Available online at www.sciencedirect.com
SCIENCE DIRECT®
ELSEVIER
Journal of Computational and Applied Mathematics 187 (2006) 142–170
JOURNAL OF COMPUTATIONAL AND APPLIED MATHEMATICS
www.elsevier.com/locate/cam

High-order compact solvers for the three-dimensional Poisson equation

Godehard Sutmann*, Bernhard Steffen

Central Institute for Applied Mathematics (ZAM) and John von Neumann Institute for Computing (NIC), Research Centre Jülich (FZJ), D-52425 Jülich, Germany

Received 3 December 2003; received in revised form 21 February 2005

Abstract

New compact approximation schemes for the Laplace operator of fourth- and sixth-order are proposed. The schemes are based on a Padé approximation of the Taylor expansion for the discretized Laplace operator. The new schemes are compared with other finite difference approximations in several benchmark problems. It is found that the new schemes exhibit a very good performance and are highly accurate. Especially on large grids they outperform noncompact schemes.

Keywords: Poisson equation; Compact solvers; Iterative solvers; Padé approximation

1. Introduction

Consider the three-dimensional Poisson equation with Dirichlet boundary conditions

$$\Delta u(x, y, z) = -f(x, y, z), \quad x, y, z \in \Omega, \tag{1}$$

$$u(x, y, z) = u_0(x, y, z), \quad x, y, z \in \partial\Omega, \tag{2}$$

* Corresponding author. Tel.: +49 2461 61 6746; fax: +49 2461 61 6656.
E-mail addresses: g.sutmann@fz-juelich.de (G. Sutmann), b.steffen@fz-juelich.de (B. Steffen).

0377-0427/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.cam.2005.03.041

where $u(x, y, z)$ is a field defined in the volume Ω with prescribed values $u_0(x, y, z)$ on the domain boundaries $\partial\Omega$ and $f(x, y, z)$ is a known source function. In a discretized form this can be written as

$$(u_{xx})_{ijk} + (u_{yy})_{ijk} + (u_{zz})_{ijk} = -f_{ijk}, \tag{3}$$

where (i, j, k) denote three-dimensional lattice indices and $(u_{\alpha\alpha})_{ijk}$ is an approximation to the second partial derivative with respect to the coordinate direction α. The simplest approximation is obtained by

$$(u_{\alpha\alpha})_{ijk} = \frac{1}{h_\alpha^2}\,\delta_\alpha^2 u_{ijk}, \tag{4}$$

where h_α is the grid spacing in direction α and δ_α^2 is the second-order difference operator, i.e.

$$\delta_x^2 = u_{i-1,j,k} - 2u_{i,j,k} + u_{i+1,j,k}. \tag{5}$$

Higher-order finite difference operators are derived from the approximation [1]

$$\left.\frac{\partial^2 u}{\partial\alpha^2}\right|_{\alpha=ih_\alpha} = \frac{4}{h_\alpha^2}\left[\sinh^{-1}\left(\frac{\delta_\alpha}{2}\right)\right]^2 \tag{6}$$

$$= \frac{1}{h_\alpha^2}\,\delta_\alpha^2\left\{1 - \frac{1}{12}\,\delta_\alpha^2 + \frac{1}{90}\,\delta_\alpha^4 - \frac{1}{560}\,\delta_\alpha^6 \pm \cdots\right\} u_{i,j,k}. \tag{7}$$

A fourth-order accurate scheme may be derived from Eq. (7) when considering only the first two terms in the expansion

$$(u_{\alpha\alpha})_{ijk} = \frac{1}{h_\alpha^2}\,\delta_\alpha^2\left(1 - \frac{1}{12}\delta_\alpha^2\right). \tag{8}$$

The explicit expression in terms of $u_{i,j,k}$ reads for the x-component

$$(u_{xx})_{ijk} = -\frac{1}{h_x^2}\,\frac{1}{12}\,(u_{i-2,j,k} - 16u_{i-1,j,k} + 30u_{i,j,k} - 16u_{i+1,j,k} + u_{i+2,j,k}). \tag{9}$$

If $\Delta_{i,j,k}$ is defined as the difference scheme and $h_\alpha = h\ \forall\alpha$, the resulting 13-point stencil can be written as

$$\Delta_{i,0,k} = -\frac{1}{12h^2}\begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -16 & 0 & 0 \\ 1 & -16 & 90 & -16 & 1 \\ 0 & 0 & -16 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}, \tag{10}$$

$$\Delta_{i,\pm1,k} = \frac{1}{12h^2}\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 16 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \tag{11}$$

$$\Delta_{i,\pm 2,k} = -\frac{1}{12h^2}\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{12}$$

This representation of a fourth-order scheme points onto a problem arising in the straight forward derivation of higher-order schemes: increasing the order of the finite difference schemes implies an inclusion of neighbor grid points of increasing extent. In the case of the fourth-order scheme one has to operate on the next-nearest gridpoint in the solver. This may give rise to a problem especially at those points close to the boundary. Here, one might use an asymmetric solver. Other possibilities are either to switch to a different solver of lower order, which may influence the overall accuracy of the solution, or to introduce a second boundary layer. This, however, is often impossible due to limited information. In the case of multigrid methods [8] this problem becomes even more obvious, since one should have to work consistently with two boundary layers on the coarse grid levels, which is not included in the formalism.

It is therefore of considerable interest to construct compact higher-order schemes, which need less information from neighbor grid points in space than those schemes derived directly from Eq. (7). In the ideal case compact schemes only need information from the next nearest grid points and therefore do not get into conflict with one-layer boundary conditions.

For the case of fourth-order solvers a compact scheme, using only one prescribed boundary layer, was derived by Spotz and Carey [7], who use a 19-point stencil for the left-hand side of Eq. (3) (henceforth called D4cc). In addition, also the right-hand side is modified in a way. It also takes into account nearest-neighbor source terms

$$\begin{aligned} &\frac{1}{h^2}\Bigg(4u_{ijk} - \frac{1}{3}\left(u_{i-1,j,k} + u_{i+1,j,k} + u_{i,j-1,k} + u_{i,j+1,k} + u_{i,j,k-1} + u_{i,j,k+1}\right) \\ &\quad - \frac{1}{6}\left(u_{i-1,j-1,k} + u_{i-1,j+1,k} + u_{i-1,j,k-1} + u_{i-1,j,k+1} + u_{i,j-1,k-1} + u_{i,j-1,k+1}\right. \\ &\quad \left. + u_{i+1,j-1,k} + u_{i+1,j+1,k} + u_{i+1,j,k-1} + u_{i+1,j,k+1} + u_{i,j+1,k-1} + u_{i,j+1,k+1}\right)\Bigg) \\ &= f_{i,j,k} + \frac{1}{12}\left(f_{i-1,j,k} + f_{i+1,j,k} + f_{i,j-1,k} + f_{i,j+1,k} + f_{i,j,k-1} + f_{i,j,k+1}\right). \end{aligned} \tag{13}$$

Furthermore a sixth-order solver was derived in Spotz and Carey [7] (henceforth called D6cc) which only needs nearest grid point information for the approximation of the Laplace operator. This approximation, however, is only obtained via a modified source term in the Poisson equation, where derivatives up to fourth-order appear. In the case, where derivatives of the sources are known this scheme is rather attractive. Otherwise, either an approximation for the derivative of the source function has to be made on the boundaries or two boundary layers have to be taken into account for a finite difference solution. The stencil notation for this approximation is

$$\Delta_{i,0,k} = \frac{1}{30h^2}\begin{bmatrix} 3 & 14 & 3 \\ 14 & -128 & 14 \\ 3 & 14 & 3 \end{bmatrix}, \tag{14}$$

$$\Delta_{i,\pm1,k} = \frac{1}{30h^2}\begin{bmatrix} 1 & 3 & 1 \\ 3 & 14 & 3 \\ 1 & 3 & 1 \end{bmatrix}. \tag{15}$$

The source term function is thereby modified to

$$f \to f + \frac{h^2}{12}\left(\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} + \frac{\partial^2 f}{\partial z^2}\right) + \frac{h^4}{360}\left(\frac{\partial^4 f}{\partial x^4} + \frac{\partial^4 f}{\partial y^4} + \frac{\partial^4 f}{\partial z^4}\right) + \frac{h^4}{180}\left(\frac{\partial^4 f}{\partial x^2 \partial y^2} + \frac{\partial^4 f}{\partial x^2 \partial z^2} + \frac{\partial^4 f}{\partial y^2 \partial z^2}\right). \tag{16}$$

This modified right-hand side (RHS) may be calculated analytically on each grid point (if possible) or evaluated numerically with a sixth-order finite difference approximation.

In Ref. [5] an approach, based on a Padé approximation was developed for two-dimensional reaction –diffusion equations. This approach was extended to three dimensions in Ref. [4]. In the present paper the Padé approximation technique will be used to derive several forms of Poisson solvers of higher order. In Section 2 the solvers will be derived. In Section 3 test cases are considered to check the validity of the new solvers. An error analysis is given and also the performance, efficiency and convergence characteristics are discussed. Section 4 will give some conclusions and an outview for future research.

2. Theory

In this section a Padé approximation to the bracketed expression in Eq. (7) will be used to derive different forms of compact stencils for the Poisson equation. The term *compact* will be used in the following for numerical schemes, which need less neighbor grid points than the straight-forward expansion approach of Eq. (7). As shown below, the schemes of sixth-order require the inclusion of points that are not nearest neighbors, which causes problems at the boundary. Unmodified they are feasible only for problems where data for a neighboring line outside can be given, as in periodic problems or an *open* boundary problem where the boundary data results from a multipole expansion of the sources, and sources are nonzero only far from the boundary. A solution of this problem may be found by using asymmetric high-order schemes at the boundary or—if sources are only in the interior—schemes where only the right-hand side modification has an extend beyond the nearest neighbors. Such schemes exist, but are not in the scope of the present investigation, as they are generally inferior to the symmetric schemes with extent of the right-hand side (Γ, see below) less or equal to the extend of the left-hand side (Δ).

For the sequel we define the $[m, n]$-Padé approximation of a function $f(x)$ as

$$\mathscr{P}_{m,n}[f(x)] = \frac{\sum_{k=0}^{m} a_k x^k}{1 + \sum_{k=1}^{n} b_k x^k} \equiv R(x), \tag{17}$$

where a_k and b_k are chosen in a way that

$$\left.\frac{\partial^k R}{\partial x^k}\right|_{x=0} = \left.\frac{\partial^k f}{\partial x^k}\right|_{x=0}, \quad k = 0, \ldots, m+n. \tag{18}$$

One could argue that there is an infinite number of finite difference schemes which may be derived on the bases of the Padé approximation. Deriving finite difference schemes of order $\mathcal{O}(n)$ with an approximation

$\mathscr{P}_{k,l}$, where $k+l+2>n$ and keeping only terms in the resulting expression which are $\mathcal{O}(n)$, will lead to increasingly worse results the larger the $k+l$. We will explicitly consider the example $\mathscr{P}_{2,4}$ for the case of a sixth-order finite difference scheme.

The test cases which are shown will be solved for the case of a Gauss–Seidel solver. Also convergence behavior and error analysis is given for that case. This is not because it is the solver of choice, but since it is the most convenient smoother of a multigrid scheme for Poisson's equation, and its convergence behavior is of central importance to the much more complicated, but far superior convergence of a multigrid scheme based on red–black Gauss–Seidel smoothing.

2.1. A $\mathscr{P}_{0,2}$ compact fourth-order scheme

Eq. (8) may be approximated by an [0, 2]-Padé approximation through

$$(u_{\alpha\alpha})_{ijk} = \frac{1}{h_\alpha^2}\,\delta_\alpha^2\left(1+\frac{1}{12}\delta_\alpha^2\right)^{-1} \tag{19}$$

$$= \frac{1}{h_\alpha^2}\,\delta_\alpha^2 (D_{[0,2],\alpha})^{-1}, \tag{20}$$

where in Eq. (20) the operator $D_{[0,2],\alpha}$ was defined. Inserting this approximation into Eq. (3), it can be written as

$$\left\{\sum_{\alpha=x,y,z}\frac{1}{h_\alpha^2}\,\delta_\alpha^2 (D_{[0,2],\alpha})^{-1}\right\} u_{i,j,k} = -f_{i,j,k}. \tag{21}$$

Simple algebraic manipulation gives

$$\begin{aligned}\frac{1}{h^2}\{&[(1+\delta_x^2(D_{[0,2],x})^{-1})(1+\delta_y^2(D_{[0,2],y})^{-1})(1+\delta_z^2(D_{[0,2],z})^{-1})\\ &-\delta_x^2\delta_y^2(D_{[0,2],x})^{-1}(D_{[0,2],y})^{-1}-\delta_x^2\delta_z^2(D_{[0,2],x})^{-1}(D_{[0,2],z})^{-1}-\delta_y^2\delta_z^2(D_{[0,2],y})^{-1}(D_{[0,2],z})^{-1}\\ &-\delta_x^2\delta_y^2\delta_z^2(D_{[0,2],x})^{-1}(D_{[0,2],y})^{-1}(D_{[0,2],z})^{-1}]-1\}u_{i,j,k} = -f_{i,j,k}.\end{aligned} \tag{22}$$

In the next step, both sides of Eq. (22) are multiplied by $D_{[0,2],x}D_{[0,2],y}D_{[0,2],z}$. Since the operators commute, this leads to

$$\frac{1}{h^2}\{\delta_x^2 D_{[0,2],y}D_{[0,2],z}+\delta_y^2 D_{[0,2],x}D_{[0,2],z}+\delta_z^2 D_{[0,2],x}D_{[0,2],y}\}u_{i,j,k} = -g_{i,j,k}, \tag{23}$$

where an effective source term $g_{i,j,k}$ was introduced

$$g_{i,j,k} = \{1+\tfrac{1}{12}(\delta_x^2+\delta_y^2+\delta_z^2)+\tfrac{1}{144}(\delta_x^2\delta_y^2+\delta_x^2\delta_z^2+\delta_y^2\delta_z^2)\}f_{i,j,k}. \tag{24}$$

Keeping terms on the left-hand side (LHS) up to fourth order, Eqs. (23) and (24) may be written finally as

$$\begin{aligned}&\frac{1}{h^2}\left\{\delta_x^2+\delta_y^2+\delta_z^2+\frac{1}{6}(\delta_x^2\delta_y^2+\delta_x^2\delta_z^2+\delta_y^2\delta_z^2)\right\}u_{i,j,k}\\ &\quad = -\left\{1+\frac{1}{12}(\delta_x^2+\delta_y^2+\delta_z^2)+\frac{c_1}{144}(\delta_x^2\delta_y^2+\delta_x^2\delta_z^2+\delta_y^2\delta_z^2)\right\}f_{i,j,k}.\end{aligned} \tag{25}$$

To have the RHS of fourth order, it is necessary to keep the first two terms only. However, a higher-order term has been formally kept in the expression. In the test cases it will be shown that it may improve accuracy in various cases. The factor c_1 takes values of zero or one. It was introduced in order to control the higher-order approximation of the RHS. Note that the higher-order term comes naturally from the Padé approximation and is not an artificial construction. Putting the constant c_1 to zero, reduces this scheme to the one introduced in Ref. [7] on the basis of a different approach. Here and in the following the extended source function is acceptable from the computational point of view since it is calculated only once before starting the iterative process. Therefore, the computational overhead is negligible.

If a stencil notation is used and the Laplace operator is denoted as $\Delta_{i,j,k}$ whereas the source term operator, appearing on the RHS is denoted as $\Gamma_{i,j,k}$, the operators take the form

$$\Delta_{i,0,k} = \frac{1}{6h^2}\begin{bmatrix} 1 & 2 & 1 \\ 2 & -24 & 2 \\ 1 & 2 & 1 \end{bmatrix}, \quad \Gamma_{i,0,k} = \frac{1}{144}\begin{bmatrix} c_1 & 12-4c_1 & c_1 \\ 12-4c_1 & 72+12c_1 & 12-4c_1 \\ c_1 & 12-4c_1 & c_1 \end{bmatrix}, \tag{26}$$

$$\Delta_{i,\pm1,k} = \frac{1}{6h^2}\begin{bmatrix} 0 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \quad \Gamma_{i,\pm1,k} = \frac{1}{144}\begin{bmatrix} 0 & c_1 & 0 \\ c_1 & 12-4c_1 & c_1 \\ 0 & c_1 & 0 \end{bmatrix}. \tag{27}$$

2.2. A $\mathscr{P}_{0,4}$ compact sixth-order scheme

In this section, the expansion of Eq. (7) is considered up to sixth order. The Padé approximation in this case is

$$(u_{\alpha\alpha})_{ijk} = \frac{1}{h_\alpha^2}\delta_\alpha^2\left(1+\frac{1}{12}\delta_\alpha^2-\frac{1}{240}\delta_\alpha^4\right)^{-1}, \tag{28}$$

$$= \frac{1}{h_\alpha^2} D^{-1}_{[0,4],\alpha}, \tag{29}$$

which leads to the following form of the Poisson equation:

$$\left\{\sum_{\alpha=x,y,z}\frac{1}{h_\alpha^2}\delta_\alpha^2 D^{-1}_{[0,4],\alpha}\right\} u_{i,j,k} = -f_{i,j,k}. \tag{30}$$

As before, both sides of Eq. (30) are multiplied by $\prod_{\alpha=x,y,z} D_{[0,4],\alpha}$ to give

$$\left\{\sum_{\alpha}\frac{1}{h_\alpha^2}\delta_\alpha^2 D_{[0,4],\beta} D_{[0,4],\gamma}\right\} = -\left\{\prod_{\alpha=x,y,z} D_{[0,4],\alpha}\right\} f_{i,j,k}, \tag{31}$$

where $\{(\beta,\gamma) \neq \alpha \wedge \beta \neq \gamma\}$. This can be written as

$$\left\{\sum_{\alpha}\frac{1}{h_\alpha^2}\delta_\alpha^2\left(1+\frac{1}{12}\sum_{\beta\neq\alpha}\delta_\beta^2-\frac{1}{240}\sum_{\beta\neq\alpha}\delta_\beta^4+\frac{1}{144}\prod_{\beta\neq\alpha}\delta_\beta^2\right)\right\} u_{i,j,k} = -g_{i,j,k}, \tag{32}$$

where an extended source term was introduced containing terms up to sixth order:

$$g_{i,j,k} = \left\{ 1 + \frac{1}{12} \sum_{\alpha} \delta_{\alpha}^2 \left(1 - \frac{1}{20} \delta_{\alpha}^2 + \frac{1}{24} \sum_{\beta \neq \alpha} \delta_{\beta}^2 - \frac{c_2}{240} \sum_{\beta \neq \alpha} \delta_{\beta}^4 \right) \right\} f_{i,j,k}. \tag{33}$$

Using the stencil notation as before, the finite difference approximation scheme for the Laplace operator results in

$$\Delta_{i,0,k} = \frac{1}{240\,h^2} \begin{bmatrix} 0 & -1 & 4 & -1 & 0 \\ -1 & 38 & 72 & 38 & -1 \\ 4 & 72 & -928 & 72 & 4 \\ -1 & 38 & 72 & 38 & -1 \\ 0 & -1 & 4 & -1 & 0 \end{bmatrix}, \tag{34}$$

$$\Delta_{i,\pm1,k} = \frac{1}{240\,h^2} \begin{bmatrix} 0 & 0 & -1 & 0 & 0 \\ 0 & 5 & 38 & 5 & 0 \\ -1 & 38 & 72 & 38 & -1 \\ 0 & 5 & 38 & 5 & 0 \\ 0 & 0 & -1 & 0 & 0 \end{bmatrix}, \quad \Delta_{i,\pm2,k} = \frac{1}{240\,h^2} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & 4 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{35}$$

Concerning the RHS of Eq. (32) it is enough to keep the first three terms in the inner brackets of Eq. (33) in order to be consistent with a sixth-order approximation. However, as before the next higher terms may be kept in order to perform a higher-order approximation for the RHS of Eq. (32). As will be shown later, a better approximation for various cases is obtained when keeping also the fourth term. Consequently, a constant c_2 is introduced which takes values zero or one. In stencil notation the source term operator then reads

$$\Gamma_{i,0,k} = \frac{1}{2880} \begin{bmatrix} 0 & -c_2 & -12+4c_2 & -c_2 & 0 \\ -c_2 & 20+8c_2 & 208-28c_2 & 20+8c_2 & -c_2 \\ -12+4c_2 & 208-28c_2 & 1464+72c_2 & 208-28c_2 & -12+4c_2 \\ -c_2 & 20+8c_2 & 208-28c_2 & 20+8c_2 & -c_2 \\ 0 & -c_2 & -12+4c_2 & -c_2 & 0 \end{bmatrix}, \tag{36}$$

$$\Gamma_{i,\pm1,k} = \frac{1}{2880} \begin{bmatrix} 0 & 0 & -c_2 & 0 & 0 \\ 0 & 0 & 20+8c_2 & 0 & 0 \\ -c_2 & 20+8c_2 & 208-28c_2 & 20+8c_2 & -c_2 \\ 0 & 0 & 20+8c_2 & 0 & 0 \\ 0 & 0 & -c_2 & 0 & 0 \end{bmatrix}, \tag{37}$$

$$\Gamma_{i,\pm2,k} = \frac{1}{2880} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -c_2 & 0 & 0 \\ 0 & -c_2 & -12+4c_2 & -c_2 & 0 \\ 0 & 0 & -c_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{38}$$

2.3. A $\mathscr{P}_{2,2}$ compact sixth-order scheme

Here the expansion of Eq. (7) is considered again up to sixth order. A [2,2]-Padé approximation is used, so that Eq. (7) is rewritten as

$$(u_{\alpha\alpha})_{i,j,k} = \frac{1}{h_\alpha}\,\delta_\alpha^2 D^{(1)}_{[2,2],\alpha}(D^{(2)}_{[2,2],\alpha})^{-1}, \tag{39}$$

where

$$D^{(1)}_{[2,2],\alpha} = 1 + \tfrac{1}{20}\,\delta_\alpha^2, \tag{40}$$

$$D^{(2)}_{[2,2],\alpha} = 1 + \tfrac{2}{15}\,\delta_\alpha^2. \tag{41}$$

The Poisson equation, Eq. (3), can then be expressed as

$$\left\{\sum_\alpha \frac{1}{h_\alpha^2}\,\delta_\alpha^2 D^{(1)}_{[2,2],\alpha} D^{(2)}_{[2,2],\beta} D^{(2)}_{[2,2],\gamma}\right\} u_{i,j,k} = -\left\{\prod_\alpha D^{(2)}_{[2,2],\alpha}\right\} f_{i,j,k}. \tag{42}$$

As before the sum on the LHS is understood for $\{(\beta,\gamma) \neq \alpha \wedge \beta \neq \gamma\}$. Keeping terms up to sixth-order in δ_α^2 on the LHS this expression is rewritten in terms of δ_α^2 as

$$\left\{\sum_\alpha \frac{1}{h_\alpha^2}\,\delta_\alpha^2\left(1 + \frac{1}{20}\,\delta_\alpha^2 + \frac{2}{15}\sum_{\beta\neq\alpha}\delta_\beta^2 + \frac{1}{150}\sum_{\beta\neq\alpha}\delta_\alpha^2\delta_\beta^2 + \frac{4}{225}\prod_{\beta\neq\alpha}\delta_\beta^2\right)\right\} u_{i,j,k} = -g_{i,j,k}, \tag{43}$$

where the effective source term

$$g_{i,j,k} = \left\{1 + \sum_\alpha \delta_\alpha^2\left(\frac{2}{15} + \frac{2}{225}\sum_{\beta\neq\alpha}\delta_\beta^2 + \frac{8c_3}{3375}\prod_{\beta\neq\alpha}\delta_\beta^2\right)\right\} f_{i,j,k} \tag{44}$$

was introduced. Again a higher-order approximation is kept formally on the RHS. Setting $c_3{=}0$ introduces the same order of accuracy on both sides of Eq. (43). For $h_\alpha = h\forall\alpha$, the finite difference approximation reads in stencil notation as

$$\Delta_{i,0,k} = \frac{1}{300h^2}\begin{bmatrix} 0 & 2 & 7 & 2 & 0\\ 2 & 32 & 40 & 32 & 2\\ 7 & 40 & -842 & 40 & 7\\ 2 & 32 & 40 & 32 & 2\\ 0 & 2 & 7 & 2 & 0\end{bmatrix}, \tag{45}$$

$$\Delta_{i,\pm1,k} = \frac{1}{300h^2}\begin{bmatrix} 0 & 0 & 2 & 0 & 0\\ 0 & 16 & 32 & 16 & 0\\ 2 & 32 & 40 & 32 & 2\\ 0 & 16 & 32 & 16 & 0\\ 0 & 0 & 2 & 0 & 0\end{bmatrix}, \quad \Delta_{i,\pm2,k} = \frac{1}{300h^2}\begin{bmatrix} 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 2 & 0 & 0\\ 0 & 2 & 7 & 2 & 0\\ 0 & 0 & 2 & 0 & 0\\ 0 & 0 & 0 & 0 & 0\end{bmatrix}, \tag{46}$$

$$\Gamma_{i,0,k} = \frac{1}{3375}\begin{bmatrix} 60-16c_3 & 210+32c_3 & 60-16c_3\\ 210+32c_3 & 1395-64c_3 & 210+32c_3\\ 60-16c_3 & 210+32c_3 & 60-16c_3\end{bmatrix}, \tag{47}$$

$$\Gamma_{i,\pm1,k} = \frac{1}{3375}\begin{bmatrix} 8c_3 & 60-16c_3 & 8c_3 \\ 60-16c_3 & 210+32c_3 & 60-16c_3 \\ 8c_3 & 60-16c_3 & 8c_3 \end{bmatrix}. \tag{48}$$

2.4. A $\mathscr{P}_{2,4}$ compact sixth-order scheme

As before the expansion of Eq. (7) is considered up to sixth order. The terms in brackets are rewritten in terms of a [2,4]-Padé approximation. Eq. (7) may then be written as

$$(u_{\alpha\alpha})_{i,j,k} = \frac{1}{h_\alpha}\,\delta_\alpha^2 D^{(1)}_{[2,4],\alpha}(D^{(2)}_{[2,4],\alpha})^{-1}, \tag{49}$$

where

$$D^{(1)}_{[2,4],\alpha} = 1 - \frac{11}{36}\,\delta_\alpha^2, \tag{50}$$

$$D^{(2)}_{[2,4],\alpha} = 1 - \frac{2}{9}\,\delta_\alpha^2 - \frac{4}{135}\,\delta_\alpha^4. \tag{51}$$

In formal analogy to the [2,2]-Padé approximation, the Poisson equation, Eq. (3) can be written as

$$\left\{\sum_\alpha \frac{1}{h_\alpha^2}\,\delta_\alpha^2 D^{(1)}_{[2,4],\alpha} D^{(2)}_{[2,4],\beta} D^{(2)}_{[2,4],\gamma}\right\} u_{i,j,k} = -\left\{\prod_\alpha D^{(2)}_{[2,4],\alpha}\right\} f_{i,j,k}, \tag{52}$$

where again on the LHS $\{(\beta,\gamma) \neq \alpha \wedge \beta \neq \gamma\}$ is understood. From this approximation it is clear that terms up to order 12 appear in the expression. Since the original approximation is of order six, in this approximation only terms up to order six are kept. This also ensures a compact representation of the Laplace operator. Rewriting Eq. (52) in terms of δ_α^2 gives

$$\left\{\sum_\alpha \frac{1}{h_\alpha^2}\,\delta_\alpha^2\left(1 - \frac{11}{36}\,\delta_\alpha^2 - \frac{2}{9}\sum_{\beta\neq\alpha}\delta_\beta^2 + \frac{31}{810}\sum_{\beta\neq\alpha}\delta_\alpha^2\delta_\beta^2 + \frac{4}{27}\prod_{\beta\neq\alpha}\delta_\beta^2\right)\right\} u_{i,j,k} = -g_{i,j,k}. \tag{53}$$

Again an effective source term was introduced:

$$g_{i,j,k} = \left\{1 - \sum_\alpha \delta_\alpha^2\left(\frac{2}{9} + \frac{4}{135}\,\delta_\alpha^2 - \frac{2}{81}\sum_{\beta\neq\alpha}\delta_\beta^2 + \frac{8c_4}{729}\prod_{\beta\neq\alpha}\delta_\beta^2\right)\right\} f_{i,j,k}, \tag{54}$$

where a higher-order approximation is kept formally on the RHS. The case $c_4 = 0$ corresponds to the same order of accuracy on both sides of Eq. (53). In stencil notation the Laplace and the source term operator read

$$\Delta_{i,0,k} = \frac{1}{1620h^2}\begin{bmatrix} 0 & 62 & -743 & 62 & 0 \\ 62 & -1696 & 9176 & -1696 & 62 \\ -743 & 9176 & -33654 & 9176 & -743 \\ 62 & -1696 & 9176 & -1696 & 62 \\ 0 & 62 & -743 & 62 & 0 \end{bmatrix}, \tag{55}$$

$$\Delta_{i,\pm1,k} = \frac{1}{1620h^2}\begin{bmatrix} 0 & 0 & 62 & 0 & 0 \\ 0 & 240 & -1696 & 240 & 0 \\ 62 & -1696 & 9176 & -1696 & 62 \\ 0 & 240 & -1696 & 240 & 0 \\ 0 & 0 & 62 & 0 & 0 \end{bmatrix},$$

$$\Delta_{i,\pm2,k} = \frac{1}{1620h^2}\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 62 & 0 & 0 \\ 0 & 62 & -743 & 62 & 0 \\ 0 & 0 & 62 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \tag{56}$$

$$\Gamma_{i,0,k} = \frac{1}{3645}\begin{bmatrix} 0 & 0 & -108 & 0 & 0 \\ 0 & 180+80c_4 & -1098-160c_4 & 180+80c_4 & 0 \\ -108 & -1098-160c_4 & 8721+320c_4 & -1098-160c_4 & -108 \\ 0 & 180+80c_4 & -1098-160c_4 & 180+80c_4 & 0 \\ 0 & 0 & -108 & 0 & 0 \end{bmatrix}, \tag{57}$$

$$\Gamma_{i,\pm1,k} = \frac{1}{3645}\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & -40c_4 & 180+80c_4 & -40c_4 & 0 \\ 0 & 180+80c_4 & -1098-160c_4 & 180+80c_4 & 0 \\ 0 & -40c_4 & 180+80c_4 & -40c_4 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \tag{58}$$

$$\Gamma_{i,\pm1,k} = \frac{1}{3645}\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -108 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{59}$$

3. Results

3.1. Test cases

In order to validate the finite difference schemes, four different test cases are considered. The first two test cases are the same ones as those which have already been applied in [7,9]. In this section, we apply the lexicographic Gauss–Seidel iteration scheme and compare the discretizations, based on Padé approximations D4c$_{P02}$ (Eqs. (26) and (27)), D6c$_{P02}$ (Eqs. (45)–(47)), D6c$_{P04}$ (Eqs. (34)–(36)) and D6c$_{P24}$ (Eqs. (55)–(57)). Also the second- (D2$_{1bl}$), fourth- (D4$_{2bl}$) and sixth-order (D6$_{3bl}$) finite difference approximations which result from Eq. (7) are checked and they require one, two or three boundary layers for the solution, respectively. In addition the sixth-order scheme, D6cc (Eqs. (14)–(16)), is compared with the new Padé solvers.

The fact that several solvers need more than one boundary layer for a solution implies that the potential on these layers must be known explicitly and according to the finite difference scheme a reduced number of unknowns has to be solved. For the test cases considered here it is no principal problem to specify two or three boundary layers as known a priori since their solution is known analytically. For general

problems, however, this may pose a problem, since one either has to *guess* the potential on a second or even third boundary layer (e.g., by extrapolation) or one has to switch to another discretization scheme, requiring only one boundary layer information (either an unsymmetric one or a symmetric one of lower order). The consequences of this procedure are not yet investigated and will be considered in a future work. Another problem related to boundary conditions may rise if the right-hand side needs information from more neighboring points than the left-hand side (e.g., if higher-order derivatives of the source term appear on the RHS which are calculated by a finite difference scheme) as it may be the case, e.g., for D6cc. In this case the unknowns near the boundary would need information about the source distribution extending to outside the computed region Ω, which gives rise to an approximation of the source function outside of Ω. Note that for all finite difference schemes developed in this work this does not pose a problem since the extent of the LHS is always larger or equal than the RHS of the Poisson equation. Therefore, knowing the source function distribution $f_{i,j,k}$ one can always calculate the necessary information on the RHS to solve the Poisson equation appropriately with the new discretization schemes. Concerning the solver D6cc we note that we treat it here as in the general case, where an analytical formulation of the second and fourth derivative of the source function is not available, i.e., these derivatives are calculated as finite difference approximations, requiring two neighbored grid points. Consequently, D6cc is treated with two boundary layers, as it is for $\mathrm{D6c}_{P04}$, $\mathrm{D6c}_{P22}$ and $\mathrm{D6c}_{P24}$.

The test problems are discretized on a normalized cube $(0, 1)^3$, where the mesh size h is varied between $h = \frac{1}{8}$ and $\frac{1}{128}$. All calculations are carried out until the norm of the residuum per grid point is reduced to $\|\mathbf{r}\| \leqslant \varepsilon_{\mathrm{res}}$ (where $\varepsilon_{\mathrm{res}}$ is a threshold value of the residuum. This value generally depends on the solver and grid size (cmp. Section 3.2). The norm of the residuum is defined as

$$\|\mathbf{r}^{(n)}\| = \sqrt{\frac{1}{N}\sum_{i,j,k}\left(\Delta_{i,j,k}u^{(n)}_{i,j,k} + f_{i,j,k}\right)^2}, \tag{60}$$

where $u^{(n)}_{i,j,k}$ is the field approximation after n iterations.

All calculations were performed on a LINUX IBM-T30 notebook with a Pentium-IV 2 GHz processor and the program was compiled with the Intel Fortran 90 compiler.

As a measure of accuracy the relative error norm is considered:

$$\varepsilon_{\mathrm{rel}} = \frac{\|\mathbf{u}^{(\mathrm{ex})} - \mathbf{u}\|}{\|\mathbf{u}^{(\mathrm{ex})}\|}, \tag{61}$$

where $u^{(\mathrm{ex})}_{i,j,k}$ is the exact solution of the model problem.

In the following the test cases are introduced. For Test Cases 1–3 results are reported in Tables 1–3 for second-, fourth- and sixth-order solvers in terms of the relative error norm, Eq. (61), discretization error, Eq. (75) and excess parameter, Eq. (86). Results for Test Case 4 are presented in Table 4 .

1. *Test Case* 1: The first test case consists in solving the Poisson equation with the following source term distribution and vanishing boundary conditions:

$$f_{i,j,k} = 3\pi^2 \sin(\pi i h_x)\sin(\pi j h_y)\sin(\pi k h_z) \tag{62}$$

for which the analytical solution is found to be

$$u_{i,j,k} = \sin(\pi i h_x)\sin(\pi j h_y)\sin(\pi k h_z). \tag{63}$$

Table 1
Discretization error norm $\varepsilon_h = \|\varepsilon_h\|$ and relative error ε_{rel} for Test Case 1 (Poisson equation where u is eigenfunction of Laplace operator), solved with Gauss–Seidel algorithm with second-, fourth- and sixth-order approximations for the Laplace operator as described in the text

	Solver grid	D2	D4$_{2bl}$	D4cP$_{0,2}$	D6$_{3bl}$	D6cP$_{0,4}$	D6cP$_{2,2}$	D6cc	D6cP$_{2,4}$
ε_h	$\frac{1}{8}$	5.594×10^{-3}	8.682×10^{-5}	1.015×10^{-4}	1.216×10^{-6}	3.980×10^{-7}	2.572×10^{-6}	9.950×10^{-7}	1.277×10^{-5}
	$\frac{1}{16}$	1.254×10^{-3}	5.771×10^{-6}	5.651×10^{-6}	3.048×10^{-8}	6.157×10^{-9}	3.783×10^{-8}	1.540×10^{-8}	2.586×10^{-7}
	$\frac{1}{32}$	2.980×10^{-4}	3.650×10^{-7}	3.352×10^{-7}	5.349×10^{-10}	9.544×10^{-11}	5.778×10^{-10}	2.386×10^{-10}	4.330×10^{-9}
	$\frac{1}{64}$	7.270×10^{-5}	2.284×10^{-8}	2.044×10^{-8}	8.181×10^{-12}	1.581×10^{-12}	8.900×10^{-12}	3.642×10^{-12}	6.660×10^{-11}
	$\frac{1}{128}$	1.775×10^{-5}	1.368×10^{-9}	1.248×10^{-9}	2.206×10^{-12}	5.006×10^{-13}	6.970×10^{-14}	3.927×10^{-13}	—[a]
ε_{rel}	$\frac{1}{8}$	1.295×10^{-2}	1.213×10^{-4}	2.349×10^{-4}	7.896×10^{-7}	5.563×10^{-7}	3.595×10^{-6}	1.391×10^{-6}	1.784×10^{-5}
	$\frac{1}{16}$	3.219×10^{-3}	1.195×10^{-5}	1.451×10^{-5}	4.915×10^{-8}	1.275×10^{-8}	7.836×10^{-8}	3.190×10^{-8}	5.358×10^{-7}
	$\frac{1}{32}$	8.036×10^{-4}	8.906×10^{-7}	9.040×10^{-7}	1.173×10^{-9}	2.329×10^{-10}	1.410×10^{-9}	5.822×10^{-10}	1.057×10^{-8}
	$\frac{1}{64}$	2.008×10^{-4}	6.012×10^{-8}	5.646×10^{-8}	2.048×10^{-11}	4.162×10^{-12}	2.342×10^{-11}	9.584×10^{-12}	1.753×10^{-10}
	$\frac{1}{128}$	5.020×10^{-5}	3.868×10^{-9}	3.529×10^{-9}	6.240×10^{-12}	1.416×10^{-12}	1.971×10^{-13}	1.111×10^{-12}	–[(1)]
x_h	$\frac{1}{8}$	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	$\frac{1}{16}$	0.16	−0.09	0.17	−0.68	0.01	0.09	0.01	−0.37
	$\frac{1}{32}$	0.07	−0.02	0.08	−0.17	0.01	0.03	0.01	−0.10
	$\frac{1}{64}$	0.04	−0.00	0.04	0.03	−0.08	0.02	0.03	0.02
	$\frac{1}{128}$	0.03	0.06	0.03	—[b]	—[b]	—[b]	—[b]	—[a]

Also shown is the *order excess parameter* x_h (Eq. (86)).

[a]Convergence of D6c$_{P24}$ for $h = \frac{1}{128}$ was so slow that it was omitted.

[b]x_h was not computed since machine precision was reached before discretization error.

Table 2
As in Table 1 for Test Case 2 (Laplace equation)

	Solver grid	D2	$D4_{2bl}$	$D4cP_{0,2}$	$D6_{3bl}$	$D6cP_{0,4}$	$D6cP_{2,2}$	D6cc	$D6cP_{2,4}$
ε_h	$\frac{1}{8}$	7.188×10^{-3}	8.473×10^{-5}	1.500×10^{-4}	2.870×10^{-6}	2.013×10^{-6}	1.226×10^{-6}	6.279×10^{-7}	1.887×10^{-5}
	$\frac{1}{16}$	1.649×10^{-3}	6.450×10^{-6}	8.511×10^{-6}	8.853×10^{-8}	3.581×10^{-8}	2.160×10^{-8}	1.146×10^{-8}	4.382×10^{-7}
	$\frac{1}{32}$	3.942×10^{-4}	4.422×10^{-7}	5.071×10^{-7}	1.793×10^{-9}	6.004×10^{-10}	3.597×10^{-10}	1.933×10^{-10}	8.000×10^{-9}
	$\frac{1}{64}$	9.632×10^{-5}	2.891×10^{-8}	3.095×10^{-8}	3.134×10^{-11}	9.655×10^{-12}	5.768×10^{-12}	3.102×10^{-12}	1.354×10^{-10}
ε_{rel}	$\frac{1}{8}$	1.197×10^{-2}	8.519×10^{-5}	2.498×10^{-4}	1.341×10^{-6}	2.024×10^{-6}	1.232×10^{-6}	6.313×10^{-7}	1.897×10^{-5}
	$\frac{1}{16}$	3.416×10^{-3}	1.078×10^{-5}	1.763×10^{-5}	1.152×10^{-7}	5.984×10^{-8}	3.609×10^{-8}	1.915×10^{-8}	7.322×10^{-7}
	$\frac{1}{32}$	9.119×10^{-4}	9.254×10^{-7}	1.173×10^{-6}	3.372×10^{-9}	1.257×10^{-9}	7.530×10^{-10}	4.046×10^{-10}	1.674×10^{-8}
	$\frac{1}{64}$	2.356×10^{-4}	6.737×10^{-8}	7.570×10^{-8}	6.946×10^{-11}	2.250×10^{-11}	1.344×10^{-11}	7.228×10^{-12}	3.156×10^{-10}
x_h	$\frac{1}{8}$	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	$\frac{1}{16}$	0.12	−0.28	0.14	−0.98	−0.19	−0.17	−0.22	−0.57
	$\frac{1}{32}$	0.06	−0.13	0.07	−0.37	−0.10	−0.09	−0.11	−0.22
	$\frac{1}{64}$	0.03	−0.07	0.03	−0.16	−0.04	−0.04	−0.04	−0.12

Table 3
As in Table 1 for Test Case 3 (Poisson equation where u is not eigenfunction of Laplace operator)

	Solver grid	D2	$D4_{2bl}$	$D4cP_{0,2}$	$D6_{3bl}$	$D6cP_{0,4}$	$D6cP_{2,2}$	D6cc	$D6cP_{2,4}$
ε_h	$\frac{1}{8}$	1.318×10^{-2}	5.255×10^{-3}	1.392×10^{-3}	4.124×10^{-3}	1.137×10^{-3}	1.050×10^{-3}	1.230×10^{-3}	2.677×10^{-3}
	$\frac{1}{16}$	2.757×10^{-3}	2.551×10^{-4}	6.573×10^{-5}	4.747×10^{-5}	1.116×10^{-5}	8.525×10^{-6}	1.224×10^{-5}	4.851×10^{-5}
	$\frac{1}{32}$	6.453×10^{-4}	1.422×10^{-5}	3.781×10^{-6}	6.183×10^{-7}	1.495×10^{-7}	1.071×10^{-7}	1.639×10^{-7}	8.263×10^{-7}
	$\frac{1}{64}$	1.568×10^{-4}	8.358×10^{-7}	2.288×10^{-7}	8.818×10^{-9}	2.173×10^{-9}	1.530×10^{-9}	2.381×10^{-9}	1.297×10^{-8}
ε_{rel}	$\frac{1}{8}$	8.596×10^{-2}	2.069×10^{-2}	9.080×10^{-3}	7.546×10^{-3}	4.475×10^{-3}	4.134×10^{-3}	4.841×10^{-3}	1.054×10^{-2}
	$\frac{1}{16}$	1.995×10^{-2}	1.489×10^{-3}	4.755×10^{-4}	2.156×10^{-4}	6.514×10^{-5}	4.975×10^{-5}	7.144×10^{-5}	2.831×10^{-4}
	$\frac{1}{32}$	4.902×10^{-3}	9.776×10^{-5}	2.872×10^{-5}	3.819×10^{-6}	1.028×10^{-6}	7.361×10^{-7}	1.127×10^{-6}	5.680×10^{-6}
	$\frac{1}{64}$	1.221×10^{-3}	6.198×10^{-6}	1.780×10^{-6}	6.219×10^{-8}	1.611×10^{-8}	1.134×10^{-8}	1.766×10^{-8}	9.617×10^{-8}
x_h	$\frac{1}{8}$	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	$\frac{1}{16}$	0.26	0.36	0.40	0.44	0.67	0.94	0.65	−0.21
	$\frac{1}{32}$	0.10	0.16	0.12	0.26	0.22	0.31	0.22	−0.12
	$\frac{1}{64}$	0.04	0.09	0.05	0.13	0.10	0.13	0.11	−0.01

Table 4
Discretization error norm $\varepsilon_h = \|\varepsilon_h\|$ of the near- and far-field parts of Test Case 4 (field of a point charge) as well as the *order excess parameter* x_h

	Solver grid	D2	D4$_{2bl}$	D4cP$_{0,2}$	D6$_{3bl}$	D6cP$_{0,4}$	D6cP$_{2,2}$	D6cc	D6cP$_{2,4}$
ε_h (near)	$\frac{1}{8}$	3.471×10^{-1}	1.855×10^{-1}	3.645×10^{-1}	1.475×10^{-1}	2.153×10^{-1}	1.390×10^{-1}	1.601×10^{-1}	1.069×10^{-1}
	$\frac{1}{16}$	3.530×10^{-1}	1.805×10^{-1}	3.360×10^{-1}	1.452×10^{-1}	1.988×10^{-1}	1.280×10^{-1}	1.475×10^{-1}	9.868×10^{-2}
	$\frac{1}{32}$	2.968×10^{-1}	1.494×10^{-1}	2.774×10^{-1}	1.198×10^{-1}	1.641×10^{-1}	1.057×10^{-1}	1.217×10^{-1}	8.180×10^{-2}
	$\frac{1}{64}$	2.291×10^{-1}	1.150×10^{-1}	2.136×10^{-1}	9.227×10^{-2}	1.264×10^{-1}	8.137×10^{-2}	9.374×10^{-2}	6.299×10^{-2}
ε_h (far)	$\frac{1}{8}$	7.332×10^{-2}	3.699×10^{-2}	6.931×10^{-2}	2.787×10^{-2}	4.103×10^{-2}	2.641×10^{-2}	3.045×10^{-2}	2.040×10^{-2}
	$\frac{1}{16}$	2.678×10^{-2}	9.826×10^{-3}	6.015×10^{-3}	4.175×10^{-3}	4.085×10^{-3}	2.156×10^{-3}	2.576×10^{-3}	2.633×10^{-3}
	$\frac{1}{32}$	4.946×10^{-3}	4.849×10^{-4}	2.544×10^{-4}	5.039×10^{-5}	3.264×10^{-5}	1.563×10^{-5}	1.719×10^{-5}	4.061×10^{-4}
	$\frac{1}{64}$	1.104×10^{-3}	2.548×10^{-5}	1.341×10^{-5}	1.150×10^{-6}	3.517×10^{-7}	2.225×10^{-7}	1.928×10^{-7}	6.028×10^{-6}
x_h (near)	$\frac{1}{8}$	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	$\frac{1}{16}$	−2.02	−3.96	−3.88	−5.98	−5.88	−5.88	−5.88	−5.88
	$\frac{1}{32}$	−1.75	−3.73	−3.72	−5.72	−5.72	−5.72	−5.72	−5.73
	$\frac{1}{64}$	−1.63	−3.62	−3.62	−5.62	−5.62	−5.62	−5.62	−5.62
x_h (far)	$\frac{1}{8}$	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	$\frac{1}{16}$	−0.55	−2.09	−0.47	−3.26	−2.67	−2.39	−2.44	−3.05
	$\frac{1}{32}$	0.44	0.34	0.56	0.37	0.97	1.11	1.23	−3.30
	$\frac{1}{64}$	0.16	0.25	0.25	−0.55	0.54	0.13	0.48	0.07

2. *Test Case* 2: The second test case consists of the solution of the Laplace equation, i.e., $f_{i,j,k} = 0$, with the following Dirichlet boundary conditions:

$$\begin{aligned} u_{i,j,k} &= \sin(\pi j h_y)\sin(\pi k h_z), \quad i = 0, \\ u_{i,j,k} &= 2\sin(\pi j h_y)\sin(\pi k h_z), \quad i = n, \\ u_{i,j,k} &= 0, \quad j,k = \{0,n\}. \end{aligned} \tag{64}$$

The analytical form of the field is found to be

$$u_{i,j,k} = \frac{\sin(\pi j h_y)\sin(\pi k h_z)}{\sinh(\sqrt{2}\pi)}[2\sinh(\sqrt{2}\pi i h_x) + \sinh(\sqrt{2}\pi(1 - i h_x))]. \tag{65}$$

3. *Test Case* 3: This test problem consists of the Gaussian potential field which is not an eigenfunction of the Laplace operator

$$u_{i,j,k} = \exp\left\{-\frac{r_{i,j,k}^2}{\sigma^2}\right\} \tag{66}$$

with

$$r_{i,j,k}^2 = (i h_x - 1/2)^2 + (j h_y - 1/2)^2 + (k h_z - 1/2)^2. \tag{67}$$

For the calculations $\sigma = 0.2$ was chosen. According to the potential field, the source function distribution is given as

$$f_{i,j,k} = \frac{2}{\sigma^2}\left(3 - 2\frac{r_{i,j,k}^2}{\sigma^2}\right)\exp\left\{-\frac{r_{i,j,k}^2}{\sigma^2}\right\}. \tag{68}$$

Boundary conditions are easily calculated from the analytical potential prescription.

4. *Test Case* 4: This test case consists of solving the Poisson equation for a unit point charge, located in the center of the cube. The source term is thereby given as

$$f_{i,j,k} = \begin{cases} \dfrac{4\pi}{h_x h_y h_z} & (x = i h_x,\, y = j h_y,\, z = k h_z) = 0.5, \\ 0 & \text{else} \end{cases} \tag{69}$$

and the analytical solution is

$$u_{i,j,k} = \frac{1}{\sqrt{(i h_x - 0.5)^2 + (j h_y - 0.5)^2 + (k h_z - 0.5)^2}}. \tag{70}$$

The special feature of this test case is the singularity at $(i h_x, j h_y, k h_z) = \mathbf{0.5}$. This singularity cannot accurately be described on a discrete grid. In contrast, considering, e.g., the norm of the total error, $\|\mathbf{u}^{(\mathrm{ex})} - \mathbf{u}_h^{(\infty)}\|^*$ (where $\|\ldots\|^*$ means that the singularity is omitted from the norm), the total discretization error may formally increase. This is due to the fact that a finer grid better resolves the singular point. Since the discretized solution at the singular point is finite, the discrete solution around the singularity will underestimate the true solution. Consequently, this test case will show relatively large errors for each approximation. In order to estimate the accuracy of the solvers, the computational domain is subdivided

into a near field and a far-field region. As a convention the near-field part is defined via the spatial resolution of the coarsest grid with $h = \frac{1}{8}$, i.e., $u_{ijk}^{(\mathrm{near})} = u_{ijk}(r|\alpha \in [3/8, 5/8], \alpha = x, y, z)$ and $u_{ijk}^{(\mathrm{far})} = u_{ijk}(r|\alpha \notin [3/8, 5/8], \alpha = x, y, z)$. It is to be expected that due to the unresolved divergence the far-field part will show a very much better error reduction than the near-field part. However, the effect that the nearest far-field point of a fine grid is closer to the singularity is still visible. Nevertheless, it will be interesting to see how good the solutions will be in the neighborhood of the singularity. Results for the near- and far-field contributions are given in Table 4 and selected potential values in x-direction $(x, y = \frac{1}{2}, z = \frac{1}{2})$ are shown in Fig. 1.

3.2. Errors

In order to validate the accuracy of the finite difference schemes an error analysis is given. Let $\mathbf{u}^{(\mathrm{ex})}$ be the exact solution of the continuous Poisson equation $\Delta\mathbf{u} = -\mathbf{f}$. Discretizing the differential operator $(\Delta \to (1/h^2)\mathbf{A}_h)$ and writing the equation for a grid with mesh size h gives

$$\frac{1}{h^2}\mathbf{A}_h\mathbf{u}_h = -\mathbf{f}_h. \tag{71}$$

Solving this equation using an iteration scheme will result in the solution $\mathbf{u}_h^{(n)}$ after n iterations with the residuum

$$\mathbf{r}_h^{(n)} = \frac{1}{h^2}\mathbf{A}_h\mathbf{u}_h^{(n)} + \mathbf{f}_h. \tag{72}$$

The ideally converged finite difference solution will be formally obtained in the limit $n \to \infty$ (assuming that $\lim_{n\to\infty} \mathbf{r}_h^{(n)} = 0$), so that the residuum may also be written as

$$\mathbf{r}_h^{(n)} = \frac{1}{h^2}\mathbf{A}_h\left(\mathbf{u}_h^{(n)} - \mathbf{u}_h^{(\infty)}\right). \tag{73}$$

With respect to the sampled exact continuous solution there will remain the discretization error $\boldsymbol{\varepsilon}_h = \mathbf{u}_h^{(\infty)} - \mathbf{u}^{(\mathrm{ex})}$. Rearranging and inserting into Eq. (73) gives

$$\mathbf{r}_h^{(n)} = \frac{1}{h^2}\mathbf{A}_h\left(\mathbf{u}_h^{(n)} - \mathbf{u}_h^{(\mathrm{ex})} - \boldsymbol{\varepsilon}_h\right). \tag{74}$$

Therefore the error norm of the difference between the discrete solution after n iterations and the exact solution can be written as

$$\|\mathbf{u}_h^{(n)} - \mathbf{u}_h^{(\mathrm{ex})}\| = \|h^2\mathbf{A}_h^{-1}\mathbf{r}_h^{(n)} + \boldsymbol{\varepsilon}_h\|, \tag{75}$$

$$\leqslant \|h^2\mathbf{A}_h^{-1}\mathbf{r}_h^{(n)}\| + \|\boldsymbol{\varepsilon}_h\|, \tag{76}$$

$$\approx \frac{h^2}{|\lambda_{\min}|}\|\mathbf{r}_h^{(n)}\| + \|\boldsymbol{\varepsilon}_h\|, \tag{77}$$

showing that the error between iterated and exact solution is a linear function of the residuum norm, where the slope is approximately determined by the inverse of the smallest eigenvalue of the stencil matrix of the finite difference operator, $\lambda_{\min}(\mathbf{A})$. The eigenvalues therefore determine the convergence speed towards the discretization error of the scheme. On the other hand, for a given scheme the discretization error determines

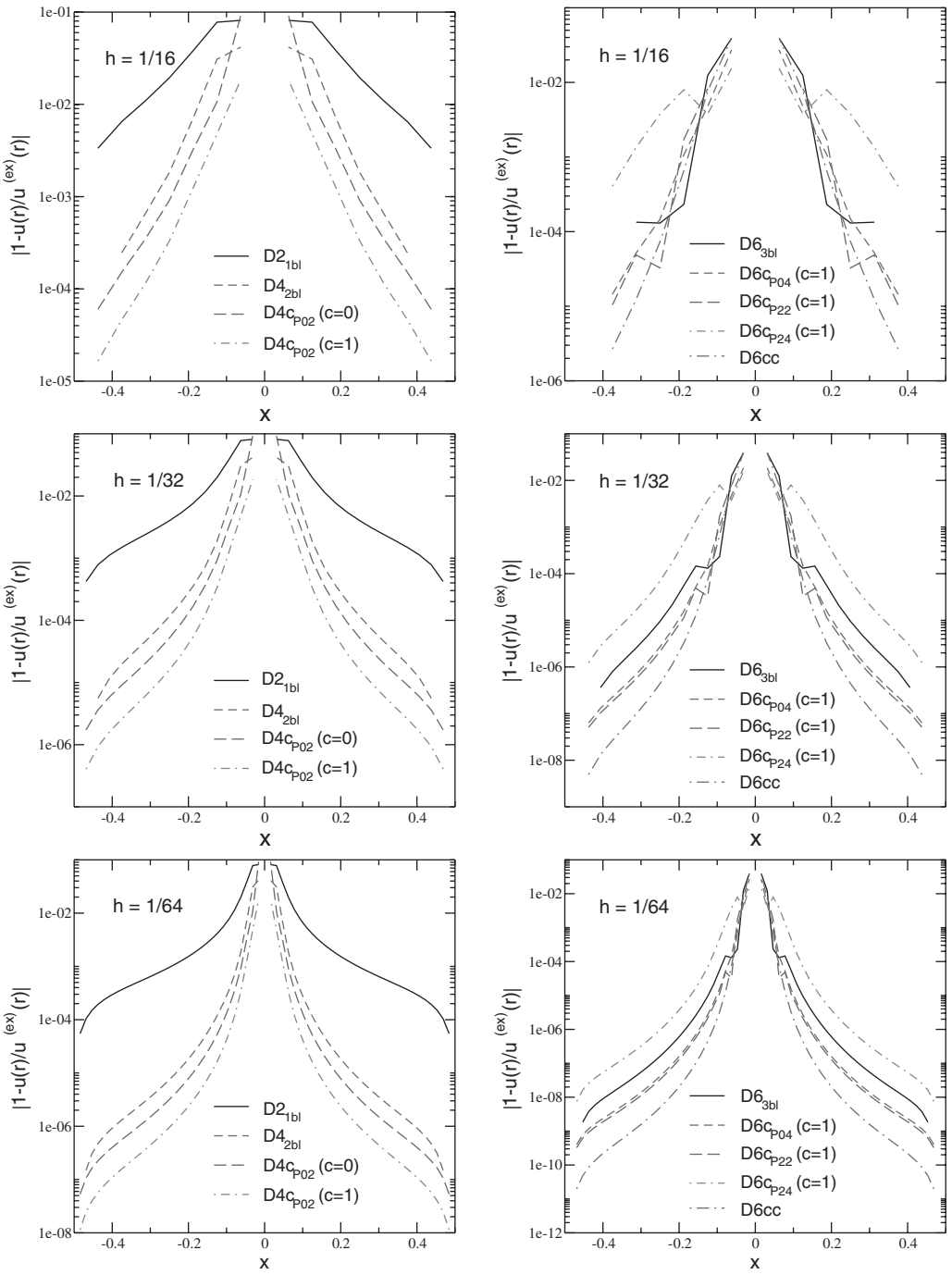

Fig. 1. Relative error $|1-u(r)/u^{(\mathrm{ex})}(r)|$ for a unit point charge in the center of the unit cube (Test Case 4) as function of distance from the charge. Shown is the behavior along the x-direction with fixed y- and z-coordinates $(x, y=\frac{1}{2}, z=\frac{1}{2})$. Left: second- and fourth-order solvers $\mathrm{D2}_{1bl}$,$\mathrm{D4}_{2bl}$, $\mathrm{D4cP}_{0,2}(c_1=0)$ and $\mathrm{D4cP}_{0,2}(c_1=1)$; Right: sixth-order solvers $\mathrm{D6}_{3bl}$, D6cc, $\mathrm{D6cP}_{0,4}$, $\mathrm{D6cP}_{2,2}$, $\mathrm{D6cP}_{2,4}$ with $c=1$ (cmp. Eqs. (33), (44) and (54)).

the threshold level of the residuum norm, i.e., for a smaller residuum, no further improvement of the solution is achieved. Therefore $\|\boldsymbol{\varepsilon}_h\|$ in combination with $\lambda_{\min}(\mathbf{A})$ determines the number of iterations to reach the discretization error of the scheme. Fig. 2 shows the development of the error norm $\|\mathbf{u}_h^{(n)} - \mathbf{u}_h^{(\mathrm{ex})}\|$ as a function of the residuum. A linear decrease of the error is found until the discretization error is reached from where the accuracy saturates. Also in Fig. 2 explicit calculations are compared with the parametrization from Eq. (77) which shows a fairly good agreement (Fig. 3).

In Table 5 the eigenvalues of the finite difference matrices $\mathbf{A}$ are shown. It is clear that a smaller eigenvalue corresponds to a faster decay of the total error norm and therefore show a faster convergence towards the discretization error. It is found that with increasing order the approximations which are directly based on Eq. (7) show larger eigenvalues $\lambda_{\min}$ than the compact schemes (except $\mathrm{D6c}_{P24}$). For fine grids the compact solvers exhibit eigenvalues nearly 50% smaller than the extended ones, therefore showing a faster convergence towards $\|\boldsymbol{\varepsilon}_h\|$. Furthermore, it is found that the discretization error is smaller for compact schemes, demonstrating a more accurate treatment of the problems. This, however, influences the computational work which has to be performed for a more accurate solution. Although compact schemes have better convergence due to a smaller eigenvalue $\lambda_{\min}(\mathbf{A})$, it is not clear whether the overall performance of the schemes will be superior. This issue will be discussed in Section 3.5.

A natural question concerning accuracy is related to the smallest error which can be reached by the schemes and this is related to the machine precision. The minimum absolute error norm for different grids can be estimated via

$$\|\delta\mathbf{u}^{(\infty)}\|_{(\mathrm{mach})} \approx \varepsilon_{\mathrm{mach}}\kappa\|\mathbf{u}^{(\mathrm{ex})}\|, \tag{78}$$

where $\delta\mathbf{u}^{(n)} = \mathbf{u}^{(\mathrm{ex})} - \mathbf{u}_h^{(n)}$ was introduced. Furthermore, $\varepsilon_{\mathrm{mach}} = 2.22 \times 10^{-16}$ is the machine epsilon for 64-bit precision and κ is the condition number of the stencil matrix $\mathbf{A}$, which gives a measure of how strong the inverse of the matrix changes, if the matrix itself is changed.

The condition number of a matrix is calculated via

$$\kappa = \frac{|\lambda_{\max}|}{|\lambda_{\min}|}, \tag{79}$$

where $\lambda_{\max}$ is the largest eigenvalue of the matrix. Since for the smallest mesh size the matrices are of size $32^3 \times 32^3$ to $128^3 \times 128^3$, methods working on the whole matrix are doomed to fail due to memory limits of the computers. Fortunately, the matrices are sparse (N^*/h^3 entries, where N^* is the number of entries of the stencils) and consequently index field addressing methods are used. Since $\mathbf{A}$ is symmetric, the *Rayleigh quotient* method is used here [6], which gives a rather reliable estimate for the largest eigenvalue. The smallest eigenvalue of $\mathbf{A}$ is calculated with the help of a shifted matrix $\mathbf{B}$ [3]

$$\mathbf{B} = \mathbf{A} - |\lambda_{\max}(\mathbf{A})|\mathbf{I}. \tag{80}$$

For this matrix the largest eigenvalue is constructed again by the *Rayleigh quotient* method. To obtain an estimate for $\lambda_{\min}(\mathbf{A})$ one has to shift back [2]

$$\lambda_{\min}(\mathbf{A}) = |\lambda_{\max}(\mathbf{A})| - |\lambda_{\max}(\mathbf{B})|. \tag{81}$$

From these calculations it is first of all found that (except for $\mathscr{P}_{2,4}$) the Padé approximation schemes have smaller condition numbers than the ones based on Eq. (7). This is mainly due to the fact that the largest eigenvalues are reduced. This fact is also observed for the corresponding iteration matrices which have smaller spectral radii (cf. Table 6).

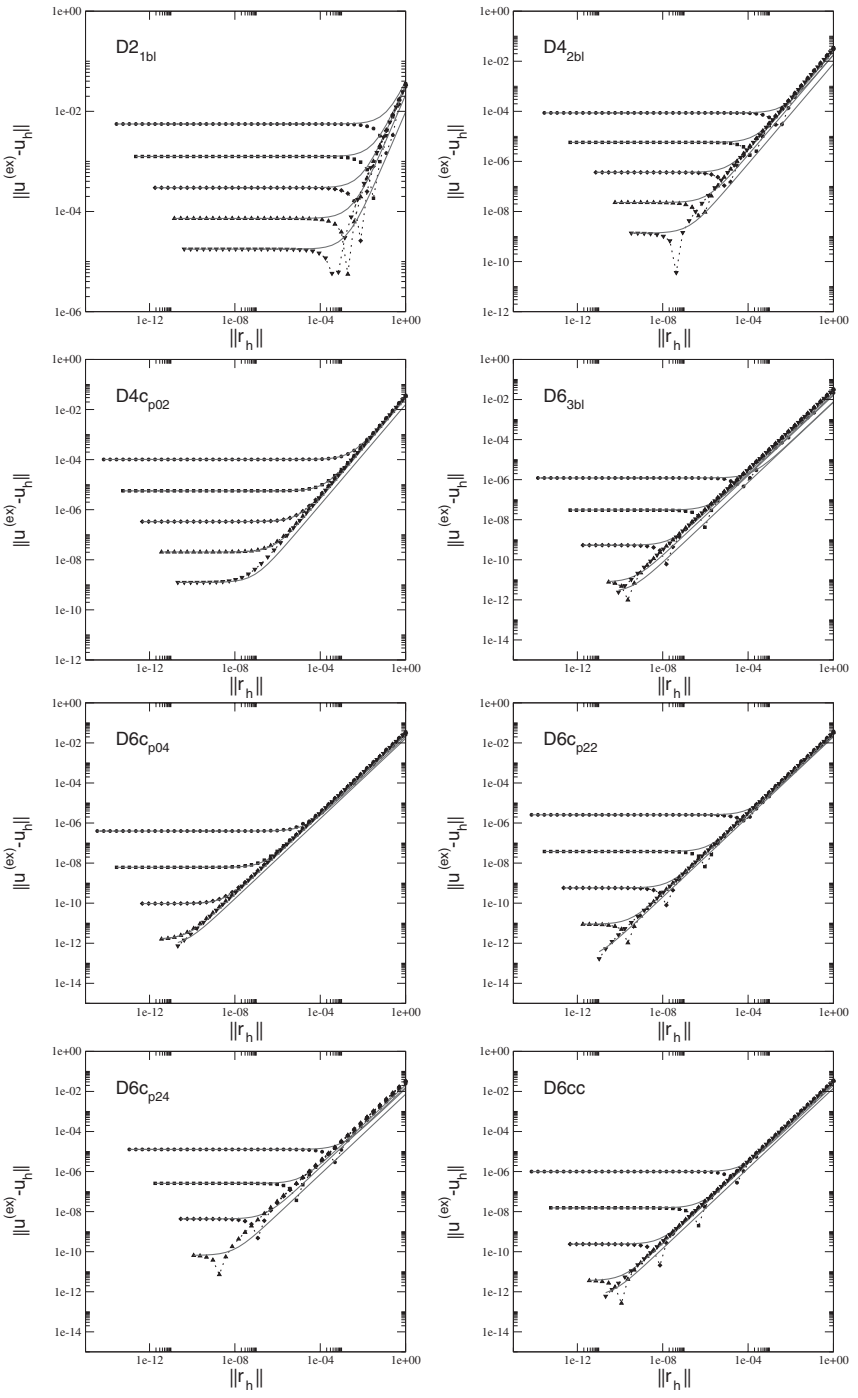

Fig. 2. Error reduction and discretization error $\varepsilon_h = \|\varepsilon_h\|$ for second-, fourth- and sixth-order solvers for Test Case 1. Compared are numerical results, obtained from explicit calculations (line-symbols) and from theoretical estimates, Eq. (77) (lines).

Table 5
Smallest eigenvalues of the stencil matrix **A** and convergence factors for second-, fourth- and sixth-order solvers

	Solver grid	D2	$D4_{2bl}$	$D4cP_{0,2}$	$D6_{3bl}$	$D6cP_{0,4}$	$D6cP_{2,2}$	$D6cP_{2,4}$	D6cc
$\lambda_{\min}(\mathbf{A})$	$\frac{1}{8}$	0.459	0.865	0.446	2.026	0.767	0.713	1.137	0.769
	$\frac{1}{16}$	0.1178	0.158	0.115	0.215	0.150	0.146	0.186	0.150
	$\frac{1}{32}$	3.2×10^{-2}	3.76×10^{-2}	3.01×10^{-2}	4.32×10^{-2}	3.39×10^{-2}	3.32×10^{-2}	6.104×10^{-2}	3.42×10^{-2}
	$\frac{1}{64}$	1.15×10^{-2}	1.36×10^{-2}	8.83×10^{-3}	1.50×10^{-2}	9.08×10^{-3}	8.52×10^{-3}	3.37×10^{-2}	9.578×10^{-3}
	$\frac{1}{128}$	6.29×10^{-3}	7.66×10^{-3}	3.86×10^{-3}	8.40×10^{-3}	3.69×10^{-3}	3.02×10^{-3}	2.05×10^{-2}	4.233×10^{-3}
$\frac{\|\mathbf{r}^{(n)}\|}{\|\mathbf{r}^{(n-1)}\|}$	$\frac{1}{8}$	0.8535 0.8536	0.7826 0.7827	0.7921 0.7922	0.5638 0.5639	0.6494 0.6506	0.5778 0.5655	0.8912 0.8889	0.6775 0.6775
$\rho(\mathbf{C})$	$\frac{1}{16}$	0.9619 0.9619	0.9592 0.9592	0.9437 0.9436	0.9486 0.9486	0.9246 0.9265	0.9021 0.8951	0.984 0.9802	0.9316 0.9304
	$\frac{1}{32}$	0.9904 —	0.9911 —	0.9856 —	0.9906 —	0.9831 —	0.9776 —	0.997 —	0.9847 —
	$\frac{1}{64}$	0.9976 —	0.9979 —	0.9964 —	0.9980 —	0.9960 —	0.9947 —	0.999 —	0.9964 —
	$\frac{1}{128}$	0.9994 —	0.9995 —	0.9991 —	0.99953 —	0.99904	0.9986 —	0.99982 —	0.99913 —
	n_D	7	13	19	19	57	57	27	57

Convergence factors are calculated from the ratio of residuum norms (left) and spectral radii of the iteration matrix **C**, Eq. (88) (right). Due to memory limitations $\rho(\mathbf{C})$ was only calculated explicitly for smaller grids. Also shown is the number of entries, n_D, of the stencils.

Table 6
CPU time τ_h to reach the discretization error and CPU time τ_{10} and number of iterations n_{10} to reduce the residuum by a factor of 10

	Solver grid	D2	$D4_{2bl}$	$D4cP_{0,2}$	$D6_{3bl}$	$D6cP_{0,4}$	$D6cP_{2,2}$	D6cc	$D6cP_{2,4}$
τ_h	$\frac{1}{8}$	2.2×10^{-3}	3.5×10^{-3}	4.2×10^{-3}	4.0×10^{-3}	5.6×10^{-3}	4.5×10^{-3}	4.7×10^{-3}	9.4×10^{-3}
	$\frac{1}{16}$	5.3×10^{-2}	7.4×10^{-2}	8.7×10^{-2}	7.7×10^{-2}	0.15	0.11	0.09	0.80
	$\frac{1}{32}$	6.3	10.1	8.2	11.9	15.5	11.8	9.3	81.8
	$\frac{1}{64}$	720.6	1493.3	928.3	2118.6	1649.7	1143.7	1277.0	7829.3
	$\frac{1}{128}$	25405	66473	34661	93295	60529	54408	62967	—[a]
τ_{10}	$\frac{1}{8}$	6.0×10^{-4}	4.7×10^{-4}	5.8×10^{-4}	3.6×10^{-4}	4.6×10^{-4}	4.5×10^{-4}	4.1×10^{-4}	9.5×10^{-4}
	$\frac{1}{16}$	8.6×10^{-3}	6.6×10^{-3}	7.7×10^{-3}	4.7×10^{-3}	8.5×10^{-3}	7.1×10^{-3}	5.4×10^{-3}	5.3×10^{-2}
	$\frac{1}{32}$	0.81	0.71	0.57	0.57	0.70	0.58	0.44	4.28
	$\frac{1}{64}$	78.9	88.6	53.8	84.9	62.8	46.8	50.2	343.3
	$\frac{1}{128}$	2522	3471	1750	3619	2234	1872	2307	—[a]
n_{10}	$\frac{1}{8}$	6.3	4.1	4.3	1.7	2.3	1.8	2.6	8.7
	$\frac{1}{16}$	25	24	17	19	12	9	13	61
	$\frac{1}{32}$	103	111	68	105	58	44	64	302
	$\frac{1}{64}$	416	475	277	499	249	188	277	1249
	$\frac{1}{128}$	1666	1999	1110	2127	1041	713	1148	—[a]

Shown are results for Test Case 1. For other test problems results do not differ qualitatively and are rather close to these values.
[a]Convergence of $D6c_{P24}$ for $h = \frac{1}{128}$ was so slow that it was omitted.

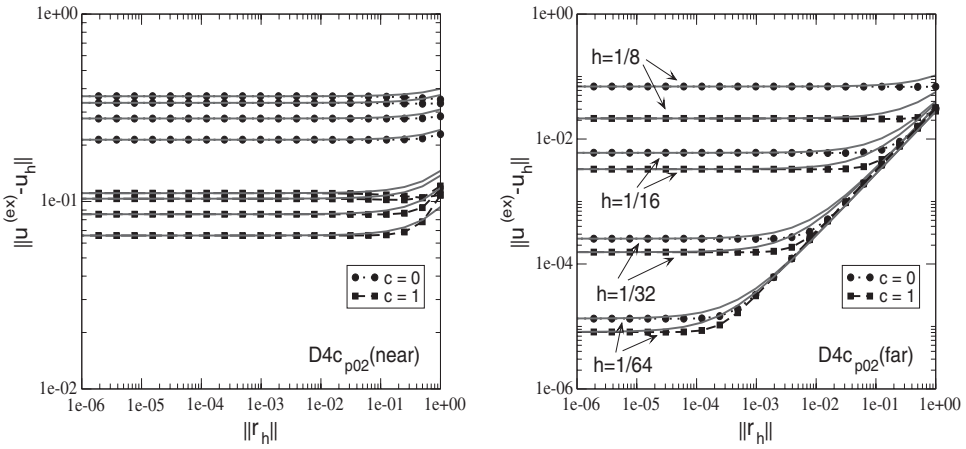

Fig. 3. Error reduction and discretization error $\varepsilon_h = \|\boldsymbol{\varepsilon}_h\|$ for D4c$_{P02}$ with $c_1 = 0$ and 1 (cmp. Eq. (25)) for Test Case 4 (point charge); left: near-field part; right: far-field part. Compared are numerical results, obtained from explicit calculations (line-symbols) and from theoretical estimates, Eq. (77) (lines).

Fig. 4a shows results for the smallest errors obtained for a residuum threshold $\varepsilon_{\text{res}} = \|\mathbf{r}_h\|_{(\min)} = 10^{-15}$. Shown is the absolute error norm, $\|\mathbf{u}^{(\text{ex})} - \mathbf{u}_h^{(\infty)}\|$. Results are compared to the machine error, Eq. (78). It is found from these calculations that on the finest grid the machine resolution is already reached for the sixth-order solvers and therefore the absolute error norm stops to decrease. In contrast, it will even increase for smaller mesh sizes since the minimal norm $\|\delta\mathbf{u}^{(\infty)}\|_{(\text{mach})}$ becomes larger due to an increasing condition number κ of the stencil matrix.

Another question related to accuracy of results and efficiency of the solver is the choice of the residuum threshold. Results so far were obtained for a fixed residuum threshold of $\varepsilon_{\text{res}} = 1 \times 10^{-15}$, which either was small enough to reach $\|\boldsymbol{\varepsilon}_h\|$ or even led to results approaching machine precision (e.g., in the case of sixth-order solvers on the finest grid). Choosing ε_{res} too small will result in the fact that the discretization error is reached far before stopping iterations and therefore a lot of iterations are done in vain. On the other hand, choosing a too large value for ε_{res} results in the fact that the discretization error is not yet reached and therefore the numerical order of the solver is not obtained. To get an estimate of the effect of choosing an inappropriate threshold value for the residuum norm, one can introduce a *residual error bound*

$$\|\delta\mathbf{u}^{(n)}\|_{(\text{res})} = \frac{h^2\varepsilon_{\text{res}}}{|\lambda_{\min}|}. \tag{82}$$

Fig. 4b shows results of calculations for Test Case 1 where $\varepsilon_{\text{res}} = 10^{-9}$. It is found that the second- and fourth-order solvers show a linear decrease of the error norm $\|\mathbf{u}^{(\text{ex})} - \mathbf{u}_h^{(\infty)}\|$ with increasing grid size. Therefore, the results are not affected by the choice of the residuum which is clear from Fig. 4b showing that the discretization error is already reached for $\varepsilon_{\text{res}} = 10^{-9}$ on all investigated grid sizes. The situation becomes different for the sixth-order solvers which start to saturate on the $h \leqslant \frac{1}{64}$, showing that discretization error is not yet reached. Compared to these findings is the error resulting from the theoretical estimate Eq. (82), where—as a limit—the eigenvalue $\lambda_{\min}(\mathbf{A})$ corresponding to D6c$_{P22}$ was

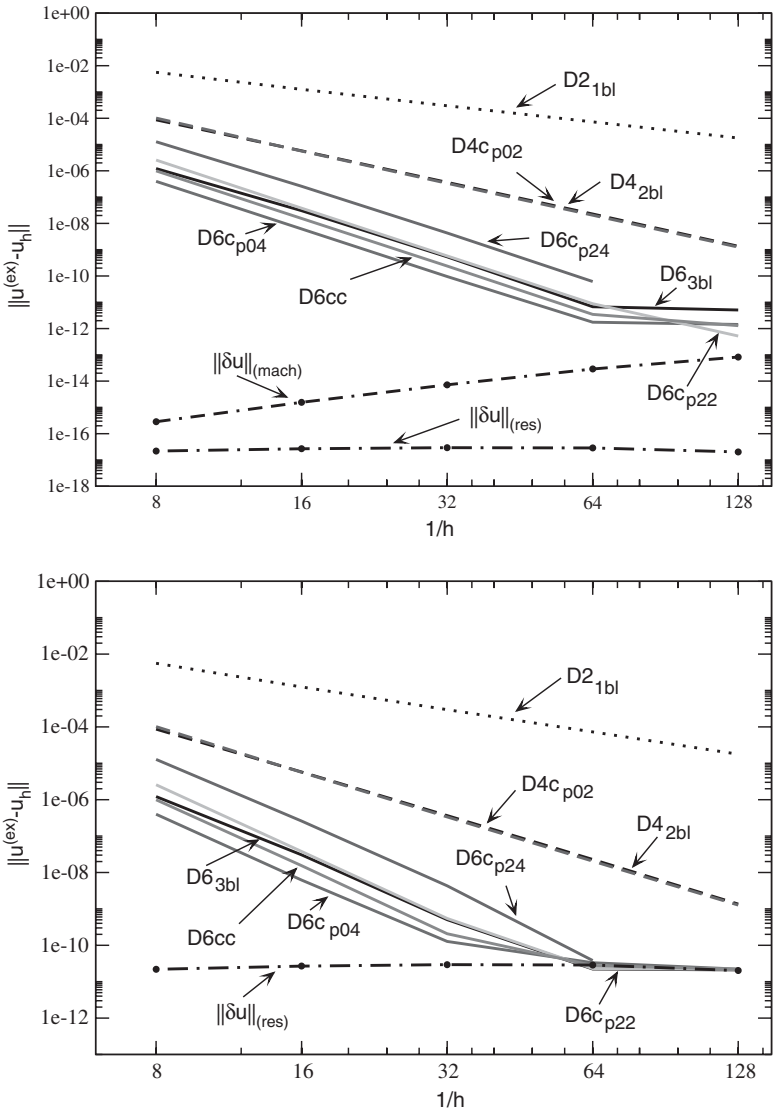

Fig. 4. Comparison of error norm $\|\mathbf{u}_h - \mathbf{u}^{(\mathrm{ex})}\|$ for Test Case 1 with *machine error bound* $\|\delta\mathbf{u}\|_{(\mathrm{mach})}$ (Eq. (78)) and *residuum error bound* $\|\delta\mathbf{u}\|_{(\mathrm{res})}$ (Eq. (82)) as function of grid size for second-, fourth- and sixth-order solvers. Compared are results for two different criteria to stop iterations: (a) residuum threshold $\|\mathbf{r}\|_{\min} = 10^{-15}$ (upper figure) and (b) residuum threshold $\|\mathbf{r}\|_{\min} = 10^{-9}$ (lower figure).

chosen (it has smallest $\lambda_{\min}(\mathbf{A})$ on the fine grids). It can be clearly seen that the error norm follows this numerical limit and therefore gives a reliable estimate for the minimum error which can be achieved on a given grid and a given threshold value $\varepsilon_{\mathrm{res}}$.

For the new Padé solvers, derived in Section 2, a higher-order approximation of the RHS of the Poisson equation was kept formally by introducing a factor c, which takes values either of zero (both sides have

same order in h) or one. Allowing $c = 1$ it is found that in some cases the solution is improved. However, there exist also cases where the solution becomes worse, i.e. there is no clear rule for which solvers and test cases there is an improvement. For example in Test Case 1, the solver D4c$_{P02}$ is $\approx 13\%$ less accurate with $c = 1$, while the solution of the Coulomb problem is improved by a factor of 3 in the near field and by a factor of 1.5 in the far field (cmp. Fig. 3). On the other hand for D6c$_{P22}$ the solution of Test Case 1 is improved by a factor of 3 when $c = 1$, while the solution of the Coulomb problem becomes less accurate (near field a factor 0.7, far field a factor of 0.9). It is therefore not decisive from the beginning to keep the factor c or not. However, it is found that the same factor in improvement (or worsening) is observed for all grid sizes for a given solver and test problem. Therefore, one may compare results on the coarsest grid and may decide whether to keep $c = 0$ or 1 on fine grids.

3.3. Order of finite difference approximations

In Section 2 a formal derivation of high-order schemes, based on a Padé approximation was given. Although the Padé approximations should have the desired order in h, a formal proof for this fact should be given.

As a matter of fact the eigenfunctions of the exact form of the Laplace operator are known to be $\phi_{k,l,m} = \cos(kx)\cos(ly)\cos(mz)$ with corresponding eigenvalues $\lambda = -(k^2 + l^2 + m^2)$. For the usual stencils, the eigenvectors of the discrete problem are simply the sampled continuous eigenvectors with eigenvalues changed according to the order of accuracy. To test this feature for stencils with a modified RHS, a slightly more involved procedure is needed. In the derivation of the Padé approximations two operators appear in the final results. The one acting on the field u, the other acting on the source term f, i.e. $\tilde{\Delta}\mathbf{u} = \tilde{\Gamma}\mathbf{f}$, where $\tilde{\Delta}$ and $\tilde{\Gamma}$ are approximation operators. Therefore, a generalized eigenvalue problem has to be solved

$$\tilde{\Delta}\mathbf{u} = \lambda\tilde{\Gamma}\mathbf{u}. \tag{83}$$

Both $\tilde{\Delta}$ and $\tilde{\Gamma}$ have the desired property that eigenfunctions are sampled continuous eigenvectors. Consequently only the eigenvalues need to be tested. For practical purposes one may insert the eigenfunctions of the *exact* operator and controls the order in h of the approximation, i.e.,

$$(\tilde{\Delta}\phi_{k,l,m} + (k^2 + l^2 + m^2)\tilde{\Gamma}\phi_{k,l,m})\phi_{k,l,m}^{-1} = \mathcal{O}(h^n), \tag{84}$$

where n is the order of the approximation. Due to the linearity of the operators, this should give the order of the operator itself. For all approximations developed here, the expected order was recovered. Note, however, that this procedure is first of all true for the Poisson equation, i.e., a nonvanishing RHS. The derivation of the solvers introduces in a natural way a modified RHS of the Poisson equation. One therefore could argue that the order for a Laplace equation will be reduced. In fact, formally $\tilde{\Delta}$ is only a second-order approximation of the exact Laplace operator for arbitrary functions ψ. However, for those functions $\tilde{\psi}$, solving the Laplace equation $\Delta\tilde{\psi} = 0$ it has full order.

The order of the schemes may also be checked empirically along with a given problem for which the discretization error should follow

$$\frac{\|\boldsymbol{\varepsilon}_{h_{i-1}}\|}{h_{i-1}^n} \geqslant \frac{\|\boldsymbol{\varepsilon}_{h_i}\|}{h_i^n}, \tag{85}$$

where h_i denotes the mesh size for a grid with $(2^i + 1)^3$ grid points and n is the expected order of the numerical scheme. Since $h_i = h_{i-1}/2$, the order condition for a numerical scheme may be reformulated with an excess parameter x_h which can be defined via

$$x_h = \log_2 \left(\frac{\|\varepsilon_{h_{i-1}}\|}{\|\varepsilon_{h_i}\|} \right) - n. \tag{86}$$

For $x_h \geqslant 0$ the numerical scheme exhibits the prescribed order $\mathcal{O}(h^n)$. In Tables 1–4 the values for x_h are shown for all solvers, applied to Test Cases 1–4. For the first three test cases it is found that x_h is close to zero, i.e. the calculations fulfill the prescribed order of the solvers. Test Case 4 is different in this respect. The near field part of the Coulomb problem definitely shows a very poor improvement with increasing the mesh size, i.e. in this region the prescribed order is not reached. This is understandable from the fact that the source function as well as the exact potential are not smooth functions. In fact, a better resolution of the singularity in the potential may result in larger relative errors near the singular point and therefore there is no improvement of the error norm. On the other hand, the far field part does not exhibit these problems and the potential there may be considered as a smooth part of the potential. Consequently, the excess parameter is very much closer to zero or even positive, indicating a better order behavior than prescribed by the solver scheme. In contrast to other properties there is no clear evidence that compact schemes exhibit larger excess parameters than extended ones.

3.4. Convergence

It is clear that the stencil and iteration matrices of the finite difference schemes have to be positive definite. This is equivalent to the statement $\mathbf{x}^{\mathrm{T}}\mathbf{A}\mathbf{x} > 0$, $\forall \mathbf{x}$ or that all eigenvalues λ of $\mathbf{A}$ are greater zero. The second requirement was checked by calculating the smallest eigenvalue of the matrices $\mathbf{A}$ which were found to be positive for all investigated cases.

Another important characteristic is the convergence behavior of the finite difference schemes. Empirically, the rate of convergence, r_c, is calculated as the asymptotic decay of the residuum

$$r_c = \lim_{n\to\infty} \frac{\|\mathbf{r}^{(n)}\|}{\|\mathbf{r}^{(n-1)}\|}. \tag{87}$$

On the other hand it may also be calculated as the spectral radius of the iteration matrix $\mathbf{C}$ of the numerical scheme. In the case of the Gauss–Seidel scheme, $\mathbf{C}$ is given by

$$\mathbf{C} = (\mathbf{D} - \mathbf{L})^{-1}\mathbf{U} \tag{88}$$

where $\mathbf{D}$ is the diagonal, $-\mathbf{L}$ the lower triangular and $-\mathbf{U}$ the upper triangular part of the operator matrix $\mathbf{A}$. The spectral radius is then given as $\rho(\mathbf{C}) = \max_i \lambda_i(\mathbf{C})$, where λ_i is an eigenvalue of matrix $\mathbf{C}$. Both methods were applied to calculate the convergence rate. Table 5 shows the comparison of the different schemes. First of all it is found that the empirical determination of r_c is in a very good agreement with the calculation of the spectral radius. Furthermore it is found that for the different schemes a different performance has to be expected. Especially for large lattices, the schemes based on the Padé approximation perform fairly superior with respect to approximations based on the Taylor expansion, Eq. (7). For the smallest lattices this seems to be different. However, it has to be noted that in the case of $D6_{3bl}$ three boundary layers are specified, which reduces the number of unknowns in that case from $(2^3 + 1)^3 = 729$

to $(2^3-5)^3=27$! In the case, where two boundary layers have to be specified ($D4_{2bl}$, $D6c_{P22}$, $D6c_{P04}$ and $D6c_{P24}$) the number of unknowns on the smallest lattice is $(2^3-3)^3=125$, whereas for one specified boundary layer ($D2_{1bl}$,$D4c_{P02}$, D6cc) there are $(2^3-1)^3=343$ unknowns. This imbalance is strongly reduced if one moves to larger lattices. To find the proper spectral radii the iteration matrix was constructed consequently with eliminated boundary conditions.

3.5. Computational efficiency

The decision to use a certain finite difference scheme does not only depend on the given order of the method but also on its efficiency, i.e., the computational effort which has to be spent to reach a given level of accuracy. This work strongly depends on the order of the discretization error, the spectral radius $\rho(\mathbf{C})$ and the work per step. The latter one may be considered in a first approximation as proportional to the number of entries, n_D, in the finite difference stencil (Table 5). Therefore the efficiency will be different for solvers of the same order.

To estimate the work per step, the operations on the RHS of the Poisson equation may be disregarded, as they are performed only once at the beginning of the iteration procedure. Therefore the computational work, W_{cpu}, could be estimated as $W_{\mathrm{cpu}}=\alpha N^* n_h n_D$, where α is a proportionality constant, n_h is the number of iterations until the residuum threshold is reached and $N^*=(1/h+1-2n_b)^3$ is the number of unknowns. n_b is the number of boarder layers, where the solution is prescribed exactly in order to fulfill the boundary conditions, i.e. these grid points are not iterated. This simple approximation, however, does not take into account machine specific characteristics, e.g. cache effects, which may result in a different performance for compact data structures. Therefore, also α will be a function of the solver at hand.

In addition, the computational work also depends on the accuracy of a given scheme. A higher accuracy implies a smaller discretization error which usually results in an increasing number of iteration steps until $\|\boldsymbol{\varepsilon}_h\|$ is reached. The number of iteration steps n_h can be estimated via

$$n_h=\log\left(\frac{1}{\rho(\mathbf{C})}\frac{\|\mathbf{r}_h\|_{(\min)}}{\|\mathbf{r}_h^{(0)}\|}\right), \tag{89}$$

where $\|\mathbf{r}_h\|_{(\min)}$ is the threshold of the residuum norm for which the discretization error is reached. According to Eq. (77) this can be estimated via

$$\|\mathbf{r}_h\|_{(\min)}\approx\frac{|\lambda_{\min}(\mathbf{A})|}{h^2}\|\boldsymbol{\varepsilon}_h\|. \tag{90}$$

Calculating n_h it is found that the number of iterations increases roughly by a factor five to eight if the number of grid points is increased by a factor of eight, showing a nearly sub-linear increase of the number of iterations as function of grid points. It is found that the compact schemes exhibit a better *internal efficiency*, i.e., have smaller ratios $n_{h_i}/n_{h_{i-1}}$ than the ones based on Eq. (7). In addition, they also show a superior behavior in absolute numbers of n_h.

However, these better characteristics in terms of convergence behavior are partly compensated by a higher computational work W_{cpu} since the compact schemes have larger values n_D as the extended ones. To get a realistic estimate of the computational efficiency, timings were recorded for each solver to reach the discretization error, i.e. the time for n_h iterations. Results are shown in Table 6 for Test Case 1. It is seen that for solvers of a given order results are comparable for small grid sizes ($h\geqslant\frac{1}{32}$). For large

grids, however, it is found that compact solvers are clearly favorable with respect to extended solvers, e.g., D6cP$_{22}$ is $\approx 50\%$ faster than D6$_{3bl}$ and D6cP$_{0,4}$, D6cc are still $\approx 25\%$ faster than D6$_{3bl}$. Assuming a constant α the ratio, e.g., $W_{\mathrm{cpu}}(\mathrm{D6}_{3bl})/W_{\mathrm{cpu}}(\mathrm{D6cP}_{22})$, is larger than one for coarse grids and approaches one on fine grids. The fact that execution times for compact solvers nevertheless are considerably better on fine grids than for extended schemes, demonstrates that machine specific effects, e.g. cache effects, play an important role which make compact schemes superior with respect to extended ones (cmp. Table 6).

Assuming that a solver has a smaller discretization error with respect to a concurrent scheme, the time to reach the level of discretization error will be probably longer than for a less accurate solver. This would mean that the computational efficiency, as characterized above, would be smaller. Therefore another measure of efficiency is obtained by calculating the number of intervals, n_{10}, and time, τ_{10}, to reduce the residuum by a factor of 10, i.e.

$$n_{10} = n_h \log\left(\frac{\|\mathbf{r}_h^{(0)}\|}{\|\mathbf{r}_h\|_{(\min)}}\right)^{-1}, \quad \tau_{10} = \tau_h \log\left(\frac{\|\mathbf{r}_h^{(0)}\|}{\|\mathbf{r}_h\|_{(\min)}}\right)^{-1}. \tag{91}$$

Using the data from Test Case 1 it is also evident for this criterion that compact schemes become clearly superior with respect to extended ones for grid sizes $N \geqslant 64^3$.

4. Conclusions

In the present article different forms of high-order compact solvers for the three-dimensional Poisson equation have been derived on the basis of a Padé approximation which was applied to the Taylor expansion of the finite difference operator, Eq. (7). It was shown that both the accuracy and the convergence characteristics obtained with these new approximations is superior to the *straight forward* implementations of the Laplace stencil (Eq. (7)). The fourth-order Padé approximation led to a similar expression as found in Ref. [7]. However, here an extended expression was obtained, where the RHS of the Poisson equation was modified according to a higher-order approximation scheme. This turned out to give more accurate results in certain cases, e.g., in the case of discontinuous source functions, i.e., the case of a unit point charge. In the case of continuous source distributions, this higher-order approximation of the RHS did not improve the accuracy. A similar extension was made for the sixth-order approximations. It is not yet clear in which cases results may be improved with the inclusion of higher-order terms. The results do not offer a clear picture which solvers applied to certain test problems benefit from this inclusion. It was found, however, that a potential improvement factor (in some cases also worsening factor) were constant for all grid sizes for a given solver and test problem. Clarification of these findings may be a hint for future research.

The new sixth-order Padé solvers were compared to the sixth-order solver D6cc, Eqs. (14)–(16). It was found that the new schemes D6c$_{P04}$ and D6c$_{P22}$ are either competitive with D6cc or even superior with respect to accuracy and performance.

As a matter of fact $[m, n]$-Padé approximations of the bracketed expression in Eq. (7) approximate the Laplace operator with order $\mathcal{O}(h^{m+n+2})$. For the fourth-order approximation, the only nontrivial choice is a [0,2]-approximation. For the case of sixth-order the two cases [0,4] and [2,2] were investigated. However, as a test, also a [2,4]-Padé approximation was used, where terms only up to sixth-order were taken into

account in the resulting expression. This case demonstrated that with higher-order Padé-approximations no better solutions are found. It was shown that this approximation turns out to lead to a larger condition number of the operator matrix **A** and to larger spectral radii. The former property leads to a less stable solution, the latter one to a slower convergence. It may be expected that this behavior is even more pronounced for higher-order Padé approximations, where terms are kept only up to order six.

In this article the sixth-order approximations require two boundary layers for a solution of a boundary value problem. This may lead to problems where the *true* potential function is only known exactly on one boundary layer. In this case, one has to switch to asymmetric stencils or to lower-order representations on and near to the boundary. It is not clear how this local change of order does affect the overall solution of the problem. Investigations in this direction will be done in the future. In addition the finite difference schemes of sixth-order cannot yet be used in a straightforward way for multigrid techniques. However, double discretization techniques, i.e., different discretization of field and residuum equations, will lead to the possibility to apply also the new high-order approximations to multigrid methods. Work in this direction is in progress.

References

[1] W.F. Ames, Numerical Methods for Partial Differential Equations, Academic Press, New York, 1977.
[2] For the smaller matrices this was checked with routines DSYTRD and DSTERF from LAPACK.
[3] G.H. Golub, C.F. van Loan, Matrix Computations, Johns Hopkins University Press, Baltimore, 1989.
[4] Y. Gu, W. Liao, J. Zhu, An efficient high order algorithm for solving systems of 3-D reaction-diffusion equations, J. Comput. Appl. Math. 155 (2003) 1–17.
[5] W. Liao, J. Zhu, A.Q.M. Khaliq, An efficient high order algorithm for solving systems of reaction-diffusion equations, J. Numer. Methods Partial Differential Equations 18 (2002) 340–354.
[6] Y. Saad, Numerical Methods for Large Eigenvalue Problems, Manchester University Press, Manchester, 1992.
[7] W.F. Spotz, G.F. Carey, A high-order compact formulation for the 3d Poisson equation, Numer. Methods Partial Differential Equations 12 (1996) 235–243.
[8] U. Trottenberg, C.W. Oosterlee, A. Schüller, Multigrid, Academic Press, San Diego, 2001.
[9] J. Zhang, Fast and high accuracy multigrid solution of the three dimensional Poisson equation, J. Comput. Phys. 143 (1998) 449–461.