SIAM J. SCI. COMPUT.
Vol. 24, No. 3, pp. 924–944

A GENETIC SEARCH FOR OPTIMAL MULTIGRID COMPONENTS WITHIN A FOURIER ANALYSIS SETTING*

C. W. OOSTERLEE† AND R. WIENANDS‡

Abstract. In this paper, Fourier analysis is used for finding efficient multigrid components. The individual multigrid components for several discrete partial differential operators are chosen automatically by a genetic optimization method. From a set of multigrid components, such as different smoothers, coarse grid correction components, cycle types, number of smoothing iterations, and relaxation parameters, an optimal three-grid Fourier convergence factor corrected for computational complexity is obtained by the genetic search. The resulting methods can be tuned for optimal efficiency or toward robustness. The analysis results are verified by numerical experiments.

Key words. Fourier three-grid analysis, genetic optimization algorithm, optimal multigrid components, standard coarsening

AMS subject classifications. 65N55, 65F10, 65Y20, 92D99

PII. S1064827501397950

1. Introduction.

An important analysis tool for multigrid methods is Fourier analysis. Since the early days of multigrid, Fourier smoothing and two-grid analysis have been used for quantitative estimates of the smoothing properties of basic iterative methods and for quantitative evaluation of the other multigrid components in a two-grid method; see, for example [2, 16, 20]. The two-grid analysis is the basis for the classical asymptotic multigrid convergence estimates. Furthermore, the local Fourier analysis (also called the local mode analysis [2]) is, in fact, the main multigrid analysis tool for nonsymmetric problems. An overview of the analysis is presented in [17]. Recently, we generalized the classical analysis in two ways. First, in [21] we provided a Fourier analysis framework for analyzing the use of multigrid as a preconditioner for restarted GMRES [14]. Second, we proposed the so-called three-grid Fourier analysis in [22]. Compared to the usual two-grid analysis, the three-grid analysis can yield additional insight, especially for singularly perturbed and nonelliptic equations. An issue that can be analyzed in more detail with a third grid is the coarse grid correction. If a large difference in convergence factors occurs between the two-grid and the three-grid convergence factors from Fourier analysis, this indicates a problematic coarse grid correction.

Although Fourier analysis tools can give good insight into multigrid convergence, it is not easy in the case of convergence problems to extract from the analysis improved or even optimal multigrid components for a PDE under consideration. This is even more true if the Galerkin coarse grid operators are employed, since it is difficult to foresee their effect for various PDEs. Tuning of multigrid parameters is, however, often necessary for fast convergence (see, for example, [17, 20]). Optimizing even just a few parameters can be a painstaking procedure that can only be done analytically for a very small set of PDE problems; see [20, 23].

*Received by the editors November 13, 2001; accepted for publication (in revised form) June 5, 2002; published electronically December 19, 2002.
http://www.siam.org/journals/sisc/24-3/39795.html
†Faculty of Information Technology and Systems, Delft University of Technology, Delft, the Netherlands (c.w.oosterlee@math.tudelft.nl).
‡Fraunhofer Institute for Algorithms and Scientific Computing (SCAI), D-53754 Sankt Augustin, Germany (wienands@scai.fraunhofer.de).

Our main idea is to employ an optimization technique to help us search for optimal multigrid components. We use a genetic algorithm (GA) in a three-grid analysis setting to find improved combinations of smoothers, coarse grid correction components, and relaxation parameters for several PDE operators. We choose genetic optimization because the objective function, i.e., finding the best three-grid convergence factors, is not at all a smooth function in the "parameters" of the optimization, such as different smoothers, transfer operators, and coarse grid discretization. Conventional local optimization methods cannot be used for this purpose. On the other hand, it is also not possible to simply apply an optimization by enumeration of all possibilities, as we can have a search space of, for example, 10^{12} possibilities with about 10 varying parameters. Thus, the GA [9, 8, 12] is the method of choice here: the objective function is well defined and can be evaluated in a relatively cheap way.

This paper is organized as follows. In section 2, we start with a detailed discussion of multigrid for nonelliptic problems to further motivate the work. In section 3 the details of the GA are listed. The three-grid analysis objective function is explained in section 3.1, and the multigrid components that are part of the search space are discussed in section 3.2. Section 4 presents the corresponding experiments and results.

Our research is oriented toward sets of satisfactory components, say, those that guarantee a three-grid convergence factor of less than a certain value, instead of toward simply searching for the best components. By this it becomes possible to observe trends within the different parameters. It is, of course, also possible to fix certain multigrid components, like the transfer operators, and search for the optimal remaining components under these "constraints."

Finally, one can also optimize the case where multigrid is used as a preconditioner since the Fourier analysis has also been generalized to this situation.

2. Discussion of multigrid for PDEs. Traditional understanding of multigrid often comes with the knowledge that a pointwise smoothing method reduces high frequency components of an error between numerical approximation and the exact numerical solution, whereas the coarse grid correction based on standard grid coarsening handles the low frequency error components. Whereas this view is perfectly valid for nicely elliptic equations like the Poisson equation, it generally does not hold for singularly perturbed and nonelliptic PDEs. Contrary to uniformly elliptic equations, these equations are governed by strong couplings that need to be taken into account by a smoother and/or by a coarse grid correction in efficient multilevel solution methods. In principle, three different strategies exist to overcome possible multigrid convergence degradation.

An efficient solution is to change the coarsening strategy to a problem-dependent coarsening while retaining a point smoother. The convergence can also be improved by more robust smoothing methods that also take care of problematic low frequencies. In this situation, the standard grid coarsening is typically applied, which may restrict the applicability of this approach, to some extent, to structured grid applications.

A third possible solution is found in the use of multigrid as a preconditioner for a Krylov subspace acceleration method like GMRES [14]. This possibility is often the easiest one, as the multigrid components, including the coarsening strategy, need not be changed. The resulting algorithm is, however, not always as efficient as the ones resulting from the two other strategies. Combining the third strategy with the first or the second leads to very efficient methods.

The second strategy is the basis for our multigrid methods. For most equations considered here, the efficient multigrid methods are based on smoothers that also take

care of certain low frequency components. In this work these smoothers are obtained "automatically" by an optimization method. As we will see in section 4.1, even for the Poisson equation, where a pointwise relaxation method is sufficient to guarantee excellent multigrid convergence, an overrelaxation parameter in a red-black Gauss–Seidel smoother leads to the most efficient solver (as in [23]). Similar to the classical SOR situation, this smoother then also reduces low frequency error components more efficiently. (The effect of smoothers on low frequency error components is not completely covered by classical theoretical methods analyzing multigrid convergence. These methods typically split a multigrid operator into two parts and analyze the smoothing and approximation properties of an operator separately. They therefore do not take into account the effect of fine grid relaxation methods on low frequency error components.)

3. The GA. GAs [9, 8] are algorithms for searching through parameter spaces and for optimizing a given function. When the objective function is not smooth, traditional search techniques like gradient methods tend to stay in one of the local minima, which is not necessarily close to the global minimum, if the function has multiple local minima. A GA can, in principle, deal with the above-mentioned problem effectively. It is based on the survival-of-the-fittest principle. An initial population of N possible solutions is created from a random selection. The strength of each of the population's individuals is evaluated by a fitness function, i.e., the objective function to be optimized. After this, a new generation of individuals is set up. This is done, with the help of genetics, by coding the parameters in binary strings and by combining them with the other binary strings to create better approximations toward the optimal solution. Each bit of the binary string is called a chromosome.

There are basically three operators that occur in GAs to define the next generation: selection, crossover, and mutation [8]. Within the GA context, several advanced operators can be chosen, for example, advanced selection and crossover operators. To make the GA more robust, operators such as tournament selection and uniform crossover have been proposed [8, 12, 5]. Tournament selection means that two individuals of the population are chosen at random, and the one with better fitness will be one of the parents. Two parents create one new individual. The crossover operator used is uniform crossover: the new individual's chromosomes are set equal to those of one of the parents. Then, for each bit of the newly created solution, the bit from the other parent is taken if a randomly selected number between 0 and 1 is less than the crossover probability p_c. p_c is set to 0.5 here. Furthermore, an elitist strategy copies the best binary string in the current population to the next generation.

We use a variant of the GA, called the microgenetic algorithm (micro-GA), as it is proposed in [11] and successfully evaluated, for example, in [5]. This GA is based on a relatively small population and avoids premature convergence to suboptimal states. If less than 5% of the total number of bits in a population is different from the best individual, then the population is restarted with randomly chosen new individuals, keeping, however, the best individual. This is an easy option for evaluating new search directions and avoiding stagnation of the convergence in a local optimum. Mutation is not applied. Most conventional GA applications operate with 50 to 200 individuals in each population; the micro-GA has $N = 5$ individuals in [11] and [5]. We will use population sizes of 10 and 15. Krishnakumar [11] compared his micro-GA against a simple GA, with a population size of 50, $p_c = 0.6$, and a mutation rate of 0.001, and reported faster and better results with his micro-GA. Similar observations are found in [5]. Furthermore, the related GA software is freely available for download from

http://cuaerospace.com/carroll/ga.html.

Here are the detailed steps of the micro-GA used.

1. Choose randomly a pre- and postsmoother, relaxation parameters, number of smoothing steps, cycle type, coarse grid discretization (in the case of Galerkin, choose restriction, prolongation, and a scaling parameter), and transfer operators. Compute the quality of the parameter set by evaluating it by the fitness function, i.e., the worst spectral radius of the block matrices appearing in the three-grid Fourier analysis; see section 3.1.
2. Code the parameter set into a binary string.
3. Repeat steps 1 and 2 to set up the initial population of size N.
4. Select from the population two members. This is done using the tournament selection process. These two will act as the parents.
5. Combine the binary strings of the two parents from step 4 to create a new string. This is done by the uniform crossover operator.

   Decode the binary string and evaluate its fitness with the help of three-grid Fourier analysis.
6. Repeat steps 4 and 5 until there are as many new individuals as in the initial population.
7. Apply the elitism operator: remove one of the individuals at random and replace him with the individual having the best fitness from the previous generation in case the current maximum fitness is not larger than the previous maximum.
8. Avoid stagnation: if less than 5% of all bits in the strings in a population differ from the string of the best individual, then restart the population with the best and $N-1$ randomly chosen new individuals.
9. Repeat steps 4–8 until the fitness function after several restarts no longer changes.

For clarity and as a simple example, the GA convergence of finding the maximum of a given function is presented in the appendix.

3.1. The fitness function, three-grid Fourier analysis. The fitness function is based on the three-grid Fourier analysis—a straightforward generalization of the well-known two-grid Fourier analysis [2]—which is briefly outlined here for 2D problems and standard coarsening. We consider a discrete problem, $L_h u_h = f_h$, where u_h represents the exact discrete solution. The main idea in the Fourier analysis is to formally extend all occurring multigrid components to an infinite grid $G_h := \{\boldsymbol{x} = (k_x h, k_y h) : k_x, k_y \in \mathbb{Z}\}$ and to restrict the considerations to discrete linear operators with constant coefficients. On G_h, we have a unitary basis of grid functions called the Fourier components,

$$\varphi_h(\boldsymbol{\theta}, \boldsymbol{x}) := e^{i\boldsymbol{\theta}\boldsymbol{x}/h} = e^{i\boldsymbol{k}\boldsymbol{\theta}} = e^{i(k_x\theta_x + k_y\theta_y)}, \tag{1}$$

with $\boldsymbol{x} \in G_h$, $\boldsymbol{k} = (k_x, k_y)$, and Fourier frequencies $\boldsymbol{\theta} = (\theta_x, \theta_y) \in \mathbb{R}^2$. These components are eigenfunctions of any discrete operator L_h with constant coefficients.

Whereas the error $v_h^m = u_h^m - u_h$ is transformed by a two-grid cycle as

$$\begin{gathered} v_h^{m+1} = M_h^{2h}\, v_h^m \quad \text{with} \\ M_h^{2h} = S_h^{\nu_2} K_h^{2h} S_h^{\nu_1} \quad \text{and} \quad K_h^{2h} = I_h - P_{2h}^h (L_{2h})^{-1} R_h^{2h} L_h, \end{gathered} \tag{2}$$

the error transformation by a three-grid cycle is given by

$$
\begin{aligned}
& v_h^{m+1} = M_h^{4h}\, v_h^m \quad \text{with} \\
& M_h^{4h} = S_h^{\nu_2} K_h^{4h} S_h^{\nu_1} \quad \text{and} \quad K_h^{4h} = I_h - P_{2h}^h (I_{2h} - (M_{2h}^{4h})^\gamma)(L_{2h})^{-1} R_h^{2h} L_h.
\end{aligned} \tag{3}
$$

Here, M_{2h}^{4h} defined by (2) reads $M_{2h}^{4h} = S_{2h}^{\nu_2}(I_{2h} - P_{4h}^{2h}(L_{4h})^{-1} R_{4h}^{2h}) S_{2h}^{\nu_1}$. L_h, L_{2h}, and L_{4h} correspond to discretizations on the h-, $2h$-, and $4h$-grids of the PDE under consideration. S_h and S_{2h} are the smoothing operators on the fine grid and the first coarse grid, and ν_i $(i = 1, 2)$ represents the number of pre- and postsmoothing steps. R_h^{2h}, R_{2h}^{4h} and P_{2h}^h, P_{4h}^{2h} denote restriction and prolongation operators, respectively, between the different grids. I_h and I_{2h} are the identity operators w.r.t. the h-grid and the $2h$-grid.

Instead of inverting L_{2h}, as is done in (2), the $2h$-equation is solved approximately in a three-grid cycle (3) by performing γ two-grid iterations M_{2h}^{4h} with zero initial approximation.

In the two-grid analysis, one assumes an exact solve of the coarse grid problem on the first coarser grid level. This simplifying assumption, however, cannot resolve the actual *multi*grid behavior for several multigrid components and cycle variants. For example, if one uses different discretizations on different grids, it may occur that an operator on the coarse grids is not favorable for the smoothing method applied. An important example for such a coarse grid discretization is the Galerkin discretization. Its entries are not known in advance and depend on the fine grid discretization and the transfer operators in use. This cannot be taken into account by a two-grid Fourier analysis but can be by a three-grid analysis.

The basic idea in the three-grid analysis is a recursive application of the two-grid analysis. Similarly as in the two-grid case, we distinguish between low and high frequencies, but now w.r.t. three grids G_h, G_{2h}, and G_{4h}. Then, it is appropriate to divide the Fourier space $\mathcal{F}$ into a direct sum of 16D subspaces, the so-called $4h$-harmonics $\mathcal{F}_\theta^{3g}$ [22]. As a consequence, M_h^{4h} is unitarily equivalent to a block diagonal matrix with 16×16 blocks:

$$
M_h^{4h}|_{\mathcal{F}_{\boldsymbol{\theta}}^{3g}} \triangleq M^{3g}(\boldsymbol{\theta}, h) \in \mathbb{C}^{16\times 16}.
$$

Then, we obtain the three-grid convergence factor $\rho_{3g}(h)$ as the supremum of the spectral radii from the block matrices. For the explicit representation of the block matrices $M^{3g}(\boldsymbol{\theta}, h)$, and more details about the three-grid analysis, we refer to [22]. The three-grid Fourier analysis software is freely available for download from http://www.mgnet.org/mgnet-codes-wienands.html. It is also available from http://www.gmd.de/SCAI/people/wienands.html.

In the GA optimization we use a three-grid convergence factor that is corrected for computational complexity costs, as in [2, 18]. Point smoothing for a stencil consisting of $nelm$ elements costs $nelm$ multiplications and $nelm - 1$ additions, i.e., $2nelm - 1$ operations per unknown (9 operations for a 5-point stencil). Multiplication and addition are assumed equally expensive. The reference work unit is the cost of a point smoother without relaxation parameter for the PDE stencil on the finest grid. Contrary to [2], we do not take special situations into account, such as the case $a_{i,j} = 1$ where one can save one multiplication, etc., in the operation count. The scaled three-grid convergence factor is defined as

$$
\sigma := (\rho_{3g}(h))^{1/wu} \tag{4}
$$

with wu the number of work units relative to one fine grid pointwise relaxation per unknown. The work unit cost increases with the number of relaxations ν_1 and ν_2.

The work for a V-cycle is a factor 1/3 higher because coarse grid points also are processed; for a W-cycle an additional factor of 1 is used, which is the additional total number of points processed by this cycle [17].

With an under/over relaxation parameter, 3 extra operations are counted per smoothing iteration. The cost of the LU factorization and the backward substitution for a linewise relaxation depends on the number of diagonals $ndiag$ involved. Tridiagonal solves cost 8 operations (see, for example, [10, 17]); for solves with 4 diagonals we count 17 operations, and for pentadiagonal matrices we count 19 operations. Furthermore, the set-up of the right-hand side costs $2(nelm - ndiag)$ operations per grid point. The cost for alternating line relaxation is twice as much as that for simple linewise relaxation, etc. The cost of transfer operators is not taken into account.

We give two calculation examples. In the case of a 9-point Galerkin coarse grid discretization and a 5-point fine grid discretization, pointwise smoothing without underrelaxation and a V(1,0)-cycle, we obtain the work unit count $wu = 1*1+1/3*17/9 = 44/27$, whereas in the case of linewise smoothing with relaxation parameter and a W(1,0)-cycle, we obtain $wu = 1 * 15/9 + 1 * 23/9 = 38/9$.

3.2. The search parameters: The multigrid components. In this section, we detail the multigrid components that can be specified in the local Fourier analysis setting. They represent the search space parameters in the GA.

We choose for standard coarsening, i.e., doubling the mesh size h in every direction. For the coarse grid operators L_{2h}, L_{4h}, a first choice is to use the direct analogue of the PDE discretization L_h on the coarse grids. The Galerkin coarse grid operator, however, has become standard in the context of problems with discontinuous coefficients, but also in the context of algebraic multigrid (AMG), where algebraic systems of equations without a grid-oriented background are being solved. These operators are defined by

$$L_{2h} := \eta R_h^{2h} L_h P_{2h}^h, \quad L_{4h} := \eta R_{2h}^{4h} L_{2h} P_{4h}^{2h}. \tag{5}$$

η is a parameter that offers a rescaling possibility of the Galerkin operator. For certain choices of R_h^{2h}, R_{2h}^{4h} and P_{2h}^h, P_{4h}^{2h} and L_h, η is necessary for fast convergence; see, for example, [1]. The transfer operators and η become parameters in a GA's search space.

The choices for the restriction and prolongation operators R_h^{2h} and P_{2h}^h, for the intergrid transfer of grid functions and as part of the Galerkin coarse grid operator, can be determined differently. They are described below. As restriction operators, we can choose from among the simple injection operator, the frequently used full weighting (FW) operator, the half-weighting (HW) operator, and the 7-point restriction operator [17]. Other possibilities include the operator-dependent restriction belonging to the so-called nonsymmetric blackbox multigrid method of Dendy [6] and, more generally, the transpose, $(P_{2h}^h)^T$, of a chosen prolongation operator. The operator-dependent restriction from [6] is based on the transpose of operator L_h.

Well-known prolongation operators are bilinear interpolation, constant (upwind) interpolation, and 7-point prolongation. Other possibilities include the prolongation operator of de Zeeuw [24] and the prolongation belonging to nonsymmetric blackbox multigrid [6]. For symmetric problems, the latter two prolongation operators coincide. We do not allow the bicubic interpolation operator or a high order restriction here, since it might be difficult to implement these in general multigrid codes.

Several smoothers can be chosen in the optimization process. We adopt the general stencil notation as in [18]. Thus, a grid point, instead of a matrix description, is used. The damped pointwise Jacobi smoother ω-JAC then reads

$$\begin{bmatrix} & \bullet & \\ & & \end{bmatrix} u^*_{i,j} = -\begin{bmatrix} \bullet & \bullet & \bullet \\ \bullet & & \bullet \\ \bullet & \bullet & \bullet \end{bmatrix} u^m_{i,j} + f_{i,j},$$

$$u^{m+1}_{i,j} = \omega u^*_{i,j} + (1-\omega) u^m_{i,j}, \qquad i,j = 1,n. \tag{6}$$

The $\bullet$ represents the operator element in a 9-point stencil. ω (representing ω_1 or ω_2 for pre- or postsmoothing) is the relaxation parameter. The four-color relaxation reads as follows:

For grid points with i odd, j odd:

$$\begin{bmatrix} & \bullet & \\ & & \end{bmatrix} u^*_{i,j} = -\begin{bmatrix} \bullet & \bullet & \bullet \\ \bullet & & \bullet \\ \bullet & \bullet & \bullet \end{bmatrix} u^m_{i,j} + f_{i,j},$$

$$u^{m+1}_{i,j} = \omega u^*_{i,j} + (1-\omega) u^m_{i,j}.$$

For grid points with i even, j even:

$$\begin{bmatrix} & \bullet & \\ & & \end{bmatrix} u^*_{i,j} = -\begin{bmatrix} \bullet & & \bullet \\ & & \\ \bullet & & \bullet \end{bmatrix} u^{m+1}_{i,j} - \begin{bmatrix} & \bullet & \\ \bullet & & \bullet \\ & \bullet & \end{bmatrix} u^m_{i,j} + f_{i,j},$$

$$u^{m+1}_{i,j} = \omega u^*_{i,j} + (1-\omega) u^m_{i,j}.$$

For grid points with i even, j odd:

$$\begin{bmatrix} & \bullet & \\ & & \end{bmatrix} u^*_{i,j} = -\begin{bmatrix} & \bullet & \\ \bullet & & \bullet \\ & \bullet & \end{bmatrix} u^{m+1}_{i,j} - \begin{bmatrix} \bullet & & \bullet \\ & & \\ \bullet & & \bullet \end{bmatrix} u^m_{i,j} + f_{i,j},$$

$$u^{m+1}_{i,j} = \omega u^*_{i,j} + (1-\omega) u^m_{i,j}.$$

For grid points with i odd, j even:

$$\begin{bmatrix} & \bullet & \\ & & \end{bmatrix} u^*_{i,j} = -\begin{bmatrix} \bullet & \bullet & \bullet \\ \bullet & & \bullet \\ \bullet & \bullet & \bullet \end{bmatrix} u^{m+1}_{i,j} + f_{i,j},$$

$$u^{m+1}_{i,j} = \omega u^*_{i,j} + (1-\omega) u^m_{i,j}.$$

(7)

The four-color relaxation is identical to the red-black Gauss–Seidel smoother for 5-point stencils. For larger stencils, it is more appropriate to call these smoothers four-color and red-black *Jacobi* smoothers, respectively. We will use these names here. A red-black Jacobi smoother uses for arbitrary stencils in the first partial (the "red") step the old values from all neighboring unknowns, whereas a Gauss–Seidel relaxation should use, on previously computed grid points that are addressed, the most recent unknown values. The same is true for the second partial step.

As an example for a linewise smoothing method, in the following we describe the x-line zebra Jacobi smoother (commonly known as x-zebra line Gauss–Seidel for

9-point stencils) with relaxation parameter:

For grid lines with j odd:

$$\begin{bmatrix} \bullet & \bullet & \bullet \\ & & \end{bmatrix} u^*_{i,j} = -\begin{bmatrix} \bullet & \bullet & \bullet \\ & & \\ \bullet & \bullet & \bullet \end{bmatrix} u^m_{i,j} + f_{i,j},$$

$$u^{m+1}_{i,j} = \omega u^*_{i,j} + (1-\omega) u^m_{i,j}, \qquad i = 1, n.$$

For grid lines with j even:

$$\begin{bmatrix} \bullet & \bullet & \bullet \\ & & \end{bmatrix} u^*_{i,j} = -\begin{bmatrix} \bullet & \bullet & \bullet \\ & & \\ \bullet & \bullet & \bullet \end{bmatrix} u^{m+1}_{i,j} + f_{i,j},$$

$$u^{m+1}_{i,j} = \omega u^*_{i,j} + (1-\omega) u^m_{i,j}, \qquad i = 1, n. \tag{8}$$

All other linewise smoothers considered, of Jacobi, lexicographic Gauss–Seidel, or alternating direction type, can be described similarly. Furthermore, we have included Kaczmarz [6] variants of the smoothers presented above.

Summarizing, we consider 5 prolongation, 6 restriction, and 15 smoothing operators. We use the following abbreviations for the restriction and prolongation operators:

bl : bilinear interpolation
cu : constant (upwind) interpolation
ze : de Zeeuw's interpolation [24]
di : Dendy's interpolation [6]
7p : 7-point interpolation
fw : full weighting
hw : half-weighting
in : injection
dr : operator-dependent restriction [6]
7r : 7-point restriction
adj : adjoint of interpolation

The direct coarse grid discretization is denoted by **dir**; the Galerkin coarse grid discretization by **gal**. In the latter case, the scaling factor η also is given.

We use the following notation for the smoothing operators:

pjac : pointwise Jacobi
pgs : pointwise lexicographic (lex.) Gauss–Seidel
rbj : red-black Jacobi
4cj : four-color Jacobi
kpjac : Kaczmarz point Jacobi
kpgs : Kaczmarz point lex. Gauss–Seidel
krbj : Kaczmarz red-black Jacobi
kalgs : Kaczmarz alternating (alt.) line Gauss–Seidel
xljac : x-line Jacobi
xlgs : x-line lex. Gauss–Seidel
xzj : x-zebra line Jacobi
alj : alt. line Jacobi
algs : alt. line lex. Gauss–Seidel
azj : alt. zebra line Jacobi
kalj : Kaczmarz alt. line Jacobi

y-line smoothers are not included here; for the test problems below (except for linear elasticity) we will not place the anisotropy in the y-direction. The pre- and postsmoothing operators are allowed to differ. The relaxation parameters are denoted by $\omega_{\mathbf{1}}$ and $\omega_{\mathbf{2}}$, respectively, and can be different for pre- and postsmoothing. The relaxation parameters and scaling η vary between 0.5 and 2.05 in steps of 0.05; for each of these parameters the GA has 2^5 possibilities.

4. Applications. To demonstrate the potential use of a GA in the three-grid analysis setting, we consider several equations and discretizations. The applications range from nicely elliptic to singular perturbation problems.

The first test problem, the Laplace operator, serves to present reference values and algorithms and to give insight into GA optimization. The other problems then show the potential of the method.

We use a 32^2 grid for the genetic optimization and a Fourier spectrum divided into 32^2 modes. The best algorithms on this relatively coarse grid are, in a postprocessing step, evaluated on a 256^2 grid with 256^2 modes. The 256^2 results are presented in the tables below (unless mentioned otherwise).

For all the methods, we repeated the three-grid analysis also on the coarse grid discretization to be sure that coarser discretizations also have favorable smoothing and coarse grid correction properties. Especially for singular perturbation problems, the repeated analysis gives valuable insight into whether or not a three-grid factor can be maintained. We will denote the three-grid factor on grid nr. 3 (grid nr. 1 being the finest grid) by $\widetilde{\rho}_{5g}$. The tilde indicates that it is not a real five-grid analysis. That would be too expensive as the size of the block matrices grows.

In addition, numerical multigrid experiments were performed on a 256^2 grid with 7 grid levels for all variants. The corresponding average reduction factors are indicated by r_{7g} in the tables.

Different runs with varying initial random number seeds were performed. The GA typically runs with a population size of $N = 10$ or $N = 15$. These population sizes appear to be a good compromise with respect to robustness and efficiency in the interplay between the restart and the other GA components. Instead of advocating a certain multigrid variant here, we prefer to show various "exotic" variants, as outcome of the Fourier analysis, with excellent convergence factors.

4.1. $O(h^2)$ Laplace discretization. The first example deals with the well-known 5-point discretization of the Laplace operator,

$$-\Delta_h u \stackrel{\wedge}{=} \frac{1}{h^2} \begin{bmatrix} & -1 & \\ -1 & 4 & -1 \\ & -1 & \end{bmatrix} u_{i,j}. \tag{9}$$

This problem serves as a reference for seeing which σ (4) can be achieved, since multigrid methods are known to achieve excellent convergence in only a very small number of operations for (9). We expect a σ-landscape with many local optima, for each smoother at least one exists, and a general overall good fitness for many parameter sets. Furthermore, there are many identical local optima, since for (9) several interpolation operators are identical. For example, bilinear interpolation **bl** is identical to the operator-dependent prolongation operators **di** and **ze**.

A well-known classical multigrid method [16] for this problem consists of the direct $2h$-, $4h$-, etc., coarse grid discretizations, bilinear interpolation of corrections, FW of residuals, and red-black Jacobi relaxation. The three-grid convergence factor for the W(1,1)-cycle is 0.074, as presented in Table 1. The computed scaled factor, however, is relatively large at $\sigma = 0.52$; for the F-cycle it is 0.48. The results for the V(1,1)-cycle are also presented in Table 1. Actually, the smallest scaled convergence factor σ was obtained by the genetic optimization for the same method, with one of the smoothing steps containing an overrelaxation parameter of $\omega_1 = 1.25$, $\omega_2 = 1.0$; see Table 1.

The best three-grid convergence factor, $\rho_{3g} = 0.033$ with a pointwise smoother, was found for a somewhat exotic variant. It is in the 4th row of Table 1. We call this variant exotic because of the scaling factor $\eta \neq 1$ in the Galerkin operator (5). The four-color relaxation is useful here, as we deal with 9-point coarse grid operators.

TABLE 1

Different multigrid components with convergence factors for the 5-point Poisson stencil, $h = 1/256$.

Cycle (ν_1, ν_2)	Smoother pre-, post-, ω_1, ω_2	Coarse grid discr. Gal. transf., η	Transfer op. prol., restr.	ρ_{3g}	σ
W(1,1)	rbj, rbj, 1.0, 1.0	dir	bl, fw	0.074	0.52
V(1,1)	rbj, rbj, 1.0, 1.0	dir	bl, fw	0.106	0.43
V(1,1)	rbj, rbj, 1.25, 1.0	dir	bl, fw	0.051	0.38
V(1,1)	4cj, 4cj, 1.0, 1.15	gal, bl, hw, 0.67	bl, fw	0.033	0.40
V(1,1)	4cj, 4cj, 1.15, 1.0	gal, bl, fw, 1.0	bl, fw	0.037	0.41

TABLE 2

Multigrid components and convergence for the 5-point Poisson stencil with prolongation and restriction prescribed, $h = 1/256$.

Smoother pre-, ω_1,	Coarse grid discr. Gal. transf., η	Transfer op. prol., restr.	W(1,0) σ	W(1,0) ρ_{3g}	F(1,0) r_{7g}
rbj, 1.0	gal, cu, adj, 1.35	cu, in	0.68	0.47	0.43

Finally, we present a more well-known variant based on **gal**, **bl**, **fw**, and $\eta = 1$ in the 5th row of Table 1. A general tendency for the very small three-grid convergence factors ρ_{3g} appears to be the four-color relaxation with an overrelaxation parameter.

All analysis results in Table 1 were confirmed by numerical multigrid computations on a 256^2 grid.

We continue with an optimization under constraints. As the prolongation operator, we allow only **cu**; as the restriction operators, we allow **in** and **adj**. These choices are also fixed in the definition of the Galerkin operator. A reason for this choice could be the need to design a solution method that also solves the Poisson equation efficiently on an unstructured grid. In this setting, the GA should find the optimal remaining multigrid components. Table 2 presents the resulting algorithm with corresponding factors. The multigrid F(1,0)-convergence is very similar to the three-grid factor.

4.2. $O(h^4)$ Laplace discretization. The next discrete operator discussed is the fourth order "long stencil" discretization of the Laplace operator,

$$-\widehat{\Delta}_h u \triangleq \frac{1}{12h^2} \begin{bmatrix} & & 1 & & \\ & & -16 & & \\ 1 & -16 & 60 & -16 & 1 \\ & & -16 & & \\ & & 1 & & \end{bmatrix} u_{i,j}. \tag{10}$$

For this type of long stencil with positive and negative entries, it is already difficult to foresee the effect of certain transfer operators on a Galerkin coarse grid discretization. Here, the matrix-dependent interpolation and restriction variants **ze**, **di**, and **dr** are not allowed in the optimization because they have only been defined for compact 9-point stencils. Figure 1 presents the convergence of the GA for the discrete 32^2 operator with two different populations $N = 10$ and $N = 15$. For both populations, two convergence curves are shown from two different initial random number seeds. Fast multigrid methods are obtained after about 70 GA iterations. Note that 1 GA iteration in the case of $N = 10$ represents 10 evaluations of the fitness function and 15 for $N = 15$. In the final convergence stage of the GA in Figure 1, where the fitness function is less than about 0.55, only local changes in the algorithm are made

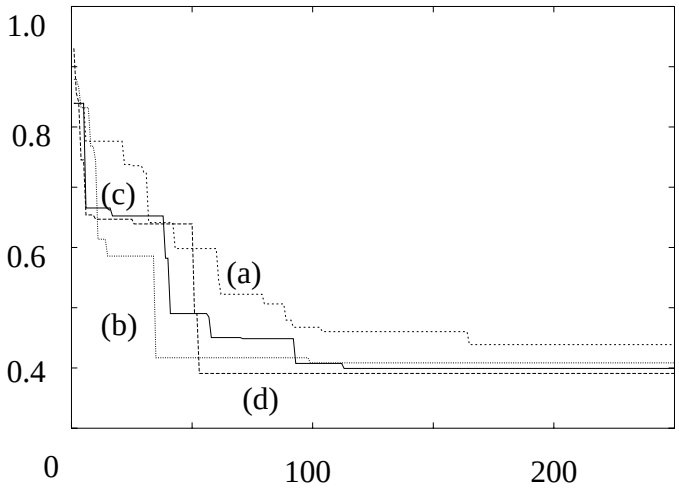

FIG. 1. *GA convergence of the fitness function with two random number seeds on a* 32^2 *grid.* (a) *run* 1, $N = 10$*;* (b) *run* 2, $N = 10$*;* (c) *run* 1, $N = 15$*;* (d) *run* 2, $N = 15$.

TABLE 3
Different multigrid components with convergence factors for the $O(h^4)$ *Poisson discretization,* $h = 1/256$.

Smoother	Coarse discr.	Prol./	V(1,0)			F(1,0)	
pre-, ω_1	Gal. transf., η	rest.	σ	ρ_{3g}	r_{7g}	σ	r_{7g}
rbj, 1.15	dir	bl, fw	0.41	0.25	0.35	0.52	0.24
4cj, 1.05	gal, 7p,fw,1.05	bl, fw	0.44	0.21	0.30	0.58	0.19
4cj, 1.15	gal, bl, inj, 0.5	bl, fw	0.40	0.28	0.35	0.48	0.22

such as in the value of a relaxation parameter. Figure 1 shows that the GA with such a large parameter space *does not* converge toward the same optimal method in different runs. This may seem somewhat surprising at first. All the resulting methods obtained by the GA are, however, very similar with respect to efficiency. This is in accordance with the statement that the GA will converge toward *or very close to* the optimum in such large parameter spaces within a reasonable number of iterations. In an extremely large number of GA iterations, it is expected that the global optimum is found independent of the type of run. For our purpose, however, it is particularly interesting to find different algorithms with similar efficiency.

Table 3 presents the three corresponding multigrid algorithms and the fitness characteristics on a 256^2 grid. The algorithm in the last row of the table contains only 5-point coarse grid discretizations. One observes, however, in the V(1,0)-cycle results of Table 3, that the V-cycle convergence for the algorithms obtained cannot be maintained, contrary to the F-cycle convergence. This phenomenon is often observed. It is in accordance with the well-known fact that one cannot prove a V-cycle convergence from a two- (or three-) grid iteration. (The computation of σ for the F-cycle is based on $\rho_{3g}(h)$ with $\gamma = 2$. In a three-grid cycle, the F- and W-cycles coincide. The work in an F-cycle is 16/9 times the work on the finest grid, which is taken into account in the computation of σ in Tables 3 and 4.)

Based on this experience, we perform a W-cycle optimization with $N = 15$ in the following. An interesting variant that also leads to 5-point coarse grid discretizations

TABLE 4
Different multigrid components with 5-point coarse grid discretization for the $O(h^4)$-discretization, $h = 1/256$.

Smoother	Coarse discr.	Transf.	F(1,0)			W(1,0)	
pre-, ω_1	Gal. transf., η		σ	ρ_{3g}	r_{7g}	σ	r_{7g}
rbj, 1.15	gal, cu, adj, 0.5	bl, fw	0.52	0.23	0.22	0.57	0.22

is obtained and is presented in Table 4. It is thus necessary to consider V-cycle convergence factors from the three-grid analysis with some care, compared to F- or W-cycle convergence. With these algorithms, the $O(h^4)$ Laplace discretization can be solved almost as efficiently as the 5-point Laplacian.

4.3. Biharmonic operator. In this subsection, we consider the discrete biharmonic operator,

$$-\Delta_h^2 u \triangleq \frac{1}{h^4} \begin{bmatrix} & & 1 & & \\ & 2 & -8 & 2 & \\ 1 & -8 & 20 & -8 & 1 \\ & 2 & -8 & 2 & \\ & & 1 & & \end{bmatrix} u_{i,j}. \tag{11}$$

A common multigrid treatment is to rewrite the operator as a system of two Poisson equations to obtain better smoothing properties and smaller stencils. Here, however, we will not do this but rather will search for optimal convergence factors directly for (11). For this equation, the GA optimization with several populations and random number seeds always detects the same basic multigrid components. They are **dir**, **bl**, **fw**, **rbj**, i.e., the direct coarse grid discretization, FW and bilinear interpolation as the transfer operators, and red-black Jacobi as both the pre- and postsmoothing operators. Furthermore, the V(3,1)-cycle shows the smallest σ-values. Table 5 presents two typical algorithms with relaxation parameters ω_1 and ω_2 and the convergence characteristics on a 256^2 grid. The results in the third row of Table 5 with parameters $\omega_1 = \omega_2 = 1$ are included only for comparison.

TABLE 5
Two relaxation parameters and convergence factors for the biharmonic operator, $h = 1/256$.

Smoother			V(3,1)		
	ω_1	ω_2	σ	ρ_{3g}	r_{7g}
rbj	1.25	0.55	0.68	0.096	0.075
rbj	1.25	0.7	0.66	0.085	0.080
rbj	1.0	1.0	0.78	0.27	0.32

The r_{7g} factors in Table 5 show that the V-cycle convergence is confirmed by the numerical experiments on 7 grid levels. The relaxation parameters give an impressive convergence improvement compared to the standard case without parameters. (A similar statement w.r.t. damped Jacobi has already been given in [2].) Figure 2 presents ρ_{3g} as a function of the two ω-parameters. In this figure it can be seen that the three-grid factor is not very sensitive with respect to ω_2.

Based on these results, we use the GA to further improve the relaxation parameters. The GA can, of course, be used merely for a parameter optimization. We allow all four relaxation parameters, $\omega_{1,1}$, $\omega_{1,2}$, $\omega_{1,3}$, and $\omega_{2,1}$ in the V(3,1)-cycle to be different. Table 6 presents the parameters and the corresponding impressive convergence factors.

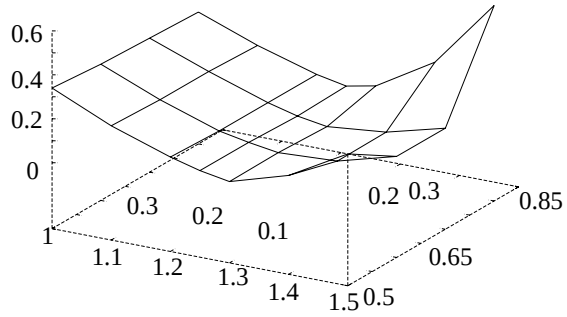

FIG. 2. *Plot of ρ_{3g}, dependence of ω_1 and ω_2.*

TABLE 6
Different relaxation parameters and convergence factors for the biharmonic operator, $h = 1/256$.

	Smoother				V(3,1)		
	$\omega_{1,1}$	$\omega_{1,2}$	$\omega_{1,3}$	$\omega_{2,1}$	σ	ρ_{3g}	r_{7g}
rbj	1.3	1.35	1.25	0.7	0.63	0.063	0.055
rbj	1.3	1.35	1.35	0.65	0.61	0.052	0.051

Comparing results in Tables 5 and 6, we see that the best variant, $\rho_{3g} = 0.052$, is about 2.5 times faster than the basic variant, with $\rho_{3g} = 0.32$. In order to reduce a residual by 5 orders of magnitude, the best variant needs about 4 iterations, whereas the basic variant costs 10.

If the biharmonic equation is, however, split into a system of Poisson equations, it is possible to benefit from the multigrid variants described in subsections 4.1 and 4.2.

4.4. Operator with a mixed derivative. Here we start with an equation with a coarse grid correction problem in a standard grid coarsening setting, and we aim to achieve optimal three-grid factors. The operator with a mixed derivative discretized by a second order 9-point operator reads

$$-(u_{xx} + u_{yy} + \tau u_{xy})_h \triangleq \frac{1}{h^2} \begin{bmatrix} \tau/4 & -1 & -\tau/4 \\ -1 & 4 & -1 \\ -\tau/4 & -1 & \tau/4 \end{bmatrix} u_{i,j}. \tag{12}$$

For $|\tau| \to 2$, this equation is singularly perturbed. Using standard grid coarsening, the direct PDE coarse grid discretization and lexicographic Gauss–Seidel smoothing lead to coarse grid correction difficulties which limit the two-grid convergence to a factor of 0.75; see, for instance, [21]. A recursive argument yields a lower bound for the V-cycle convergence on k grids, which is given by $\rho_{kg}(h) \geq 1 - 4^{-k+1}$ ($\rho_{2g}(h) \geq 0.75$, $\rho_{3g}(h) \geq 0.9375, \dots$).

The fact that we deal with a coarse grid correction difficulty for $\tau \to 2$ is clear in the algorithmic variants in Table 7 obtained by the GA: in all variants especially the coarse grid discretization is involved, i.e., a Galerkin discretization with a scaling parameter $\eta \neq 1$. Moreover, the transfer operators are matrix-dependent (**ze** equals

TABLE 7
Different multigrid components with convergence factors for the operator with mixed derivative, $h = 1/256$.

Smoother post-, ω_2	Coarse discr. Gal. transf., η	Prol./ rest.	W(0,1) σ	ρ_{3g}	$\widetilde{\rho}_{5g}$	r_{3g}	r_{7g}
4cj, 1.1	gal, ze, adj, 0.75	ze, adj	0.67	0.39	0.81	0.32	0.76
xzj, 1.15	gal, ze, adj, 0.85	ze, fw	0.66	0.34	0.81	0.29	0.79

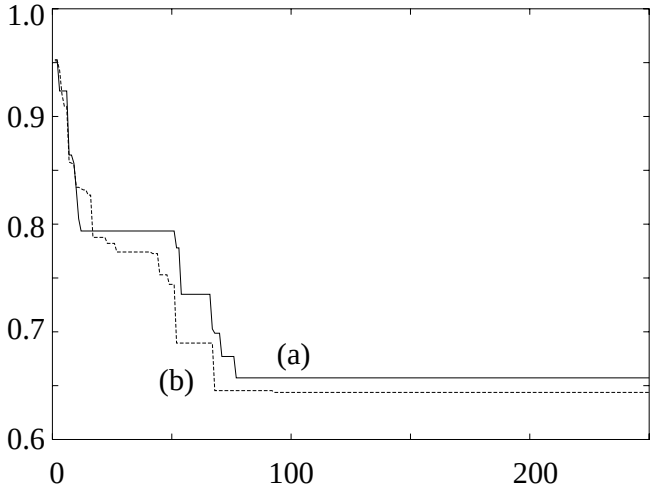

FIG. 3. *Convergence of the fitness function σ by the GA with two different initial random number seeds on a 32^2 grid.* (a) *run* 1, $N = 15$; (b) *run* 2, $N = 15$.

di for this problem) and the W-cycle is selected. Furthermore, only one smoothing iteration appears to be necessary.

Figure 3 shows, for the W-cycle, the σ-convergence for $N = 15$ of the GA with different initial random number seeds.

The convergence factors from the repeated three-grid analysis $\widetilde{\rho}_{5g}$ on the (coarse) 64^2 grid in Table 7, however, indicate that the convergence factors will increase if more than three grids are involved. This indeed is observed in the last column of Table 7. A simple and cheap cure for the multigrid convergence degradation is to apply a "recombination of iterants" technique [19] only on the third grid. By this, we have regained the three-grid convergence factor by a W-cycle multigrid algorithm with 7 grid levels.

Another possible way to avoid the convergence factors growing with the number of grids is to exclude these solutions within the optimization. In the following, we have optimized, instead of objective function (4), another objective function under the constraints that the difference between ρ_{2g} and ρ_{3g}, and between ρ_{3g} and $\widetilde{\rho}_{5g}$, should not be larger than 0.1:

$$\sigma' := (\rho_{2g})^{1/wu} + (\rho_{3g})^{1/wu} + (\widetilde{\rho}_{5g})^{1/wu}, \tag{13}$$

subject to

$$|\rho_{3g} - \rho_{2g}| \le 0.1, \quad |\rho_{3g} - \widetilde{\rho}_{5g}| \le 0.1.$$

If a possible solution does not satisfy these inequalities in a GA iteration, it is given

TABLE 8
Different multigrid components with convergence factors for the operator with mixed derivative, $h = 1/256$.

Smoother post-, ω_2	Coarse discr. Gal. transf., η	Prol./ rest.	W(0,2) σ	ρ_{3g}	r_{7g}
4pj, 1.3	gal, ze, adj, 0.75	ze, adj	0.78	0.30	0.30

a very small fitness function value.

Table 8 presents the multigrid components obtained by a GA with $N = 15$ and the convergence factors on a 256^2 grid.

The W-cycle convergence of this highly specialized multigrid variant is impressive for this singular perturbation problem. The convergence cannot, however, be maintained for the F-cycle. We find $r_{7g} = 0.6$ for the F(0,2)-cycle.

4.5. Helmholtz equation. We consider the Helmholtz equation,

$$-\Delta u - cu = f \quad (c \geq 0), \tag{14}$$

on the unit square with Dirichlet boundary conditions. First, we consider the standard 5-point discretization stencil,

$$\frac{1}{h^2}\begin{bmatrix} & -1 & \\ -1 & 4-ch^2 & -1 \\ & -1 & \end{bmatrix}. \tag{15}$$

The corresponding eigenvalues,

$$\lambda_{k,l} = \widetilde{\lambda}_{k,l} - c \equiv \frac{2}{h^2}(2 - \cos k\pi h - \cos l\pi h) - c \quad (k,l = 1,2,\ldots,n-1) \tag{16}$$

are not equal to zero as long as c is not equal to any of the eigenvalues of the corresponding discrete Laplace operator $\widetilde{\lambda}_{k,l}$. Otherwise, the matrix is singular and its nullspace is spanned by the corresponding eigenfunctions,

$$\varphi_h^{k,l} = \sin k\pi x \sin l\pi y. \tag{17}$$

The Helmholtz operator leads to smoothing and coarse grid approximation difficulties for standard multigrid depending on the particular value of c. The matrix is positive definite as long as $c < \widetilde{\lambda}_{1,1}$. However, we have to expect a performance degradation of multigrid if c approaches this first eigenvalue. For $c > \widetilde{\lambda}_{1,1}$, the matrix is no longer positive definite. Gauss–Seidel iteration does not *converge*, but since its smoothing properties are satisfactory, the multigrid convergence will deteriorate gradually for c increasing (provided that a direct solver is employed on the coarsest grid). By the time c approaches the sixth eigenvalue $\widetilde{\lambda}_{k,l}$ ($c \approx 150$), standard multigrid (**dir**, **bl**, **fw**, **rbj**, $\omega_i = 1$) diverges. The reason for this degradation is that Gauss–Seidel relaxation now diverges for smooth eigenfrequencies $\varphi_h^{k,l}$ with $\widetilde{\lambda}_{k,l} < c$. Consequently, multigrid will still converge as long as the coarsest level used is fine enough to sufficiently represent these smooth eigenfrequencies. That is, the size of the coarsest level limits the convergence when c gets larger: the more variables on the coarsest level, the higher the value of c will be for which standard multigrid converges. If $c \to \infty$, then the smoothing properties of relaxation methods fail and standard multigrid does not converge.

The Helmholtz operator belongs, as the Laplace operator, to the class of symmetric or "even" stencils [20]. For these stencils, it is possible to apply Fourier analysis on the basis of discrete sine-eigenfunctions $\varphi_h^{k,l}$ (see (17)) instead of the exponential functions (1). For problems with even stencils and Dirichlet boundary conditions, this analysis is rigorous [16, 17, 21] in the sense that it can predict h-dependent convergence factors. From the discussion above, this is particularly important for the Helmholtz operator. Therefore, we apply the rigorous analysis in this section. A disadvantage is that we can only include operators in the multigrid algorithm which can be represented by even stencils. Therefore, the use of lexicographic Gauss–Seidel type smoothers or certain (unsymmetric) transfer operators is not allowed by the analysis.

We look for multigrid convergence for increasing values of c on a relatively coarse grid with $h = 1/32$. It is more difficult to achieve good three-grid convergence factors on this grid than on a finer grid for $-50 < c < -200$. In Table 9, the standard multigrid components (**dir**, **bl**, **fw**, **rbj**, $\omega_i = 1$) are evaluated.

TABLE 9
Standard multigrid Fourier analysis convergence factors for the $O(h^2)$ discrete Helmholtz operator (15) *and different c-values, $h = 1/32$.*

	$c = 50$		$c = 100$		$c = 150$		$c = 200$	
Factor	V(1,0)	V(1,1)	V(1,0)	V(1,1)	V(1,0)	V(1,1)	V(1,0)	V(1,1)
ρ_{2g}	0.35	0.35	0.52	0.52	0.51	0.48	0.61	0.61
ρ_{3g}	0.75	0.75	0.87	0.87	DIV	DIV	DIV	DIV
σ	0.80	0.90	0.90	0.95	–	–	–	–

The benefits of the three-grid analysis are presented here. Whereas the two-grid convergence factor ρ_{2g} is satisfactory for all values of c considered, the inclusion of the third grid leads, on such a coarse grid, to worse convergence or even to divergence. All three-grid analysis results in Table 9 were confirmed by numerical experiments on three grids with $h = 1/32$ as the finest grid. In contrast to the model elliptic equations, we observe in Table 9 that two smoothing iterations do not improve the three-grid convergence compared to one smoothing iteration.

For the same c-values, we perform the GA optimization of multigrid components. The best algorithms obtained are presented in Table 10.

TABLE 10
Optimized multigrid components with three-grid convergence factors for the $O(h^2)$ discrete Helmholtz operator, $h = 1/32$.

c	Cycle	Smoother pre-, ω_1	Coarse discr. Gal. transf., η	Prol./ rest.	ρ_{3g}	σ
50	V(1,0)	rbj, 0.7	dir	bl, fw	0.60	0.75
100	W(1,0)	rbj, 0.7	dir	bl, fw	0.59	0.82
	W(1,0)	rbj, 1.0	dir	bl, hw	0.59	0.82
150	W(1,0)	4cj, 0.8	gal, bl, fw, $\eta = 1$	bl, fw	0.58	0.85
	V(1,0)	rbj, 0.8	gal, bl, fw, $\eta = 1.1$	bl, fw	0.72	0.86

Except for $c = 200$, for which convergent multigrid methods were not obtained by the GA, the σ-values in Table 10 show an improved efficiency in the resulting algorithms. Again, these results were confirmed by numerical experiments.

Another discretization of the Helmholtz operator is the $O(h^4)$-discretization based

on the Padé approximation, called the HO discretization in [15]:

$$\frac{1}{h^2}\begin{bmatrix} -\frac{1}{6}-\frac{\delta ch^2}{144} & -\frac{2}{3}-\frac{ch^2}{12}\left(1-\frac{\delta}{6}\right) & -\frac{1}{6}-\frac{\delta ch^2}{144} \\ -\frac{2}{3}-\frac{ch^2}{12}\left(1-\frac{\delta}{6}\right) & \frac{10}{3}-ch^2\left(\frac{2}{3}+\frac{\delta}{36}\right) & -\frac{2}{3}-\frac{ch^2}{12}\left(1-\frac{\delta}{6}\right) \\ -\frac{1}{6}-\frac{\delta ch^2}{144} & -\frac{2}{3}-\frac{ch^2}{12}\left(1-\frac{\delta}{6}\right) & -\frac{1}{6}-\frac{\delta ch^2}{144} \end{bmatrix}. \tag{18}$$

Parameter δ is set to 0 here (other values did not lead to better convergence).

Similar to the "Mehrstellen" discretization for the Poisson equation [17], the right-hand side is weighted among neighboring grid nodes for the $O(h^4)$-accuracy. In Table 11, the two- and three-grid Fourier analysis convergence factors for the standard multigrid components and (18) are presented.

TABLE 11
Standard multigrid Fourier analysis convergence factors for the $O(h^4)$ discrete Helmholtz operator (18) *and different c-values, $h = 1/32$.*

	$c = 50$		$c = 100$		$c = 150$		$c = 200$	
Factor	V(1,0)	V(1,1)	V(1,0)	V(1,1)	V(1,0)	V(1,1)	V(1,0)	V(1,1)
ρ_{2g}	0.53	0.15	0.77	0.23	0.27	0.12	0.94	0.27
ρ_{3g}	0.67	0.17	0.70	0.28	0.31	0.21	0.72	0.57
σ	0.74	0.52	0.77	0.62	0.42	0.56	0.78	0.81

These factors are much more satisfactory than those for the 5-point discretization. Two relaxation iterations further improve the multigrid convergence in this case. The convergence for $c = 150$ (not "close" to an eigenvalue) is surprisingly good. Table 12 presents improved multigrid algorithms for (18) obtained by the GA optimization.

TABLE 12
Optimized multigrid components with three-grid convergence factors for the $O(h^4)$ discrete Helmholtz operator, $h = 1/32$.

c	Cycle	Smoother pre-, post-, ω_i	Coarse discr. Gal. transf., η	Prol./ rest.	ρ_{3g}	σ
50	V(1,0)	xzj, 1.1	dir	bl, fw	0.21	0.42
100	V(1,0)	4cj, 0.9	dir	bl, fw	0.36	0.52
150	V(1,0)	4cj, 1.0	dir	bl, fw	0.29	0.39
200	V(1,0)	4cj, 0.9	dir	bl, fw	0.49	0.64
300	V(1,0)	4cj, 1.2	dir	bl, fw	0.75	0.83

On the 32^2 grid, it is possible to obtain a convergent multigrid Helmholtz solver as far as $c = 300$. Especially interesting is $c = 300$, for which the overrelaxation parameter $\omega_1 = 1.2$ is obtained. It indicates that the smoother also will reduce low frequency error components in this case. For discretization (18) all three-grid analysis results were confirmed by numerical experiments on three grids, with $h = 1/32$ as the finest grid. The GA finds converging three-grid methods on a 64^2 grid for c-values as far as $c = 750$ (**dir**, **fw**, **bl**, V(1,0), 4cj, $\omega_1 = 1.05$, $\rho_{3g} = 0.55, \sigma = 0.68$). These results are promising for discretization (18). The multigrid components will fail for c-values very close to eigenvalues. In that case, subspace correction techniques should be employed [4].

The c-range for which multigrid converges remains limited, however. A fast solution method for the Helmholtz equation with very large values of c is the wave-ray multigrid method [3]. Another efficient treatment in which multigrid is combined

with a Krylov subspace iteration method is proposed in [7]. The improvements in the method of [7] can be easily combined with discretization (18), leading to an efficient and accurate Helmholtz solver.

4.6. Linear elasticity. We finish the applications section with the 2D *system* of linear elasticity equations, considering the plain strain situation (no strain in the z-direction),

$$\begin{aligned} L_{1,1}u + L_{1,2}v &:= 2\frac{1-\overline{\nu}}{1-2\overline{\nu}}u_{xx} + u_{yy} + \frac{1}{1-2\overline{\nu}}v_{xy} = f_1, \\ L_{2,1}u + L_{2,2}v &:= \frac{1}{1-2\overline{\nu}}u_{xy} + v_{xx} + 2\frac{1-\overline{\nu}}{1-2\overline{\nu}}v_{yy} = f_2, \end{aligned} \tag{19}$$

with $\overline{\nu}$ denoting the Poisson ratio here, and (u, v) the displacement vector.

A comprehensive form for a discretization in which a finite difference as well as a finite element discretization with bilinear elements is included is based on

$$-u_{xx,h} \triangleq \frac{1}{1+2\delta}\begin{bmatrix} -\delta & 2\delta & -\delta \\ -1 & 2 & -1 \\ -\delta & 2\delta & -\delta \end{bmatrix}.$$

For $\delta = 0$, we find the well-known (h^2-scaled) finite differences, whereas for $\delta = 1/4$ it is the bilinear finite element discretization. In fact, the linear elasticity equations also consist of this building block for both types of discretization,

$$L_{1,1,h} \triangleq \frac{2(1-\overline{\nu})}{(1-2\overline{\nu})(1+2\delta)}\begin{bmatrix} -\delta & 2\delta & -\delta \\ -1 & 2 & -1 \\ -\delta & 2\delta & -\delta \end{bmatrix} + \frac{1}{1+2\delta}\begin{bmatrix} -\delta & -1 & -\delta \\ 2\delta & 2 & 2\delta \\ -\delta & -1 & -\delta \end{bmatrix},$$

block $L_{2,2,h}$ is obtained correspondingly, and

$$L_{1,2,h} = L_{2,1,h} \triangleq \frac{1}{4(1-2\overline{\nu})}\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}. \tag{20}$$

The equations are anisotropic; there is a significant difference between the coefficients in the x- and y-directions. We merely use this discrete example to find multigrid components by the GA optimization for different values of $\overline{\nu}$. The operator-dependent transfer operators **di** and **ze** also can be used for systems if they are based on the diagonal blocks $L_{i,i,h}$. For linear elasticity, however, they are equal to **bl**, their transpose to **fw**.

We choose *decoupled* relaxation, i.e., block $L_{1,1,h}$ is used for relaxing the u-variables, with v-iterants from a previous iteration; v is updated afterward in a second partial step, where block $L_{2,2,h}$ is the main operator and the u-unknowns from the previous partial step are used. The calculation of the number of work units is easy in this case.

We search for the multigrid components from the GA that are efficient for the finite element and finite difference discretization (20) for two $\overline{\nu}$-values, $\overline{\nu} = 0.33$, $\overline{\nu} = 0.495$. An often investigated case is $\overline{\nu} = 1/3$, describing steel material. Materials that are almost incompressible are described by $\overline{\nu}$ close to $1/2$. The algorithms and the scaled three-grid factors are presented in Table 13.

As expected, due to the anisotropy in the equations, the best three-grid factors are obtained with linewise smoothers. The convergence with the decoupled relaxation

TABLE 13
Optimized multigrid components with three-grid convergence factors for the linear elasticity system, $\overline{\nu} = 0.33, 0.495$, $h = 1/256$.

Discr.	Cycle	Smoother pre-, post-, ω_i	Coarse discr. Gal. transf., η	Prol./ rest.	ρ_{3g}	σ
$\overline{\nu} = 0.33$:						
FE	V(1,0)	azj, 1.15	gal, bl, fw, 1.0	bl, fw	0.062	0.37
FD	V(1,0)	azj, 1.0	gal, bl, fw, 1.0	bl, fw	0.081	0.43
$\overline{\nu} = 0.495$:						
FE	W(1,0)	azj, 1.4	gal, bl, fw, 0.8	bl, fw	0.45	0.82
FD	W(1,0)	azj, 1.2	gal, bl, fw, 0.8	bl, fw	0.35	0.80

is very satisfactory. The Galerkin coarse grid discretization is always found to be the most efficient. Indeed, when compared to the direct discretization and the same remaining components, the three-grid convergence differs significantly.

5. Conclusions. In this paper, we advocate the use of a genetic algorithm (GA) for finding efficient multigrid methods. The objective function to be optimized is the three-grid convergence factor within the Fourier analysis setting. It is corrected for computational complexity. Three-grid Fourier analysis yields additional insight compared to the usual two-grid Fourier analysis, w.r.t. the quality of a coarse grid correction and the effect of a smoother on *low* frequency error components. By a GA, the Fourier analysis can be used to *actively* search for satisfactory multigrid components. One result from the optimization is that for most of the equations considered, the optimal components include smoothing methods that also reduce certain low frequency error components by means of an overrelaxation parameter.

The GA optimization can be directed toward optimal components for individual equations or toward the robustness of a multigrid solver, i.e., solving several equations efficiently. One could aim for the following software framework: with the increasing speed of computers, one could provide optimized analysis software with a (black-box) multigrid algorithm. The available efficient components are then chosen by the analysis before (or components are adapted during) the actual numerical solution of a PDE, at least for structured applications. In this way, it is not necessary to tune multigrid components anew by hand for each following application, as the components are chosen automatically by the GA.

Appendix: GA optimization of a simple function. In this appendix, we give an example for the optimization of a simple function. The reason for this is because some readers may not be very familiar with GAs. Therefore, we show in some detail how the micro-GA works. Here, we search for the maximum of the analytic function,

$$f(x) = -x\left(\frac{1}{4}x^3 - \frac{7}{3}x^2 + 7x - 8\right).$$

One parameter, x, is varied from -7 until 8.75 in steps of 0.25. These comprise 64 possibilities. Thus, -7 is coded by 000000, -6.75 by 000001, and 8.75 by 111111, etc. The maximum is obtained at $x = 4$, for which $f(x) = 5.3333$.

The GA from section 3 proceeds as follows. At first, 5 random individuals are generated. They follow from steps 1, 2, and 3 of the algorithm in section 3:

```
nr:    bin. code:   x-value:   f(x):
 1     1 0 0 0 1 1   1.75     2.72298177
 2     1 1 1 1 0 1   8.25    -258.360352
 3     0 1 0 0 0 1  -2.75    -137.761393
 4     0 0 1 0 0 0  -5.      -662.916667
 5     0 0 1 1 0 1  -3.75    -300.922852
```

Individual 1 is the best so far. After steps 4, 5, and 6, we find the second generation. This run is as follows:

```
nr:    bin. code:   x-value:   f(x):
 1     1 0 1 0 0 1   3.25     4.26985677
 2     1 0 0 0 1 1   1.75     2.72298177
 3     0 0 0 0 0 1  -6.75    -1609.53223
 4     1 1 0 1 1 1   6.75    -66.3134766
 5     1 1 1 0 0 1   7.25    -111.45931
```

After selection and crossover, new individual 1 is best. This individual remains strongest for a large number of GA iterations. After iteration 6, we have the following population:

```
nr:    bin. code:   x-value:   f(x):
 1     1 0 1 0 0 1   3.25     4.26985677
 2     1 0 0 0 0 1   1.25     3.0094401
 3     1 0 1 0 0 1   3.25     4.26985677
 4     1 0 1 0 0 1   3.25     4.26985677
 5     1 0 1 0 0 1   3.25     4.26985677
```

At this stage, step 8 of the algorithm is applied for the first time. A restart takes place; only the fittest individual so far remains with 4 randomly chosen new individuals. It takes until iteration 13 in this run, right after a restart, before the global maximum 5.333 is obtained.

Acknowledgment. C.O. gratefully acknowledges J. Van lent for many pointers to the literature.

REFERENCES

[1] D. Braess, *Towards algebraic multigrid for elliptic problems of second order*, Computing, 55 (1995), pp. 379–393.

[2] A. Brandt, *Multi-level adaptive solutions to boundary-value problems*, Math. Comp., 31 (1977), pp. 333–390.

[3] A. Brandt and I. Livshits, *Wave-ray multigrid methods for standing wave equations*, Electron. Trans. Numer. Anal., 6 (1997), pp. 162–181.

[4] A. Brandt and S. Ta'asan, *Multigrid method for nearly singular and slightly indefinite problems*, in Proc. EMG'85 (Cologne), Multigrid Methods II, W. Hackbusch and U. Trottenberg, eds., Springer, Berlin, 1986, pp. 99–121.

[5] D.L. Carroll, *Genetic algorithms and optimizing chemical oxygen-iodine lasers*, in Developments in Theoretical and Applied Mechanics, Vol. XVIII, H.B. Wilson, R.C. Batra, C.W. Bert, A.M.J. Davis, R.A. Schapery, D.S. Stewart, and F.F. Swinson, eds., School of Engineering, University of Alabama, Tuscaloosa, AL, 1996, pp. 411–424.

[6] J.E. Dendy, Jr., *Blackbox multigrid for nonsymmetric problems*, Appl. Math. Comput., 13 (1983), pp. 261–283.

[7] H.C. Elman, O.G. Ernst, and D.P. O'Leary, *A multigrid method enhanced by Krylov subspace iteration for discrete Helmholtz equations*, SIAM J. Sci. Comput., 23 (2001), pp. 1291–1315.

[8] D.E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Reading, MA, 1989.
[9] J.H. Holland, *Adaptation in Natural and Artificial Systems*, MIT Press, Cambridge, MA, 1992.
[10] A. Krechel, H-J. Plum, and K. Stüben, *Parallelization and vectorization aspects of the solution of tridiagonal linear systems*, Parallel Comput., 14 (1990), pp. 31–49.
[11] K. Krishnakumar, *Microgenetic algorithms for stationary and nonstationary function optimization*, in SPIE Proc. Vol. 1196: Intelligent Control and Adaptive Systems, G. Rodriguez, ed., International Society for Optical Engineering, Bellingham, WA, 1990, pp. 289–296.
[12] M. Mitchell, *An Introduction to Genetic Algorithms*, MIT Press, Cambridge, MA, 1996.
[13] J.W. Ruge and K. Stüben, *Algebraic multigrid*, in Multigrid Methods, Frontiers in Appl. Math. 3, S.F. McCormick, ed., SIAM, Philadelphia, 1987, pp. 73–130.
[14] Y. Saad and M.H. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Comput., 7 (1986), pp. 856–869.
[15] I. Singer and E. Turkel, *High order finite difference methods for the Helmholtz equation*, Comput. Methods Appl. Mech. Engrg., 163 (1998), pp. 343–358.
[16] K. Stüben and U. Trottenberg, *Multigrid methods: Fundamental algorithms, model problem analysis and applications*, in Multigrid Methods, Lecture Notes in Math. 960, W. Hackbusch and U. Trottenberg, eds., Springer, Berlin, 1982, pp. 1–176.
[17] U. Trottenberg, C.W. Oosterlee, and A. Schüller, *Multigrid*, Academic Press, London, 2001.
[18] J. Van lent and S. Vandewalle, *Multigrid waveform relaxation for anisotropic partial differential equations*, Numer. Algorithms, 31 (2002), pp. 361–380.
[19] T. Washio and C.W. Oosterlee, *Krylov subspace acceleration for nonlinear multigrid schemes*, Electron. Trans. Numer. Anal., 6 (1997), pp. 271–290.
[20] P. Wesseling, *An Introduction to Multigrid Methods*, Wiley, Chichester, 1992.
[21] R. Wienands, C.W. Oosterlee, and T. Washio, *Fourier analysis of GMRES(m) preconditioned by multigrid*, SIAM J. Sci. Comput., 22 (2000), pp. 582–603.
[22] R. Wienands and C.W. Oosterlee, *On three-grid Fourier analysis for multigrid*, SIAM J. Sci. Comput., 23 (2001), pp. 651–671.
[23] I. Yavneh, *On red-black SOR smoothing in multigrid*, SIAM J. Sci. Comput., 17 (1996), pp. 180–192.
[24] P.M. de Zeeuw, *Matrix-dependent prolongations and restrictions in a blackbox multigrid solver*, J. Comput. Appl. Math., 33 (1990), pp. 1–27.
[25] O.C. Zienkiewicz and R.L. Taylor, *The Finite Element Method, Vol.* 1: *Basic Formulation and Linear Problems*, 4th ed., McGraw-Hill, London, 1988.