion where the solution has a nearly circularly symmetric behaviour, it makes sense to use polar coordinates rather that cartesian coordinates there. As a consequence there will be two types of discretisation grids, see Figure 2b, which need to be combined appropriately. Section 2 contains a description of discretisation techniques for both cartesian and polar coordinates. The Poisson equation in polar

*Scientific Computing Group, Faculteit der Wiskunde en Informatica, Technische Universiteit Eindhoven
coordinates is singular at the origin. This fact must be taken into account when constructing a discretisation scheme. In the third section we give a detailed formulation of LDC for the two coordinates systems situation. A single-matrix approach towards LDC is described in section 4, where we show, that LDC can be treated as an iterative way of solving a single approximation of the Poisson equation on a composite cartesian-polar grid. The last section contains numerical results demonstrating the efficiency of LDC. Besides illustrating the convergence we also consider a problem with a small parameter $\epsilon$ and demonstrate how our method yields error estimates de facto uniform in $\epsilon$.

2 The Poisson equation in cartesian and polar coordinates

In this paper we are interested in solving the Poisson equation numerically, the exact solution of which has locally higher activity areas having a more or less of a circular shape. This may occur when the right-hand side has a sudden change in a such an area causing a peak shape solution like in Figure 1. A natural idea is then to try and solve the equation in the region of higher activity of the solution in polar coordinates.
The global domain is, however, rectangular. To avoid problems associated with a complicated discretisation scheme on a composite grid and even more the resulting data structure, we intend to solve the equation on a rectangular global domain and a circular local domain separately and then combine the results by means of LDC, discussed in the next section. First, we need discretisation and solution methods for both rectangular and polar regions, which will therefore be the main concern of this section. Without loss of generality we may assume the global domain \( \Omega \) to be \((-1, 1) \times (-1, 1)\). In cartesian coordinates \((x, y)\) the Poisson equation reads

\[
-\nabla^2 u = -\frac{\partial^2 u}{\partial x^2} - \frac{\partial^2 u}{\partial y^2} = f(x, y), \quad (x, y) \in \Omega. \tag{1}
\]

Let equation (1) be complemented with a boundary condition of Dirichlet type

\[
u = g \text{ on } \Gamma := \partial \Omega. \tag{2}\]

Next, we define uniform grids \( \Omega_h \) and \( \Gamma_h \) for \( \Omega \) and \( \Gamma \) respectively by

\[
\Omega_h := \{(x, y) | (x, y) \in \Omega, \frac{x}{h} \in \mathbb{Z}, \frac{y}{h} \in \mathbb{Z}\}, \tag{3}
\]

\[
\Gamma_h := \{(x, y) | (x, y) \in \Gamma, \frac{x}{h} \in \mathbb{Z}, \frac{y}{h} \in \mathbb{Z}\}.
\]

The standard five-point stencil is used to discretise (1) at \((x, y) \in \Omega_h\), i.e.

\[
\frac{4u(x, y) - u(x - h, y) - u(x + h, y) - u(x, y - h) - u(x, y + h)}{h^2} = f(x, y). \tag{4}
\]
Assembling equations (4) for all points in $\Omega_h$ we obtain the discrete system

$$L_h u_h = f_h,$$  

(5)

where $L_h$ denotes the discrete Poisson operator and $u_h$ and $f_h$ are grid functions defined on $\Omega_h$

$$u_h(x, y) := u(x, y), \ f_h(x, y) := f(x, y), \ (x, y) \in \Omega_h.$$  

The next step is to deduce the system leading to the solution of the problem on the local domain. We are thus looking for the discrete solution of the Poisson problem on a circle $B_R := \{(x, y)|x^2 + y^2 \leq R^2, R < 1\}$

$$\nabla^2 \bar{u} = \bar{f} \text{ in } \bar{\Omega} := B_R, \ \bar{u} = u \text{ on } \bar{\Gamma} := \partial \bar{\Omega}.$$  

(6)

Tildes indicate that the problem is formulated in a local region (where the solution might exhibit a higher activity); $\bar{u}$ is a restriction of $u$ on $\bar{\Omega}$

$$u(x, y) \equiv \bar{u}(x, y), \ (x, y) \in \bar{\Omega}.$$  

Since we assumed that the higher activity occurs in a circular region it is reasonable to reformulate the problem there in polar coordinates $(r, \phi)$

$$- \frac{\partial^2 \bar{u}}{\partial r^2} - \frac{1}{r} \frac{\partial \bar{u}}{\partial r} - \frac{1}{r^2} \frac{\partial^2 \bar{u}}{\partial \phi^2} = \bar{f}(r, \phi), \ (r, \phi) \in \bar{\Omega} = [0, R] \times [0, 2\pi),$$

(7)

$$\bar{u}(R, \phi) = u(R, \phi), \ \phi = [0, 2\pi].$$

Keeping in mind the periodicity of the solution we have

$$\bar{u}(r, 0) = \bar{u}(r, 2\pi).$$  

(8)

We included the origin $(0, \phi)$ in $\bar{\Omega}$, though, strictly speaking, it belongs to the boundary. One reason is that, in cartesian coordinates, the origin is located inside $B_R$. We still have no condition on $\bar{u}(0, \phi)$. An extra condition can be derived for some special cases [7], [8]. For instance, when the problem possesses rotational symmetry, i.e. $\partial^2 \bar{u}/\partial \phi^2 \equiv 0$, the radial derivative of the solution is equal to zero at the origin

$$\frac{\partial \bar{u}}{\partial r}(0, \phi) = 0.$$  

Another condition can be obtained if $\bar{f}(r, \phi) \equiv 0$. Then, using the mean value property of the harmonic functions, we obtain

$$\bar{u}(0, \phi) = \int_0^{2\pi} u(R, \phi) d\phi.$$  

It turns out that the proper choice of the grid and the discretisation ensures the fact that we do not need the condition at the origin. Indeed, let us introduce a tensor grid in polar coordinates

$$\tilde{\Omega}_h := \{(r, \phi)|(r, \phi) \in \bar{\Omega}, \ r \frac{\phi}{\Delta \phi} \in \mathbb{Z}, \ \frac{\phi}{\Delta \phi} \in \mathbb{Z}\}.$$  

(9)
Though $\tilde{\Omega}_h$ depends on two parameters, $\tilde{h}$ and $\Delta \phi$, we can eliminate dependence on $\phi$ assuming that

$$R \Delta \phi \sim h,$$

i.e. the length of the curve with angle equal to $\Delta \phi$ is approximately equal to $\tilde{h}$.

The next assumption we make on $\Delta \phi$ is needed for construction of our discretisation. We assume that

$$\frac{1}{4} \frac{2\pi}{\Delta \phi} \in \mathbb{N},$$

i.e. the total number of sectors is proportional to 4. We will need this condition later, when constructing the discrete equation for the origin. Beside the grid $\tilde{\Omega}_h$, we introduce a grid on the boundary $\tilde{\Gamma}$ of $\tilde{\Omega}_h$,

$$\tilde{\Gamma}_h = \{(R, \phi) | (R, \phi) \in \tilde{\Gamma}, \frac{\phi}{\Delta \phi} \in \mathbb{Z}\}.$$

Let us now consider the point $(r, \phi) \in \tilde{\Omega}_h$ away from the origin. In order to obtain a discrete Poisson equation we employ four neighbouring points as well, see Figure 5

$$\begin{align*}
\frac{2\tilde{u}(r, \phi) - \tilde{u}(r - \tilde{h}, \phi) - \tilde{u}(r + \tilde{h}, \phi)}{\tilde{h}^2} + \frac{1}{r} \frac{\tilde{u}(r - \tilde{h}, \phi) - \tilde{u}(r + \tilde{h}, \phi)}{2\tilde{h}} \\
+ \frac{1}{r^2} \frac{2\tilde{u}(r, \phi) - \tilde{u}(r, \phi - \Delta \phi) - \tilde{u}(r, \phi + \Delta \phi)}{(\Delta \phi)^2} = \tilde{f}(r, \phi).
\end{align*}$$

(11)

Periodicity of the solution should be taken into account for the points $(r, 0)$. At the origin (11) can not be applied due to the singular behaviour of the coefficients $1/r$ and $1/r^2$. We use a discretisation of the cartesian formulation instead. At the point $(0, 0)$ we have

$$\frac{4\tilde{u}(0, 0) - \tilde{u}(\tilde{h}, 0) - \tilde{u}(\tilde{h}, \pi/2) - \tilde{u}(\tilde{h}, \pi) - \tilde{u}(\tilde{h}, 3\pi/2)}{\tilde{h}^2} = \tilde{f}(0, 0).$$

(12)

Because of (10), the points $(\tilde{h}, 0), (\tilde{h}, \pi/2), (\tilde{h}, \pi), (\tilde{h}, 3\pi/2)$ all belong to $\tilde{\Omega}_h$. As we recall, in polar coordinates the origin is represented not only by the point $(0, 0)$ but also by the points of the form $(0, \phi)$. For these points we have

$$\tilde{u}(0, \phi) = \tilde{u}(0, 0), \quad (0, \phi) \in \tilde{\Omega}_h.$$

(13)
The discrete equations (11), (12), (13) when written for all points of $\tilde{\Omega}_h$, form a linear system

$$\tilde{L}_h \tilde{u}_h = \tilde{f}_h,$$

where $\tilde{L}_h$ is a discrete Laplace operator in polar coordinates, $\tilde{u}_h$ and $\tilde{f}_h$ are grid functions that attain values $\tilde{u}(r, \phi)$ and $\tilde{f}(r, \phi)$ for $(r, \phi) \in \tilde{\Omega}_h$ correspondingly.

### 3 Local Defect Correction

We have derived a discretisation of the Poisson equation (1) on $\Omega$ and on $\tilde{\Omega}$ a circular subdomain of $\Omega$, where we have assume the solution to have higher gradients. One way to resolve the problem is to make the mesh covering the whole domain, i.e. $\Omega_h$, fine enough to resolve the large gradients. Of course, this might lead to a situation where the number of points of $\Omega_h$ is so large that it becomes unpractical, or even impossible, to solve the discrete problem (5). We thus opt for a different approach: we solve the problem in $\Omega$, then solve the problem in $\tilde{\Omega} \subset \Omega$ and combine results obtained in an appropriate way. We begin by solving (5). Find $u_h$ such that

$$L_h u_h = f_h, \quad u_h \in V(\Omega_h), \quad f_h \in V(\Omega_h),$$

$$u(x, y) = g(x, y).$$

(15)

The size $h$ of $\Omega_h$ should satisfy two requirements. First, $h$ should be large enough, so that (15) is a relatively small problem, i.e. can be solved at low computational costs. Secondly, $u_h$ should yield a good approximation of $u^*(x, y)$ – the exact solution of (1,2) – for $(x, y) \in \Omega \setminus \tilde{\Omega}$, i.e. the area where the exact solution does not have rapid changes.

The second step is to solve (14), i.e. find $\tilde{u}_h$ such that

$$\tilde{L}_h \tilde{u}_h = \tilde{f}_h, \quad \tilde{u}_h \in V(\tilde{\Omega}_h), \quad \tilde{f}_h \in V(\tilde{\Omega}_h).$$

(16)

Since we assumed $\tilde{u}$ to have large gradients in $\tilde{\Omega}$ we need $\tilde{h}$ to be rather small. We need to complement (16) with a boundary condition of some sort. The boundary conditions for the continuous analogue of (16) look like

$$\tilde{u} = u \text{ on } \tilde{\Gamma}.$$  

(17)

We cannot use the discrete version of (17), because, $u_h$ is not defined at the grid points belonging to $\tilde{\Gamma}_h$. We construct the boundary conditions for (16) by interpolation: linear or quadratic, see Figure 6. Denoting the interpolation operator as $p : V(\Omega_h) \to V(\tilde{\Gamma}_h)$ we have the following boundary conditions for (16)

$$\tilde{u}_h(r, \phi) = pu_h, \quad (r, \phi) \in \tilde{\Gamma}_h.$$  

(18)

In general, the composite approximation $(u_h, \tilde{u}_h)$ turns out to be not accurate enough. This fact can easily be explained by the observation that no information about the solution on $\tilde{\Omega}_h$ was transferred to $\Omega_h$. To mend this situation we create a correction for the right-hand side of (15) with the help of $\tilde{u}_h$. First, we introduce
Figure 6: Points used in linear (a) and quadratic (b) interpolation.

Figure 7: $\Omega_h'$ (a) and $\Omega''_h$ (b).
a splitting of $\Omega_h$ into two subgrids $\Omega'_h$ (Figure 7a) and $\Omega''_h$ (Figure 7b). The subgrid $\Omega'_h$ is formed by the points of $\Omega_h$, the distance from which to the center is less than $R - h$, the rest constitutes $\Omega''_h$. Formally, we have
\[
\Omega'_h = \{(x,y) | (x,y) \in \Omega_h, x^2 + y^2 < (R - h)^2\}, \quad \Omega''_h = \Omega_h \setminus \Omega'_h
\]  
Using the splitting (19) we define a grid function $w_h$
\[
w_h(x,y) := \begin{cases} 
\begin{array}{l}
  u_h(x,y), \quad (x,y) \in \Omega''_h \\
  r\tilde{u}_h, \quad (x,y) \in \Omega'_h,
\end{array}
\end{cases}
\]  
where $r$ is the restriction operator $r : V(\tilde{\Omega}_h) \rightarrow V(\Omega'_h)$, yielding the values of the grid functions on $\Omega'_h$ by means or linear of quadratic interpolation from $\tilde{\Omega}_h$. The grid function $w_h$ can be considered as an "updated approximation" of the solution on a global grid. Next, we introduce the defect $d_h \in V(\Omega_h)$ defined by
\[
d_h = \chi_{\Omega'_h}(L_h w_h - f_h).
\]  
In (21) $\chi_{\Omega'_h}$ is a characteristic function of $\Omega'_h$
\[
\chi_{\Omega'_h}(x,y) := \begin{cases} 
1, \quad (x,y) \in \Omega'_h \\
0, \quad (x,y) \notin \Omega'_h.
\end{cases}
\]  
By construction $d_h$ is not zero in $\Omega'_h$ only. The defect contains information about the behaviour of $\tilde{u}_h$ in $\tilde{\Omega}_h$. By adding the defect to the right-hand side of (15) we transfer this information to the problem on a global grid
\[
L_h u^1_h = f_h + d_h, \quad u^1_h \in V(\Omega_h), \quad f^h \in V(\Omega_h),
\]
\[
u^1_h(x,y) = g(x,y), \quad (x,y) \in \Gamma_h.
\]  
The solution of (22) provides the boundary conditions for the new problem on a local mesh
\[
\tilde{L}_h \tilde{u}^1_h = \tilde{f}^1_h, \quad \tilde{u}^1_h \in V(\tilde{\Omega}_h), \quad \tilde{f}^1_h \in V(\Omega_h),
\]
\[
\tilde{u}^1_h(r,\phi) = pu^1_h, \quad (r,\phi) \in \tilde{\Gamma}_h.
\]  
The pair $(u^1_h, \tilde{u}^1_h)$ forms the next composite approximation. The solution on local mesh $\tilde{u}^1_h$ is again used to create the defect for the next update of the right-hand side of the global grid problem. The LDC algorithm can be thus summarised as follows:

Algorithm 1. Local Defect Correction
Step 0. $i := 0; \quad d^0 = 0;$

Step 1. Solve for $u^1_h$
\[
L_h u^1_h = f^1_h + d^h, \quad u^1_h \in V(\Omega_h), \quad u^1_h(x,y) \in g(x,y), \quad (x,y) \in \Gamma_h.
\]  
Step 2. Create the boundary conditions for the problem on a local mesh
\[
\tilde{u}_h(R,\phi) = pu^1_h, \quad (R,\phi) \in \tilde{\Gamma}_h.
\]
Step 3. Solve for $\tilde{u}_h^i$

$$\tilde{L}_h \tilde{u}_h^i = \tilde{f}_h.$$ 

Step 4. Create grid function $w_h \in V(\Omega_h)$

$$w_h(x, y) = \begin{cases} u_h(x, y), & (x, y) \in \Omega'_h \\ r\tilde{u}_h^i, & (x, y) \in \Omega''_h \end{cases}$$

Step 5. Compute defect $d_h \in V(\Omega_h)$

$$d_h = \chi_{\Omega''_h} (L_h u_h - f_h).$$

Step 6. Form composite approximation $(u_h^i, \tilde{u}_h^i)$; return to step 1 if residuals are not small enough.

4 Algebraic properties of LDC

In the previous section we introduced LDC as an iterative method which combines the solution of the problem discretised on the global (relatively) coarse mesh and the solution of the subproblem discretised on a local finer mesh of a different kind. It turns out that the solution, yielded as a result of LDC iterations, is at the same time the solution of a linear system, which we will derive later on in this section. Thus, we intend to show that to find the LDC solution is equivalent to solving the larger linear system in a special iterative manner. Before we proceed with the formalities we need a new notation for the splitting of the discrete operators. We define the splitting $\tilde{L}_h^{(1)} : V(\tilde{\Omega}_h) \rightarrow V(\tilde{\Omega}_h)$ and $\tilde{L}_h^{(2)} : V(\tilde{\Omega}_h) \rightarrow V(\tilde{\Gamma}_h)$ according to

$$\left(\tilde{L}_h^{(1)} \tilde{z}_h|_{\tilde{\Omega}_h}\right)(r, \phi) + \left(\tilde{L}_h^{(2)} \tilde{z}_h|_{\tilde{\Gamma}_h}\right)(r, \phi) = \left(\tilde{L}_h \tilde{z}_h\right)(r, \phi), \quad (r, \phi) \in \tilde{\Omega}_h, \quad \tilde{z}_h \in V(\tilde{\Omega}_h). \quad (25)$$

The splitting indicates explicitly that $\tilde{L}_h$ acts on grid point points belonging to both $\tilde{\Omega}_h$ and $\tilde{\Gamma}_h$. We need a similar notation for the parts of $L_h$ at the points of $\Omega_h$ and $\Omega''_h$. The operators $L_h^{(1,1)} : V(\Omega_h) \rightarrow V(\Omega'_h)$ and $L_h^{(1,2)} : V(\Omega_h) \rightarrow V(\Omega''_h)$ together constitute $L_h$ acting on an arbitrary grid function $z_h$ at the points $\Omega_h$, i.e

$$\left(L_h^{(1,1)} z_h|_{\Omega'_h}\right)(x, y) + \left(L_h^{(1,2)} z_h|_{\Omega''_h}\right)(x, y) = \left(L_h z_h\right)(x, y), \quad (x, y) \in \Omega_h, \quad z_h \in V(\Omega_h). \quad (26)$$

One more splitting we shall employ is again the splitting of $L_h$, but now acting on an arbitrary grid function at the points of $\Omega''_h$. Similar to (26) $L_h^{(2,1)} : V(\Omega''_h) \rightarrow V(\Omega'_h)$ and $L_h^{(2,2)} : V(\Omega''_h) \rightarrow V(\Omega''_h)$ form

$$\left(L_h^{(2,1)} z_h|_{\Omega''_h}\right)(x, y) + \left(L_h^{(2,2)} z_h|_{\Omega''_h}\right)(x, y) = \left(L_h z_h\right)(x, y), \quad (x, y) \in \Omega''_h, \quad z_h \in V(\Omega_h). \quad (27)$$

The grid splitting $\Omega_h = \Omega'_h \cup \Omega''_h$ thus induces the direct sum of grid function space

$$V(\Omega_h) = V(\Omega'_h) \oplus V(\Omega''_h). \quad (28)$$
Using definitions (26) and the splitting (28) we can formally write the global grid iteration of LDC as

\[ L_h u_h^i = \left[ \begin{array}{c}
L_h^{(1,1)} r \tilde{u}_h^i + L_h^{(1,2)} u_h|_{\Omega''_h} \\
fh|_{\Omega''_h}
\end{array} \right], \quad i \geq 1. \]  

The local grid iteration can be in turn formalised using (25) as

\[ \tilde{L}_h^{(1)} \tilde{u}_h^i + \tilde{L}_h^{(2)} p u_h|_{\Omega''_h} = \tilde{f}_h, \quad i \geq 0. \]

Let us assume that the LDC iteration converges and yield as a result a fixed point \((u_h, \bar{u}_h)\). From (29,30) it follows that \((u_h, \bar{u}_h)\) satisfies the coupled system

\[ L_h u_h - \left[ \begin{array}{c}
L_h^{(1,1)} r \bar{u}_h + L_h^{(1,2)} u_h|_{\Omega''_h} \\
fh|_{\Omega''_h}
\end{array} \right] = \left[ \begin{array}{c}
0 \\
fh|_{\Omega''_h}
\end{array} \right] \text{ on } \Omega_h, \tag{31} \]

\[ \tilde{L}_h^{(1)} \tilde{u}_h^i + \tilde{L}_h^{(2)} p u_h|_{\Omega''_h} = \tilde{f}_h \text{ on } \tilde{\Omega}_h. \tag{32} \]

The subdomain \(\tilde{\Omega}\) on which we have defined \(\tilde{\Omega}_h\) is a subset of the global domain \(\Omega\). The exact solution of (6) coincides with the exact solution of (1) on \(\tilde{\Omega}\), i.e.

\[ u \equiv \bar{u} \text{ on } \tilde{\Omega}. \]

A similar relation should couple the LDC iterates as well. The elements of the LDC fixed point \(u_h\) and \(\bar{u}_h\) are defined at different grid points, thus they can not be simply equal to each other. Instead, it turns out \(u_h\) can be obtained as restriction of \(\bar{u}_h\) at the grid points belonging to \(\Omega''_h\). We designate this result as Lemma 1

**Lemma 1** If \((u_h, \bar{u}_h)\) satisfies the coupled system (31,32), then

\[ u_h|_{\Omega''_h} = r \bar{u}_h \tag{33} \]

**Proof.** From (31) and (26) we have

\[ (L_h u_h)(x,y) - (L_h^{(1,1)} r \bar{u}_h)(x,y) - (L_h^{(1,2)} u_h|_{\Omega''_h})(x,y) = 0, \quad x \in \Omega_h', \]

\[ (L_h u_h)(x,y) - (L_h^{(1,1)} u_h|_{\Omega''_h})(x,y) - (L_h^{(1,2)} u_h|_{\Omega''_h})(x,y) = 0, \quad x \in \Omega'_h. \]

Thus, after subtracting, the following holds

\[ L_h^{(1,1)} (u_h|_{\Omega''_h} - r \bar{u}_h) = 0. \tag{34} \]

Since \(L_h^{(1,1)}\) is nonsingular (34) is equivalent to (33). If the pair \((u_h, \bar{u}_h)\) is a fixed point of LDC iteration, we can construct the grid function defined on a composite grid \(\Omega''_h \oplus \tilde{\Omega}_h\), which satisfies a composite approximation of the Poisson equation constructed explicitly in Lemma 2
Lemma 2 If \((u_h, \bar{u}_h)\) satisfies the coupled system \((31,32)\), the composite grid function
\[
\hat{u} := \begin{bmatrix} u_h|\Omega_h' \\ \bar{u}_h \end{bmatrix}
\]
satisfies
\[
\hat{L}\hat{u} = \hat{f}
\]
with
\[
\hat{L} := \begin{bmatrix} L_h^{(2,2)} & L_h^{(2,1)} \\ \bar{L}_h^{(2)} & \bar{L}_h^{(1)} \end{bmatrix},
\]
\[
\hat{f} := \begin{bmatrix} f_h|\Omega_h' \\ \bar{f}_h \end{bmatrix}.
\]

Proof. The first part of \((36)\) follows from \((32)\). And \((27), (31)\) results in
\[
L_h^{(2,2)} u_h|\Omega_h' + L_h^{(2,1)} r\bar{u}_h = f_h|\Omega_h'.
\]
Now \((36)\) follows by the definition of \(\hat{u}\).

The next lemma provides us with the result, which is an inverse statement of Lemma 2.

Lemma 3 If \(\hat{u}\) satisfies the composite grid problem \((36)\), then \((u_h, \bar{u}_h)\) defined by
\[
u_h|\Omega_h' = \hat{u}|\Omega_h', \quad u_h|\Omega_h' = r\hat{u}|\Omega_h',
\]
\[
\bar{u}_h = \hat{u}|\Omega_h',
\]
satisfies the coupled system \((31,32)\).

Proof. Since \(\hat{u}\) satisfies \(\hat{L}\hat{u} = \hat{f}\) we obtain from
\[
\bar{L}_h^{(1)} \hat{u}|\Omega_h' + \bar{L}_h^{(2)} p\hat{u}|\Omega_h' = \bar{f}_h,
\]
\[
L_h^{(2,2)} u_h|\Omega_h' + L_h^{(2,1)} r\bar{u}_h|\Omega_h' = f_h|\Omega_h'.
\]
Using definitions \((39)\) and \((40)\) we obtain
\[
\bar{L}_h^{(1)} \hat{u}_h + \bar{L}_h^{(2)} p u_h|\Omega_h' = \bar{f}_h,
\]
\[
L_h^{(2,2)} u_h|\Omega_h' + L_h^{(2,1)} u|\Omega_h' = f_h|\Omega_h',
\]
\[
L_h^{(1,1)} u_h|\Omega_h' + L_h^{(1,2)} u_h|\Omega_h' = L_h^{(1,1)} r\bar{u}_h + L_h^{(1,2)} u_h|\Omega_h'.
\]
Now it follows from \((26), (27)\) that \((u_h, \bar{u}_h)\) satisfies \((31)\) and \((32)\). Combination of the results of Lemmas 1, 2 and 3 proves the following theorem:

Theorem 1 The fixed points of LDG algorithm iff they are the solutions of the composite grid problem \((36)\).
5 Numerical Experiments

In this section we consider two examples. The first example illustrates a typical rate of convergence of LDC and approximation order of the LDC solution. In the second example we demonstrate a typical phenomenon occurring in a problem with a small parameter determining a (local) high activity of the solution. We show, that the latter type of problems requires a number of grid points which is de facto independent of this parameter.

Example 1.

Let us consider a domain $\Omega = (-1,1) \times (-1,1)$ and on it the Poisson equation

$$-\frac{\partial^2 u}{\partial x^2} - \frac{\partial^2 u}{\partial y^2} = 3200x(1 - 20(x^2 + y^2))e^{-40(x^2+y^2)}, \quad (x,y) \in \Omega,$$

$$u(-1,y) = -10e^{-40(1+y^2)}, \quad u(1,y) = 10e^{-40(1+y^2)},$$

$$u(x,-1) = 10xe^{-40(x^2+1)}, \quad u(x,1) = 10xe^{-40(x^2+1)}.$$

The exact solution $u^*$ of (41) problem appears to be

$$u^* = 10xe^{-40(x^2+y^2)}.$$

Clearly, the area of the higher activity of the solution is concentrated around (0,0) and has a more or less circular shape. Now assume we would like to approximate $u^*$ with an accuracy in the range $10^{-3} - 10^{-4}$. From (42) we see that the solution has damped out below $10^{-4}$ for $x^2 + y^2 > 0.25$. So we determine $\hat{\Omega}$ as the circle $B_{1/2}$. Of course, in practice one should find a high activity area by some estimation method; this is not the issue here, however. As described in section 2 we introduce $\Omega_h$ and $\hat{\Omega}_h$ corresponding to $\Omega$ and $\hat{\Omega}$, respectively. We then solve (15) and (16) by a direct network method [6] (for large linear systems, iterative linear solvers can be an option, [9]) and combine the two solutions by means of LDC. In order to illustrate the convergence speed of LDC we introduce $d_i$, which measures the distance between consecutive iterates on $\Omega_h$

$$d_i := ||u_h^i - u_h^{i-1}||_{\infty},$$

and the convergence factor $\rho_i$

$$\rho_i = \frac{d_i}{d_{i-1}}.$$

The values of $d_i$ and $\rho_i$ are computed for various values of $h$. If we take the ratio of mesh sizes $\hat{h}/h$ to be equal to 1/2 we obtain Table 1 and for $\hat{h}/h$ equal to 1/4 we find Table 2. As can be seen from these tables, LDC converges very fast. Note, that the discretisation error is constant for fixed $h$; it also appears to become smaller for smaller $h$. Furthermore, a comparison between the Table 1 and Table 2 seems to indicate that the convergence factor is independent on $\hat{h}$.

Next, we examine the global approximation error of the LDC iterates. The immediate problem is a stopping criterion for LDC. Note, that the discretisation methods used are of second order, so we anticipate discretisation errors not larger than $10^{-4}$. 

12
Table 1: The difference between the consecutive iterates $d^i$ and the convergence factor $\rho^i$ for $\tilde{h}/h = 1/2$. 

<table>
<thead>
<tr>
<th>$i$</th>
<th>$d^i$</th>
<th>$\rho^i$</th>
<th>$d^i$</th>
<th>$\rho^i$</th>
<th>$d^i$</th>
<th>$\rho^i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$2.21 \cdot 10^{-4}$</td>
<td>$4.87 \cdot 10^{-5}$</td>
<td>$1.12 \cdot 10^{-5}$</td>
<td>$2.3 \cdot 10^{-4}$</td>
<td>$5.36 \cdot 10^{-9}$</td>
<td>$1.1 \cdot 10^{-4}$</td>
</tr>
<tr>
<td>2</td>
<td>$5.26 \cdot 10^{-7}$</td>
<td>$2.3 \cdot 10^{-4}$</td>
<td>$1.1 \cdot 10^{-4}$</td>
<td>$5.36 \cdot 10^{-9}$</td>
<td>$1.1 \cdot 10^{-4}$</td>
<td>$4.8 \cdot 10^{-10}$</td>
</tr>
<tr>
<td>3</td>
<td>$1.21 \cdot 10^{-10}$</td>
<td>$2.4 \cdot 10^{-4}$</td>
<td>$1.1 \cdot 10^{-4}$</td>
<td>$5.90 \cdot 10^{-13}$</td>
<td>$1.1 \cdot 10^{-4}$</td>
<td>$4.2 \cdot 10^{-5}$</td>
</tr>
</tbody>
</table>

Table 2: The difference between the consecutive iterates $d^i$ and the convergence factor $\rho^i$ for $\tilde{h}/h = 1/4$. 

<table>
<thead>
<tr>
<th>$i$</th>
<th>$d^i$</th>
<th>$\rho^i$</th>
<th>$d^i$</th>
<th>$\rho^i$</th>
<th>$d^i$</th>
<th>$\rho^i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$1.17 \cdot 10^{-4}$</td>
<td>$2.42 \cdot 10^{-5}$</td>
<td>$9.65 \cdot 10^{-6}$</td>
<td>$2.3 \cdot 10^{-4}$</td>
<td>$6.67 \cdot 10^{-3}$</td>
<td>$1.44 \cdot 10^{-3}$</td>
</tr>
<tr>
<td>2</td>
<td>$2.68 \cdot 10^{-8}$</td>
<td>$2.3 \cdot 10^{-4}$</td>
<td>$1.71 \cdot 10^{-14}$</td>
<td>$2.3 \cdot 10^{-4}$</td>
<td>$1.38 \cdot 10^{-3}$</td>
<td>$3.52 \cdot 10^{-4}$</td>
</tr>
<tr>
<td>3</td>
<td>$6.19 \cdot 10^{-12}$</td>
<td>$2.3 \cdot 10^{-4}$</td>
<td>$1.71 \cdot 10^{-14}$</td>
<td>$2.3 \cdot 10^{-4}$</td>
<td>$1.38 \cdot 10^{-3}$</td>
<td>$3.52 \cdot 10^{-4}$</td>
</tr>
</tbody>
</table>

Table 3: Global approximation error of LDC, $||\tilde{u}_h^i - (u^*)_h||_\infty$.
Thus, we can terminate LDC after the first iteration, as appears from Table 1,2. Therefore we can view $u_h^1$ and $\tilde{u}_h^1$ as the LDC solution. In Table 3 we have displayed this approximation error for various values of $h$ and $\tilde{h}$. For $h$ small enough we see that the LDC solution approximates $u^\star$ within second order of accuracy, indeed. Note, that we have second order approximation inside $n_h$, even though we used a linear interpolation for the boundary conditions, cf [1].

For our last test we compare $\|\tilde{u}_h^1 - (u^\star)_h\|_\infty$ against the error between the exact solution restricted to $\tilde{n}_h$ and the solution of (14) with the boundary conditions obtained not through interpolation from $u_h^1$ but taken from the exact solution. We denote this solution by $\tilde{u}_h^\star$. In other words, $\tilde{u}_h^\star$ is a solution we obtain by solving (14). As can be seen from Table 4 the results produced by LDC are close to optimal.

Next, we like to illustrate how LDC blends in nicely with the idea that a suitable transformation of the variable, stretching depending on a problem parameter $\epsilon$ say, would take the problem into a reference area, independent of $\epsilon$. Thus a suitable approximation would, in turn, also be independent of $\epsilon$. We shall give an example to illustrate this.

**Example 2.**
Consider the following boundary value problem

$$
-\frac{\partial^2 u}{\partial x^2} - \frac{\partial^2 u}{\partial y^2} = \frac{4}{\epsilon} (1 - \frac{x^2 + y^2}{\epsilon})e^{-(x^2+y^2)/\epsilon} \sin(x) + \sin(y), \quad (x, y) \in \Omega, \quad (43)
$$

$$
u(x, y) = e^{-(x^2+y^2)/\epsilon} \sin(x) + \sin(y), \quad (x, y) \in \Gamma.
$$

The exact solution of (43) is given by

$$u^\star = e^{-(x^2+y^2)/\epsilon} \sin(x) + \sin(y).
$$

We can decompose the exact solution $u^\star$ into two components: the smooth part $-\sin(x) + \sin(y)$ and the layer solution $e^{-(x^2+y^2)/\epsilon}$. Note, that the gradients of the layer solution become steeper as $\epsilon$ increases. We intend to resolve the smooth solution by using a global coarse mesh and the layer solution by using a local fine mesh. From a practical point of view we need to to decide where the layer solution has damped out sufficiently. For this we may say that if for a given tolerance, say TOL,

$$e^{-R^2/\epsilon} = \text{TOL},$$

Table 4: $\|\tilde{u}_h^1 - (u^\star)_h\|_\infty$ versus the optimal error $\|u_h^\star - (u^\star)_h\|_\infty$, $h = 10^{-1}$. 

| $\tilde{h}$ | $\|\tilde{u}_h^1 - (u^\star)_h\|_\infty$ | $\|u_h^\star - (u^\star)_h\|_\infty$ |
|----------|
| 20^{-1}  | 9.11 \cdot 10^{-2}  | 5.74 \cdot 10^{-2}  |
| 40^{-1}  | 2.32 \cdot 10^{-2}  | 1.42 \cdot 10^{-2}  |
| 60^{-1}  | 5.63 \cdot 10^{-3}  | 3.46 \cdot 10^{-3}  |

the $R$ determines the radius of the local circular subdomain $\tilde{\Omega}$. So, from (44) we obtain that $\epsilon$ and $R$ are coupled by

$$R = -\epsilon^{1/2} \ln^{1/2} \text{TOL}.$$  (45)

Next, we restrict (43) to $\tilde{\Omega}$. In polar coordinates we have

$$-\frac{\partial^2 \tilde{u}}{\partial r^2} - \frac{1}{r} \frac{\partial \tilde{u}}{\partial r} - \frac{1}{r^2} \frac{\partial^2 \tilde{u}}{\partial \phi^2} = \frac{4}{\epsilon} \left(1 - \frac{r^2}{\epsilon}\right)e^{-r^2/\epsilon} + s(r, \phi), \quad (r, \phi) \in \tilde{\Omega}$$  (46)

where $s$ denotes the part of the right-hand side corresponding to the smooth solution.

We now introduce a transformation of the variables, 'stretching'

$$\sigma = \frac{r}{\sqrt{\epsilon}}.$$

In new variables $(\sigma, \phi)$ (46) reads

$$-\frac{\partial^2 \tilde{u}}{\partial \sigma^2} - \frac{1}{\sigma} \frac{\partial \tilde{u}}{\partial \sigma} - \frac{1}{\sigma^2} \frac{\partial^2 \tilde{u}}{\partial \phi^2} = 4(1 - \sigma)e^{-\sigma^2} + \epsilon s, \quad (\sigma, \phi) \in B(-\ln \text{TOL}).$$  (47)

The new problem (47) is independent of $\epsilon$. For the discrete analogue of (47) we can choose as many grid points as are needed to resolve it within the prescribed tolerance TOL. Scaling (47) back by setting $r = \sqrt{\epsilon} \sigma$, we obtain a set of discretisations which provide us with an error bound uniform in $\epsilon$. Furthermore, by construction, the number of grid points stays the same as $\epsilon$ varies. However we need to keep in mind, that we are solving the two-scale problem (smooth+layer solution) by means of LDC, which iteratively enhances the solution both inside $\Omega$ and $\tilde{\Omega}$. If the grid $\Omega'_h \subset \Omega_h$, i.e. the grid where we need to compute the defect, does not contain enough points, we reach the limits of this approach, of course. In Table 5 we illustrate this independence of $\epsilon$ where we display the discretisation error for various $\epsilon$. We also observe that the lack of grid points in $\Omega'_h$, which happens for larger $h$, indeed corrupts the error. For small $h$, however, the error remains uniformly bounded with respect to $\epsilon$. The size of $\tilde{\Omega}$ is determined according to (45), see Table 6.

**References**

Table 6: Values of $R$ for various $\epsilon$, $TOL = 10^{-4}$.

\[
\begin{array}{lcccc}
\epsilon = 50^{-1} & \epsilon = 100^{-1} & \epsilon = 200^{-1} & \epsilon = 1000^{-1} \\
0.45 & 0.32 & 0.22 & 0.1 \\
\end{array}
\]


