MATHEMATICS OF COMPUTATION
VOLUME 43, NUMBER 167
JULY 1984, PAGES 89–115

Higher Order Multi-Grid Methods*

By Steve Schaffer

Abstract. This paper is concerned with the treatment of higher order multi-grid techniques for obtaining accurate finite difference approximations to partial differential equations. The three basic techniques considered are a multi-grid process involving smoothing via higher order difference approximations, iterated defect corrections with multi-grid used as an inner loop equation solver, and tau-extrapolation. Efficient versions of each of these three basic schemes are developed and analyzed by local mode analysis and numerical experiments. The numerical tests focus on fourth and sixth order discretizations of Poisson's equations and demonstrate that the three methods performed similarly yet substantially better than the usual multi-grid method, even when the right-hand side lacked sufficient smoothness.

Introduction. The goal of the numerical solution of partial differential equations is to obtain the highest accuracy possible within the constraints imposed by limitations in computer time and storage. Using higher order discretizations of the differential equation provides a means of obtaining this accuracy without requiring large amounts of storage. Higher order approximations, however, are more expensive to obtain due to the added complexity of the resulting discretized equations. Recent advances in fast solution techniques have made it more feasible to attempt these super accurate approximations. The Multi-Grid method is one such fast solver which is easily adapted to accommodate higher order processes. This paper treats three higher order multi-grid methods applied to finite difference discretizations of a linear partial differential equation.

This paper is primarily concerned with comparing the numerical performance of the three higher order multi-grid solution processes. The model problem used for the experiments is given by

$$\begin{aligned} LU &= F \quad \text{in the interior of } \Omega, \\ U &= G \quad \text{on the boundary of } \Omega(\partial\Omega), \end{aligned} \tag{1}$$

where L is the Laplace operator, Ω is the unit square in R^2, F is a continuous function defined on the interior of Ω and G is a continuous function defined on $\partial\Omega$. Several discretizations of (1), using second, fourth and sixth order differences, are considered. A number of algorithms based on these higher order multi-grid methods are tested in numerical experiments for their efficiency of solution (i.e., the work required to obtain a given accuracy). The algorithms are defined by various choices of discretizations and multi-grid processes, grid transfers, types of relaxation, etc.

Received June 21, 1982; revised November 2, 1983.
1980 *Mathematics Subject Classification*. Primary 65N20, 65F10.
* This work was supported by the National Science Foundation under grant no. MCS78-03847 and the Air Force Office of Scientific Research under grant no. F33615-79-3223.

0025-5718/84 $1.00 + $.25 per page

The results of these experiments are used for comparison of the three higher order methods as well as the discretizations.

Other work involving higher order multi-grid schemes has focused on some combination of multi-grid and defect correction. (See Chapter 2, Section 1 for comments regarding the methods treated by Brandt [5] and Hackbusch [8].) Auzinger and Stetter [1] have treated several implementations of the defect correction iteration and the multi-grid solution process.

The current paper is an outgrowth of the work presented in the report [10] and is contained in the author's thesis [11].

In Chapter 1, the discretizations of (1) that are used in the experiments are described. The definitions of the various multi-grid processes are then given, followed by a description of the second order multi-grid algorithm. Finally, the full multi-grid solution process, generalized to incorporate higher orders, is discussed. The three higher order multi-grid methods are described in Chapter 2. The first section of Chapter 3 deals with estimating the convergence rates of the higher order multi-grid algorithms through a local mode analysis; the next section presents the numerical results of the experiments; and the last section summarizes the results of the numerical experiments.

The graphs that are discussed in the text are available in a supplement at the end of this issue.

CHAPTER 1. DISCRETIZATIONS AND MULTI - GRID PROCESSES

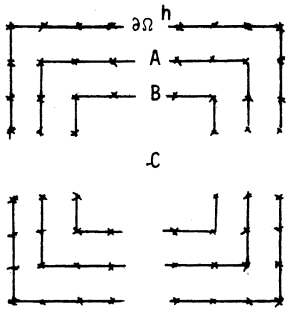
This chapter is concerned with describing the finite difference discretizations and the various multi-grid processes used in the higher order methods described in Chapter 2.

1.1. The Finite Difference Discretizations. For each $h \in \{2^{-K}: K = 1,2,\ldots,6\}$, a uniform grid, Ω^h, is defined on the unit square with mesh spacing h. Finite difference equations approximating (1) will be denoted by:

$$L^h U^h = F^h \quad \text{on the interior of } \Omega^h, \tag{1.1.1}$$

$$U^h = G^h \quad \text{on the boundary of } \Omega^h(\partial\Omega^h), \tag{1.1.2}$$

where L^h is one of the finite difference operators described below, G^h is the pointwise restriction of G onto $\partial\Omega^h$ and F^h is a grid function representing the function F. It is convenient to partition Ω^h as follows:



where the sets A and B consist of points whose distance is h and $2h$ from $\partial\Omega^h$, respectively, and C consists of all remaining interior points.

The purpose here is to study high order multi-grid methods for solving (1). There are many possible versions for these methods due in part to the range of choices for discretizations (1.1.1)–(1.1.2), relaxation processes and grid transfer operators. In this section the particular finite difference operators and discretizations of (1) considered in this paper will be described. In what follows, F^h will represent the pointwise restriction of F onto Ω^h.

L_2^h: This operator is defined by the usual five point star differencing scheme

$$\frac{1}{h^2}\begin{pmatrix} & 1 & \\ 1 & -4 & 1 \\ & 1 & \end{pmatrix}_h$$

at each interior point of Ω^h. Two corresponding discretizations are considered for this operator and are given by

$$L_2^h U^h(x, y) = F^h(x, y), \tag{1.1.3}$$

$$L_2^h U^h(x, y) = I_{h/2}^h F^{h/2}(x, y) = \frac{1}{16}\begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix}_{h/2} F^{h/2}(x, y), \tag{1.1.4}$$

where $F^{h/2}(x, y)$ is the pointwise restriction of $F(x, y)$ onto $\Omega^{h/2}$, and $I_{h/2}^h$, representing here the "full weighting operator", restricts functions on $\Omega^{h/2}$ onto Ω^h by averaging as indicated.

L_M^h: This is the Mehrstellen Verfahren difference operator (see Collatz [6]) given by the difference stencil

$$\frac{1}{6h^2}\begin{pmatrix} 1 & 4 & 1 \\ 4 & -20 & 4 \\ 1 & 4 & 1 \end{pmatrix}_h.$$

This operator is used in conjunction with the following weightings of F to produce, respectively, the fourth and sixth order difference approximations,

$$L_M^h U^h(x, y) = \frac{1}{12}\begin{pmatrix} & 1 & \\ 1 & 8 & 1 \\ & 1 & \end{pmatrix}_h F^h(x, y) \tag{1.1.5}$$

and

$$L_M^h U^h(x, y) = \frac{1}{360}\begin{pmatrix} 1 & & 4 & & 1 \\ & 48 & & 48 & \\ 4 & & 148 & & 4 \\ & 48 & & 48 & \\ 1 & & 4 & & 1 \end{pmatrix}_{h/2} F^{h/2}. \tag{1.1.6}$$

L_4^h: This operator is derived by approximating the two differential operators, $\partial^2/\partial x^2$ and $\partial^2/\partial y^2$, separately by fourth order finite differences, in their respective directions. In regions B and C, the differencing in both directions is given by

$$\frac{1}{12h^2}\begin{pmatrix} -1 & 16 & -30 & 16 & -1 \\ & & \bullet & & \end{pmatrix}_h, \tag{1.1.7}$$

where "$\bullet$" is used to indicate the central coefficient. In region A, (1.1.7) is used whenever possible and, in directions where it is not, the fourth order difference

$$\frac{1}{12h^2}\begin{pmatrix} 10 & -15 & -4 & 14 & -6 & 1 \\ & \bullet & & & & \end{pmatrix}_h \tag{1.1.8}$$

is used.

$L^h_{4,2}$: This difference operator agrees with L^h_4 except that the difference, (1.1.8), is replaced by the second order difference

$$\frac{1}{h^2}\begin{pmatrix} 1 & -2 & 1 \\ & \bullet & \end{pmatrix}_h. \tag{1.1.9}$$

$L^h_{4,M}$: This difference operator agrees with L^h_4 except that, in region A, the fourth order Mehrstellen Verfahren difference equation, (1.1.5), is enforced.

L^h_6: Again, the differential operators, $\partial^2/\partial x^2$ and $\partial^2/\partial y^2$, are differenced separately. In region C, the sixth order differencing scheme

$$\frac{1}{180h^2}\begin{pmatrix} 2 & -27 & 270 & -490 & 270 & -27 & 2 \\ & & & \bullet & & & \end{pmatrix}_h \tag{1.1.10}$$

is used for both directions. In regions A and B, the sixth order differences

$$\frac{1}{180h^2}\begin{pmatrix} -11 & 214 & -378 & 130 & 85 & -54 & 16 & -2 \\ & & \bullet & & & & & \end{pmatrix}_h \tag{1.1.11}$$

and

$$\frac{1}{180h^2}\begin{pmatrix} 126 & -70 & -486 & 855 & -670 & 324 & -90 & 11 \\ & \bullet & \bullet & & & & & \end{pmatrix}_h \tag{1.1.12}$$

are used in directions where the central point has, respectively, only one or two neighboring points in Ω^h. At all other points of regions A and B, (1.1.10) is used.

$L^h_{6,4}$: This operator agrees with L^h_6 in regions B and C. In region A, (1.1.12) is replaced by the fourth order difference (1.1.8).

$L^h_{6,M}$: This operator agrees with L^h_6 in region C. In regions A and B, the sixth order Mehrstellen Verfahren difference equation (1.1.6) is enforced.

The discretization equations for the operators L^h_4, $L^h_{4,2}$, $L^h_{4,M}$, L^h_6, $L^h_{6,4}$, and $L^h_{6,M}$ each use the pointwise restriction of F onto Ω^h with the exception of the points where the Mehrstellen Verfahren is used in operators $L^h_{4,M}$ and $L^h_{6,M}$. The boundary conditions will henceforth be incorporated with the interior equations so that (1.1.1) and (1.1.2) are now written as

$$L^h U^h = F^h \text{ on } \Omega^h. \tag{1.1.13}$$

It is understood that this equation holds at points interior to Ω^h.

1.2. Relaxation, Restriction and Interpolation Operators. Two relaxation processes are considered here for solving equations of the form (1.1.13) (the subgrids Ω^h_{even} and Ω^h_{odd} are defined by $\Omega^h_{\text{even}} = \{(x, y) \in \Omega^h: (x + y)/h \equiv 0 \pmod 2\}$ and $\Omega^h_{\text{odd}} = \Omega^h - \Omega^h_{\text{even}}$).

LEX: This refers to the usual Gauss-Seidel relaxation method, where the points in Ω^h are processed lexicographically.

CHJ: This refers to Gauss-Seidel/Checkerboard relaxation with even-odd "Jacobi-like" processing of the points. New values are determined for the approximation to U^h on Ω^h_{even}, the approximation being displaced by these new values at the completion of the sweep. A similar sweep over Ω^h_{odd} is then made beginning with the updated approximation.

The restriction operator, denoted by I_h^{2h}, maps a function defined on Ω^h to some representation of this function on Ω^{2h}. The three restriction operators used here are:

$$\begin{aligned}
&\text{injection} && - \quad (1)_h \\
&\text{injection by } 1/2 && - \quad \frac{1}{2}(1)_h \\
&\text{full weighting} && - \quad \frac{1}{16}\begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix}_h .
\end{aligned}$$

The prolongation operator, denoted by I_{2h}^h, interpolates a function defined on Ω^{2h} to Ω^h where point evaluation is used at coextensive points. The prolongations considered here are defined by bilinear, cubic and quintic interpolation.

The identity operator on each grid is denoted simply by I, where it will be clear from the context on which grid it is to operate. Likewise, U will denote the solution of (1) and also its restrictions onto each grid.

1.3. The Method MG2(n_1, n_2). The Correction Cycle Multi-Grid scheme is described briefly as follows (cf., Brandt [3] for more details). Let (1.1.13) represent the second order discretization, (1.1.3), on some grid Ω^h, and suppose the choices are made for restriction, prolongation and relaxation from those described in Section 1.2. To avoid cumbersome notation, the current approximation to the solution, U^h, of (1.1.13) will be denoted by the lower case u^h at each step of the process and the symbol " $\leftarrow$ " will be used throughout to indicate replacement. The correction cycle multi-grid scheme, denoted by MG2(n_1, n_2), will first be described for two grids in the following four steps where the initial guess is given:

1. Perform n_1 relaxation sweeps on Eq. (1.1.13) ($u^h \leftarrow$ resulting approximation).
2. On Ω^{2h}, solve the "coarse grid correction" equation

$$L_2^{2h} V^{2h} = I_h^{2h}\left(F^h - L_2^h u^h\right). \tag{1.3.1}$$

3. Correct u^h by

$$u^h \leftarrow u^h + I_{2h}^h V^{2h}. \tag{1.3.2}$$

4. Perform n_2 relaxation sweeps on (1.1.13) ($u^h \leftarrow$ resulting approximation).

The correction cycle, as described, consists of two main parts, namely, smoothing (steps 1 and 4) and coarse grid correction (steps 2 and 3). The latter involves the approximation of the fine grid residual equation

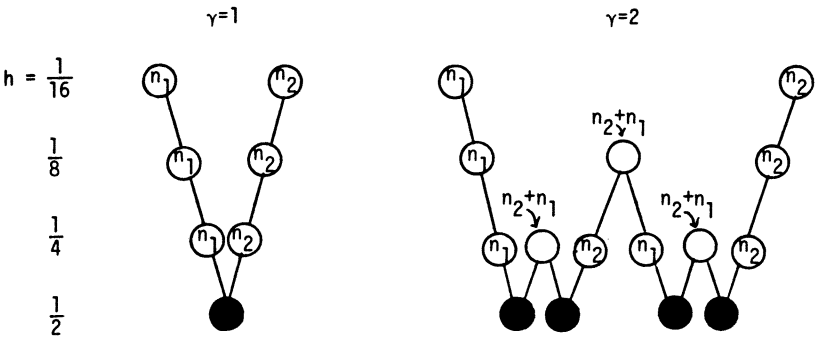
$$L_2^h V^h = F^h - L_2^h u^h \tag{1.3.3}$$

by the coarse grid correction equation (1.3.1) on Ω^{2h} so that the correction, V^{2h}, used in (1.3.2) acts as a coarse grid approximation to the fine grid error, $V^h = U^h - u^h$. This approximation is meaningful only when V^h is smooth enough, that is, when the high frequency components in V^h, which have no representation on Ω^{2h}, have been sufficiently reduced. The smoothing part of the correction cycle attempts to accomplish this by relaxation, which quickly damps the high frequency information in V^h.

On more than two grids, MG2(n_1, n_2), is defined recursively as follows. Since the coarse grid correction equation, (1.3.2), is of the same form as the original equation, (1.1.13), on Ω^h, it is natural to attempt to solve it by a similar application of steps 1

through 4. This would involve performing n_1 relaxations on Ω^{2h} followed by a correction from the next coarser grid, Ω^{4h}, on which the coarse grid correction equation is again of the same form as (1.1.13). It follows that this process can be continued recursively by applying in turn the correction cycle multi-grid scheme to the coarse grid correction equations defined on successively coarser grids. The process continues until a sufficiently coarse grid is reached, at which point the problem on this grid would be solved exactly. The approximate corrections are then passed back to successively finer grids via step 3 where n_2 relaxation sweeps are performed on the existing equation on each grid. When correction and relaxation have occurred on the finest grid, the correction cycle is complete.

The cycling structure for MG2(n_1, n_2) is determined by the number, γ, of applications of steps 1 through 4 on each coarse grid before passing the approximate correction back to the next finer grid. The cycling structures considered here correspond to $\gamma = 1$ (the so-called V-cycle) and $\gamma = 2$ (the so-called W-cycle). These are depicted for $h = 1/16$ by



where the symbols $\backslash$, $/$, $\textcircled{n}$ and $\bullet$ denote, respectively, forming the coarse grid correction equation, performing the coarse-to-fine correction (step 3), performing n relaxation sweeps and solving the grid equation exactly.

For completeness, the iteration matrix for the second MG2(n_1, n_2) is derived. An analysis of MG2(n_1, n_2) is found in [9] and [13]. The relaxation method used in steps 1 and 4 will now be represented in terms of a matrix R^h and vector $\tilde{f}^h$ by

$$u^h \leftarrow R^h u^h + \tilde{f}^h. \tag{1.3.4}$$

Since $\tilde{f}^h$ is constructed in terms of F^h so that U^h is a fixed point of (1.3.4), then the actual error $V^h = U^h - u^h$ satisfies

$$V^h \leftarrow R^h V^h. \tag{1.3.5}$$

From (1.3.1) and (1.3.3) it follows that

$$V^{2h} = \left(L_2^{2h}\right)^{-1} I_h^{2h} L_2^h V^h, \tag{1.3.6}$$

where $(L_2^{2h})^{-1}$ is the inverse of L_2^{2h}. Thus, step 3 yields

$$u^h \leftarrow u^h + I_{2h}^h \left(L_2^{2h}\right)^{-1} I_h^{2h} L_2^h V^h. \tag{1.3.7}$$

Subtracting both sides in (1.3.7) from U^h gives

$$V^h \leftarrow V^h - I_{2h}^h \left(L_2^{2h}\right)^{-1} I_h^{2h} L_2^h V^h, \tag{1.3.8}$$

and, combining (1.3.5) and (1.3.8), it follows that the error after one cycle of MG2(n_1, n_2) on two levels is governed by

$$V^h \leftarrow (R^h)^{n_2}\left(I - I_{2h}^h\left(L_2^{2h}\right)^{-1} I_h^{2h} L_2^h\right)(R^h)^{n_1} V^h \equiv M_2^h V^h, \tag{1.3.9}$$

where M_2^h is called the two level multi-grid iteration matrix. The multi-grid iteration matrix, $M^h = M_K^h$, can now be described recursively according to the number of levels, K, that are used (see Stüben [13] for details of a similar development) by

$$M^h = \begin{cases} M_2^h, & K = 2, \\ M_2^h + (R^h)^{n_2} I_{2h}^h \left(M_{K-1}^h\right)^{\gamma} \left(L_2^{2h}\right)^{-1} I_h^{2h} L_2^h (R^h)^{n_1}, & K > 2, \end{cases} \tag{1.3.10}$$

where γ is the cycling parameter described above. The residual $L_2^h V^h$ satisfies

$$L^h V^h \leftarrow \left(L_2^h M^h \left(L_2^h\right)^{-1}\right) L_2^h V^h, \tag{1.3.11}$$

where the matrix $L_2^h M^h (L_2^h)^{-1}$ is called the iteration matrix for the residual.

1.4. The Full Multi-Grid (FMG) Algorithm. The method MG2(n_1, n_2) and the three higher order multi-grid methods to be described in Chapter 2 are implemented here as FMG algorithms, introduced by Brandt [3]. The basic purpose of FMG is to efficiently produce a good initial approximation for some basic multi-grid iteration process on a given fine grid by using an approximate solution to the corresponding problem on the next coarser grid. The approximation on the coarser grid is obtained by a multi-grid process which again is initiated by an approximate solution from a still coarser grid and so on. This differs from that of the coarse grid correction step used in the multi-grid scheme basically in that the latter attempts to approximate the actual error of the fine grid approximation, whereas the fine grid solution itself is approximated in FMG.

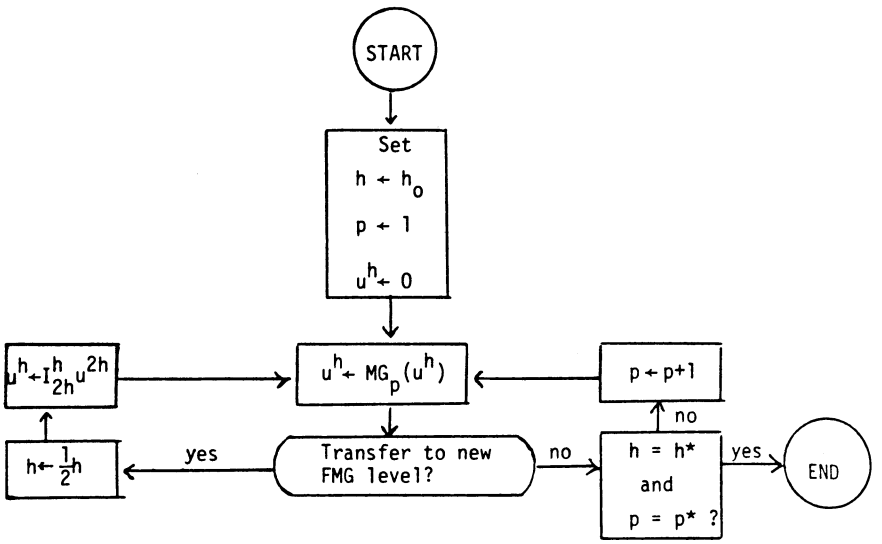


FIGURE 1

In addition to the mesh refinement process described above, the FMG algorithm used here incorporates the possibility of increasing the order of approximation. Specifically, let the increasing set of positive integers, $\{\sigma_p: p = 1,2,\ldots,p^*\}$, and the mesh size parameters, h_0 and h^*, where $h^* = 2^{-L}h_0$ for some positive integer L, be given. Furthermore, for each σ_p, let MG_p denote a given σ_pth order multi-grid method and let $u^h \leftarrow \mathrm{MG}_p(u^h)$ denote the result of applying MG_p on grid h with initial approximation u^h. The general FMG algorithm for obtaining a σ_{p^*}th order approximation solution on grid h^* is then given in Figure 1.

Chapter 2. The Higher Order Methods

This chapter describes three multi-grid methods for obtaining high order approximations to (1).

2.1. The Method MGH. This method consists of a smoothing step on the fine grid via the higher order difference equation and a coarse grid correction based on the higher order residual. Specifically, let one of the higher order finite difference discretizations described in Section 1.1 be denoted by

$$\mathscr{L}^h \mathscr{U}^h = F^h, \tag{2.1.1}$$

and let u^h be an initial approximation obtained as described in the FMG algorithm. One iteration of the method $\mathrm{MGH}(n_1, n_2, n_3, n_4)$ is described in four steps as follows:

1. Perform n_1 relaxation sweeps on equation (2.1.1) ($u^h \leftarrow$ resulting approximation).

2. Starting with an initial approximation $v^{2h} = 0$ on grid $2h$, apply the method $\mathrm{MG2}(n_3, n_4)$ to the MGH coarse grid correction equation

$$L_2^{2h} V^{2h} = I_h^{2h}(F^h - \mathscr{L}^h u^h) \tag{2.1.2}$$

($v^{2h} \leftarrow$ the final approximation to V^{2h}). (When the W-cycle is used, two iterations of $\mathrm{MG2}(n_3, n_4)$ are made on (2.1.2).)

3. Correct u^h by

$$u^h \leftarrow u^h + I_{2h}^h v^{2h}. \tag{2.1.3}$$

4. Perform n_2 relaxation sweeps on Eq. (2.1.1) ($u^h \leftarrow$ resulting approximation).

As in the conventional second order case, relaxation is performed in steps 1 and 4 to smooth the error $e^h = \mathscr{U}^h - u^h$. The higher order relaxations used here have the advantage of dealing directly with this error. It is important to note that, in addition to the added expense of these higher order relaxations, the higher order difference operators usually produce poorer smoothing rates and are more likely to become unstable in relaxation. This is due to the more complicated stencils defining these difference operators. However, in the present experiments, it was found that, with the exception of L_6^h, the higher order difference operators considered for the model problem performed reasonably well in the MGH method. (See Chapter 3.)

The replacement of the higher order relaxation in steps 1 and 4 in the above algorithm by second order relaxation results in a version of the multi-grid defect correction method (cf. [4] and [8]). This method, which uses relaxation on a lower order approximation to smooth the error of the higher order approximation, is

similar in structure to the method of τ-extrapolation. (See Section 2.3.) The experimental performance of both methods on the Model Problem was found to be nearly equivalent, so this method is not treated here.

The iteration matrix governing the error in the method MGH(n_1, n_2, n_3, n_4), which will be used in Section 3.1, is derived as follows. Let $\mathscr{R}^h$ be the iteration matrix for the relaxation in steps 1 and 4 so that the error after a relaxation sweep is given by

$$e^h \leftarrow \mathscr{R}^h e^h. \tag{2.1.4}$$

In step 2, the error $V^{2h} - v^{2h}$ after γ iterations of MG2(n_3, n_4) applied to (2.1.2) is related to the initial error, V^{2h} (v^{2h} is initially zero), by

$$V^{2h} - v^{2h} = (M^{2h})^{\gamma} V^{2h}, \tag{2.1.5}$$

where M^{2h} is the multi-grid iteration matrix (1.3.10) defined on grid $2h$. From (2.1.1), (2.1.2) and (2.1.5) it follows that

$$v^{2h} = \big(I - (M^{2h})^{\gamma}\big)V^{2h} = \big(I - (M^{2h})^{\gamma}\big)(L^{2h})^{-1} I_h^{2h} \mathscr{L}^h e^h. \tag{2.1.6}$$

Thus, the new approximation obtained in step 3 can be expressed according to

$$u^h \leftarrow u^h + I_{2h}^h \big(I - (M^{2h})^{\gamma}\big)(L^{2h})^{-1} I_h^{2h} \mathscr{L}^h e^h. \tag{2.1.7}$$

Subtracting both sides of (2.1.7) from $\mathscr{U}^h$ and using (2.1.4), it follows that

$$e^h \leftarrow (\mathscr{R}^h)^{n_2} \Big(I - I_{2h}^h \big(I - (M^{2h})^{\gamma}\big)(L^{2h})^{-1} I_h^{2h} \mathscr{L}^h\Big)(\mathscr{R}^h)^{n_1} e^h. \tag{2.1.8}$$

The matrix on the right side of (2.1.8) is the iteration matrix for the method MGH(n_1, n_2, n_3, n_4). This matrix will be used in Section 3.1 to analyze the local properties of this method.

2.2. The Method MGD. This method is based on Iterated Defect Corrections (IDEC) where an "inner loop" multi-grid iteration process is implemented to approximately solve the discrete equations resulting from the "outer loop" IDEC iterations. Specifically, let the higher order approximation (2.1.1) be given and let u^h be an initial approximation obtained as described in the higher order FMG algorithm. One iteration of the method MGD(n_1, n_2, n_3) is described in three steps as follows:

0. If the method is to start on a new FMG level, that is, if the initial approximation u^h is the result of interpolating the final approximation on the next coarser level, then perform n_3 relaxation sweeps on the second order difference equation ($u^h \leftarrow$ resulting approximation).

1. Starting with an initial approximation $v^h = 0$ apply MG2(n_1, n_2) to the higher order defect equation

$$L_2^h V^h = F^h - \mathscr{L}^h u^h \tag{2.2.1}$$

($v^h \leftarrow$ the final approximation to the correction, V^h).

2. Correct u^h by

$$u^h \leftarrow u^h + v^h. \tag{2.2.2}$$

The purpose of the second order relaxations in step 0 is to attempt to smooth the higher order error $e^h = \mathscr{U}^h - u^h$ which initially contains high frequency components, due in part to the FMG interpolation $u^h \leftarrow I_{2h}^h u^{2h}$. These high frequency

components in the error contaminate the defect $F^h - \mathscr{L}^h u^h$ and can reduce the effectiveness of the higher order correction of steps 1 and 2. This was confirmed by experiments reported in Chapter 3.

An important feature of the method MGD is that it involves only lower order relaxation on all grids. It differs from the coarse grid defect correction method mentioned in Section 2.1 in that the latter immediately passes the higher order defect to the coarse grid and the unperturbed second order equation is then used to approximately smooth the higher order error on the fine grid. In the method MGD, the second order relaxation on the fine grid does not deal directly with the actual higher order error but is designed to smooth the error of the approximation to the solution of the defect correction equation (2.2.1). This makes the fine grid smoothing step consistent with the coarse grid correction step.

The iteration matrix for the method MGD, which will be used in Section 3.1, is derived as follows. In step 1, applying the method MG2(n_1, n_2) to Eq. (2.2.1) transforms the error $\varepsilon^h = V^h - v^h$ according to

$$\varepsilon^h \leftarrow M^h \varepsilon^h, \tag{2.2.3}$$

where M^h is the multi-grid iteration matrix in (1.3.10). From (2.2.1) and the fact that the initial approximation is identically zero, it follows that the initial error is given in terms of the higher order error e^h by

$$\varepsilon^h = V^h = \left(L_2^h\right)^{-1} \mathscr{L}^h e^h. \tag{2.2.4}$$

Combining (2.2.3) and (2.2.4) and rearranging terms, it follows that the final approximation in step 1 is given by

$$v^h = V^h - M^h V^h = (I - M^h)\left(L_2^h\right)^{-1} \mathscr{L}^h e^h. \tag{2.2.5}$$

Thus, correction (2.2.2) gives the new approximation

$$u^h \leftarrow u^h + (I - M^h)\left(L_2^h\right)^{-1} \mathscr{L}^h e^h, \tag{2.2.6}$$

and by subtracting both sides in (2.2.6) from $\mathscr{U}^h$ it follows that

$$e^h \leftarrow \left(I - (I - M^h)(L^h)^{-1} \mathscr{L}^h\right) e^h. \tag{2.2.7}$$

The matrix on the right-hand side of (2.2.7) is the iteration matrix governing the higher order error for MGD.

2.3. The Method of τ-Extrapolation. τ-extrapolation is based on the assumption that for h small enough the truncation error, τ^h, of the second order discretization (1.1.3) can be expressed locally by

$$\tau^h \equiv L_2^h U - F^h = Ah^2 + O(h^4), \tag{2.3.1}$$

where the function A depends on the points $(x, y) \in \Omega$, but is independent of h. Thus, the truncation errors τ^h and τ^{2h} can be extrapolated to yield the fourth order approximation

$$\tau_h^{2h} \equiv \frac{4}{3}\left(\tau^{2h} - I_h^{2h} \tau^h\right) = \tau^{2h} + O(h^4). \tag{2.3.2}$$

The method of τ-extrapolation uses an approximation to the extrapolation (2.3.2) as follows. Let u^h be a second order approximation to U on grid h and define

$$r^h = L_2^h u^h - F^h, \tag{2.3.3}$$

$$r^{2h} = L_2^{2h} I_h^{2h} u^h - F^{2h} \tag{2.3.4}$$

and

$$r_h^{2h} = \frac{4}{3}\left(r^{2h} - I_h^{2h} r^h\right). \tag{2.3.5}$$

Then combining (2.3.1) through (2.3.5) it follows that

$$\tau_h^{2h} - r_h^{2h} = \frac{4}{3}\left(L_2^{2h} I_h^{2h} - I_h^{2h} L_2^h\right)(U - u^h) = O(h^4), \tag{2.3.6}$$

where the $O(h^4)$ term depends on fourth differences of the error $e^h = U - u^h$. Equations (2.3.2) and (2.3.6) imply that the equation

$$L_2^{2h} \hat{U}^{2h} = F^{2h} + r_h^{2h} \tag{2.3.7}$$

will produce a fourth order approximation to U on grid $2h$. Setting $u^h = U^h$, the exact solution of the second order equation, and solving the equation (2.3.7) then yields the solution

$$\begin{aligned} \hat{U}^{2h} &= \left(L_2^{2h}\right)^{-1}\left(F^{2h} + \frac{4}{3}\left(L_2^{2h} I_h^{2h} U^h - F^{2h}\right)\right) \\ &= U^{2h} + \frac{4}{3}\left(I_h^{2h} U^h - U^{2h}\right). \end{aligned} \tag{2.3.8}$$

This is equivalent to applying fourth order Richardson's extrapolation from grid h to grid $2h$, although τ-extrapolation is more general since the expression (2.3.1) can hold even when U^h itself does not have a similar expansion in h.

Also considered here is a sixth order τ-extrapolation onto grid $4h$ which assumes that the truncation error can be locally expanded in the form

$$\tau^h = L_2^h U - F^h = Ah^2 + Bh^4 + O(h^6), \tag{2.3.9}$$

where the functions A and B are independent of h. Using the extrapolated values τ_h^{2h} and τ_{2h}^{4h}, where τ_{2h}^{4h} is obtained from the extrapolation of τ^{2h} and τ^{4h}, a second extrapolation yields

$$\tau_h^{4h} \equiv \frac{11}{15}\tau_{2h}^{4h} + \frac{16}{15} I_{2h}^{4h} \tau_h^{2h} = \tau^{4h} + O(h^6). \tag{2.3.10}$$

Let u^h and u^{2h} be fourth order approximations to U on grid h and grid $2h$, respectively, and define

$$r^{4h} = L_2^{4h} I_{2h}^{4h} u^{2h} - F^{4h}, \tag{2.3.11}$$

$$r_{2h}^{4h} = \frac{4}{3}\left(r^{4h} - I_{2h}^{4h} r^{2h}\right) \tag{2.3.12}$$

and

$$r_h^{4h} = \frac{11}{15} r_{2h}^{4h} + \frac{16}{15} I_{2h}^{4h} r_h^{2h}. \tag{2.3.13}$$

Combining (2.3.10) through (2.3.13), it follows that

$$\begin{aligned} \tau_h^{4h} - r_h^{4h} &= \frac{44}{45}\left(L_2^{4h} I_{2h}^{4h} - I_{2h}^{4h} L_2^{2h}\right)(U - u^{2h}) \\ &\quad + \frac{48}{45} I_{2h}^{4h}\left(L_2^{2h} I_h^{2h} - I_h^{2h} L_2^h\right)(U - u^h) \\ &= O(h^6), \end{aligned} \tag{2.3.14}$$

where the $O(h^6)$ term depends on a combination of fourth differences of the errors $U - u^h$ and $U - u^{2h}$. Equations (2.3.10) and (2.3.14) now imply that the equation

$$L^{4h}\hat{U}^{4h} = F^{4h} + r_h^{4h} \tag{2.3.15}$$

will produce a sixth order approximation to U on grid $4h$.

The full weighting restriction operator is an $O(h^2)$ perturbation of the injection operator given explicitly by

$$\frac{1}{16}\begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix}_h = (1)_h + \frac{1}{16}\left[\frac{1}{h^2}\begin{pmatrix} 1 & 2 & 1 \\ 2 & -12 & 2 \\ 1 & 2 & 1 \end{pmatrix}_h\right]h^2, \tag{2.3.16}$$

where the bracketed term is a second order difference approximation to the Laplacian operator. It follows from (2.3.16) that the truncation error of the second order discretization (1.3.4) will have an expansion of the same form as (2.3.1) or (2.3.9). The τ-extrapolation applied to the discretization (1.3.4) will be referred to as weighted τ-extrapolation. Expression (2.3.16) can also be used to show that the extrapolations (2.3.2) and (2.3.10) and the order of approximation in (2.3.6) and (2.3.14) continue to hold when the full weighting restriction operator is used in these expressions. Thus, full weighting of the residuals will be considered in both the τ-extrapolation and weighted τ-extrapolation methods.

In the following descriptions of the τ-extrapolation methods, the initial fine grid approximation is assumed to have been obtained as described in the FMG algorithm. Equation (1.1.13) will represent the discretization (1.1.3) in the case of τ-extrapolation or the discretization (1.1.4) in the case of weighted τ-extrapolation. The method of (weighted) τ-extrapolation to fourth order, denoted by (W)MGT4(n_1, n_2, n_3, n_4, n_5), is described in four steps as follows.

1. Perform n_1 relaxation sweeps on Eq. (1.1.13) ($u^h \leftarrow$ resulting approximation).

2. Perform n_3 relaxation sweeps on the fourth order extrapolation equation (2.3.7) formed on grid $2h$ with $u^{2h} = I_h^{2h}u^h$ as the initial approximation. Here, I_h^{2h} represents injection. Then use the resulting approximation to initialize the application of MG2(n_4, n_5) to (2.3.7) ($u^h \leftarrow$ the final approximation).

3. Correct u^h by

$$u^h \leftarrow u^h + I_{2h}^h\left(u^{2h} - I_h^{2h}u^h\right). \tag{2.3.17}$$

4. Perform n_2 relaxation sweeps on Eq. (1.1.13) ($u^h \leftarrow$ resulting approximation).

The correction (2.3.17) is made in order to preserve the high frequency information held in the approximation u^h which could not be approximated on the coarse grid. This information would be lost if the approximation u^h were simply replaced by the interpolation of the extrapolated function u^{2h}. The correction (2.3.17) corresponds to the Full Approximation Scheme multi-grid algorithm (see [3]). The second order relaxation in step 4 is designed to reduce high frequency components in the error on grid h which would adversely effect a second extrapolation. Experiments support the use of this step, although the disparity between the two solution processes, higher order extrapolation and second order relaxation, makes an analysis of this method difficult (see [8]). The approximation obtained in step 3 will be considered as the final approximation of the fourth order τ-extrapolation method.

(W)MGT6(n_1, n_2, n_3, n_4, n_5, n_6), the method of (weighted) τ-extrapolation to sixth order, is described in six steps as follows:

1. Perform n_1 relaxation sweeps on Eq. (1.1.13) ($u^h \leftarrow$ resulting approximation).

2. Perform n_3 relaxation sweeps on the fourth order extrapolation equation (2.3.7) formed on grid $2h$ with $u^{2h} = I_h^{2h} u^h$ as the initial approximation. Here I_h^{2h} is injection ($u^{2h} \leftarrow$ resulting approximation).

3. Apply MG2(n_5, n_6) to the sixth order extrapolation equation (2.3.15) formed on grid $4h$ with $u^{4h} = I_{2h}^{4h} u^{2h}$ as the initial approximation. Here I_{2h}^{4h} is injection ($u^{4h} \leftarrow$ resulting approximation).

4. Correct u^{2h} by

$$u^{2h} \leftarrow u^{2h} + I_{4h}^{2h}\left(u^{4h} - I_{2h}^{4h} u^{2h}\right), \tag{2.3.18}$$

and perform n_4 relaxation sweeps on the fourth order extrapolation equation formed in step 2 ($u^{2h} \leftarrow$ resulting approximation).

5. Correct u^h by

$$u^h \leftarrow u^h + I_{2h}^h\left(u^{2h} - I_h^{2h} u^h\right). \tag{2.3.19}$$

6. Perform n_2 relaxation sweeps on Eq. (1.1.13) ($u^h \leftarrow$ resulting approximation).

The resulting approximation obtained in step 5 will be considered the final approximation of the sixth order τ-extrapolation method.

Chapter 3. Mode Analysis and Numerical Results

Numerical experiments were run on three problems to test the relative efficiency of the four multi-grid methods considered here and the accuracy of the various discretizations described in Chapter 1. The convergence rate estimates obtained from the local mode analysis are compared with the rates observed in the numerical experiments. For each method, comparisons are made for various choices of restriction, interpolation, relaxation and cycle structure.

3.1. Local Mode Analysis. In this section the convergence rate of the two-grid MG2, MGH and MGD methods is estimated through a local mode analysis (see [3]). These estimates will then be compared in the following sections to the global convergence rates that were observed in the experiments performed here. For completeness, the description of the local mode analysis of the method MG2 given by Stüben [13] is briefly outlined below. The methods MGH and MGD will be incorporated into this discussion.

The local convergence of a multi-grid cycle, that is, the behavior of the error in the interior of the domain, is analyzed by neglecting the boundaries of the domain and the boundary conditions of the original problem. The restriction, interpolation and finite difference operators are then viewed as acting on the infinite grids $\bar{\Omega}^h = \{\nu h: \nu \in Z^2\}$ and $\bar{\Omega}^{2h} = \{\nu 2h: \nu \in Z^2\}$, where the extensions of these operators are denoted by $\bar{I}_h^{2h}$, $\bar{I}_{2h}^h$ and $\bar{L}^h$, respectively. The extensions of the multi-grid iteration matrices that were described in Chapters 1 and 2 are then analyzed on the complex spaces:

$$\varepsilon^h = \left\langle e^{i\theta \cdot X/h}: |\theta| \leqslant \pi \right\rangle, \tag{3.1.1}$$

$$\varepsilon^{2h} = \left\langle e^{i\theta \cdot X/h}: |\theta| \leqslant \pi/2 \right\rangle, \tag{3.1.2}$$

where $\theta = (\theta_1, \theta_2) \in R^2$ and $|\theta| = \max(|\theta_1|, |\theta_2|)$. Specifically, these matrices are studied on four-dimensional subspaces, the so-called harmonic subspaces, given by

$$H(\theta) = \left\langle e_{\alpha\beta}^h \equiv e^{i\theta_{\alpha\beta} \cdot X/h}: \alpha, \beta = 0, 1 \right\rangle, \tag{3.1.3}$$

where $\theta = (\theta_1, \theta_2)$, $|\theta| \leqslant \pi/2$ and

$$\theta_{\alpha\beta} = (\theta_{1(\alpha)}, \theta_{2(\beta)}) = (\theta_1 - \alpha \cdot \operatorname{sign}(\theta_1) \cdot \pi, \theta_2 - \beta \cdot \operatorname{sign}(\theta_2) \cdot \pi).$$

To each harmonic subspace, $H(\theta)$, there is a uniquely determined Fourier mode in ε^{2h} given by $e_{00}^{2h} = e_{00}^{h}|_{\bar{\Omega}^{2h}}$.

The Fourier modes in $H(\theta)$ are mapped into the one-dimensional subspace $\langle e_{00}^{2h} \rangle \subset \varepsilon^{2h}$ under the restriction operator according to

$$\bar{I}_h^{2h} e_{\alpha\beta}^h = r_{\alpha\beta} e_{00}^{2h}, \qquad \alpha, \beta = 0, 1. \tag{3.1.4}$$

Thus, the restriction of $\bar{I}_h^{2h}$ to $H(\theta)$ can be represented by the 4×1 matrix

$$[r_\theta] \equiv (r_{00}\, r_{11}\, r_{10}\, r_{01}). \tag{3.1.5}$$

The constants $r_{\alpha\beta}$, $\alpha, \beta = 1, 0$, for the injection and full weighting operators are given by

$$(1)_h\colon r_{\alpha\beta} = 1,$$

$$\frac{1}{16}\begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix}_h : r_{\alpha\beta} = \frac{1}{4}\big(1 + \cos(\theta_{1(\alpha)}) + \cos(\theta_{2(\beta)}) + \cos(\theta_{1(\alpha)})\cos(\theta_{2(\beta)})\big)$$

$$= \begin{cases} 1 - O(\theta_1^2 + \theta_2^2), & \alpha + \beta > 0, \\ O(\theta_1^2 + \theta_2^2), & \alpha = \beta = 0. \end{cases}$$

The above description shows that the full weighting operator is more effective in dampening the high frequency components which have no representation on $\bar{\Omega}^{2h}$. It is important to note, however, that in practice, injection of the residuals involves the calculation of the residuals at only one quarter of the points whereas full weighting involves the calculation of the residuals at every point of the fine grid.

The Fourier mode $e_{00}^{2h} \in \varepsilon^{2h}$ is mapped into the corresponding four-dimensional harmonic subspace $H(\theta) \subset \varepsilon^h$ under the interpolation operators according to

$$\bar{I}_{2h}^h e_{00}^{2h} = S_{00} e_{00}^h + S_{11} e_{11}^h + S_{10} e_{10}^h + S_{01} e_{01}^h. \tag{3.1.6}$$

Thus, $\bar{I}_{2h}^h$ restricted to the subspace $\langle e_{00}^{2h} \rangle$ can be represented in matrix form by

$$[s_\theta] = \begin{pmatrix} S_{00} \\ S_{11} \\ S_{10} \\ S_{01} \end{pmatrix}. \tag{3.1.7}$$

The constants in the matrix $[s_\theta]$ are of the form

$$\begin{aligned} S_{00} &= (1 - S(\theta_1))(1 - S(\theta_2)), & S_{11} &= S(\theta_1)S(\theta_2), \\ S_{10} &= (1 - S(\theta_1))S(\theta_2), & S_{01} &= S(\theta_1)(1 - S(\theta_2)), \end{aligned} \tag{3.1.8}$$

where, for each interpolation operator, $S(\theta_i)$, $i = 1, 2$, is given by

$$\text{Linear:} \quad S(\theta_i) = \frac{1}{2}(1 - \cos(\theta_i)),$$

$$\text{Cubic:} \quad S(\theta_i) = \frac{1}{16}(8 - 9\cos(\theta_i) + \cos(3\theta_i)),$$

$$\text{Quintic:} \quad S(\theta_i) = \frac{1}{256}(128 - 150\cos(\theta_i) + 25\cos(3\theta_i) - 3\cos(5\theta_i)).$$

For the extended finite difference operators, the Fourier modes are eigenfunctions:

$$\bar{L}^h e^{i\theta \cdot X/h} = \lambda(\theta, h) e^{i\theta \cdot X/h}, \qquad |\theta| \leqslant \pi, \tag{3.1.9}$$

$$\bar{L}^{2h} e^{i2\theta X/2h} = \lambda(2\theta, 2h) e^{i2\theta X/2h}, \quad |\theta| \leqslant \pi/2. \tag{3.1.10}$$

The matrix representations for the operators $\bar{L}^h$ and $\bar{L}^{2h}$ acting on the subspaces $H(\theta)$ and $\langle e_{00}^{2h} \rangle$, respectively, are given by

$$[\lambda_\theta^h] = \begin{bmatrix} \lambda(\theta_{00}, h) & & & \\ & \lambda(\theta_{11}, h) & & 0 \\ & & \lambda(\theta_{10}, h) & \\ & 0 & & \\ & & & \lambda(\theta_{01}, h) \end{bmatrix} \tag{3.1.11}$$

and

$$[\lambda_{2\theta}^{2h}] = [\lambda(2\theta, 2h)]. \tag{3.1.12}$$

The eigenvalues, $\lambda(\theta, h)$, for the four extended finite difference operators are given by

$$\bar{L}_2^h: \quad \lambda(\theta, h) = 2(\cos(\theta_1) + \cos(\theta_2) - 2)/h^2,$$

$$\bar{L}_4^h: \quad \lambda(\theta, h) = \frac{1}{6}(-(\cos(2\theta_1) + \cos(2\theta_2)) + 16(\cos(\theta_1) + \cos(\theta_2)) - 30)/h^2,$$

$$\bar{L}_6^h: \quad \lambda(\theta, h) = \frac{1}{90}(2(\cos(3\theta_1) + \cos(3\theta_2)) - 27(\cos(2\theta_1) + \cos(2\theta_2)) + 270(\cos(\theta_1) + \cos(\theta_2)) - 490)/h^2,$$

$$\bar{L}_M^h: \quad \lambda(\theta, h) = \frac{1}{3}(4(\cos(\theta_1) + \cos(\theta_2)) + 2\cos(\theta_1)\cos(\theta_2) - 10)/h^2.$$

The relaxation procedures are formally extended to $\bar{\Omega}^h$. First, let $\bar{R}^h$ denote the amplification matrix corresponding to the extended LEX relaxation method using the difference operator $\bar{L}_2^h$. Let the functions V^h and $\hat{V}^h \in \varepsilon^h$ satisfy

$$\hat{V}^h = \bar{R}^h V^h. \tag{3.1.13}$$

At each point $(x, y) \in \bar{\Omega}^h$, V^h and $\hat{V}^h$ are related as follows

$$V^h(x + h, y) + V^h(x, y + h) + \hat{V}^h(x - h, y) + \hat{V}^h(x, y - h) - 4\hat{V}^h(x, y) = 0. \tag{3.1.14}$$

Setting $V^h(X) = e^{i\theta \cdot X/h}$, then relation (3.1.14) becomes

$$\hat{V}^h = \mu(\theta) V^h, \tag{3.1.15}$$

where the complex number $\mu(\theta)$ is called the convergence factor corresponding to the frequency θ. It follows from (3.1.15) that the restriction of $\bar{R}^h$ to the subspace $H(\theta)$ is represented by the 4×4 diagonal matrix

$$[\mu_\theta^h] = \begin{bmatrix} \mu(\theta_{00}) & & & \\ & \mu(\theta_{11}) & & 0 \\ & & \mu(\theta_{10}) & \\ & 0 & & \\ & & & \mu(\theta_{01}) \end{bmatrix}. \tag{3.1.16}$$

For the four extended finite difference operators, the convergence factor, $\mu(\theta)$, of the LEX relaxation method is given by

$$\bar{L}_2^h\colon \mu(\theta) = \frac{-e^{i\theta_1} - e^{i\theta_2}}{e^{-i\theta_1} + e^{-i\theta_2} - 4},$$

$$\bar{L}_4^h\colon \mu(\theta) = \frac{-16(e^{i\theta_1} + e^{i\theta_2}) + (e^{i2\theta_1} + e^{i2\theta_2})}{-60 + 16(e^{-i\theta_1} + e^{-i\theta_2}) - (e^{-i2\theta_1} + e^{-i2\theta_2})},$$

$$\bar{L}_6^h\colon \mu(\theta) = \frac{-270(e^{i\theta_1} + e^{i\theta_2}) + 27(e^{i2\theta_1} + e^{i2\theta_2}) - 2(e^{i3\theta_1} + e^{i3\theta_2})}{-980 + 270(e^{-i\theta_1} + e^{-i\theta_2}) - 27(e^{-i2\theta_1} + e^{-i2\theta_2}) + 2(e^{-i3\theta_1} + e^{-i3\theta_2})},$$

$$\bar{L}_M^h\colon \mu(\theta) = \frac{-4(e^{i\theta_1} + e^{i\theta_2}) - e^{i(\theta_1+\theta_2)} - e^{i(\theta_1-\theta_2)}}{-20 + 4(e^{-i\theta_1} + e^{-i\theta_2}) + e^{-i(\theta_1+\theta_2)} + e^{-i(\theta_1-\theta_2)}}.$$

The smoothing rate, μ, of the relaxation process is defined as

$$\mu = \sup_{\pi/2 \leqslant |\theta| \leqslant \pi} |\mu(\theta)|. \tag{3.1.17}$$

Thus, μ is a measure of the rate at which the high frequency components are reduced by the relaxation method. $(-(\log \mu)^{-1}$ relaxation sweeps are needed to reduce all the high frequency components by an order of magnitude.) The following smoothing rates were computed for the LEX relaxation using the four extended finite difference operators

$$\begin{aligned} L_2^h\colon &\quad \mu = .5, \\ L_4^h\colon &\quad \mu = .534, \\ L_6^h\colon &\quad \mu = .553, \\ L_M^h\colon &\quad \mu = .464. \end{aligned}$$

A complete relaxation sweep of the extended even/odd CHJ relaxation method, starting with an initial function $V^h \in \varepsilon^h$, can be written as

$$\tilde{V}^h = \bar{R}_{\text{even}}^h V^h, \tag{3.1.18}$$

$$\hat{V}^h = \bar{R}_{\text{odd}}^h \tilde{V}^h, \tag{3.1.19}$$

where $\bar{R}_{\text{even}}^h$ and $\bar{R}_{\text{odd}}^h$ are the amplification matrices corresponding to the partial Jacobi-like relaxation sweeps over the subsets $\bar{\Omega}_{\text{even}}^h$ and $\bar{\Omega}_{\text{odd}}^h$, respectively. Here, $\bar{\Omega}_{\text{even}}^h$ is defined to be the set $\{(x, y) \in \bar{\Omega}^h\colon (x, y)/h \equiv 0 \pmod 2\}$ and $\bar{\Omega}_{\text{odd}}^h = \bar{\Omega}^h - \bar{\Omega}_{\text{even}}^h$. Using the operator $\bar{L}_2^h$, the functions V^h and $\tilde{V}^h$ in (3.1.18) are related by

$$\begin{aligned} &V^h(x+h, y) + V^h(x-h, y) + V^h(x, y+h) \\ &\qquad + V^h(x, y-h) - 4\tilde{V}^h(x, y) = 0, \qquad (x, y) \in \bar{\Omega}_{\text{even}}^h, \\ &\tilde{V}^h(x, y) = V^h(x, y), \qquad (x, y) \in \bar{\Omega}_{\text{odd}}^h. \end{aligned} \tag{3.1.20}$$

Setting $V^h(X) = e^{i\theta X/h}$, relation (3.1.20) becomes

$$\begin{aligned} &\tilde{V}^h(x, y) = \mu(\theta) V^h(x, y), \quad (x, y) \in \bar{\Omega}_{\text{even}}^h, \\ &\tilde{V}^h(x, y) = V^h(x, y), \qquad (x, y) \in \bar{\Omega}_{\text{odd}}^h, \end{aligned} \tag{3.1.21}$$

where $\mu(\theta)$ is the convergence factor corresponding to the Jacobi relaxation method. By a similar argument, it can be shown that the functions $\hat{V}^h$ and $\tilde{V}^h$ in (3.1.19) are related by

$$\begin{aligned} \hat{V}^h(x, y) &= \tilde{V}^h(x, y), && (x, y) \in \bar{\Omega}^h_{\text{even}}, \\ \hat{V}^h(x, y) &= \mu(\theta)\tilde{V}^h(x, y), && (x, y) \in \bar{\Omega}^h_{\text{odd}}, \end{aligned} \tag{3.1.22}$$

where $\mu(\theta)$ is defined as in (3.1.21). Using the relations (3.1.21) and (3.1.22), Stüben shows that the restriction of $\bar{R}^h_{\text{even}}$ and $\bar{R}^h_{\text{odd}}$ to the subspace $H(\theta)$ can be represented by the four by four matrices

$$[\mu^h_{\theta,\text{even}}] = \frac{1}{2}\begin{bmatrix} \mu_{00}+1 & -(\mu_{11}-1) & 0 & 0 \\ -(\mu_{00}-1) & \mu_{11}+1 & 0 & 0 \\ 0 & 0 & \mu_{10}+1 & -(\mu_{01}-1) \\ 0 & 0 & -(\mu_{10}-1) & \mu_{01}+1 \end{bmatrix}, \tag{3.1.23}$$

$$[\mu^h_{\theta,\text{odd}}] = \frac{1}{2}\begin{bmatrix} \mu_{00}+1 & \mu_{11}-1 & 0 & 0 \\ \mu_{00}-1 & \mu_{11}+1 & 0 & 0 \\ 0 & 0 & \mu_{10}+1 & \mu_{01}-1 \\ 0 & 0 & \mu_{10}-1 & \mu_{01}+1 \end{bmatrix}, \tag{3.1.24}$$

where $\mu_{\alpha\beta} = \mu(\theta_{\alpha\beta})$. Thus the matrix $[\mu^h_\theta]$ describing the effect of a complete even/odd Checkerboard relaxation sweep on the subspace $H(\theta)$ is given by

$$[\mu^h_\theta] = [\mu^h_{\theta,\text{odd}}][\mu^h_{\theta,\text{even}}]. \tag{3.1.25}$$

Since the matrices (3.1.23) and (3.1.24) couple the high and low frequency components in $H(\theta)$, a smoothing rate for the CHJ relaxation method analogous to that defined in (3.1.17) is not possible. For the four extended finite difference operators, the convergence factor $\mu(\theta)$ of the Jacobi relaxation method is given by

$$\bar{L}^h_2\colon \quad \mu(\theta) = \frac{1}{2}(\cos(\theta_1) + \cos(\theta_2)),$$

$$\bar{L}^h_4\colon \quad \mu(\theta) = \frac{1}{30}(16(\cos(\theta_1) + \cos(\theta_2)) - \cos(2\theta_1) - \cos(2\theta_2)),$$

$$\bar{L}^h_6\colon \quad \mu(\theta) = \frac{1}{490}(270(\cos(\theta_1) + \cos(\theta_2)) - 27(\cos(2\theta_1) + \cos(2\theta_2)) + 2(\cos(3\theta_1) + \cos(3\theta_2))),$$

$$\bar{L}^h_M\colon \quad \mu(\theta) = \frac{1}{10}(4(\cos(\theta_1) + \cos(\theta_2)) + 2\cos(\theta_1)\cos(\theta_2)).$$

It follows from the above arguments that the extended multi-grid iteration matrices corresponding to the methods MG2, MGH and MGD can be represented on each of the subspaces $H(\theta)$, $|\theta| \leqslant \pi/2$, by the following 4×4 matrices

$$[\text{MG2}_\theta] = [\mu_\theta]^{n_2}\big(I - [s_\theta][\lambda^{2h}_{2\theta}]^{-1}[r_\theta][\lambda^h_\theta]\big)[\mu_\theta]^{n_1}, \tag{3.1.26}$$

$$[\text{MGH}_\theta] = [\hat{\mu}_\theta]^{n_2}\big(I - [s_\theta][\lambda^{2h}_{2\theta}][r_\theta][\hat{\lambda}^h_\theta]\big)[\hat{\mu}_\theta]^{n_1} \tag{3.1.27}$$

and

$$[\text{MGD}_\theta] = I - (I - [\text{MG2}_\theta])[\lambda^h_\theta]^{-1}[\hat{\lambda}^h_\theta]. \tag{3.1.28}$$

The symbol " $\hat{}$ " indicates matrices defined by one of the extended higher order difference operators. The corresponding extended iteration matrices for the residual (see (1.3.11)) are represented on each subspace $H(\theta)$, $|\theta| \leqslant \pi/2$, by the following 4×4 matrices

$$[\mathrm{MG2}_\theta]_R = [\lambda_\theta^h][\mathrm{MG2}_\theta][\lambda_\theta^h]^{-1}, \tag{3.1.29}$$

$$[\mathrm{MGH}_\theta]_R = [\hat{\lambda}_\theta^h][\mathrm{MGH}_\theta][\hat{\lambda}_\theta^h]^{-1}, \tag{3.1.30}$$

$$[\mathrm{MGD}_\theta]_R = [\hat{\lambda}_\theta^h][\mathrm{MGD}_\theta][\hat{\lambda}_\theta^h]^{-1}. \tag{3.1.31}$$

Since the performance of the multi-grid methods in the first few iterations is of interest here, the spectral norm of the extended iteration matrices $\bar{M}^h$ will be used for comparison. The spectral norm of $\bar{M}^h$ is defined here by

$$\|\bar{M}^h\|_2 = \sup\{\|[m_\theta]\|_2 : |\theta| \leqslant \pi/2\}, \tag{3.1.32}$$

where $[m_\theta]$ is the 4×4 matrix which represents the restriction of $\bar{M}^h$ to the subspace $H(\theta)$. In the experiments reported here the discrete approximation to $\|\bar{M}^h\|_2$ given by

$$\sigma(\bar{M}^h) = \max\{\|m_{\theta_{i,j}}\|_2 : \theta_{i,j} = h\pi(i, j), i, j \in Z, |\theta_{i,j}| \leqslant \pi/2\}$$

is used. $\sigma(\bar{M}^h)$ will be referred to as the σ-norm of the matrix $\bar{M}^h$.

The σ-norm of the extended iteration matrices corresponding to the following three algorithms is presented in Table 1.

A1–MG2; $n_1 = 2, n_2 = 1$, full weighting restriction, linear interpolation.
A2–MGH; $n_1 = 3, n_2 = 1$, full weighting restriction, cubic interpolation.
A3-MGD; $n_1 = 2, n_2 = 1$, full weighting restriction, cubic interpolation.

The convergence rates were calculated for the case $h = 1/32$. These results will be compared to the actual convergence rates of the corresponding algorithms in the experiments in Section 3.2.

The rates given in Table 1 are not the best possible since algorithms A1, A2 and A3 were designed only for the purpose of comparison. The rates for algorithm A2 are better than those for algorithm A1 due to the extra relaxation and the higher order interpolation used in A2. It was observed in the experiments that the performance of the MGH algorithms generally improved with the additional relaxation and the use of cubic interpolation whereas the efficiency of the MG2 algorithm deteriorated with these additions.

The extended iteration matrix for MGD is given by

$$I - (\bar{L}_2^h)^{-1}\bar{\mathscr{L}}^h + \bar{M}^h(\bar{L}_2^h)^{-1}\bar{\mathscr{L}}^h, \tag{3.1.33}$$

where $\bar{\mathscr{L}}^h$ denotes one of the operators $\bar{L}_4^h$, $\bar{L}_6^h$, or $\bar{L}_M^h$, and $\bar{M}^h$ is the extended iteration matrix for MG2. The term $I - (\bar{L}_2^h)^{-1}\bar{\mathscr{L}}^h$ occurring in (3.1.33) corresponds to the outer loop IDEC iteration of MGD. The σ-norm of $I - (\bar{L}_2^h)^{-1}\bar{\mathscr{L}}^h$ for $\bar{\mathscr{L}}^h = \bar{L}_4^h$, $\bar{L}_6^h$ and $\bar{L}_M^h$ was calculated to be .33, .51 and .33, respectively. Comparing these results with the results for algorithm A3 in Table 1, it is clear that the convergence rate of MGD is essentially determined by the IDEC iteration.

TABLE 1
Local mode analysis estimates

Algorithm	A1		A2						A3					
Operator	$\bar{L}_2^h$		$\bar{L}_4^h$		$\bar{L}_6^h$		$\bar{L}_M^h$		$\bar{L}_4^h$		$\bar{L}_6^h$		$\bar{L}_M^h$	
Relaxation Type	LEX	CHJ	LEX	CHJ	LEX	CHJ	LEX	CHJ	LEX	CHJ	LEX	CHJ	LEX	CHJ
σ-Norm, Error	.119	.080	.0627	.044	.070	.0481	.0452	.033	.38	.33	.57	.51	.33	.33
σ-Norm, Residual	.123	.080	.0632	.044	.076	.0476	.0454	.025	.38	.33	.57	.51	.33	.33

3.2. Numerical Results. Three problems are presented in this section. They are discretized using the finite differences described in Section 1.1. For each problem, the discretization errors are presented and several algorithms based on the methods MG2, MGH, MGD and MGT are then described. The numerical performance of these algorithms, implemented on each of the three problems, is compared. Finally, the methods of Richardson extrapolation and τ-extrapolation are compared in the multi-grid context.

The three problems were determined by preselecting a solution to Eq. (1) and defining the right-hand side and boundary conditions accordingly.

Problem 1. This problem is well suited for higher order approximations on the grids considered here.

SOLUTION: $U(x, y) = \mathrm{SIN}(\pi x)\mathrm{SIN}(2\pi y)$.

Problem 2. The solution to this problem has high oscillations relative to the mesh spacing of the grids considered here (on grid $h = 1/32$, the solution contains an average of 6 grid points per wavelength).

SOLUTION: $U(x, y) = \mathrm{SIN}(17(x + y))$.

Problem 3. The solution to this problem has a jump discontinuity in the 6th derivatives along the parabola given by

$$T(x, y) = y - x^2 + x - .75 = 0.$$

It is well suited for second and fourth order approximations, although the solution lacks the differentiability that, in theory, is needed for a sixth order approximation.

SOLUTION: $U(x, y) = \mathrm{EXP}(C(x, y)T^6(x, y))$ where $C(x, y)$ is defined by

$$C(x, y) = \begin{cases} 1, & T(x, y) > 0, \\ -1, & T(x, y) \leqslant 0. \end{cases}$$

In the experiments reported here, the discrete analog of the continuous Euclidean norm is used to measure the error, $E^h \equiv U - u^h$, where U is the continuous solution projected onto Ω^h and u^h is a given approximation to U. Thus, the error norm, denoted by σ^h, is given by

$$\sigma^h = \|E^h\|_2^h = \left[\sum_{(x,y)\in\Omega^h} \left(E^h(x, y)\right)^2 h^2 \right]^{1/2}.$$

The norm

$$\delta^h = \left[\sum_{(x,y)\in\Omega^h} \left(\left(\delta_x^h(x, y)\right)^2 + \left(\delta_y^h(x, y)\right)^2 \right) h^2 \right]^{1/2},$$

where

$$\delta_x^h(x, y) = \big(E^h(x + h, y) - E^h(x, y)\big)/h$$

and

$$\delta_y^h(x, y) = \big(E^h(x, y + h) - E^h(x, y)\big)/h,$$

will be used to test how well the derivatives of U are approximated by u^h. The norms, σ^h and δ^h, are defined so that, for any two grids, the norm defined on one grid is compatible with the corresponding norm defined on the next.

For Problems 1, 2, and 3, the converged solutions corresponding to each of the finite difference discretizations were obtained by performing several cycles of MG2, MGH or MGD on grid $1/32$ and grid $1/64$. In Graphs 1, 2, and 3 the negative log of the error norm, σ^h, measured at convergence on grid $1/32$ and grid $1/64$, is plotted for each discretization (solid lines). The negative log of the norm δ^h, measured at convergence on grid $1/32$ and $1/64$, is also plotted for each discretization (dotted lines).

The discretizations are ordered in the graphs according to accuracy of solution, Mehrstellen Verfahren consistently giving the most accurate solution in both the fourth order and sixth order cases. The accuracy generally deteriorated when lower order differences were used near the boundaries as seen from $L_{4,2}^h$ and $L_{6,4}^h$. The solutions obtained by L_4^h and $L_{4,M}^h$ were equally as accurate, whereas the solutions obtained by $L_{6,M}^h$ were appreciably more accurate than those for L_6^h. In the second order case, the weighting of the function F onto the grid (Eq. (1.3.4)) produced a more accurate discretization than the usual injection (Eq. (1.3.3)) for the three problems considered here.

In the algorithms considered in the experiments, the error, E^h, of the approximation is related to the amount of work, W^h, used to obtain it by the expression $E^h = C(W^h)^{-r}$. The constants C and $r > 0$ depend on the method, the finite difference discretization and the multi-grid processes (interpolation, restriction and relaxation), but are independent of h. Here, the work was measured by a cumulative operation count which included interpolations and restrictions as well as relaxations. One multiplication was counted as two additions. Therefore, the graphs $-\log(\sigma^h)$ versus $\log(W^h)$ and $-\log(\delta^h)$ versus $\log(W^h)$ will be used to compare the four multi-grid methods as well as the effects of varying the interpolations, restrictions, etc., within each method.

A preliminary set of experiments was run for each of the higher order methods using the various choices of interpolation, restriction, number of relaxations, discretizations, multi-grid cycling and FMG cycling. From these experiments, a set of multi-grid processes which produced the best general performance in all three higher order methods was selected for comparison purposes. Unless otherwise stated, each method uses the following:

1. The W-cycle (see Section 1.3);
2. Full weighting of the residuals;
3. Cubic interpolation of the approximation to a new level (FMG);
4. Linear interpolation in the coarse grid correction step for the method MG2; and

5. Cubic interpolation in the coarse grid correction step for the higher order methods.

It was found that the performance of the fourth and sixth order Mehrstellen Verfahren discretizations in the methods MGH and MGD were nearly equivalent. Therefore, only the sixth order Mehrstellen Verfahren discretization was considered in these experiments. The following four FMG cycles were used:

h	FMG_a Order of Approx.	FMG_b Order of Approx.	FMG_c Order of Approx.	FMG_d Order of Approx.
$\frac{1}{4}$	2	2	2	2
$\frac{1}{8}$	2	2	2	2
$\frac{1}{16}$	2	2	2	$2 \rightarrow 6$
$\frac{1}{32}$	2	$2 \rightarrow 4$	$2 \rightarrow 6$	$2 \rightarrow 6$
$\frac{1}{64}$	2	$4 \rightarrow 6$	6	6.

The following nine algorithms using both LEX and CHJ relaxation were applied to Problems 1, 2, and 3. MG2(2, 1) was used in the second order steps of FMG_b, FMG_c and FMG_d.

Alg 1: MG2(2, 1); FMG_a
Alg 2: WMG2(2, 1); FMG_a
Alg 3: MGH(3, 1, 2, 1); FMG_b; $(L^h_{4,2}, L^h_{6,4})$
Alg 4: MGH(3, 1, 2, 1); FMG_b; $(L^h_{4,M}, L^h_{6,M})$
Alg 5: MGH(3, 1, 2, 1); FMG_c; (L^h_M–sixth order discretization)
Alg 6: MGD(2, 1, 2); FMG_b; (L^h_4, L^h_6)
Alg 7: MGD(2, 1, 2); FMG_c; (L^h_M–sixth order discretization)
Alg 8: MGT4(1, 0, 1, 2, 1);
MGT6(1, 0, 3, 0, 3, 3); V-cycle, FMG_b
Alg 9: WMGT4(1, 0, 1, 2, 1);
WMGT6(1, 0, 3, 0, 3, 3); V-cycle, FMG_b.

All of the graphs, unless otherwise stated, start at the second order solution obtained on grid 1/32. The subsequent points on each graph represent measurements taken at the completion of each FMG step (i.e., just before refining the grid or increasing the order of approximation). In the first set of graphs, the MGH algorithms (i.e., Algorithms 3–5) and the second order Algorithm 1 are compared. In the graphs, LEX and CHJ relaxation will be denoted by the symbols L and C, respectively.

Experiments were also run on the algorithms

Alg 3a: MGH(3, 1, 2, 1); FMG_c; $(L^h_{6,4})$

and

Alg 4a: MGH(3, 1, 2, 1); FMG_c; $(L^h_{6,M})$.

The results of these experiments showed that both Alg 3a and Alg 4a were less efficient than the corresponding algorithms Alg 3 and Alg 4.

TABLE 2
Residual reduction factors for MGH

		Alg 3				Alg 4				Alg 5	
		$L^h_{4,2}$		$L^h_{6,4}$		$L^h_{4,M}$		$L^h_{6,M}$		L^h_M	
Problem	Grid	LEX	CHJ	LEX	CHJ	LEX	CHJ	LEX	CHJ	LEX	CHJ
LMA	1/32	.063	.044	.076	.048	.063	.044	.077	.048	.045	.033
1	1/32	.01	.077			.016	.01			.007	.026
1	1/64	.004	.005	.055	.011	.002	.001	.066	.028	.001	.002
2	1/32	.042	.027			.045	.021			.038	.029
2	1/64	.011	.009	.05	.072	.008	.003	.037	.012	.004	.002
3	1/32	.046	.067			.062	.056			.03	.038
3	1/64	.023	.026	.029	.071	.01	.006	.062	.016	.008	.003

The rates in Table 2 generally stayed within the predicted values (LMA). The exceptions occurred in algorithms which used CHJ relaxation.

Referring to Graphs 1, 2 and 3, it can be seen that the efficiency of the MGH algorithms in Graphs 4 through 9 is related to the accuracy of the discretizations used. In general, the algorithms with CHJ relaxation performed somewhat more efficiently than with LEX relaxation. The first derivatives of the solution were approximated in the MGH algorithms equally as well as the solution itself. In the experiments, the residual reduction factors were calculated. These and the corresponding residual reduction factors predicted by the Local Mode Analysis (LMA) are presented in Table 2.

In Graphs 10–15, the MGD algorithms, Alg 6 and Alg 7, using both LEX and CHJ relaxation and Alg 1 with CHJ relaxation, are compared.

The MGD algorithms using CHJ relaxation initially produced better results. However, in subsequent iterations the efficiency of these algorithms degenerated. This is seen particularly in the derivative graphs, Graphs 13–15. It was shown in Section 3.1 that the outer loop IDEC iteration dominates the rate of convergence of the MGD method. Thus, high frequency information in the error on the fine grid will tend to slow down the convergence of the MGD algorithms to that determined by IDEC. CHJ relaxation leaves highly oscillatory residuals which are reflected in the error as well. In Table 3, the residual reduction rates for the two MGD algorithms are presented. It is seen there that the rates for the algorithms using CHJ relaxation attain and sometimes surpass the predicted rates.

In Graphs 16–21, the τ-extrapolation algorithms, Alg 8 and Alg 9, using both LEX and CHJ relaxation and algorithms Alg 1 and Alg 2 with CHJ relaxation are compared.

TABLE 3
Residual reduction factors for the MGD algorithms

		Alg 6				Alg 7	
		L_4^h		L_6^h		L_M^h	
Problem	Grid	LEX	CHJ	LEX	CHJ	LEX	CHJ
LMA	1/32	.38	.33	.57	.51	.33	.33
1	1/32	.073	.028			.012	.013
1	1/64	.24	.33	.34	.52	.071	.33
2	1/32	.055	.046			.069	.039
2	1/64	.23	.34	.31	.75	.067	.30
3	1/32	.11	.075			.044	.044
3	1/64	.25	.33	.28	.82	.065	.23

Algorithms Alg 2 and Alg 9, based on the more accurate weighted second order discretization (1.3.4), produced consistently better approximations than the corresponding algorithms, Alg 1 and Alg 8. Extrapolating to fourth order, the algorithms produced nearly the same results when either CHJ or LEX relaxation were used. In the sixth order case, a substantial improvement in the accuracy of the approximation is observed after the extrapolation in the algorithms using LEX relaxation. The graphs show that the extrapolation to sixth order in the algorithms using CHJ relaxation produce an approximation still of order four. This is reflected even more strongly in the derivative graphs. This behavior is similar to that observed in the MGD algorithms.

In the next set of graphs, the previous algorithms based on the methods MGH, MGD and MGT are compared. Included in these graphs is the following, nearly optimal, second order algorithm:

Alg 1a: MG2(1, 1); FMG_a; injection (by $\frac{1}{2}$) of residuals; V-cycling; CHJ.

In Graphs 22–27, the performance of the three higher order algorithms on the three problems is seen to be nearly equivalent. In terms of efficiency, the three higher order multi-grid algorithms, MGH, MGD and MGT are clearly superior to the second order multi-grid algorithm MG2.

The standard algorithms, Alg 1 through Alg 9, were modified and experiments were run on Problem 1 to test the effects of alternate choices of residual transfer, interpolation, numbers of relaxation and/or FMG cycling. Results of the following MGH algorithms are plotted in Graphs 28–31.

Alg H1: Alg 4; LEX; injection of residuals,

Alg H2: Alg 4; CHJ; injection (by $\frac{1}{2}$) of residuals,

Alg H3: Alg H1; quintic interpolation of high order approximation,

Alg H4: Alg 5; LEX; quintic interpolation of high order approximation,
Alg H5: Alg 5; CHJ; quintic interpolation of high order approximation,
Alg H6: Alg 5; CHJ; FMG_d,
Alg H7: Alg H6; quintic interpolation of high order approximation.

In Graphs 28 and 29, the algorithms using injection of the residuals are seen to be less efficient than Alg 4; LEX, especially Alg H2 where CHJ relaxation was used. The accuracy improved considerably when quintic interpolation of the approximation was introduced (Alg H3). The marked increase in accuracy obtained by introducing quintic interpolation is seen again in Graphs 30 and 31 (compare Alg 5; CHJ to Alg H5 and Alg H6 to Alg H7). Comparing Alg H6 to Alg 5; CHJ and Alg H7 to Alg H5, it is seen that no appreciable gain in efficiency is obtained by introducing higher orders earlier in the FMG algorithm.

The first two MGD algorithms below are designed to test the effect of the second order smoothing step used on grid $1/64$ prior to forming the higher order defect. The next four algorithms investigate the performance of MGD when less expensive multi-grid procedures are used.

Alg D1: Alg 6; MGD(2, 1, 1); LEX ,
Alg D2: Alg 6; MGD(2, 1, 3); LEX,
Alg D3: Alg 6; MGD(1, 1, 2); LEX,
Alg D4: Alg D3; injection of residuals,
Alg D5: Alg D4; V-cycle,
Alg D6: Alg D5; linear interpolation of corrections.

In Graphs 32 and 33, it is seen that the second order smoothing step strongly influences the subsequent higher order MGD iterations. The less expensive algorithms presented in Graphs 34 and 35 were correspondingly less efficient with the exception of algorithm Alg D3. The nearly equivalent performance of Alg 7; LEX and Alg D3 suggests that, similar to the results seen on Graphs 32 and 33, the most critical smoothing step on the fine grid in the MGD algorithm occurs just before forming a new defect.

The following MGT algorithms were designed to investigate the effects of quintic interpolation and the use of injection of residuals on the extrapolation.

Alg T1: Alg 9; LEX; quintic interpolation of higher order approximations,
Alg T2: Alg T1; injection of residuals,
Alg T3: Alg 9; LEX; injection of residuals.

A significant gain in accuracy is seen for the extrapolations made when quintic interpolation was used. The algorithms using injection of residuals performed less efficiently. This is due not only to the slower convergence observed for algorithms using injection of residuals, but also to the fact that high frequencies in the fine grid residual, which would adversely affect the extrapolation on the coarse grid, are directly injected onto the coarse grid.

The Richardson extrapolation to fourth and sixth orders from grid $h = 1/64$ to grids $h = 1/32$ and $h = 1/16$, respectively, is compared with the corresponding τ-extrapolations in Alg 8; LEX. Richardson extrapolation was performed on the second order solution obtained by the following three MG2 algorithms.

Alg R1: FMG_a; CHJ; MG2(2, 1),
Alg R2: FMG_a; CHJ; MG2(2, 1) applied twice on each grid,
Alg R3: FMG_a; CHJ; MG2(2, 1) applied three times on each grid.

In Table 4, the results of these algorithms is presented. For comparison the σ-norm of the fourth and sixth order extrapolated solutions obtained by Alg 8; LEX on grid 1/32 and grid 1/16, respectively, are presented along with the work counts corresponding to these points in the algorithm.

The equivalence of the two extrapolation methods is apparent from the results in Table 4. The accuracy of the Richardson's extrapolation approached that of the τ-extrapolation as the accuracy of the second order approximations used in the Richardson extrapolation increased. The σ-norm of the second order residuals on grid 1/16, grid 1/32, and grid 1/64 resulting in the three Richardson's extrapolation algorithms are presented in Table 5. The work counts for the Richardson's extrapolations are somewhat high as the second order algorithms used were not optimal.

TABLE 4

Richardson's extrapolation versus τ-extrapolation
σ-norms of the error after extrapolation

Order	Grid	Alg R1	Alg R2	Alg R3	Alg 9;L
4	1/32	.12(-4)	.64(-6)	.52(-6)	.93(-6)
		78 ops	136 ops	194 ops	87 ops
6	1/16	.47(-5)	.22(-6)	.13(-7)	.24(-7)
		78 ops	136 ops	194 ops	119 ops

TABLE 5

σ-norm of the second order residuals
in the Richardson's extrapolation algorithms

Grid	Alg R1	Alg R2	Alg R3
1/16	.25(-2)	.94(-4)	.40(-5)
1/32	.11(-3)	.30(-5)	.16(-6)
1/64	.39(-5)	.13(-6)	.73(-8)

3.3. Conclusions. To summarize some of the basic implications of the previous study, note first that all three higher order multi-grid methods produced very accurate approximations with about the same efficiency. The performance of the algorithms based on these higher order methods depended on the particular choices of multi-grid processes used, although each was consistently more efficient than the conventional second order algorithms. The use of full weighting restriction, higher order interpolation of corrections and *W*-cycling produced more efficient higher order algorithms.

The convergence rates observed in the experiments generally stayed within the optimal rates predicted by the local mode analysis. The notable exception to this was

the MGD algorithms using Checkerboard relaxation. This relaxation introduces high frequency information into the error on the fine grid, which, in turn, contaminates the higher order approximation in the Defect Correction equation used in MGD. The sixth order τ-extrapolation algorithms using Checkerboard relaxation produced similar results to that of MGD.

The Mehrstellen Verfahren discretizations were the most accurate of the discretizations considered here. The algorithms using the Mehrstellen Verfahren discretization were consistently more efficient than the corresponding algorithms using other alternatives.

APPENDIX

List of algorithms used in the experiments

Alg 1: MG2(2, 1); FMG_a

Alg 1a: MG2(1, 1); FMG_a; injection (by $\frac{1}{2}$) of residuals; V-cycle; CHJ

Alg 2: WMG2(2, 1); FMG_a

Alg 3: MGH(3, 1, 2, 1); FMG_b; $(L^h_{4,2}, L^h_{6,4})$

Alg 3a: MGH(3, 1, 2, 1); FMG_c; $(L^h_{6,4})$

Alg 4: MGH(3, 1, 2, 1); FMG_b; $(L^h_{4,M}, L^h_{6,M})$

Alg 4a: MGH(3, 1, 2, 1); FMG_c; $(L^h_{6,M})$

Alg 5: MGH(3, 1, 2, 1); FMG_c; L^h_M—sixth order discretization)

Alg H1: Alg 4; LEX; injection of residuals

Alg H2: Alg 4; CHJ; injection (by $\frac{1}{2}$) of residuals

Alg H3: Alg H1; quintic interpolation of high order approximation

Alg H4: Alg 5; LEX; quintic interpolation of high order approximation

Alg H5: Alg 5; CHJ; quintic interpolation of high order approximation

Alg H6: Alg 5; CHJ; FMG_d

Alg H7: Alg H6; quintic interpolation of high order approximation

Alg 6: MGD(2, 1, 2); FMG_b; (L^h_4, L^h_6)

Alg 7: MGD(2, 1, 2); FMG_c; (L^h_M—sixth order discretization)

Alg D1: Alg 6; MGD(2, 1, 2); LEX

Alg D2: Alg 6; MGD(2, 1, 3); LEX

Alg D3: Alg 6; MGD(1, 1, 2); LEX

Alg D4: Alg D3; injection of residuals

Alg D5: Alg D4; V-cycle

Alg D6: Alg D5; linear interpolation of corrections

Alg 8: MGT4(1, 0, 1, 2, 1);
MGT6(1, 0, 3, 0, 3); V-cycle; FMG_b

Alg 9: WMGT4(1, 0, 1, 3, 1);
WMGT6(1, 0, 3, 0, 3); V-cycle; FMG_b

Alg T1: Alg 9; LEX; quintic interpolation of high order approximation

Alg T2: Alg T1; injection of residuals

Alg T3: Alg 9; LEX; injection of residuals.

Institute for Computational Research
Colorado State University
Fort Collins, Colorado 80523

1. W. AUZINGER & H. J. STETTER, "Defect corrections and multigrid iterations." Preliminary report.

2. J. H. BRAMBLE & B. E. HUBBARD, "Approximation of derivatives by finite difference methods in elliptic boundary value problems," *Contrib. Differential Equations*, v. 3, 1963, pp. 399–410.

3. A. BRANDT, "Multi-level adaptive solutions to boundary value problems," *Math. Comp.*, v. 31, 1977, pp. 333–390.

4. A. BRANDT, "Numerical stability and fast solutions to boundary value problems," in *Boundary and Interior Layers—Computational and Asymptotic Methods* (J. J. H. Miller, Ed.), Boole Press, Dublin, 1980.

5. A. BRANDT & N. DINAR, "Multi-grid solutions to elliptic flow problems," in *Numerical Methods for Partial Differential Equations* (V. S. Parter, Ed.), Academic Press, New York, 1979.

6. L. COLLATZ, *Numerical Treatment of Differential Equations*, 3rd ed., Springer-Verlag, Berlin, 1960.

7. H. FÖRSTER, K. STÜBEN & V. TROTTENBERG, "Non-standard multi-grid techniques using checkered relaxation and intermediate grid," in *Elliptic Problem Solvers* (M. Schultz, Ed.), Academic Press, New York, 1980.

8. W. HACKBUSCH, "Bemerkungen zur iterierten Defektkorrektur und zu ihrer Kombination mit Mehrgitterverfahren," *Rev. Roumaine Math. Pures Appl.*, v. 26, 1981, pp. 1319–1329.

9. W. HACKBUSCH, "Survey of convergence proofs for multi-grid iterations," in *Special Topics of Applied Mathematics* (J. Frehse, et. al., Eds.), North-Holland, Amsterdam, 1980.

10. S. SCHAFFER, *High Order Multi-Grid Methods to Solve the Poisson Equation*, Proc. NASA-Ames Res. Center Symp. on Multigrid Methods, Moffett Field, Oct., 1981.

11. S. SCHAFFER, *Higher Order Multi-Grid Methods*, Ph.D. Thesis, Colorado State University, May, 1982.

12. H. J. STETTER, "The defect correction principle and discretization methods," *Numer. Math.*, v. 29, 1978, pp. 425–443.

13. K. STÜBEN, *Local Mode Analysis for the Solution of Elliptic Problems by Multigrid Methods*, Internal Report, GMD-IMA, St. Augustin, Germany, 1982.