Analysis of the Local Defect Correction and High Order Compact Finite Differences

M. Sizov, M.J.H. Anthonissen, R.M.M. Mattheij

November 22, 2004

1 Introduction

Many boundary value problems produce solutions that have highly localized properties. In this paper we consider boundary value problems with solutions that have one or a few small regions with high activity.

We study a method based on a combination of high order compact finite difference discretizations on several uniform grids with different grid sizes that cover different parts of the domain. At least one grid should cover the entire domain; the mesh size of this global coarse grid is chosen in agreement with the relatively smooth behaviour of the solution outside the high activity regions. Apart from this global coarse grid, one or several local grids are used which are also uniform. Each of these local grids covers only a (small) part of the domain and contains a high activity region. The mesh sizes of the local grids are chosen in agreement with the behaviour of the solution in the corresponding high activity region. In this way every part of the domain can be covered by a (locally) uniform grid with a mesh size that is in agreement with the behaviour of the continuous solution in that part of the domain. This refinement strategy is known as local uniform grid refinement. The solution is approximated on the composite grid, which consists of the uniform coarse grid and subgrid(s). Note that such composite grids are highly structured and hence very simple data structures can be used.

The boundary value problem is solved on the composite grid by the local defect correction method (LDC) (see [7, 10]). In this method, which is an iterative process, a basic global discretization is improved by local discretizations defined in the subdomains. This update of the coarse grid solution is achieved by performing a defect correction on the right hand side of the coarse grid problem. At each iteration step the process yields a discrete approximation of the continuous solution on the composite grid. The discrete problem that is actually being solved is an implicit result of the iterative process. Therefore the LDC method is both an iterative discretization and solution method.

An analysis of the LDC technique in combination with finite difference discretizations is presented in [7–9]. The LDC method is combined with finite volume discretizations in [3, 4, 24]. An application of the LDC algorithm in a finite volume context was presented in [17] for numerical simulations of the flow and heat transfer in a glass tank. Finally, LDC is studied in combination with finite element discretizations in [25].

LDC was previously used with standard finite difference schemes like first order upwind or second order central differences. Nowadays, especially for Direct Numerical Simulation (DNS) of turbulent flows, high order compact finite difference (HOCFD) schemes are becoming more and more popular. In this paper we combine the LDC technique with such types of discretization.

Implicit finite difference relations for the first and second derivatives have been given a variety of names. Many can be found in [6], under the names of *Mehrstellen* or *Hermitian* methods by analogy with Hermitian finite elements. They are also known as Padé differencing approximation [12]. In the 70-80's a large number of applications for solving fluid-mechanics equations have been developed in [11, 13, 19, 20]. From the more recent papers the works [14] and [16] should be

mentioned. Following [18] all implicit formulae can be derived in a systematic way from a Taylor series expansion.

Finite difference methods have received much attention in the beginning of 90's for the DNS of transitional and turbulent flows. The classical paper [14] was the pioneering work in this field. Most high order finite difference methods used in direct numerical simulations are central difference schemes [14, 16] which introduce only phase errors but no dissipative errors in numerical solutions. The drawback of central schemes is that they are not robust for convection dominated flow simulations. In order to overcome this difficulty, compact upwind schemes were developed, which have numerical dissipation to control the aliasing errors. In the past years, many upwind schemes were proposed. The main articles are [1, 5, 22, 26]. The analysis of the stability of numerical boundary treatment for high order compact finite difference schemes was done in [15].

In this paper the major objective is to construct a proper LDC framework for high order compact finite difference schemes and to extend the results on LDC found for classical central difference and upwind schemes. The paper is built up as follows. In Section 2, we formulate a stationary convection diffusion problem and describe the LDC algorithm. In Section 3, we give a brief overview of high order compact finite differences. In Section 4 we construct an LDC framework first for one-dimensional problems and then we present a general approach, which is applicable for two and three-dimensional problems. In Section 5 we present some numerical results.

2 Problem description and formulation of the LDC algorithm

The problem we study in this paper is given by

$$\begin{cases} Lu = -\epsilon \nabla^2 u(x, y) + \mathbf{c} \cdot \nabla u(x, y) = f(x, y) \text{ in } \Omega, \\ u = g \text{ on } \Gamma. \end{cases} \quad (1)$$

In (1), L is a linear elliptic differential operator, and f and g are the source term and Dirichlet boundary condition, respectively, $\epsilon > 0$ is the diffusion coefficient, $\mathbf{c}$ is the convection coefficient, u is the unknown function of (x, y) , Ω is the domain of interest and Γ is the boundary of this domain.

Following [2], we briefly introduce the basics of the Local Defect Correction technique.

In order to discretize (1), we first choose a global coarse grid (grid size H), which we denote by Ω^H . The next step is to find an initial approximation $\mathbf{u}_0^H$ on Ω^H by solving the system

$$\mathbf{L}^H \mathbf{u}_0^H = \mathbf{f}^H, \quad (2)$$

which is a discretization of boundary value problem (1). In (2), the right hand side $\mathbf{f}^H$ incorporates the source term f as well as the Dirichlet boundary condition g .

Now, assume that the continuous solution u of (1) has a high activity region in some (small) part of the domain. This high activity could either be caused by the boundary conditions or the source term. We would like to capture this high activity of u by discretizing (1) on a composite grid. So we choose $\Omega_l \subset \Omega$ such that the high activity region of u is contained in Ω_l . If we have more than one high activity region, one may take more regions of refinement. In Ω_l , we choose a local fine grid (with the grid size h), which we denote by Ω_l^h . The fine grid is chosen such that $\Omega^H \cap \Omega_l \subset \Omega_l^h$, i.e., grid points of the global coarse grid that lie in the area of refinement belong to the local fine grid too.

Now we have to define a local discrete problem on Ω_l . So we define artificial boundary conditions on Γ , the interface between Ω_l and $\Omega \setminus \Omega_l$. Since on Γ we have more fine grid points than coarse grid ones, we prescribe artificial Dirichlet boundary conditions by applying an interpolation operator.

In this way, we find the following approximation $\mathbf{u}_{l,i}^h$, $i = 0$, on Ω_l^h

$$\mathbf{L}_l^h \mathbf{u}_{l,i}^h = \mathbf{f}_l^h (u_i^H|_\Gamma). \quad (3)$$

In (3), the matrix $\mathbf{L}_l^h$ is a discrete approximation of the differential operator L on the subdomain Ω_l , and the first term on the right hand side $\mathbf{f}_l^h$ incorporates the source term f as well as the Dirichlet boundary condition g on $\partial\Omega_l \setminus \Gamma$ given in (1). The dependence on the coarse grid approximation via the artificial Dirichlet boundary condition is made explicit by writing $\mathbf{f}_l^h(u_i^H|_\Gamma)$.

When boundary value problem (1) has been discretized and solved on a coarse grid, and when an area of the coarse grid has been refined and a local solution has been calculated on the finer grid, we can define a composite grid approximation $\mathbf{u}_0^{H,h}$ as

$$\mathbf{u}_0^{H,h}(x, y) := \begin{cases} \mathbf{u}_{l,0}^h(x, y), & (x, y) \in \Omega_l^h, \\ \mathbf{u}_0^H(x, y), & (x, y) \in \Omega^H \setminus \Omega_l^h. \end{cases} \quad (4)$$

So for the coarse grid points within the region of the refinement we have two solutions, one coming from the coarse grid and another from the fine grid. We will now use the local fine grid solution to update the coarse grid approximation. This update can be achieved by projecting the more accurate fine grid solution onto the local coarse grid, and by calculating the residual of the projected solution; this residual is an estimate of the local discretization error of the coarse grid discretization. The estimate is used to formulate a modified discrete problem on the coarse grid. This is considered in more detail below.

The grid points of the coarse grid will be partitioned as $\Omega^H = \Omega_l^H \cup \Gamma^H \cup \Omega_c^H$, where $\Omega_l^H := \Omega^H \cap \Omega_l$, $\Gamma^H := \Omega^H \cap \Gamma$ and $\Omega_c^H := \Omega^H \setminus (\Omega_l^H \cup \Gamma^H)$. If we would substitute the projection on Ω^H of the exact solution u of boundary value problem (1) into the coarse grid discretization (1), we would find the local discretization error or defect $\mathbf{d}^H$, given by $\mathbf{L}^H(u|_{\Omega^H}) = \mathbf{f}^H + \mathbf{d}^H$. If we would know the values of the defect $\mathbf{d}^H$, we could use them to find a better approximation on the coarse grid. This could be achieved by putting the defect vector on the right hand side of (1). However, as we do not know the exact solution of the boundary value problem, we cannot calculate $\mathbf{d}^H$. What we can do though, is to use the approximation $\mathbf{u}_{l,0}^h$ calculated on the local fine grid to estimate $\mathbf{d}^H$ at the coarse grid points $(x, y) \in \Omega_l^H$. Define $\mathbf{w}_0^H \in G(\Omega^H)$ as the global coarse grid function of best approximations sofar, i.e.

$$\mathbf{w}_0^H(x, y) := \begin{cases} \mathbf{u}_{l,0}^h(x, y), & (x, y) \in \Omega_l^H, \\ \mathbf{u}_0^H(x, y), & (x, y) \in \Gamma^H \cup \Omega_c^H. \end{cases} \quad (5)$$

Next, we estimate the defect by $\mathbf{d}^H = \mathbf{L}^H(u|_{\Omega^H}) - \mathbf{f}^H \approx \mathbf{L}^H \mathbf{w}_0^H - \mathbf{f}^H =: \mathbf{d}_0^H$. Assuming that the stencil at grid point (x, y) involves (at most) function values at $(x + iH, y + jH)$ with $i, j \in \{-1, 0, 1\}$, $\mathbf{d}_0^H$ provides an estimate of the local discretization error of the coarse grid discretization at all points of Ω_l^H . Therefore, we can update the coarse grid approximation by placing the estimate at the right hand side of the coarse grid equation (1). This leads to the coarse grid correction step to find $\mathbf{u}_i^H$, $i = 1$, on the coarse grid

$$\mathbf{L}^H \mathbf{u}_i^H = \mathbf{f}_{i-1}^H, \quad (6)$$

where

$$\mathbf{f}_i^H(x, y) := \begin{cases} \mathbf{f}^H(x, y) + \mathbf{d}_i^H(x, y), & (x, y) \in \Omega_l^H, \\ \mathbf{f}^H(x, y), & (x, y) \in \Gamma^H \cup \Omega_c^H. \end{cases}$$

The correction step (6) produces a new solution $\mathbf{u}_1^H$ on the coarse grid. Because (6) incorporates estimates of the local discretization error of the coarse grid discretization, the new solution $\mathbf{u}_1^H$ is assumed to be more accurate than $\mathbf{u}_0^H$. Hence, the new solution $\mathbf{u}_1^H$ provides a better boundary condition on the interface. A better solution on the local fine grid can be found as before by solving (3) with $i = 1$.

To summarize, we have the following iterative method.

Algorithm 1.

Two-grid LDC algorithm with area of refinement chosen *a priori*

Initialization

- Solve the basic coarse grid problem (2).
- Solve the local fine grid problem (3).

Iteration, $i = 1, 2, \dots$

- Solve the updated coarse grid problem (6).
- Solve the local fine grid problem (3).

3 High order compact finite difference schemes

The basic idea of high order compact finite difference schemes is to employ not only the function values but also the values of the derivatives as unknowns. This gives us a possibility to get higher accuracy or better spectral resolution while keeping the stencil relatively small.

With a limitation to three-point expressions the general form of an implicit finite difference relation between a function of one variable and its first two derivatives would read

$$\begin{aligned} a_+ u_{i+1} + a_0 u_i + a_- u_{i-1} + b_+ (u_x)_{i+1} + b_0 (u_x)_i + b_- (u_x)_{i-1} \\ + c_+ (u_{xx})_{i+1} + c_0 (u_{xx})_i + c_- (u_{xx})_{i-1} = 0. \end{aligned} \quad (7)$$

By imposing different constraints on the coefficients a , b and c , we can tune the numerical scheme in some sense. For example we can get a higher order or a better spectral resolution of the scheme [21]. In the following we address some of the possible high order compact finite difference schemes, which will later be used in numerical examples.

So, for the one-dimensional problem we have three sets of equations: a discretized version of our differential equation (1) which looks like

$$Lu_i \equiv -\epsilon u_i'' + cu_i' = f_i, \quad i = 1, \dots, N-1; \quad (8)$$

an expression for the second derivative (see Section 3.1); an expression for the first derivative (see Section 3.2).

3.1 Diffusive term

For the discretization of the diffusive term in (1) we can use the following schemes

- Padé scheme (3-point scheme [18])

$$u_{i+1}'' + 10u_i'' + u_{i-1}'' - \frac{12}{h^2}(u_{i+1} - 2u_i + u_{i-1}) = 0. \quad (9)$$

To close this relation at the boundary, we use

$$\epsilon u_1'' + au_1' + cu_1 = f_1, \quad \epsilon u_N'' + au_N' + cu_N = f_N.$$

- High order compact upwind scheme of Zhong (5-point scheme [26])

$$25u_{i-1}'' + 60u_i'' + 15u_{i+1}'' = \frac{1}{h} \left(-\frac{5}{2}u_{i-1}' - \frac{160}{3}u_{i-1}' + 15u_i' + 40u_{i+1}' + \frac{5}{6}u_{i+2}' \right). \quad (10)$$

To close this relation at the boundary, we use ([26], equations (23)-(27))

$$60u_0'' + 180u_1'' = \frac{1}{h}(-170u_0' + 90u_1' + 90u_2' - 10u_3'), \quad (11)$$

$$15u_0'' + 60u_1'' + 15u_2'' = \frac{1}{h}(-45u_0' + 45u_2'), \quad (12)$$

$$15u''_N + 60u''_{N-1} + 15u''_{N-2} = -\frac{1}{h}(-45u'_N + 45u'_{N-2}), \quad (13)$$

$$60u''_N + 180u''_{N-1} = -\frac{1}{h}(-170u'_N + 90u'_{N-1} + 90u'_{N-2} - 10u'_{N-3}). \quad (14)$$

Another possibility is to use the same scheme as for convective term (see relation (16) and related closing relations in Section 3.2), but modified for second derivative

$$u''_{i+1} + 4u''_i + u''_{i-1} - \frac{3}{h}(u'_{i+1} - u'_{i-1}) = 0. \quad (15)$$

To close this relation at the boundary, we use

$$2u''_1 + u''_0 - \frac{1}{2h}(u'_2 + 4u'_1 - 5u'_0) = 0, \quad u''_N + 2u''_{N-1} - \frac{1}{2h}(5u'_N - 4u'_{N-1} - u'_{N-2}) = 0.$$

3.2 Convective term

Central difference type schemes do not behave well when applied to convection-dominated problems. A way to overcome this difficulty is to use upwind schemes. In [15] stability of the numerical boundary treatment for compact high-order finite difference schemes was investigated and it was shown that the upwind compact schemes give better results. A lot of information about non-centered high-order compact finite difference schemes one can find in the article [22].

For the discretization of the convective term in (1) we can use following schemes

- Padé scheme (3-point scheme [18])

$$u'_{i+1} + 4u'_i + u'_{i-1} - \frac{3}{h}(u_{i+1} - u_{i-1}) = 0. \quad (16)$$

To close this relation at the boundary, we use

$$2u'_1 + u'_0 - \frac{1}{2h}(u_2 + 4u_1 - 5u_0) = 0, \quad 2u'_N + u'_{N-1} - \frac{1}{2h}(5u_N - 4u_{N-1} - u_{N-2}) = 0.$$

- High order compact upwind scheme of Zhong (5-point scheme [26]). This is the scheme (10)-(14), modified for the first derivative:

$$25u'_{i-1} + 60u'_i + 15u'_{i+1} = \frac{1}{h} \left(-\frac{5}{2}u_{i-1} - \frac{160}{3}u_{i-1} + 15u_i + 40u_{i+1} + \frac{5}{6}u_{i+2} \right). \quad (17)$$

To close this relation at the boundary, we use

$$\begin{aligned} 60u'_0 + 180u'_1 &= \frac{1}{h}(-170u_0 + 90u_1 + 90u_2 - 10u_3), \\ 15u'_0 + 60u'_1 + 15u'_2 &= \frac{1}{h}(-45u_0 + 45u_2), \\ 15u'_N + 60u'_{N-1} + 15u'_{N-2} &= -\frac{1}{h}(-45u_N + 45u_{N-2}), \\ 60u'_N + 180u'_{N-1} &= -\frac{1}{h}(-170u_N + 90u_{N-1} + 90u_{N-2} - 10u_{N-3}). \end{aligned}$$

4 Combination of LDC with HOCFD

In this section we present an algorithm which combines the Local Defect Correction technique with the schemes presented in Section 3. First we start with the one-dimensional version of equation (1) and after that we extend the algorithm to two and three-dimensional problems.

4.1 One-dimensional problems

We can rewrite the system of equations (9),(16) and a discrete form of the differential equation (8) with corresponding boundary conditions in the following matrix-vector form

$$\mathbf{A}\mathbf{x} = \mathbf{b}, \quad (18)$$

where $\mathbf{A}$ is a 3×3 block-diagonal matrix representing the discrete operator, $\mathbf{x} = (\mathbf{u}'', \mathbf{u}', \mathbf{u})^T$ is the vector of unknowns and $\mathbf{b} = (0, 0, \mathbf{f}(x))^T$ is the right-hand side. The matrix $\mathbf{A}$ has the following structure:

$$\mathbf{A} = \begin{pmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} & \mathbf{A}_{13} \\ \mathbf{0} & \mathbf{A}_{22} & \mathbf{A}_{23} \\ \mathbf{A}_{31} & \mathbf{A}_{32} & \mathbf{A}_{33} \end{pmatrix}. \quad (19)$$

The condition number of the matrix $\mathbf{A}$ is quite large, even for "small" problems, due to the fact that the matrix is highly unbalanced (we have $O(1)$, $O(h)$ and $O(h^2)$ terms on the diagonal). Moreover, we should point out that the LDC algorithm is not applicable to the equation in the form (18). So we need a reformulation. One of the ways to solve this problem is to rearrange the matrix $\mathbf{A}$. We can write our system (18) as

$$\mathbf{A}_{11}\mathbf{u}'' + \mathbf{A}_{12}\mathbf{u}' + \mathbf{A}_{13}\mathbf{u} = \mathbf{0}, \quad (20)$$

$$\mathbf{A}_{22}\mathbf{u}' + \mathbf{A}_{23}\mathbf{u} = \mathbf{0}, \quad (21)$$

$$\mathbf{A}_{31}\mathbf{u}'' + \mathbf{A}_{32}\mathbf{u}' + \mathbf{A}_{33}\mathbf{u} = \mathbf{f}. \quad (22)$$

Rearranging the terms we can get following equation:

$$\mathbf{A}_{11}^{-1} (\mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{A}_{23} - \mathbf{A}_{13}) \mathbf{u} - \mathbf{A}_{32}\mathbf{A}_{22}^{-1}\mathbf{A}_{23}\mathbf{u} + \mathbf{A}_{33}\mathbf{u} = \mathbf{f}, \text{ or}, \quad (23)$$

$$\mathbf{A}_m \mathbf{u} = \mathbf{f}, \quad (24)$$

with $\mathbf{A}_m := \mathbf{A}_{11}^{-1} (\mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{A}_{23} - \mathbf{A}_{13}) - \mathbf{A}_{32}\mathbf{A}_{22}^{-1}\mathbf{A}_{23} + \mathbf{A}_{33}$. Matrices $\mathbf{A}_{11}$ and $\mathbf{A}_{22}$ are non-singular. After these rearrangements our matrix $\mathbf{A}_m$ does not suffer from ill conditioning. After the reformulation like (24) it is possible to directly apply the algorithm presented in Section 2.

4.2 Two- and more dimensional problems

We would like to discretize (1) using the schemes from Section 3. We introduce the vector of unknowns $\mathbf{x} = (\mathbf{u}_{xx}^T, \mathbf{u}_{yy}^T, \mathbf{u}_x^T, \mathbf{u}_y^T, \mathbf{u}^T)^T$, with the size $5N$ where N is the number of grid points.

The corresponding matrix $\mathbf{A}$ is a 5×5 block matrix with the following structure

$$\mathbf{A} = \begin{pmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} & \mathbf{A}_{13} & \mathbf{A}_{14} & \mathbf{A}_{15} \\ \mathbf{A}_{21} & \mathbf{A}_{22} & \mathbf{A}_{23} & \mathbf{A}_{24} & \mathbf{A}_{25} \\ \mathbf{A}_{31} & \mathbf{A}_{32} & \mathbf{A}_{33} & \mathbf{A}_{34} & \mathbf{A}_{35} \\ \mathbf{A}_{41} & \mathbf{A}_{42} & \mathbf{A}_{43} & \mathbf{A}_{44} & \mathbf{A}_{45} \\ \mathbf{A}_{51} & \mathbf{A}_{52} & \mathbf{A}_{53} & \mathbf{A}_{54} & \mathbf{A}_{55} \end{pmatrix} \quad (25)$$

Entries $\mathbf{A}_{1,i}$ represent the discretization of u_{xx} by one of the possible discretization schemes (16)-(17), entries $\mathbf{A}_{2,i}$ of u_{yy} by (16)-(17), entries $\mathbf{A}_{3,i}$ of u_x by (9)-(10), entries $\mathbf{A}_{4,i}$ of u_y by (9)-(5), entries $\mathbf{A}_{5,i}$ of u ; the latter represent the equation (1) as well as the boundary conditions. Depending on the type of discretization used, some of the off-diagonal submatrices $\mathbf{A}_{i,j}$ could be zero or singular (see Figure 1). The matrix $\mathbf{A}$ has quite a large condition number, so we use equilibration of rows in order to reduce it. The basic idea of the equilibration of rows is to multiply rows of the matrix such that we get $O(1)$ values on the main diagonal. More detailed information one can find in [23].

For two-dimensional problems it is quite difficult to perform explicit substitution like (23). We still do the same reduction of the matrix $\mathbf{A}$, but instead of explicitly expressing the matrix subblocks, we use block Gaussian elimination. In order to solve our problem we need to perform the following steps

Figure 1: Possible matrix structures depending on the scheme used

1. Solve the coarse grid problem

- (a) Construct matrix $\mathbf{A}^H$ and right hand side $\mathbf{f}^H$. Matrix $\mathbf{A}^H$ has the form (19).
- (b) Perform block LU-decomposition of the matrix $\mathbf{A}^H$. As a result we have $\mathbf{A}^H = \mathbf{L}^H \mathbf{U}^H$ and we get $\mathbf{L}^H$ and $\mathbf{U}^H$ in the following form

$$\mathbf{L}^H = \begin{pmatrix} \mathbf{I} & 0 & 0 & 0 & 0 \\ \mathbf{L}_{21} & \mathbf{I} & 0 & 0 & 0 \\ \mathbf{L}_{31} & \mathbf{L}_{32} & \mathbf{I} & 0 & 0 \\ \mathbf{L}_{41} & \mathbf{L}_{42} & \mathbf{L}_{43} & \mathbf{I} & 0 \\ \mathbf{L}_{51} & \mathbf{L}_{52} & \mathbf{L}_{53} & \mathbf{L}_{54} & \mathbf{I} \end{pmatrix},$$

$$\mathbf{U}^H = \begin{pmatrix} \mathbf{U}_{11} & \mathbf{U}_{12} & \mathbf{U}_{13} & \mathbf{U}_{14} & \mathbf{U}_{15} \\ 0 & \mathbf{U}_{22} & \mathbf{U}_{23} & \mathbf{U}_{24} & \mathbf{U}_{25} \\ 0 & 0 & \mathbf{U}_{33} & \mathbf{U}_{34} & \mathbf{U}_{35} \\ 0 & 0 & 0 & \mathbf{U}_{44} & \mathbf{U}_{45} \\ 0 & 0 & 0 & 0 & \mathbf{U}_{55} \end{pmatrix}.$$

- (c) Define the vector $\mathbf{y}^H := \mathbf{U}^H \mathbf{x}$.
- (d) Solve $\mathbf{L}^H \mathbf{y}^H = \mathbf{f}^H$. Get $\mathbf{y}_5^H$.
- (e) Solve

$$\mathbf{U}_{55} \mathbf{u}^H = \mathbf{y}_5^H \quad (26)$$

and get $\mathbf{u}^H$, the coarse grid solution.

2. Solve the fine grid problem

- (a) Get the boundary conditions for the fine grid boundary value problem and construct $\mathbf{A}_l^h$ and $\mathbf{f}_l^h$.
- (b) Solve the local grid problem

$$\mathbf{A}_l^h \mathbf{x}_l^h = \mathbf{f}_l^h \quad (27)$$

and get $\mathbf{x}_l^h = (\mathbf{u}_{xx}^h, \mathbf{u}_{yy}^h, \mathbf{u}_x^h, \mathbf{u}_y^h, \mathbf{u}^h)^T$. Extract $\mathbf{u}_l^h$.

3. Calculate the defect

- (a) Construct the vector $\mathbf{w}^H$

$$\mathbf{w}^H(x, y) = \begin{cases} \mathbf{u}_l^h(x, y) & (x, y) \in \Omega_l^H \\ \mathbf{u}^H(x, y) & (x, y) \in \Omega^H \setminus \Omega_l^H \end{cases}$$

- (b) Construct the defect $\mathbf{d}_0^H = \mathbf{U}_{55}\mathbf{w}^H - \mathbf{y}_5^H$. Restrict the defect by setting it to zero outside the area of refinement.

4. Solve the updated coarse grid problem

$$\mathbf{U}_{55}\mathbf{u}_1^H = \mathbf{y}_5^H + \mathbf{d}_0^H \quad (28)$$

and get the new coarse grid solution $\mathbf{u}_1^H$.

We have outlined all steps of the new LDC method. The iteration is as in Algorithm 1; the basic coarse grid problem, the local fine grid problem and the updated coarse grid problem are given by (26), (27) and (28), respectively.

5 Numerical results

In this section we present some typical numerical results for the convection-diffusion problem (1). As in the previous section, we start with a one-dimensional problem and then present results for a two-dimensional model problem.

5.1 Example 1

With the use of numerical simulations we want to investigate the following properties of our LDC method for high order compact finite difference schemes: convergence behaviour; accuracy; efficiency.

For our numerical test we solve a one-dimensional variant of the boundary value problem (1) and we choose the source term f and the boundary conditions such that

$$u(x) = \frac{1}{2} (\tanh(50(x - 1/8)) + 1). \quad (29)$$

The following parameters are chosen for the computations: $c = \mp 0.99$ (convection dominated), $c = \mp 0.1$ (diffusion dominated); interface point $\gamma = 0.3$; different number of coarse grid points N . We use the Padé scheme given by (16), (9).

The typical results for the convergence behaviour for the LDC technique one can find in Figures 2-3. In Figure 2 we plot $\|\mathbf{u}^i - \mathbf{u}^{i-1}\|_\infty$ against the iteration number i , where $\mathbf{u}^i$ is our numerical coarse grid solution on the i th iteration. As one can see in Figure 2, the LDC algorithm shows fast convergence. In Figure 3 we plot $\|\mathbf{u}^i - \mathbf{u}^{exact}\|_\infty$ against the iteration number, where *exact* stands for the exact solution of the problem (which is known in our case). As one can see from Figure 3, we only need a small number of the LDC iterations to reach the fixed point solution, and, as will be shown later, this is the solution we get on a fine uniform grid. These results are typical and do not change much for the different values of N and c .

In order to compare the results of the LDC technique with those we get using a fine uniform grid, we performed a number of calculations. Under fine uniform grid we understand a grid which covers the whole domain with the mesh size equal to the mesh size of the fine LDC grid. First we compare the error. In Table 1 one can find $\|\mathbf{u}^{exact} - \mathbf{u}^H\|_\infty$ for the LDC solution and for the fine uniform grid solution. For LDC we stop the iteration if $\|\mathbf{u}_i^{H,h} - \mathbf{u}_{i-1}^{H,h}\|_\infty < 1 * 10^{-5}$. The errors for the LDC algorithm and the fine uniform grid are of the same order.

(a) $c = 0.9$, $N = 11$

(b) $c = 0.9$, $N = 21$

Figure 2: Convergence of the LDC algorithm

(a) $c = 0.9$, $N = 11$

(b) $c = 0.9$, $N = 21$

Figure 3: Convergence of the LDC algorithm

We measure the CPU time spent on the computation by the LDC algorithm with refinement factor 4 and equivalent uniform grid solution with space step $H/4$. It should be noted that the measurements of the CPU times cannot be considered as absolute, since they are machine-dependent; even on the same computer they could differ depending on the load of the machine at that particular moment. However, the results in Table 2 were obtained on the same machine and conditions (that is system activity, background processes, etc.) were approximately the same. Moreover the data in Table 2 is the averaged data after 25 runs of the same test.

As one can see from Table 2, LDC algorithm even for one-dimensional problems gives considerable savings in wall calculation time compared with equivalent uniform grid. Taking into account that the accuracy of both methods is the same, we can conclude that for problems with some high activity regions and smooth solutions in the rest of the domain even in one-dimension, LDC is the method to use.

5.2 Example 2

In this section, we consider the LDC algorithm combined with the high order finite difference schemes in the algorithm described in Section 4.2. For the numerical experiment, we apply the

c	N	LDC	fine uniform grid	c	N	LDC	fine uniform grid
0.0	11	0.067	0.056	0.5	11	0.071	0.052
	21	$2.52 * 10^{-4}$	$2.55 * 10^{-4}$		21	$2.21 * 10^{-4}$	$2.49 * 10^{-4}$
	41	$9.92 * 10^{-6}$	$9.92 * 10^{-6}$		41	$1.02 * 10^{-5}$	$1.02 * 10^{-5}$
	81	$7.41 * 10^{-6}$	$7.41 * 10^{-6}$		81	$7.57 * 10^{-6}$	$7.57 * 10^{-6}$
	161	$3.72 * 10^{-6}$	$3.72 * 10^{-6}$		161	$3.74 * 10^{-6}$	$3.74 * 10^{-6}$
0.1	11	0.067	0.056	0.9	11	0.050	0.041
	21	$2.48 * 10^{-4}$	$2.54 * 10^{-4}$		21	$3.13 * 10^{-4}$	$1.96 * 10^{-4}$
	41	$9.95 * 10^{-6}$	$9.95 * 10^{-6}$		41	$1.12 * 10^{-5}$	$1.12 * 10^{-5}$
	81	$7.43 * 10^{-6}$	$7.43 * 10^{-6}$		81	$7.83 * 10^{-6}$	$7.83 * 10^{-6}$
	161	$3.73 * 10^{-6}$	$3.73 * 10^{-6}$		161	$4.01 * 10^{-6}$	$4.01 * 10^{-6}$

Table 1: $\|\mathbf{u}^{exact} - \mathbf{u}^H\|_\infty$ for LDC algorithm and equivalent uniform grid

c	N	LDC (# iter)	equivalent uniform grid	c	N	LDC (# iter)	equivalent uniform grid
0.0	11	0.067(5)	0.026	0.5	11	0.0457(5)	0.0248
	21	0.085(4)	0.0737		21	0.0873(4)	0.0825
	41	0.209(3)	0.3461		41	0.2271(3)	0.3786
	81	0.574(2)	1.9890		81	0.6157(2)	1.9525
	161	1.8668(1)	7.6213		161	1.73(1)	11.6560
0.1	11	0.0465(5)	0.0272	0.9	11	0.0397(5)	0.0273
	21	0.0893(4)	0.0805		21	0.0725(4)	0.0785
	41	0.2299(3)	0.3818		41	0.2264(3)	0.3778
	81	0.7467(2)	2.0971		81	0.6222(2)	2.1028
	161	1.8111(1)	11.2755		161	1.7290(1)	11.5411

Table 2: Calculation time for LDC algorithm and equivalent uniform grid

LDC algorithm to the boundary value problem

$$\begin{cases} -\epsilon_1 \frac{\partial^2 u}{\partial x^2} - \epsilon_2 \frac{\partial^2 u}{\partial y^2} + a_1 \frac{\partial u}{\partial x} + a_2 \frac{\partial u}{\partial y} + cu = f, & (x, y) \in \Omega = (0, 1) \times (0, 1), \\ u(x, 0) = g_1(x), u(x, 1) = g_2(x), u(0, y) = g_3(y), u(1, y) = g_4(y). \end{cases} \quad (30)$$

In (30), f and g have been chosen such that

$$u(x, y) = \tanh[25(x + y - 1/3)] + 1. \quad (31)$$

We choose a uniform coarse grid Ω^H in Ω with grid sizes $\Delta x = \Delta y = 1/(N - 1)$, with $N = 11, 16, 21$. The area of refinement Ω_l is chosen as $\Omega_l = (0, 1/2)^2$. We choose a uniform fine grid Ω_l^h in Ω_l with grid sizes $\Delta x = \Delta y = h$ with $h = H/2, H/4$. For the coarse grid discretization we tested different combinations of convective and diffusive terms discretizations. For the fine grid we used the same schemes as for the coarse grid, although it is not required.

The typical results for the LDC algorithm can be found in Figure 4. All the numerical tests show fast convergence of the LDC algorithm. Usually the LDC algorithm reaches the fixed point solution in one or two iterations. We also compare the accuracy of results for the LDC solution and for the fine uniform grid - the accuracy is similar, although the LDC gives a substantial savings in the size of the problem. For instance, suppose we have a 21×21 uniform grid. This leads us to a linear system with a matrix of dimension 2205×2205 . If we use LDC, we can get the same accuracy by using two 11×11 grids - coarse and fine. This leads to two linear systems with matrices of dimension 605×605 .

We compare accuracy and performance of the LDC method with the solution on a fine uniform grid. Results are in Tables 3 and 4. The accuracy of the LDC method is exactly the same as for the equivalent uniform grid. In Table 3 the results for the LDC technique with an 11×11 coarse grid and a 21×21 fine grid do not differ from those for 11×11 for both coarse and fine grids due to the fact that the main error region is no longer in the area of refinement (see Figure

Figure 4: Difference between exact and numerical results for LDC technique applied to 2D convection-diffusion equation.

Grid size	init	1 iteration	uniform grid
Coarse: 11×11 , fine: 11×11 , equiv.: 21×21	$2.56 * 10^{-2}$	$1.30 * 10^{-3}$	$1.26 * 10^{-3}$
Coarse: 11×11 , fine: 21×21 , equiv.: 41×41	$2.56 * 10^{-2}$	$1.28 * 10^{-3}$	$3.46 * 10^{-5}$
Coarse: 21×21 , fine: 21×21 , equiv.: 41×41	$1.26 * 10^{-3}$	$3.23 * 10^{-5}$	$3.46 * 10^{-5}$

Table 3: $\|\mathbf{u}^{exact} - \mathbf{u}^H\|_\infty$ for LDC algorithm and equivalent uniform grid

4 (a)). We expect that the LDC algorithm should be more efficient than the uniform fine grid method. This is indeed what we see in Table 4 - LDC is much faster than the equivalent uniform grid method.

References

- [1] N. Adam and K. Shariff. A high-resolution hybrid compact-eno scheme for shok-tubulence interaction problems. *J. Computational Physics*, 127, 1996.
- [2] M. J. H. Anthonissen, R. M. M. Mattheij, and J. H. M. ten Thije Boonkkamp. Convergence analysis of the local defect correction method for diffusion equations. *Numerische Mathematik*, 95(3):401–425, 2003.
- [3] M. J. H. Anthonissen, B. van 't Hof, and A. A. Reusken. A finite volume scheme for solving elliptic boundary value problems on composite grids. *Computing*, 61:285–305, 1998.
- [4] Martijn J. H. Anthonissen. *Local defect correction techniques: analysis and application to combustion*. PhD thesis, Eindhoven University of Technology, Eindhoven, 2001.

Grid size	1 iteration	equiv. uniform grid
Coarse: 11×11 , fine: 11×11 , equiv.: 21×21	2.60	7.71
Coarse: 11×11 , fine: 21×21 , equiv.: 41×41	1.84	530
Coarse: 21×21 , fine: 21×21 , equiv.: 41×41	9.988	530

Table 4: Calculation time for LDC algorithm and equivalent uniform grid

- [5] I. Chrisite. Upwind compact finite difference schemes. *J. Computational Physics*, 59:353–368, 1985.
- [6] L. Collatz. *The Numerical Treatment of Differential Equations*. Springer, Berlin, 1961.
- [7] Peter J. J. Ferket. *Solving boundary value problems on composite grids with an application to combustion*. PhD thesis, Eindhoven University of Technology, Eindhoven, 1996.
- [8] Peter J. J. Ferket and Arnold A. Reusken. Further analysis of the local defect correction method. *Computing*, 56:117–139, 1996.
- [9] Peter J. J. Ferket and Arnold A. Reusken. A finite difference discretization method on composite grids. *Computing*, 56:343–369, 1996.
- [10] W. Hackbush. *Elliptic Differential Equations. Theory and Numerical Treatment*. Springer, Berlin, 1992.
- [11] R. S. Hirsch. Higher order accurate difference solutions of fluid mechanics problems by a compact differencing scheme. *J. Computational Physics*, 19:20–109, 1975.
- [12] Z. Kopal. *Numerical Analysis*. John Willey, New York, 1961.
- [13] E. Krause. Mehrstellen Verfahren zur Integration der Grenzschichtgleichungen. *DLR Mitteilungen*, 71:109–40, 1971.
- [14] S. K. Lele. Compact finite difference schemes with spectral-like resolution. *J. Computational Physics*, 103:16–42, 1991.
- [15] D. Gottlieb M. Carpenter and S. Abarbanel. Stability of numerical boundary treatment for compact high-order finite-difference schemes. *J. Computational Physics*, 108:272–295, 1993.
- [16] K. Mahesh. A family of high order difference schemes with good spectral resolution. *J. Computational Physics*, 145:332–358, 1998.
- [17] Vsevolod Nefedov. *Numerical analysis of viscous flow using composite grids with application to glass furnaces*. PhD thesis, Eindhoven University of Technology, Eindhoven, 2001.
- [18] R. Peyret. *Introduction to high-order approximation methods for computational fluid dynamics. In Advanced Turbulent flow calculations*. Springer, Berlin, 2000.
- [19] S. G. Rubin and R. A. Graves. Viscous flow solutions with a cubic spline approximation. *Computer & Fluids*, 3:1–36, 1975.
- [20] S. G. Rubin and P. K. Khosla. Polynomial interpolation method for viscous flow calculations. *J. Computational Physics*, 24:217–46, 1976.
- [21] C. Tam and J. Webb. Dispersion-relation-preserving finite differences schemes for computational acoustics. *J. Computational Physics*, 107:262–281, 1993.
- [22] A. Tolstykh and M. Lipavskii. On performance of methods with third- and fifth-order compact upwind differencing. *J. Computational Physics*, 140:205–232, 1998.
- [23] A. van der Sluis. Equilibration and pivoting in linear algebraic systems. In *IFIP Congress (1)*, pages 127–129, 1968.
- [24] Bas van ’t Hof. *Numerical aspects of laminar flame simulation*. PhD thesis, Eindhoven University of Technology, Eindhoven, 1998.
- [25] Jens Uwe Wappler. *Die lokale Defektkorrekturmethode zur adaptiven Diskretisierung elliptischer Differentialgleichungen mit finiten Elementen*. PhD thesis, Christian-Albrechts-Universität, Kiel, 1999. In German.

- [26] X. Zhong. High-order finite-difference schemes for numerical simulation of hypersonic boundary-layer transition. *J. Computational Physics*, 144:662–709, 1998.