ACCELERATION OF THE CONVERGENCE IN FINITE DIFFERENCE METHOD BY PREDICTOR-CORRECTOR AND SPLITTING EXTRAPOLATION METHODS*

PEKKA NEITTAANMAKI
(University of Jyväskylä, Dept. of Mathematics Seminaarinkatu 15, SF-40100 Jyväskylä, Finland)

LIN QUN (林 群)
(Institute of Systems Science, Academia Sinica, Beijing, China)

Abstract

Two types of combination methods for accelerating the convergence of the finite difference method are presented. The first is based on an interpolation principle (correction method) and the second one on extrapolation principle. They improve the convergence from $O(h^2)$ to $O(h^4)$. The main advantage, when compared with standard methods, is that the computational work can be splitted into independent parts, which can then be carried out in parallel.

§ 1. Introduction

We study the finite difference approximation of the solution, u, to the model problem

$$\begin{cases} \Delta u=f, & \text{in } \Omega, \\ u=g, & \text{on } \partial\Omega \end{cases} \tag{1.1}$$

in the 2 or 3-dimensional domain Ω with boundary $\partial\Omega$. Suppose that Ω consists of some squares in the 2-dimensional case and of some cubes in the 3-dimensional case. Furthermore, suppose that the solution of (1.1) is smooth enough.

Let u_h be the solution of the approximate finite difference analogy to (1.1) with mesh size h:

$$\begin{cases} \Delta_h u_h=f, & \text{in } \Omega_h^d,\ d=2,3, \\ u_h=g, & \text{on } \partial\Omega_h^d. \end{cases} \tag{1.2}$$

Here Δ_h denotes the 5-point approximation of the Laplace operator Δ in the 2-dimensional case, and the 7-point approximation in the 3-dimensional case as usual.

It is well known that

$$u-u_h=h^2e+O(h^4) \text{ in } \Omega_h^d, \text{ for } h\to 0, \tag{1.3}$$

where e is the solution of a correction differential equation independent of h ([3, 7]). Function e can be estimated as

$$-h^2e=\frac{4}{3}(u_h-u_{h/2})+O(h^4). \tag{1.4}$$

The more accurate solution, $u_{h/2}$, may be corrected by $\frac{1}{3}(u_{h/2}-u_h)$, and the correction taken as an error estimate. The disadvantage of the above extrapolation procedure is the computation of another solution with a smaller parameter $(h/2)$, which involves solving once again a finite difference equation of much larger size than the one corresponding to the original h for multidimensional problems.

In this paper two methods are presented which lead to accuracy $O(h^4)$ but which are of a lower computational complexity than the standard extrapolation method. They are especially efficient when parallel architecture of the computer system is used. The methods consist of a predictor–corrector type method, and a splitting extrapolation method. The methods will be introduced in sections 2 and 3. Finally, in section 4 we give some results of numerical tests. In the two–dimensional case we have compared the presented correction and splitting extrapolation methods with the standard 5–point scheme and multigrid method; in the three–dimensional case we have compared our method with 9–point and 15–point schemes, and with the standard extrapolation method. All computations have been carried out with the conventional computer system. Consequently, the parallelization properties of the correction method and the splitting extrapolation method have not been utilized. In spite of that, the methods presented here seem to be superior compared with standard ones.

For a survey of extrapolation methods in FE and FD–schemes we refer to survey work [6]. Especially, see [1] for FEM and [11] for FD in connection with multigrid method.

§ 2. Predictor–Corrector Type Procedure

Define the (uniform regular) lattice domain

$$\Omega_h^d \equiv \Omega(h, \cdots, h)$$
$$=\{(x_1, \cdots, x_d)\in \mathbb{R}^d \mid x_i = m_i h,\ m_i = 0,\ \pm 1, \cdots, \pm n,\ i=1, \cdots, d,\ nh=1\}.$$

Let operators Δ_h^+ and $\Delta_h^\times$ be approximations for the Laplace operator $L=\Delta$ at point $x\in \operatorname{int}\Omega_h^d$ (where int Ω denotes the interior of Ω). For $d=2$, Δ_h^+ is the 5–point difference operator, where

$$\Delta_h^+ u(x_1, x_2) = L_5^h u(x_1, x_2) = \frac{1}{h^2}\{u(x_1-h, x_2)+u(x_1+h, x_2)$$
$$+u(x_1, x_2-h)+u(x_1, x_2+h)-4u(x_1, x_2)\},$$

and $\Delta_h^\times$ the 5–point difference operator, where

$$\Delta_h^\times u(x_1, x_2) = L_5^h u(x_1, x_2) = \frac{1}{2h^2}\{u(x_1-h, x_2-h)+u(x_1+h_1, x_2+h)$$
$$+u(x_1-h, x_2+h)+u(x_1+h, x_2-h)-4u(x_1, x_2)\}.$$

For $d=3$, Δ_h^+ is the 7–point difference operator, and

$$\Delta_h^+ u(x_1, x_2, x_3) = L_7^h u(x_1, x_2, x_3)$$
$$=\frac{1}{h^2}\{u(x_1-h, x_2, x_3)+u(x_1+h, x_2, x_3)+u(x_1, x_2-h, x_3)$$
$$+u(x_1, x_2+h, x_3)+u(x_1, x_2, x_3-h)$$
$$+u(x_1, x_2, x_3+h)-6u(x_1, x_2, x_3)\},$$

and with $\Delta_h^\times$ as the 9–point difference operator, we have

$$\begin{aligned}\Delta_h u(x_1, x_2, x_3) = L_5^h u(x_1, x_2, x_3) = (&u(x_1+h, x_2+h, x_3+h)+u(x_1-h, x_2+h, x_3+h)\\ &+u(x_1+h, x_2-h, x_3+h)+u(x_1-h, x_2-h, x_3+h)\\ &+u(x_1+h, x_2+h, x_3-h)+u(x_1-h, x_2+h, x_3-h)\\ &+u(x_1+h, x_2-h, x_3-h)+u(x_1-h, x_2-h, x_3-h)\\ &-8u(x_1, x_2, x_3))/4h^2.\end{aligned}$$

Corresponding to these difference operators we set the following approximation problems for (1.1)

Find u_h^+ such that

$$\begin{cases} \Delta_h^+ u_h^+ = f, & \text{in } \Omega_h^d,\ d=2, 3; \\ u_h^+ = g - \dfrac{h^2}{12} f, & \text{on } \partial\Omega_h^d. \end{cases} \tag{2.1}$$

Find $u_h^\times$ such that

$$\begin{cases} \Delta_h^\times u_h^\times = f, & \text{in } \Omega_h^d,\ d=2, 3; \\ u_h^\times = g - \dfrac{h^2}{12} f, & \text{on } \partial\Omega_h^d. \end{cases} \tag{2.2}$$

It is generally known that

$$\|u-u_h^+\|_{L^\infty(\Omega_h^d)} = O(h^2) \quad \text{for } h\to 0_+ \tag{2.3}$$

and that

$$\|u-u_h^\times\|_{L^\infty(\Omega_h^d)} = O(h^2), \quad \text{for } h\to 0_+,$$

where $\|u\|_{L^\infty(\Omega_h^d)} := \sup_{x\in\Omega_h^d} |u(x)|$, (see, for example, [4, 10]). The above accuracy results can be improved by using $u_h^\times$ as a correction to u_h^+. Indeed, we have

Theorem 2.1. *Let u_h^+ and $u_h^\times$ be the solutions of* (2.1) *and* (2.2), *respectively. Take*

$$u_h := \frac{1}{3}(2u_h^+ + u_h^\times) + \frac{h^2}{12} f. \tag{2.4}$$

Then

$$\|u-u_h\|_{L^\infty(\Omega_h^d)} = O(h^4) \quad \text{for } h\to 0_+. \tag{2.5}$$

Proof. It is easy to see that the truncation errors for operators Δ_h^+ and $\Delta_h^\times$ will be

$$\begin{cases} \Delta_h^+ u - \Delta u = h^2 l_1(u) + O(h^4), \\ \Delta_h^\times u - \Delta u = h^2 l_2(u) + O(h^4) \end{cases} \tag{2.6}$$

with

$$l_1(u) = \frac{1}{12} \sum_{i=1}^{d} \frac{\partial^4}{\partial x_i^4} u, \tag{2.7}$$

$$l_2(u) = \frac{1}{12} \sum_{i=1}^{d} \frac{\partial^4}{\partial x_i^4} u + 6 \sum_{\substack{i,j=1\\ i<j}}^{d} \frac{\partial^4}{\partial x_i^2\, \partial x_j^2} u \tag{2.8}$$

and

$$\frac{2}{3} l_1(u) + \frac{1}{3} l_2(u) = \frac{1}{12} \Delta^2 u = \Delta\left(\frac{1}{12} f\right). \tag{2.9}$$

Let w_i be the solution of the auxiliary problems

$$\begin{cases} \Delta w_i = l_i(u), & \text{in } \Omega,\ i=1, 2, \\ w_i = \dfrac{1}{12} f, & \text{on } \partial\Omega. \end{cases} \tag{2.10}$$

Then we have by (2.5)—(2.10)

$$\begin{cases} \Delta_h^+(u-u_h^+-h^2w_1)=\Delta_h^+u-\Delta_h^+u_h^+-h^2\Delta_h^+w_1 \\ \qquad =h^2l_1(u)-h^2\Delta w_1+h^2(\Delta w_1-\Delta_h^+w_1)+O(h^4)=O(h^4), \quad \text{in } \Omega_h^d, \\ u-u_h^+-h^2w_1=0, \quad \text{on } \partial\Omega_h^d \end{cases} \tag{2.11}$$

and

$$\begin{cases} \Delta_h^\times(u-u_h^\times-h^2w_2)=\Delta_h^\times u-\Delta_h^\times u_h^\times-h^2\Delta_h^\times w_2 \\ \qquad =h^2l_2(u)-h^2\Delta w_2+h^2(\Delta w_2-\Delta_h^\times w_2)+O(h^4)=O(h^4) \quad \text{in } \Omega_h^d, \\ u-u_h^\times-h^2w_2=0 \quad \text{on } \partial\Omega_h^d. \end{cases} \tag{2.12}$$

Consequently, by the maximum principle,

$$\begin{cases} u-u_h^+-w_1=O(h^4) \quad \text{in } \Omega_h^d, \\ u-u_h^\times-w_2=O(h^4) \quad \text{in } \Omega_h^d. \end{cases} \tag{2.13}$$

Hence

$$u-u_h-h^2w=O(h^4), \quad \text{for } h\to 0,$$

where, according to (2.9), $w=\frac{1}{3}(2w_1+w_2)$ satisfies

$$\begin{cases} \Delta w=\frac{2}{3}\, l_1(u)+\frac{1}{3}\, l_2(u)=\Delta\left(\frac{1}{12}f\right) \quad \text{in } \Omega, \\ w=\frac{1}{12}f \quad \text{on } \partial\Omega. \end{cases}$$

Thus $w=\frac{1}{12}f$, and (2.5) is proved. ∎

Remark 2.2. Bramble [2] has proposed for the 3-dimensional case a 19-point difference operator L_{19}^h and a difference solution u_{19}^h defined by

$$\begin{cases} L_{19}^h u_{19}^h=f+\frac{h^2}{12}\Delta f, \quad \text{in } \Omega_h^3 \\ u_{19}^h=g, \quad \text{on } \partial\Omega_h^3 \end{cases} \tag{2.14}$$

and proved that

$$u-u_{19}^h=O(h^4), \text{ in } \Omega_h^3 \text{ for } h\to 0.$$

It seems that (2.1) and (2.2) are easier to solve than the Bramble scheme (2.14) not only because of the 19-point operator L_{19}^h, but also because of the Δf appearing in (2.14) (possibly causing numerical differentiation for data).

As an alternative to the Bramble scheme we present

Remark 2.3. Let

$$L_{15}^h u=\frac{2}{3}\, L_7^h u+\frac{1}{3}\, L_9^h u$$

be the 15-point approximation, and let u_{15}^h be the difference solution defined by

$$\begin{cases} L_{15}^h u_{15}^h=f, \quad \text{in } \Omega_h^3, \\ u_{15}^h=g-\frac{h^2}{12}f, \quad \text{on } \partial\Omega_h^3. \end{cases} \tag{2.15}$$

It is easy to see that

$$u-u_{15}^h-\frac{h^2}{12}f=O(h^4), \quad \text{in } \Omega_h^3 \text{ for } h\to 0.$$

Finally, we remark that all three schemes (2.4), (2.14) and (2.15) are of the same accuracy $O(h^4)$. The essential difference in (2.4) is that u_h can be computed by two parallel schemes, and consequently can use parallel processors for finding a computer solution efficiently. In Section 4 some numerical tests have been presented where schemes (2.4) and (2.15) have been applied.

§ 3. Splitting Extrapolation

The extrapolation technique is a simple and effective tool for solving differential equations (among others), especially in a one-dimensional case. For the multidimensional problems the disadvantage of the standard isotropic extrapolation method is the need for the computation of another approximation with a small parameter, say $h/2$. This involves computing once again an approximation of a much larger size than the one corresponding to the original h. To remove this imperfection, we present, in the following, a splitting extrapolation procedure which will save computational work and storage.

Let $\Omega(h_1, \cdots, h_d)$ be the lattice domain with mesh size h_i along the variables $x_1, \cdots, x_d$. We denote by $u(\underline{h})$ the corresponding finite difference solution which have been obtained by using the central difference quotient equation (in 2-D, a 5-point, and in 3-D, a 7-point difference operator). If u is smooth enough, we have

$$u-u(h_1, \cdots, h_d) = \sum_{1\le|\alpha|\le m} c_\alpha \underline{h}^{2\alpha} + O(|\underline{h}|^{2m+1}), \tag{3.1}$$

where $m\geq 1$, $\alpha \Rightarrow (\alpha_1, \cdots, \alpha_d)$, $|\alpha| = \alpha_1+\cdots+\alpha_d$, $\underline{h}^\alpha = h_1^{\alpha_1}\cdots h_d^{\alpha_d}$, and c_α are unknowns to be determined in an approriate way.

The usual isotropic extrapolation method involves the following: make the homogeneous refinement lattices $\Omega\left(\frac{h_1}{2}, \cdots, \frac{h_d}{2}\right)$, $\Omega\left(\frac{h_1}{4}, \cdots, \frac{h_d}{4}\right)$, $\cdots$, and compute the isotropic extrapolation solutions (cf. (3.1))

$$\begin{cases} Iu(\underline{h}) = \frac{1}{3}(4u(\underline{h}/2) - u(\underline{h})), \\ Iu(\underline{h}/2) = \frac{1}{45}(64u(\underline{h}/4) - 20u(\underline{h}/2) + u(\underline{h})). \end{cases} \tag{3.2}$$

Then it is well known[2,7] that

$$\begin{cases} u - Iu(\underline{h}) = O\left(\sum_{i=1}^{d} h_i^4\right), & \text{in } \Omega(\underline{h}), \text{ for } h\to 0, \\ u - Iu(\underline{h}/2) = O\left(\sum_{i=1}^{d} h_i^6\right), & \text{in } \Omega(\underline{h}), \text{ for } h\to 0. \end{cases} \tag{3.3}$$

The splitting extrapolation method involves the following: Make the one-variable refinement meshes

$$\Omega\left(\frac{h_1}{2}, h_2, \cdots, h_d\right), \Omega\left(h_1, \frac{h_2}{2}, \cdots, h_d\right), \cdots, \Omega\left(h_1, h_2, \cdots, \frac{h_d}{2}\right)$$

and

$$\Omega\left(\frac{h_1}{4}, h_2, \cdots, h_d\right), \Omega\left(h_1, \frac{h_2}{4}, \cdots, h_d\right), \cdots, \Omega\left(h_1, h_2, \cdots, \frac{h_d}{4}\right).$$

Compute the difference solutions

$$\begin{cases} u_1\left(\frac{h_1}{2}, h_2, \cdots, h_d\right), \\ \vdots \\ u_d\left(h_1, h_2, \cdots, \frac{h_d}{2}\right). \end{cases} \tag{3.4}$$

and

$$\begin{cases} u_1\left(\frac{h_1}{4}, h_2, \cdots, h_d\right), \\ \vdots \\ u_d\left(h_1, h_2, \cdots, \frac{h_d}{4}\right), \end{cases} \tag{3.5}$$

and then compute the splitting extrapolation solutions

$$\mathrm{Su}(h/2) = \frac{1}{3}\left\{\sum_{i=1}^{d} 4u_i\left(h_1, \cdots, \frac{h_i}{2}, \cdots, h_d\right) - (4d-3)u(\underline{h})\right\} \tag{3.6}$$

and

$$\mathrm{Su}(h/4) = \frac{1}{45}\left\{\sum_{i=1}^{d} 64u_i\left(h_1, \cdots, \frac{h_i}{4}, \cdots, h_d\right) - 20u\left(h_1, \cdots, \frac{h_i}{2}, \cdots, h_d\right) - (44d-45)u(\underline{h})\right\}. \tag{3.7}$$

Then, if u is sufficiently smooth, one has (cf. (3.3))

$$\begin{cases} u - \mathrm{Su}(h/2) = O\left(\sum_{i=1}^{d} h_i^4\right), & \text{in } \Omega(h) \text{ for } h \to 0_+, \\ u - \mathrm{Su}(h/4) = O\left(\sum_{i=1}^{d} h_i^6\right), & \text{in } \Omega(h) \text{ for } h \to 0_+. \end{cases} \tag{3.8}$$

This result obviously follows from the general extrapolation theory ([3, 9]). We shall go into details only in a special case. Actually, in the asymptotic sense, the isotropic extrapolation method and the splitting extrapolation method give the same accuracy. We want, however, to emphasize the computational efficiency of the splitting extrapolation method. Keeping in mind the parallel architecture computer system, the algorithm for finding $\mathrm{Su}(h/2)$ or $\mathrm{Su}(h/4)$ seems to be a promising technique for multidimensional problems.

Finally, in order to get more insight from the results given above, we will make the situation more concrete.

Define in the interior of the lattice $\Omega(h, \cdots, h)$ the difference operator

$$\Delta^h u \equiv h^{-2}\sum_{k=1}^{d}\{u(x_1, \cdots, x_k - h, \cdots, x_d) - 2u(x_1, \cdots, x_k, \cdots, x_d) + u(x_1, \cdots, x_k + h, \cdots, x_d)\}.$$

Correspondingly we define in the interior of the lattice $\Omega\left(h, \cdots, \frac{h}{2}, \cdots, h\right)$ (in one dimension) the operators

$$\Delta_i^h u \equiv \left(\frac{h}{2}\right)^{-2}\left[u\left(x_1, \cdots, x_i - \frac{h}{2}, \cdots, x_d\right) - 2u(x_1, \cdots, x_i, \cdots, x_d) + u\left(x_1, \cdots, x_i + \frac{h}{2}, \cdots, x_d\right)\right] + h^{-2}\sum_{k \neq i}\{u(x_1, \cdots, x_k - h, \cdots, x_d) - 2u(x_1, \cdots, x_k, \cdots, x_d) + u(x_1, \cdots, x_k + h, \cdots, x_d)\}.$$

A flow chart for the splitting extrapolation method is shown in Fig. 3.1.

The splitting extrapolation method is numerically very efficient. The necessary computational effort of the isotropic extrapolation method is about $2^{3(d-1)}$ times that of the splitting extrapolation method. As is evident from Fig. 3.1 the splitting

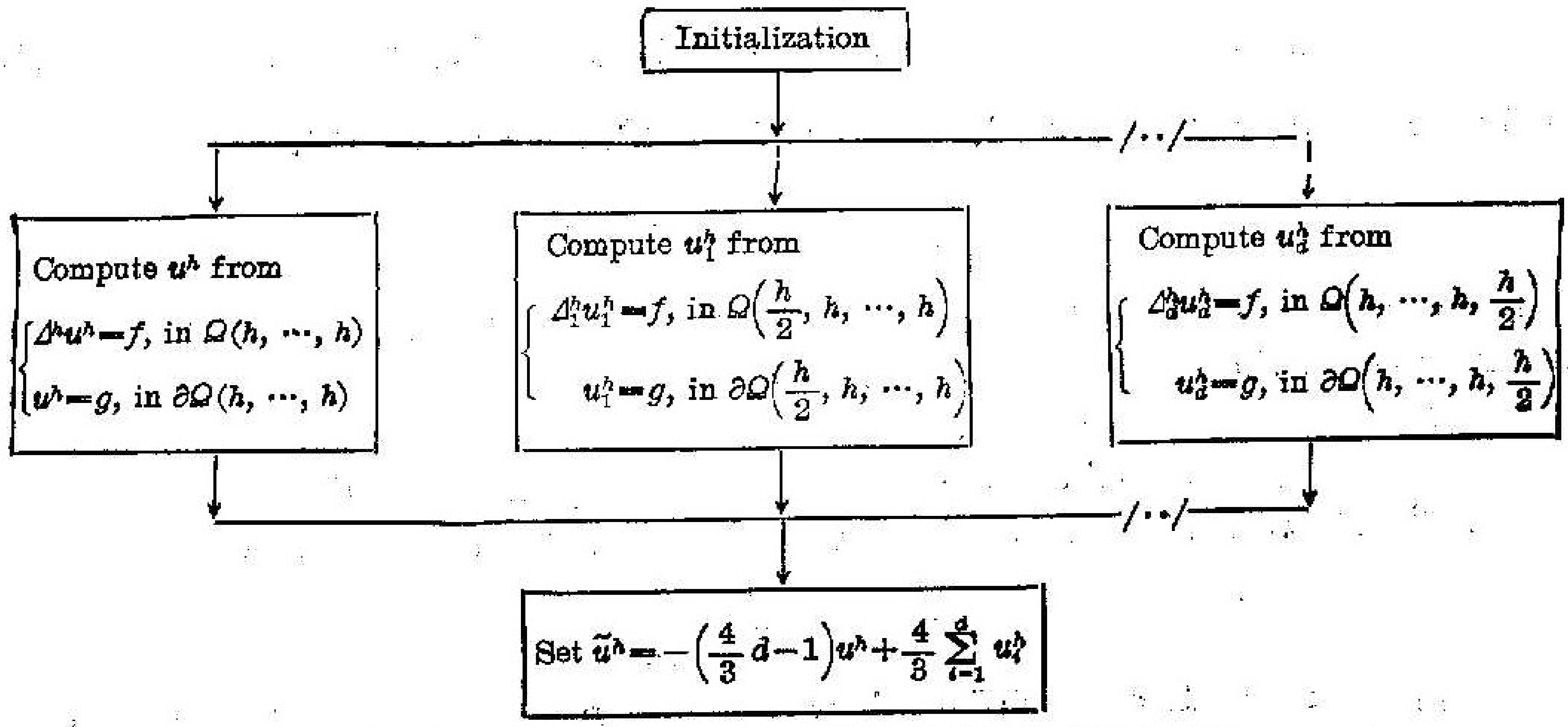

Fig. 3.1 Flow chart for the splitting extrapolation scheme

extrapolation method is very efficient in parallel computer architecture.

To end this section we shall give a rigorous proof for the convergence of the splitting extrapolation method. For the purpose of simplicity, we restrict ourselves to the two-dimensional case.

Theorem 3.1. *Suppose that $f=0$ and that*

$$\frac{\partial^2}{\partial x_1^2} f - \frac{\partial^2}{\partial x_2^2} f = 0$$

at four corners of Ω (coherence condition). Then, if $f \in C^{3+\alpha}(\overline{\Omega})$,

$$\|u - \tilde{u}^h\|_{L^\infty(\Omega(h,h))} = O(h^{3+\alpha}) \text{ for } h \to 0,\ 0<\alpha\leqslant 1. \tag{3.9}$$

Proof. We set the auxiliary problems

$$\begin{cases} \Delta e_i = l_i(u), & \text{in } \Omega, \\ e_i = 0, & \text{on } \partial\Omega, \end{cases} \tag{3.10}$$

where

$$l_0(u) = \frac{1}{12}\frac{\partial^4 u}{\partial x_1^4} + \frac{1}{12}\frac{\partial^4 u}{\partial x_2^4},$$

$$l_1(u) = \frac{1}{48}\frac{\partial^4 u}{\partial x_1^4} + \frac{1}{12}\frac{\partial^4 u}{\partial x_2^4},$$

$$l_2(u) = \frac{1}{12}\frac{\partial^4 u}{\partial x_1^4} + \frac{1}{48}\frac{\partial^4 u}{\partial x_2^4}.$$

Clearly $l_i(u) = 0$ at four corners of Ω (coherence condition). We have

$$e_i \in C^{3+\alpha}(\overline{\Omega}),\ \Delta_i^h e_i - \Delta e_i = O(h^{1+\alpha})$$

with $\Delta_0^h = \Delta^h$. Moreover,

$$\Delta_i^h u - \Delta u = h^2 l_i(u) + O(h^{3+\alpha}).$$

Hence,

$$\Delta_i^h(u - u_i^h - h^2 e_i) = \Delta_i^h u - \Delta u - h^2 \Delta e_i - h^2(\Delta_i^h e_i - \Delta e_i) = O(h^{3+\alpha}).$$

By the discrete max-min-principle

$$u - u_i^h = h^2 e_i + O(h^{3+\alpha}) \tag{3.11}$$

and

$$\frac{1}{3}(4e_1+4e_2-5e_0)=\frac{1}{3}\Delta^{-1}(4l_1(u)+4l_2(u)-5l_0(u))=0. \quad (3.12)$$

Now the assertion of Theorem 3.1 follows from (3.11) and (3.12). ∎

§ 4. Numerical Tests

Some numerical experiments are conducted to test the validity of the accuracy results given in the previous sections. Moreover, the presented methods have been compared with the standard ones. All computations are performed on Sperry 1100/70 at the computing centre of the University of Jyväskylä, using double precision (18 digits). The linear systems have been solved by S. O. R with the relaxation parameter

$$\omega=\frac{2}{1+\sin\pi h_{\min}},$$

where $h_{\min}$ is the smallest mesh length parameter in lattice $\Omega(h)$. The authors are indebted to Mr. K. Saarinen for his assistance in carrying out the numerical tests.

4.1. The two-dimensional case.

The methods tested are

(1) 5-point difference; accuracy $E(\Delta_h^+)=O(h^2)$,

(2) multigrid with 5-point difference; accuraty $\mathrm{MGE}(\Delta_h^+)=O(h^2)$,

(3) the correction method; accuracy $E(\Delta_h^*)=O(h^4)$, and

(4) splitting extrapolation; accuracy $SE(\Delta^h)=O(h^4)$.

The multigrid method tested is presented in [12] with the FORTRAN code. For multigrid methods we refer to [5].

Example 4.1. Let $\Omega=(0,1)\times(0,1)$. For

$$f(x_1,x_2)=-2\sin\pi x_2-(x_1-x_1^2)\pi^2\sin\pi x_2$$

and

$$g(x_1,x_2)=0$$

the exact solution of (1.1) is

$$u(x_1,x_2)=(x_1-x_1^2)\sin\pi x_2.$$

Table 4.1 shows the results of numerical tests; $E(\cdot)$ refers to L^∞-error, $R(h)=E(h)/E(2h)$ convergence order. The CPU times have been given in seconds.

Table 4.1 Comparison of results obtained from the four different methods

h	$E(\Delta_h^+)$ CPU, $R(h)$	$\mathrm{MGE}(\Delta_h^+)$ CPU, $R(h)$	$E(\Delta^*)$ CPU, $R(h)$	$SE(\Delta^h)$ CPU, $R(h)$
1/4	0.677×10^{-2} 0.16×10^{-1}; —	0.677×10^{-2} 0.17×10^{-1}; —	0.190×10^{-2} 0.43×10^{-1}; —	0.159×10^{-3} 0.13×10^{0}; —
1/8	0.166×10^{-2} 0.12×10^{0}; 4.1	0.166×10^{-2} 0.67×10^{-1}; 4.1	0.112×10^{-3} 0.27×10^{0}; 17.0	0.101×10^{-4} 0.11×10^{1}; 15.8
1/16	0.412×10^{-3} 0.99×10^{0}; 4.0	0.420×10^{-3} 0.27×10^{0}; 4.0	0.69×10^{-5} 0.19×10^{1}; 16.2	0.6×10^{-6} 0.85×10^{1}; 15.9
1/32	0.103×10^{-3} 0.76×10^{1}	0.103×10^{-3} 0.11×10^{1}; 4.0	0.4×10^{-6} 0.15×10^{2}; 16.1	0.4×10^{-8} 0.64×10^{2}; 16.0
1/64	—	0.253×10^{-4} 0.45×10^{1}; 4.0	—	—
accuracy	$O(h^2)$	$O(h^2)$	$O(h^4)$	$O(h^4)$

4.2. The three-dimensional case.

The methods tested are

(1) 7-point difference; accuracy $E(\Delta_h)=O(h^2)$,

(2) 15-point difference; accuracy $E(L_{15}^h)=O(h^4)$,

(3) Richardsons extrapolation (isotropic extrapolation); accuracy $RE(\Delta_h)=O(h^4)$,

(4) the correction method; accuracy $E(\Delta_h^*)=O(h^4)$, and

(5) splitting extrapolation; accuracy $SE(\Delta^h)=O(h^4)$.

Example 4.2. Let $\Omega=(0, 1)\times(0, 1)\times(0, 1)$. For

$$f(x_1, x_2, x_3)=-\pi^2\sin\pi x_1(x_2-x_2^2)(x_3-x_3^2)-2\sin\pi x_1(x_3-x_3^2)-2\sin\pi x_1(x_2-x_2^2)$$

and

$$g(x_1, x_2, x_3)=0$$

the exact solution of (1.1) is

$$u(x_1, x_2, x_3)=\sin\pi x_1(x_2-x_2^2)(x_3-x_3^2).$$

Table 4.2 shows the results of numerical tests for Example 4.2.

Example 4.3. Let $\Omega=(0, 1)\times(0, 1)\times(0, 1)$. For

$$f(x_1, x_2, x_3)=-14\pi^2\sin\pi x_1\sin 2\pi x_2\sin 3\pi x_3$$

and

Table 4.2 Comparison of the different methods used in Example 4.2

h	$E(\Delta_h^+)$ CPU, $E(h)$	$E(L_{15}^h)$ CPU, $E(h)$	$RE(\Delta_h)$ CPU, $E(h)$	$E(\Delta_h^*)$ CPU, $E(h)$	$SE(\Delta^h)$ CPU, $E(h)$
1/4	0.117×10^{-2} 0.75×10^{-1}; —	0.180×10^{-3} 0.10×10^{0}; —	0.67×10^{-5} 0.15×10^{1}; —	0.112×10^{-2} 0.17×10^{0}; —	0.300×10^{-4} 0.76×10^{0}; —
1/8	0.282×10^{-3} 0.10×10^{1}; 4.1	0.104×10^{-4} 0.16×10^{1}; 17.4	0.40×10^{-6} 0.23×10^{2}; 15.2	0.565×10^{-4} 0.21×10^{1}; 17.2	0.19×10^{-5} 0.12×10^{2}; 15.4
1/16	0.701×10^{-4} 0.16×10^{2}; 4.0	0.60×10^{-6} 0.27×10^{2}; 16.4	0.26×10^{-7} 0.35×10^{3}; 15.4	0.40×10^{-5} 0.32×10^{2}; 16.3	0.20×10^{-6} 0.74×10^{3}; 16.0
1/32	0.175×10^{-4} 0.26×10^{3}; 4.0	0.37×10^{-7} 0.37×10^{3}; 16.1	—	0.2×10^{-6} 0.51×10^{3}; 16.1	—
accuracy	$O(h^2)$	$O(h^4)$	$O(h^4)$	$O(h^4)$	$O(h^4)$

Table 4.3 Comparison of the different methods used in Example 4.3

h	$E(\Delta_h^+)$ CPU, $E(h)$	$E(L_{15}^h)$ CPU, $E(h)$	$RE(\Delta_h)$ CPU, $E(h)$	$E(\Delta_h^*)$ CPU, $E(h)$	$SE(\Delta^h)$ CPU, $E(h)$
1/4	0.439×10^{0} 0.42×10^{0}; —	0.131×10^{0} 0.17×10^{0}; —	0.205×10^{-1} 0.16×10^{1}; —	0.679×10^{0} 0.30×10^{1}; —	0.606×10^{-1} 0.10×10^{1}; —
1/8	0.944×10^{-1} 0.14×10^{1}; 4.7	0.953×10^{-2} 0.19×10^{1}; 13.7	0.113×10^{-2} 0.21×10^{2}; 18.1	0.293×10^{-1} 0.29×10^{1}; 23	0.305×10^{-2} 0.14×10^{2}; 19.9
1/16	0.228×10^{-1} 0.20×10^{2}; 4.1	0.613×10^{-3} 0.29×10^{2}; 15.6	0.669×10^{-3} 0.30×10^{3}; 16.5	0.168×10^{-2} 0.42×10^{2}; 17.6	0.181×10^{-3} 0.19×10^{3}; 16.9
1/32	0.564×10^{-2} 0.31×10^{3}; 4.0	0.386×10^{-4} 0.47×10^{3}; 15.9	—	0.103×10^{-3} 0.66×10^{3}; 16.4	—
accuracy	$O(h^2)$	$O(h^4)$	$O(h^4)$	$O(h^4)$	$O(h^4)$

$$g(x_1, x_2, x_3) = 0$$

the exact solution of (1.1) is

$$u(x_1, x_2, x_3) = \sin \pi x_1 \sin 2\pi x_2 \sin 3\pi x_3.$$

Table 4.3 shows the results of numerical tests for Example 4.3.

The above numerical results confirm the theoretical results. The parallel properties of the correction and the splitting extrapolation schems have not been utilized above. In spite of that the correction and splitting extrapolation methods seem to be superior to other methods tested.

References

[1] H. Blum, Lin, Q., R. Rannacher, Asymptotic error expansion and Richardson extrapolation for linear finite elements, *Numer. Math.*, 49, 1986, 11—37.

[2] J. Bramble, Fourth order finite difference analogues for the Dirichlet problem for Poisson's equations, *Math. Comp.*, 17, 1963, 217—222.

[3] C. Brezinski, A general extrapolation algorithm, *Numer. Math.*, 35, 1980, 175—187.

[4] G. E. Forsyte, W. R. Wasow, Finite Difference Methods for Partial Differential Equations, J. Wiley & Sons, London, 1960.

[5] W. Hackbush, Multigrid Methods and Applications, Springer series in computational mathematics, Vol. 4, Springer Verlag, Berlin, 1985.

[6] M. Křížek, P. Neittaanmäki, On superconvergence techniques, Universität Jyväskylä, Dept. Math., Preprint 33, 1984.

[7] Lin, Q., Lu, T., Splitting extrapolations for multidimensional problems, *J. Comp. Math.*, 1, 1983, 45—51.

[8] Lin, Q., Lu, T., The combination of approximate solutions for accelerating the convergence, *RAIRO Anal. Numér.*, 12, 1984, 153—160.

[9] G. I. Marchuk, Methods of Numerical Mathematics, Applications of mathematics, Vol. 2, Springer Verlag, Berlin 1975.

[10] A. R. Mitchell, Computational Methods in Partial Differential Equations, John Wiley & Sons, London, 1969.

[11] A. Schüller, Lin, Q., Efficient algorithms to accelerate the convergence in finite difference methods for elliptic boundary value problems using full multigrid technique in connection with extrapolation methods, to appear.

[12] K. Stüben, U. Trottenberg, Multigrid Methods: Fundamental Algorithms, Model Problem Analysis and Applications, in LN in Mathematics 960, Springer Verlag, Berlin, 1982.