Electronic Transactions on Numerical Analysis.
Volume 13, pp. 81-105, 2002.
Copyright © 2002, Kent State University.
ISSN 1068-9613.

ETNA
Kent State University
etna@mcs.kent.edu

MULTIGRID PRECONDITIONING AND TOEPLITZ MATRICES [*]

THOMAS HUCKLE AND JOCHEN STAUDACHER [†]

Abstract. In this paper we discuss multigrid methods for symmetric Toeplitz matrices. Then the restriction and prolongation operators can be seen as projected Toeplitz matrices. Because of the intimate connection between such matrices and trigonometric series we can express the multigrid algorithm in terms of the underlying functions with special zeros. This shows how to choose the prolongation/restriction operator in order to get fast convergence. We start by considering Toeplitz matrices with generating functions having a single zero of finite order in $]-\pi,\pi]$, and we extend previous results on multigrid for Toeplitz matrices, in particular, by introducing a natural coarse grid operator. Afterwards we carry over our reasoning to cases with more than one zero and, we study how the previous cases relate to Toeplitz systems resulting from the discretization of Fredholm integral equations of the first kind which arise in image processing. Next, we take a brief look at Block Toeplitz systems with Toeplitz Blocks. We show how the one-dimensional techniques can be carried over easily for positive definite problems with a single zero in $]-\pi,\pi]^2$ and we also present a multigrid algorithm for linear systems arising from practical image deblurring problems. Finally, we give a new characterization of the well-known difficulties encountered in the indefinite case.

Key words. multigrid methods, iterative methods, preconditioning, Toeplitz matrices, Fredholm integral equations, image deblurring.

AMS subject classifications. 65N55, 65F10, 65F22, 65F35, 65R20.

1. Introduction.

1.1. Toeplitz matrices and generating functions.

Let $f(x)$ be a real-valued continuous function on the interval $I=[-\pi,\pi]$ which is periodically extended to the whole real axis. Given the Fourier coefficients of $f(x)$

$$a_k=\frac{1}{2\pi}\int_{-\pi}^{\pi} f(\theta)e^{-ik\theta}d\theta, \qquad \text{for } k \text{ an integer,}$$

we can define the sequence of Toeplitz matrices $\{A_n \equiv T_n(f)\}_n$, associated with the generating function $f(x)$. Its entries are given by $(A_n)_{\mu,\nu}=a_{\mu-\nu}$:

$$A_n = \begin{pmatrix} a_0 & a_{-1} & \cdots & \cdots & a_{1-n} \\ a_1 & a_0 & a_{-1} & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & a_1 & a_0 & a_{-1} \\ a_{n-1} & \cdots & \cdots & a_1 & a_0 \end{pmatrix}.$$

Note that the matrices A_n are Hermitian, since $f(x)$ is real-valued. In the case when $f(x)$ is an even function, we are dealing with a sequence of real symmetric Toeplitz matrices. Furthermore, we know that the spectrum of A_n is contained in $range(f)$.

Example 1: The well-known matrix $tridiag(-0.5,1,-0.5)$, i.e., the one-dimensional Laplacian, is related to the function $f(x)=-0.5e^{-ix}+1-0.5e^{ix}=1-cos(x)$. The eigenvalues of A_n are contained in the interval $[0,2]$. The small eigenvalues of A_n, that lead to the large condition numbers, are caused by the zero $x_0=0$ of f, $f(x_0)=f(0)=0$, of multiplicity two.

[*] Received February 21, 2002. Accepted for publication June 13, 2002. Recommended by L. Reichel.

[†] Institut für Informatik, Technische Univerität München, Arcisstr. 21, D-80290 München, Germany. E-mail: huckle@in.tum.de and staudacj@in.tum.de. The work of the second author was partially supported by the DAAD via grant DD/00/20283.

If we want to solve $A_n x = b$ iteratively we could use a further Toeplitz matrix M – which should be easy to invert – as a preconditioner and consider $M^{-1}A_n x = M^{-1}b$. A good preconditioner could, e.g., be one with an underlying function $m(x)$ of M having the same zero as $f(x)$ with the same multiplicity. If we can prove that the spectrum of $M^{-1}A_n$ is contained in an interval $[a,b]$, $0 < a \leq b < \infty$, independent of n, then this guarantees fast convergence, e.g., for the preconditioned conjugate gradient method (see, e.g., [30]).

1.2. Additive and multiplicative multigrid methods. For many classes of linear systems, multigrid methods are among the fastest iterative solvers. Frequently, their computational complexity is of the same order as the multiplication of the system matrix with a vector, i.e., $O(n)$ for a sparse matrix and $O(n\log(n))$ in the case of a Toeplitz matrix, where n denotes the number of unknowns in the linear system.

Before going deeper into the discussion of multigrid preconditioners we would like to review certain basic concepts very briefly as they will be of major importance for the rest of the paper. First of all, multiplicative multigrid cycles can be used as stand-alone solvers. We would now like to give a compact version of a multigrid cycle. For more algorithmic details see, e.g., the books by Greenbaum [20], pp. 193, or Briggs [4], pp. 48.

Algorithm 1: Solving $Ax = b$ iteratively by a **multigrid cycle**:

```
Proceed with the following iteration until the stopping criterion is satisfied.
(a) Smooth (e.g., by the Richardson method) in order to get a new iterate $x^{(j)}$.
(b) Compute the residual $r = Ax^{(j)} - b$.
(c) Restrict the residual $r^{coarse} = P^T r$, using the restriction operator $P^T$.
(d) Set up the coarse grid matrix, e.g., via the Galerkin approach $A^{coarse} = P^T AP$.
(e) Solve the residual equation $A^{coarse} y = r^{coarse}$ on the coarse grid – if not already on
    coarsest level, then apply the multigrid cycle recursively.
(f) Update $x^{(j)} := x^{(j)} - Py$ using the prolongation operator $P$.
```

If within the recursive solution in step (e) we use one cycle, we get a so-called V-cycle algorithm. By applying two cycles, we recognize the W-cycle algorithm.

However, multigrid cycles can also be used as preconditioners for Krylov subspace methods, such as, e.g., the Conjugate Gradient (CG) algorithm. For complicated problems in Scientific Computing this may be favourable because the Krylov subspace method guarantees convergence. In any event, we shall see in the following that for the structured problems investigated in this paper, multigrid cycles will usually perform more efficiently when used as stand-alone solvers.

In contrast to the multiplicative multigrid algorithms discussed so far there are also additive preconditioners like the celebrated BPX-preconditioner [1] or the multilevel diagonal scaling method [39]. These methods are designed to work as preconditioners only, and although they can rarely outperform their multiplicative counterparts on a serial computer, they are highly interesting for their usually superior parallel performance. The following expression gives a simplified BPX-preconditioner without smoothing (– the matrices $P_1, \ldots, P_k$ denote the prolongation operators on the individual levels)

$$(I + P_1(I + P_2(I + \cdots (I + P_k P_k^T)\cdots)P_2^T)P_1^T). \tag{1.1}$$

One of the goals of this paper will be to develop appropriate transfer operators for solving Toeplitz systems. Hence we finally mention the well-known fact that better transfer operators for multiplicative multigrid algorithms will normally also lead to better additive preconditioners and vice versa.

1.3. Multigrid and Toeplitz systems – A brief motivation. In multigrid methods we need to apply a restriction and prolongation operator. If we use a Galerkin approach – according to step (d) in Algorithm 1 – we can in the symmetric case write the coarse grid matrix for a two-grid step as

$$A_{n/2} \;=\; I_{n,n/2}^T * B_n^T * A_n * B_n * I_{n,n/2} \;=\; P_n^T * A_n * P_n,$$

with a Toeplitz matrix B_n related to a function $b(x)$, and the elementary projection matrix

$$I_{n,n/2} \;=\; \begin{pmatrix} 1 & & \\ 0 & 0 & \\ 0 & 1 & 0 \\ & 0 & 0 \\ & 0 & 1 \\ & & . \end{pmatrix} \;=\; I(1:n, 1:2:n),$$

in MATLAB-notation with the identity matrix I. In most cases we will consider only symmetric B_n with real-valued generating function $b(x)$.

As a starting point for our paper let us introduce the following heuristics. With $\tilde{f}(x) = b(x) * f(x) * b(x)$, the entries of the matrix $B_n^T * A_n * B_n$ are "asymptotically given" by the coefficients of $\tilde{f}(x)$; therefore the coefficients of $A_{n/2}$ can – up to a perturbation of low rank – be found by deleting every second entry in $\tilde{f}(x)$.

$$f_2(x) = (1/2) * \left(b^2(\frac{x}{2})f(\frac{x}{2}) + b^2(\frac{x}{2}+\pi)f(\frac{x}{2}+\pi)\right). \tag{1.2}$$

Let us assume that $f(x)$ has a unique zero x_0 of finite order $2k$ in the interval $]-\pi,\pi]$. Now the new matrix $A_{n/2}$ should be closely related to the original A_n. Hence the related function $f_2(x)$ should have a zero with the same multiplicity as $f(x)$. In view of $f(x) \geq 0$, this is only possible if $b(x_0+\pi) = 0$. Therefore, we can easily motivate the use of a prolongation operator of the form

$$b(x) \;=\; (cos(x_0) + cos(x))^k \;. \tag{1.3}$$

Remark 1: Note that in general a suitable prolongation operator $b(x)$ may have an additional zero $b(x_1) = 0$ without generating an additional zero in $f_2(x)$ as long as $b(x_1+\pi)f(x_1+\pi) \neq 0$. More generally we could even use prolongation operators of the form $b(x) * h(x)$ with any nonnegative function h, but in most cases we are strongly interested in retaining sparsity by setting $h(x) \equiv 1$.

1.4. Existing work on multigrid for Toeplitz systems. Multigrid methods for symmetric positive definite Toeplitz matrices were first proposed by Fiorentino and Serra for the univariate and the block case in [16] and [17], respectively. In [18] they try to extend their work to indefinite symmetric Toeplitz systems via an additive algorithm. In these papers the main focus lies on Toeplitz systems with a generating function in the Wiener class having a single zero $x_0 \in [0,\pi]$ of finite order. Fiorentino and Serra use prolongations and restrictions corresponding to the function (1.3) and employ Galerkin coarse grid operators and Richardson smoothers in their algorithms. Furthermore, they were the first to come up with (1.2).

Multigrid algorithms for Toeplitz systems were also studied by R. Chan and collaborators in [9] and [35]. In [9] a convergence proof for multigrid solvers is given for a class of symmetric positive definite Toeplitz systems generated by functions with isolated zeros of orders at most 2. In particular, [9] deals with results in cases where a prolongation operator corresponding to (1.3) is no longer appropriate, such as, e.g., for Toeplitz systems generated by $f(x) = 1 - cos(2x)$. The paper [9] first establishes two-grid convergence. In analogy with the papers by Serra and Fiorentino [16], [17], this is done on the basis of a general theorem by Ruge and Stüben ([28], p. 89) which gives conditions for level-independent convergence of two-grid solvers with Galerkin coarse grid operators. Then a convergence result for V-cycle solvers is presented. However, in general the convergence rate of these multigrid solvers is level-dependent (see [9], section 4). The algorithms studied in [9] employ damped Jacobi smoothing and – as we already mentioned – use Galerkin coarse grid operators.

Recently, Serra [31] also gave a detailed proof of convergence for two-grid solvers based on the assumption that the generating function f has a single zero x_0 of finite order at the origin. Again, the proof is founded on Ruge and Stüben's theory [28]. In particular, Serra's precise analysis of two-grid optimality (see [31], section 5) also explains rigorously why prolongation operators according to (1.3) are suitable – and it suggests lower and upper bounds for k within (1.3).

The work of Serra and Fiorentino is driven by pointing out close relations between Toeplitz matrices and matrices from trigonometric algebras. In particular, they also give multigrid convergence proofs for τ-matrices, i.e., the algebra of matrices that can be diagonalized by the fast sine transform. In [32] and [33] Serra and Tablino very recently presented multigrid approaches for circulant matrices. We finally wish to mention that formula (1.2) holds the same way for these matrix algebras since it is actually based on a convolution argument.

1.5. Outline of this paper. We start by taking a look at the case of generating functions with a single zero $x_0 \in [0, \pi]$ of finite order. We report problems that can arise in a multigrid approach with the prolongation operator corresponding to (1.3) for certain positions of x_0 and briefly present a new idea on how to overcome such difficulties by projections onto every m-th column. However, we observe that such algorithms are not very recommendable computationally. Instead, we focus on another way to overcome the problems. It is trivial and well-known that we can scale every Toeplitz matrix with a zero $x_0 \in \,]-\pi, \pi]$ such that x_0 is shifted to the origin. We then turn to the problem that we will normally lose Toeplitz structure on coarser levels, if we set up our coarse grid matrices using a Galerkin approach according to Algorithm 1, step (d). In the case when our generating function has only got a single zero $x_0 \in \,]-\pi, \pi]$, our strategy of shifting the zero to the origin helps us to get rid of such inconveniences completely, because it allows us to use a natural coarse grid operator.

Then we carry over our multigrid algorithms with natural coarse grid operators to problems with equidistant zeros of finite order in $]-\pi, \pi]$. Afterwards we investigate Fredholm integral equations of the first kind arising from one-dimensional image deblurring. We are attempting to make a connection with linear systems considered previously by interpreting the system matrices to be associated with a "zero of infinite order". We extend an algorithmic idea by R. Chan, T. Chan and J. Wan [7] – again by obtaining the coarse grid operators via rediscretization – and put the new algorithm into the context of the established multigrid method of the second kind by Hackbusch (see [22], chapter 16.)

The final section deals with Block Toeplitz matrices with Toeplitz blocks (BTTB matrices). We show how the one-dimensional techniques carry over easily in the case of a single zero $x_0 \in]-\pi,\pi]^2$. Again, we follow the strategy of fixing the zero at the origin, combined with a natural coarse grid operator. Obtaining coarse grid operators via rediscretization also leads to a practical multigrid algorithm for deblurring images with Tikhonov regularization. We note that the advantages of employing a natural coarse grid operator – in terms of preserving BTTB structure – are even more striking than in one dimension. Finally, we also take a look at indefinite BTTB matrices. We give a new phenomological characterization of the problems encountered in designing multigrid algorithms for such systems which seems to be strongly related to a very recent algorithm by Brandt and Livshits [3] for Helmholtz problems with constant coefficients.

We would like to emphasize right now that – very much unlike the papers by Serra and Fiorentino and R. Chan and collaborators, respectively – the focus of our paper does not lie on mathematical proofs. Instead, we are concerned with the development of algorithms. In any event, we will point out how our new algorithms fit into the existing mathematical framework.

Due to our focus on algorithmic issues we feel the need to give the reader plenty of numerical results. We will report numerical experiments for additive and multiplicative multigrid preconditioners as well as for multigrid algorithms as stand-alone solvers. There we will put particular emphasis on W-cycle solvers. (Note also that multigrid convergence proofs can frequently only be achieved for W-cycles and not for V-cycles.)

We always employ the following stopping criterion to obtain the iteration counts we list in our tables.

$$\frac{\|r^{(j)}\|_\infty}{\|r^{(0)}\|_\infty} \leq 10^{-6}.$$

Here $r^{(j)}$ denotes the residual after j iterations and $r^{(0)}$ the original residual, i.e., we stop iterating when the relative residual corresponding to the maximum norm is less than or equal 10^{-6}. Unless otherwise stated, we use two steps of the Richardson method for pre- and post-smoothing in our multigrid cycles. According to [16] and [17] we use the damping parameters $\omega_1 = 1/\max_{\theta\in[-\pi,\pi]} f(\theta)$ for pre-smoothing and $\omega_2 = 2/\max_{\theta\in[-\pi,\pi]} f(\theta)$ for post-smoothing, respectively. We finally note that it would not be sensible to apply a variant of Gauss-Seidel for smoothing in a Toeplitz context, unless the matrix was sparse.

2. Generating functions with zeros of finite order: Simple cases.

2.1. Model problems. In this and the following two subsections we will assume that our Toeplitz matrix A is connected with a generating function f in the Wiener class having a single zero $x_0 \in [0,\pi]$ of finite order. Although we are actually interested in dense Toeplitz matrices our reasoning is most easily explained by first considering sparse linear systems.

Example 2: Our standard example in what follows will be the sparse matrix belonging to the generating function

$$f(x) = (cos(x_0) - cos(x))^2,$$

with $x_0 \in [-\pi,\pi]\backslash\{\pm\frac{\pi}{2}\}$. Thus f has the zeros $\pm x_0$. Note that we deliberately exclude the case $x_0 = \pm\frac{\pi}{2}$, which we will investigate separately in section 3.

The matrices from Example 2 are strongly related to the indefinite matrices corresponding to $\tilde{f}(x) = cos(x_0) - cos(x)$, which can be seen as the result of a uniform finite difference discretization of the 1D Helmholtz equation

$$u_{xx} \; + \; \kappa^2 u \; = \; g.$$

Note that the matrices from Example 2 will in general differ from the Helmholtz normal equations by a perturbation of low rank.

2.2. The position of the zero. Let us consider Toeplitz matrices A_n connected to $f(x) = (cos(x_0) - cos(x))^2$. According to (1.3) we use a function with zeros at $\pm x_0 + \pi$ as prolongation operator, namely $(cos(x_0) + cos(x))^k$. The corresponding prolongation matrices B_n are

$tridiag(0.5, cos(x_0), 0.5)$,
$pentadiag(0.25, cos(x_0), cos(x_0)^2 + 1/2, cos(x_0), 0.25)$,
$septadiag(\frac{1}{8}, \frac{3}{4}cos(x_0), \frac{3}{2}cos(x_0)^2 + \frac{3}{8}, cos(x_0)^3 + \frac{3}{2}cos(x_0), \frac{3}{2}cos(x_0)^2 + \frac{3}{8}, \frac{3}{4}cos(x_0), \frac{1}{8})$,

and so on. The Galerkin coarse grid matrix matrix $A_{n/2}$ of half size is – up to a low rank term – related to the function

$$f_2(x) \;=\; (1/2)\Big((cos(x_0) + cos(\frac{x}{2}))^{2k}(cos(x_0) - cos(\frac{x}{2}))^2 +$$

$$+ \, (cos(x_0) - cos(\frac{x}{2}))^{2k}(cos(x_0) + cos(\frac{x}{2}))^2\Big) \;=$$

$$(cos(x_0)^2 - cos(\frac{x}{2})^2)^2 * \Big((cos(x_0) + cos(\frac{x}{2}))^{2k-2} + (cos(x_0) - cos(\frac{x}{2}))^{2k-2}\Big)/2 =$$

$$(cos(2x_0) - cos(x))^2 * \Big((cos(x_0) + cos(\frac{x}{2}))^{2k-2} + (cos(x_0) - cos(\frac{x}{2}))^{2k-2}\Big)/8.$$

That way our heuristics points out that $f_2(x)$ has the zeros $\pm 2x_0$ with the same multiplicity as $f(x)$. The new function $f_2(x)$ can be seen as a slightly changed version of the original f with the new zeros $\pm 2x_0$. (Note that this has been studied rigorously e.g. in [16], [17], [31]). We observe that the case $x_0 = 0$ is exceptional because $2x_0 = x_0 = 0$ and we can use the same prolongation and restriction operators in every step.

Remark 2: In general, this change of the zeros $\pm x_0$, $\pm 2x_0$, $\pm 4x_0$, and so on, can lead to difficulties if in the course of the multigrid method we reach a zero near $(2j + 1)\pi/2$; this case is e.g. also related to the function $f(x) = cos(x)^2$ with two double zeros at $\pi/2$ and $3\pi/2$. Then x_0 and $x_0 + \pi$ are both zeros of f, and f_2 will have $2x_0$ as a zero of higher multiplicity than $f(x)$; then our reasoning shows that the above approach will lead to a deterioration of the condition number of the related linear system.

Remark 2 can easily be confirmed in numerical experiments. The following tables compare iteration numbers for additive multilevel preconditioners of the form (1.1) for the Conjugate Gradient method.

number of unknowns	$\epsilon = 0.2$	$\epsilon = 0.15$	$\epsilon = 0.1$	$\epsilon = 0.01$	$\epsilon = 0.001$
256	60	88	133	159	166
512	83	111	200	246	265

Table 1. CG Iteration numbers for additive preconditioners:
$f(x) = (cos(\phi_0) - cos(x))^2$, $b(x) = (cos(\phi_0) + cos(x))^2$ and $\phi_0 = \pi/4 + \epsilon$.

number of unknowns	$\epsilon = 0.2$	$\epsilon = 0.15$	$\epsilon = 0.1$	$\epsilon = 0.01$	$\epsilon = 0.001$
256	158	187	213	221	232
512	294	350	396	416	422

Table 2. CG Iteration number for additive preconditioners:
$f(x) = (cos(\phi_0) - cos(x))^2$, $b(x) = (cos(\phi_0) + cos(x))^2$ with $\phi_0 = \pi/2 + \epsilon$.

2.3. Projections onto every m-th column – the first idea for a result. In order to avoid the problem outlined in Remark 2 we could also introduce elementary projections onto every third, fourth, or general m-th column/row of A_n. Instead of reducing A_n to half size we use $A_{n/m}$. To this aim we apply elementary projections $I_{n,n/m}$. Making use of our heuristics (1.2) once again this is related to picking every m-th entry out of $\tilde{f}(x) = b(x)^2 f(x)$. Then we get

$$f_m(x) = \frac{1}{m} \sum_{j=0}^{m-1} \tilde{f}(\frac{x + 2j\pi}{m}) = \frac{1}{m} \sum_{j=0}^{m-1} b^2(\frac{x + 2j\pi}{m}) f(\frac{x + 2j\pi}{m}) \ ,$$

which is again a 2π-periodic function. If f has a zero x_0 we have to generate a zero with the same multiplicity in f_m. This can be achieved by defining

$$b(x) = \Big(\prod_{j=1}^{m-1} (cos(x_0) - cos(x - \frac{2j\pi}{m}) \Big)^k \ .$$

Then the function f_m will have a zero at mx_0 with the desired multiplicity. Therefore, by choosing m in every step of the multigrid method we could at least avoid the exceptional case $x_0 \approx (2j + 1)\pi/2$.

However, it has long been known that multigrid algorithms usually work best if the restriction yields a reduction to every second column. This has been confirmed in all our numerical experiments which have been leading us to the conclusion that the new algorithmic idea outlined in this subsection is not very recommendable for use in practice. In the following table we simply compare iteration counts for additive preconditioners for $A = tridiag(-1, 2, -1)$.

number of unknowns	128	256	512	1024	2048	4096
reduction 1:2 per step; 5 levels used	18	18	19	20	21	21
reduction 1:4 per step; 3 levels used	37	40	41	41	41	41

Table 3. CG Iteration numbers for additive preconditioners of the form (1.1): We clearly observe that a reduction to every second column is superior to a reduction to every fourth column.

2.4. Diagonal scaling – the better result. Regarding the fact that

$$(cos(x_0) - cos(x))^2 = (1 - cos(x - x_0)) * (1 - cos(x + x_0)), \tag{2.1}$$

there is a much simpler strategy to solve the problems from Example 2. The product form (2.1) allows us to devise a simple and effective preconditioner. We solve the two matrix problems related to $1 - cos(x \pm x_0)$ (e.g., by multigrid) and use the result to precondition conjugate gradients. Note that the matrices related to $1 - cos(x \pm x_0)$ can be treated very efficiently by multigrid, because they are nothing else than diagonally scaled versions of $tridiag(-0.5, 1, -0.5)$ – i.e., the one-dimensional Laplacian With the two (orthogonal) diagonal matrices

$$Q_1 = diag(1, e^{ix_0}, e^{2ix_0}, ...), \qquad Q_2 = diag(1, e^{-ix_0}, e^{-2ix_0}, ...),$$

we can write

$$Q_1 * tridiag(-0.5, 1, -0.5) * Q_2 = tridiag(-0.5\, e^{ix_0},\ 1,\ -0.5\, e^{-ix_0}). \tag{2.2}$$

It is plain that the diagonal scaling strategy (2.2) can be applied to any Toeplitz matrix in order to shift the generating function along the x-axis. Furthermore, as long as we have only a single zero $x_0 \in]-\pi, \pi]$, the whole algorithm is simplified by shifting x_0 to the origin. Then we can use the same kind of transfer operators in every step – i.e., standard prolongations and restrictions according to $b(x) = (1 + cos(x))^k$.

2.5. Natural coarse grid operator. For the rest of section 2 we shall assume that our Toeplitz matrices are related to a nonnegative generating function with a unique zero $x_0 \in]-\pi, \pi]$. (Note that this does not in general include the matrices from Example 2.)
In the previous subsections we have presented a number of arguments for scaling the Toeplitz system with the diagonal matrices $Q_{1,2} = diag(1, e^{\pm ix_0}, e^{\pm 2ix_0}, ...)$, before treating it by multigrid. However, we have not yet presented a way to handle the problem that our Galerkin coarse grid operators lose their Toeplitz structure. R. Chan and collaborators already pointed out in [9] that if we use standard linear interpolation (according to $b(x) = 1 + cos(x)$) then we can only be sure to preserve Toeplitz structure on all the coarse levels if the size n of the matrix is of the form $n = 2^q - 1$ (with q integer). Otherwise perturbations of low rank can be introduced. But note that this loss of Toeplitz structure may cause severe difficulties when we go down to lower levels.

There is a very simple result. **First scale the matrix according to (2.2) – and then employ a natural coarse grid operator!** In any event, let us start from scratch. Going back to the earliest papers on multigrid (such as, e.g., [2]), researchers did not think in terms of Galerkin coarse grid operators. Instead they used a natural coarse grid operator based on an appropriate rediscretization of the underlying partial differential equation. We wish to emphasize that using a natural coarse grid operator is still the most popular choice among multigrid practitioners. In particular, the recent 600-page monograph by Trottenberg, Oosterlee and Schüller [37] deals almost entirely with this type of coarse grids. Galerkin coarse grid operators are mainly preferred from an algebraic viewpoint for their superior stability properties, e.g., in connection with partial differential equations with highly oscillatory or discontinuous coefficients (see e.g. [25]) – and, furthermore, the underlying variational principle facilitates proving convergence theorems.

But let us take a closer look at the Toeplitz problems in question. Considering the one-dimensional Laplace problem with the system matrices

$$A_n = (1/h^2) * tridiag(-1, 2, -1) \qquad \text{with} \quad h = \frac{1}{n},$$

it is well-known that natural coarse grid operators work out perfectly and convergence proofs are not difficult.

Now let us switch over to general Toeplitz matrices belonging to a generating function f that satisfies (cf. also [9], section 3)

$$\min_{\theta \in [-\pi,\pi]} \frac{f(\theta)}{1 - cos(\theta)} > 0. \tag{2.3}$$

If f has a zero of order 2, such as, e.g., $f(x) = x^2$, then we know that the corresponding sequence of Toeplitz matrices will have the same spectral properties as the 1D Laplace matrices from Example 1 (see [21]). This fact motivates the idea to simply mimic a multigrid algorithm with a natural coarse grid operator, i.e., just as we would do it for the Laplacian, our coarse level matrix is nothing but an appropriately scaled Toeplitz matrix of half size corresponding to the same generating function $f(x)$.

Remember that, in a multigrid algorithm with natural coarse grid operators for a second order finite difference discretization of the 1D Laplacian, the fine grid matrix and the matrix representation on the following coarse grid bear the different factors $(1/h)^2$ and $(1/2h)^2$, respectively. Hence if we prefer to multiply our discrete equations by the factor h^2, then it is evident and well-known that care must be taken to use the correct factor $4h^2$ on the coarse grid, i.e., we must not forget to multiply the residual r^{coarse} in Algorithm 1, step (c), with the factor $fac = 4$ (see e.g., [37], sec. 2.7).

When we mimic this procedure for a general Toeplitz matrix generated by a function f satisfying (2.3), then the appropriate factor to be used for scaling the defects needs to reflect the order ρ of the zero of the function f, i.e. $fac = 2^\rho$. (And, as there is no physical grid associated with our Toeplitz matrix, we will most certainly prefer to have this factor on the right hand sides of our equations.) In other words, our multigrid algorithms with natural coarse grid operators need to take into account carefully the order of the zero of the generating function involved, such that defects are scaled correctly.

In order to show the computational feasibility of our approach, we will deliberately choose the matrix sizes in our numerical experiments in most cases to be of the forms $n = 2^q$ or $n = 2^q + 1$, i.e., we are mainly studying cases in which the Toeplitz structure would be lost on the coarse levels if a Galerkin operators were employed. All the numerical results presented in the subsections 2.6, 3.3 and 4.3 were obtained with multigrid W-cycles with the coarsest level matrix always chosen to be a 7×7, 8×8 or 9×9-matrix, respectively. On this coarsest level we perform a direct solve. Finally, we wish to emphasize once again that our idea of using natural coarse grid operators crucially depends on the fact that our single zero $x_0 \in]-\pi, \pi]$ of finite order is indeed shifted to the origin, for otherwise, (2.3) could not be satisfied.

2.6. Numerical results. In our numerical experiments we will show that the new multigrid algorithms with natural coarse grid operators perform very well. We will only give numerical results for dense Toeplitz matrices as the loss of Toeplitz structure on coarser levels is only an issue in this case.

Example 3: Generating functions for dense Toeplitz matrices with a single zero of order at most two at the origin.
(a) $f_1(x) = x^2$ with the Fourier expansion

$$f_1(x) = \frac{\pi^2}{3} + 4 * \sum_{j=1}^{\infty} \frac{(-1)^j}{j^2} cos(j * x);$$

(b) $f_2(x) = (x/4) * sin(x/2)$ with the Fourier expansion

$$f_2(x) = \frac{1}{\pi} + \frac{2}{\pi} * \sum_{j=1}^{\infty} \frac{(-1)^j * (4 * j^2 + 1)}{(2 * j - 1)^2 * (2 * j + 1)^2} cos(j * x);$$

(c) $f_3(x) = |x|$ with the Fourier expansion

$$f_3(x) = \frac{\pi}{2} - \frac{4}{\pi} * \sum_{j=1}^{\infty} \frac{1}{(2 * j - 1)^2} cos((2 * j - 1) * x);$$

(d) $f_4(x) = |sin(x/2)|$ with the Fourier expansion

$$f_4(x) = \frac{2}{\pi} - \frac{4}{\pi} * \sum_{j=1}^{\infty} \frac{1}{(2 * j - 1) * (2 * j + 1)} cos(j * x).$$

Note that for the matrices from Examples 3 (a)–(d) the celebrated circulant preconditioner [10] will in general not lead to optimal computational performance, because the underlying functions are not strictly positive. The matrices from Examples 3 (a) and (b) have zeros of order 2 and hence – just as we would do it for the Laplacian – we scale the residuals with a factor 4. For Examples 3 (c) and (d) we regard the zeros to be of order 1 and hence multiply the defects by 2. According to the existing theory, the prolongation operator to be used for multigrid treatment of all the matrices from Example 3 is linear interpolation corresponding to $b(x) = 1 + cos(x)$.

number of unknowns	513	1025	2049	4097	8193	16385
$f_1(x) = x^2$	9	9	9	9	9	9
$f_2(x) = (x/4) * sin(x/2)$	11	12	11	12	12	12
$f_3(x) = \|x\|$	5	5	5	5	5	5
$f_4(x) = \|sin(x/2)\|$	7	7	7	7	7	7

Table 4. Iteration numbers for the preconditioned conjugate gradient method for the dense matrices from Example 3: We use a W-cycle for preconditioning.

number of unknowns	512	1024	2048	4096	8192	16384
$f_1(x) = x^2$	11	12	12	12	12	12
$f_2(x) = (x/4) * sin(x/2)$	12	12	12	12	12	12
$f_3(x) = \|x\|$	6	6	6	6	6	6
$f_4(x) = \|sin(x/2)\|$	5	5	5	5	5	5

Table 5. Iteration numbers for W-cycle solvers for the dense matrices from Example 3.

Tables 4 and 5 show very clearly that our new multigrid algorithms lead to fast convergence with iteration counts independent of the number of unknowns involved. Hence they give very efficient solvers of optimal computational complexity $O(n \log n)$. Furthermore, our multigrid method has no problem at all with the fact that $f_3(x) = |x|$ and $f_4(x) = |sin(x/2)|$ are not differentiable at the origin. On the contrary, the fact that the order of the zero is lower than 2 leads to even faster convergence.

Finally, let us also take a brief look at zeros of higher order.

Example 4: Generating functions for dense Toeplitz matrices with a single zero of order higher than 2 at the origin.
(a) $f_5(x) = x^4$ with the Fourier expansion

$$f_5(x) = \frac{\pi^4}{5} + \sum_{j=1}^{\infty} \left(\frac{48}{(2*j-1)^4} - \frac{8\pi^2}{(2*j-1)^2}\right) \cos((2*j-1)*x) + \\ + \sum_{j=1}^{\infty} \left(\frac{2\pi^2}{j^2} - \frac{6}{2*j^4}\right) \cos(2*j*x);$$

(b) $f_6(x) = |x|^3$ with the Fourier expansion

$$f_6(x) = \frac{\pi^3}{4} + \frac{2}{\pi} * \sum_{j=1}^{\infty} \left(\frac{12}{(2*j-1)^4} - \frac{3\pi^2}{(2*j-1)^2}\right) \cos((2*j-1)*x) + \\ + \sum_{j=1}^{\infty} \frac{6\pi}{(2*j)^2} \cos(2*j*x).$$

For the matrices generated by f_5 we deal with a zero of fourth order and hence scale the defects by a factor of 16. As for f_6 the third derivative is discontinuous – we treat it as a zero of third order and use the scaling factor 8.

Now equation (1.3) tells us to employ a prolongation operator corresponding to $b(x) = (1 + cos(x))^2$ – this will be abbreviated by “Prol. squared” in the following tables. However, it might be very interesting also to try standard linear interpolation corresponding to $b(x) = 1 + cos(x)$ (abbreviated by “Prol. simple”):

number of unknowns	511	1023	2047	4095	8191	16383
$f_5(x) = x^4$, Prol. simple	29	29	29	29	29	29
$f_5(x) = x^4$, Prol. squared	33	33	33	33	33	33
$f_6(x) = \|x\|^3$, Prol. simple	14	14	14	14	14	14
$f_6(x) = \|x\|^3$, Prol. squared	19	19	19	19	19	19

Table 6. Iteration numbers for W-cycle solvers for the dense matrices from Example 4.

We observe that in practice it is sufficient to use standard linear interpolation for prolongation and restriction. Surprisingly, in this case results are even better if we use the transfer operators corresponding to $b(x) = 1 + cos(x)$ – although the convergence theory for the two-grid case presented in [31] clearly tells us to use $b(x) = (1 + cos(x))^2$. In any case, this confirms the well-known advice of multigrid practitioners that higher order interpolations might frequently not pay off. Finally, note that level-independent convergence for the matrices from Example 4 has yet only been established for two-grid solvers (see [31]) – but we also observe optimality for our W-cycle solvers.

2.7. Conclusions. In this section we have presented a new and efficient way to solve Toeplitz systems corresponding to an underlying function having a single zero $x_0 \in]-\pi, \pi]$ of finite order. One first scales the matrix with diagonal matrices of the form $diag(1, e^{\pm ix_0}, e^{\pm 2ix_0}, ...)$ in order to shift the zero to the origin and then solves the scaled system by a multigrid algorithm employing a natural coarse grid operator.

3. Generating functions with equidistant zeros of finite order.

3.1. Equidistant zeros. The case when the generating function has more than one zero of finite order is certainly more complicated. Let us start with a fairly simply example which has first been addressed in [9]. For generating functions of the form $f(x) = 1 \mp cos(m * x)$, m integer, it is no longer appropriate to use prolongation operators of the form (1.3). Instead, Chan, Chang and Sun [9] use prolongations corresponding to $b(x) = 1 \pm cos(m * x)$ and their multigrid algorithms based on Galerkin coarsening work out perfectly. Choose $f(x) = 1 - cos(m * x)$ and observe that this idea also matches the heuristics (1.2)

$$f_2(x) = (1+cos(\frac{m*x}{2}))^2 *(1-cos(\frac{m*x}{2}))+(1+cos(\frac{m*x}{2}+\pi))*(1-cos(\frac{m*x}{2}+\pi))^2 =$$

$$= (1 + cos(\frac{m*x}{2}))^2 * (1 - cos(\frac{m*x}{2})) + (1 - cos(\frac{m*x}{2}))^2 * (1 + cos(\frac{m*x}{2})) =$$

$$= 2 * (1 - (cos(\frac{m*x}{2}))^2) = 1 - cos(m * x) \ .$$

Note that for this sparse example the appropriate choice for the prolongation operator could also be written in the form $b(x) = f(x+\pi)$ whenever m is odd. This choice is closely related to the so-called “Matrix Multilevel Method” [25] recently proposed by the authors for much more general sparse matrices.

However, the prolongation operators $b(x) = 1 + cos(m * x)$ are applicable in case the generating function of our Toeplitz matrix has m equidistant zeros of order at most 2 in the interval $[0, 2\pi[$, one of which needs to be at the origin, i.e., the generating function has the zeros $x_0 = 0, x_1 = \frac{2\pi}{m}, \ldots, x_{m-1} = \frac{2*(m-1)*\pi}{m}$.

We can again apply our reasoning from section 2. In case none of our m equidistant zeros of order at most 2 is at the origin, we first scale the matrix according to (2.2). Afterwards we observe that $f(x) = 1 - cos(m * x)$ can again be interpreted as a discretization of the 1D Laplacian – and, analogously to (2.3), we can carry over our approach to Toeplitz matrices associated with a generating function f satisfying

$$\min_{\theta \in [-\pi,\pi]} \frac{f(\theta)}{1 - cos(m * \theta)} > 0. \tag{3.1}$$

In other words, we are able to use multigrid algorithms with natural coarse grid operators and the prolongations $b(x) = 1 + cos(m * x)$ for functions satisfying (3.1).

3.2. A block interpretation. The above case also leads us to an interesting observation. Let us take a look at the matrix connected with $f(x) = 1 - cos(m * x)$ and the corresponding transfer operators $b(x) = 1 + cos(m * x)$. Now we can interpret this also in terms of matrix valued functions.

$$f(x) = I_m - cos(I_m * x) = I_m * (1 - cos(x))$$

is treated by prolongations of the form

$$b(x) = I_m + cos(I_m * x) = I_m * (1 + cos(x)),$$

with I_m denoting the m-by-m identity matrix. Thus we can view this case as standard multigrid applied to Block Toeplitz matrices with m-by-m blocks. By inserting block matrices different from the identity we can carry over this idea to general Block Toeplitz matrices (i.e., also without Toeplitz blocks). This will be presented later in a separate paper [26].

However, note that the strategy outlined in subsection 3.1 also applies to other cases, e.g., Toeplitz matrices generated by $f(x) = x * sin(x)$ which are not covered by the above block interpretation (see Example 5(c) in subsection 3.3 for the Fourier expansion). As $f(x) = x * sin(x)$ has the two zeros $x_0 = 0$ and $x_1 = \pi$ we can interpret the appropriate prolongation

$$b(x) = (1 - cos(x - x_0)) * (1 - cos(x - x_1)) = \frac{1}{2} * (1 - cos(2 * x)) \tag{3.2}$$

analogously to (1.3) as the product of the two prolongations corresponding to x_0 and x_1. This interpretation has previously been given by Serra in [31], although he has not published any numerical experiments to confirm it.

3.3. Numerical results.

In the following we will test our multigrid algorithms employing natural coarse grid operators for problems with equidistant zeros in $[0, 2\pi[$.

Example 5: Generating functions for dense Toeplitz matrices with two zeros $x_0 = 0$ and $x_1 = \pi$ of order at most two.
(a) $f_7(x) = x^2 * (x - \pi)^2$ (– defined on $[0, \pi]$ and then evenly extended to $[-\pi, 0[$ –) with the Fourier expansion

$$f_7(x) = \frac{\pi^4}{30} - \sum_{j=1}^{\infty} \frac{6}{2 * (2 * j)^4} * cos(2 * j * x),$$

(b) $f_8(x) = |sin(x)|$ with the Fourier expansion

$$f_8(x) = \frac{2}{\pi} - \frac{4}{\pi} * \sum_{j=1}^{\infty} \frac{2}{(2 * j - 1) * (2 * j + 1)} * cos(2 * j * x),$$

(c) $f_9(x) = x * sin(x)$ with the Fourier expansion

$$f_9(x) = 1 - \frac{1}{2} cos(x) - 2 * \sum_{j=2}^{\infty} \frac{(-1)^j}{(j - 1) * (j + 1)} * cos(j * x).$$

f_7 has two isolated zeros of order 2 and hence we expect to treat as like $f(x) = 1 - cos(2 * x)$ and use the factor 4 for scaling the defects. For f_8 the zeros are of order 1 and defects are multiplied by 2 when going down to coarser levels. f_9 is the most challenging example. The zero $x_0 = 0$ has order 2, whereas the zero $x_1 = \pi$ has order 1. As the arithmetic mean of the orders of the zeros is 1.5 we employ the scaling factor $2^{1.5} = 2\sqrt{2}$.

number of unknowns	513	1025	2049	4097	8193	16385	32769
$f_7(x) = x^2 * (x-\pi)^2$	11	12	12	12	12	12	12
$f_8(x) = \lvert sin(x) \rvert$	5	5	5	5	5	5	5
$f_9(x) = x * sin(x)$	9	9	9	9	9	9	9

Table 7. Iteration numbers for W-cycle solvers for the dense matrices from Example 5.

We observe optimal computational behaviour of our multigrid algorithms for all problems from Example 5. Thus we can confirm numerically that the multigrid algorithms suggested in section 2 carry over to the case of generating functions with m equidistant zeros in $[0, 2\pi[$.

4. Image Deblurring.

Today image processing is may be the most eminent field of applications of Toeplitz matrices (see, e.g., [5], [6]). The best known example are dense matrices from image deblurring.

4.1. The model.

Let us start with an idealized model for one-dimensional image deblurring. There we want to solve an integral equation of the first kind of the form

$$\mathcal{K}u(x) = \int_\Omega k(x-x')u(x')dx', \tag{4.1}$$

with a convolution kernel of the form $k(x) = exp(-x^2/\sigma^2)$, with $\sigma \in]0,1[$ on the interval $\Omega = [-p,p]$. The operator $\mathcal{K}$ is often referred to as “Gaussian blur”.
We can now discretize this integral equation on a uniform grid via the midpoint quadrature rule and will end up with a Toeplitz matrix (see, e.g., [5], section 4.4, or [27], chapter 2). We work with the mesh size $h = \frac{2p}{n}$ and the midpoints

$$x_j = -p + (2*j-1)*\frac{h}{2}, \qquad j = 1, 2, \ldots, n.$$

Then we use the midpoint quadrature rule and the convolution operator (4.1) translates into

$$\mathcal{K}u(x_i) = \int_{-p}^{p} k(x_i - x')u(x')dx' \approx \sum_{j=0}^{n-1} k(x_i - x_j)u(x_j)h \approx [K\bar{u}]_i,$$

with the symmetric positive definite Toeplitz matrix

$$K = h * toeplitz(k(0), k(1*h), \ldots, k((n-1)*h)),$$

and the vector $\bar{u} = [u(x_1), \ldots, u(x_n)]^T$.

It is well-known that the blurring matrices K are highly ill-conditioned and hence deblurring algorithms are extremely sensitive to noise [19]. In other words, we are dealing with an inverse problem and hence we need to regularize.
In the following we shall only investigate Tikhonov regularization [15], [38]. If we minimize the Tikhonov functional in the L_2-norm our system matrix becomes

$$L = K + \lambda I, \tag{4.2}$$

with regularization parameter λ. If we use the H_1-norm instead, then the system matrix becomes

$$L = K + \lambda\triangle,$$

with $\triangle = tridiag(-1, 2, -1)$ denoting the one-dimensional Laplacian.

For simplicity, we will in the following only discuss the L_2-based case (4.2). However, carrying over our reasoning to the H_1-based case is straightforward.

4.2. Transfer operators and smoothers.

So far, there has essentially been only one paper by R. Chan, T. Chan and J. Wan [7] on multigrid methods for image deblurring. In the following we shall attempt to put their observations and algorithms into the context of multigrid methods for Toeplitz matrices and the so-called multigrid method of the second kind.

The paper [7] reports that, for the system matrices in question, standard relaxation methods like Richardson fail as smoothers. To overcome this difficulty, a semi-iterative smoother is used. They employ conjugate gradients with optimal cosine transform preconditioner [8]. The approach does not make any explicit use of Toeplitz structure; furthermore, standard prolongations and restrictions and Galerkin coarsening are used without any further explanation (see [7], p. 70).

Do the methods presented in section 2 relate to this case? First of all, we need to state that obviously there is no underlying function connected with our matrices K. However, if we simply assign functions to matrices of different size, we observe that our matrices behave as if we were dealing with a “single zero of infinite order” located at $x_0 = \pi$.

However, this information does not help us to devise multigrid transfer operators. The reasoning associated with (1.3) is no longer applicable as we certainly do not want to use anything like $b(x) = (1 - cos(x))^{\infty}$.

On the other hand, reasonable information on the prolongation operators P and the restriction operators R is given if we look at the ideas from [7] in the context of the multigrid method of the second kind, proposed by Hackbusch (see [22], chapter 16), i.e., W-cycle algorithms for the efficient numerical solution of Fredholm integral equations of the second kind.

As long as we discretize our integral operator (3.1) via the midpoint quadrature rule the discretization error will be of order $O(h^2)$ and in that case it has been pointed out by Hackbusch [22], p. 308, why piecewise linear interpolation – i.e. corresponding to $b(x) = 1 + cos(x)$ – should be used for prolongation. For the restrictions R trivial injections are shown to be the canonical choice – since they satisfy $R * P = I$ on every level – but it is pointed out why weighted restrictions with $P = R^T$ on every level come out to be suited as well. (See again [22], pp. 306, for the details and underlying consistency results).

From the Toeplitz point of view we would like to emphasize that Hackbusch's theory states clearly that it would be totally inappropriate to scale the system via the diagonal matrices $diag(1, e^{\pm i\pi}, e^{\pm 2\pi}, ...)$ in order to move the zero $x_0 = \pi$ to the origin for the discrete integral operators from (4.1). In this case shifting the zero would mean violating consistency conditions and thus would corrupt the whole algorithm.

Let us view our system matrix $L = K + \lambda * I$ in terms of a Fredholm integral equation of the second kind. Setting $\tilde{k}(x, y) := -k(x, y)$ and obtaining a discretization $\tilde{K}$ via the midpoint quadrature rule we can rewrite (4.2) as

$$L = \lambda * I - \tilde{K},$$

in the standard form of an integral equation of the second kind.

In standard applications of integral equations of the second kind we mostly deal with the case $\lambda = 1$ and then the multigrid method of the second kind – which usually employs one step of Richardson (pre)smoothing – works out perfectly. However, if λ becomes small, then the Richardson smoother is strongly divergent and this can cause severe problems for the W-cycle solvers, because the smoothing steps ”worsen” the approximate solution in every iteration (and on each level). Thus the standard version of the multigrid method of the second kind may also diverge rapidly as its convergence factor depends quadratically on $(1/\lambda)$ (see [22], sec. 16.2.1).

Furthermore, there are variants of the multigrid method of the second kind which are designed especially for the case of small λ (and also employ Richardson smoothing). We mention in particular a variant by Hemker and Schippers [24] and a variant by Hackbusch which basically omits every second smoothing step on all the coarse levels (see [22], section 16.2.3).

However, the convergence factors of the improved versions still depend linearly on $(1/\lambda)$ and we have checked very carefully in plenty of numerical experiments that they do not lead to convergent algorithms for the small regularization parameters λ we need to deal with in image deblurring problems. Instead, we confirm the need for a semi-iterative smoother according to [7]. We know that an appropriate smoother would be a scheme that never diverges for problems of the form (4.2) – and (preconditioned) conjugate gradients are an algorithm known to have this property. For more details and analysis we refer to the upcoming Ph.D. thesis [34].

As we are not using a stationary iterative smoother, our multigrid cycles are no longer available for standard Krylov subspace solvers. In fact, our multigrid preconditioner changes during the iteration as a result of the CG-smoothing. However, we could use so-called ”flexible” Krylov subspace methods, such as, e.g., the FGMRES variant by Saad [29], which allows to use a different preconditioner in every iteration step. But as our system matrix (4.2) is symmetric positive definite, symmetric versions of flexible Krylov solvers would also come into account (see [12]).

Finally, we emphasize that again we prefer to use a natural coarse grid operator – instead of Galerkin coarsening used by R. Chan, T. Chan and J. Wan (see [7], p. 70) – in order to preserve Toeplitz structure on the coarse levels. We will point out why this can be regarded as a major algorithmic improvement.

4.3. Numerical results. In the following we give numerical results for discretizations of the one-dimensional deblurring problem on $\Omega = [-1, 1]$ using L_2-based Tikhonov regularization with different regularization parameters λ. We have implemented a multigrid algorithm using conjugate gradients with the optimal circulant preconditioner [10] as a smoother. In all our tables the W-cycle solvers employ two pre-smoothing and no post-smoothing steps.

Within our multigrid cycles we also wanted to check whether it is preferable to use trivial injection for restriction or if the standard full-weighting operator gives better results. We will distinguish between these two choices by the abbreviations “MG (triv)” and “MG (lin)”, respectively.

The following tables 8a, 8b and 8c list the iteration counts for W-cycle solvers with natural coarse grid operators and two pre-smoothing steps of preconditioned CG for the choice $\sigma = 0.1$ in (4.1) and for the three regularization parameters $\lambda = 1e - 3$, $\lambda = 1e - 4$ and $\lambda = 1e - 5$. For comparison we also list the iteration counts of CG with circulant preconditioning as a stand-alone solver, abbreviated by ”Circ-CG”.

number of unknowns	512	1024	2048	4096	8192	16384	32768
MG (triv)	5	5	4	3	3	2	3
MG (lin)	5	4	4	3	3	3	3
Circ-CG	9	9	9	9	9	9	9

Table 8a. Iteration counts for 1D deblurring problem (4.2) with $\lambda = 1e - 3$.

number of unknowns	512	1024	2048	4096	8192	16384	32768
MG (triv)	6	5	5	5	4	3	3
MG (lin)	9	7	6	5	5	4	4
Circ-CG	15	15	16	15	15	15	15

Table 8b. Iteration counts for 1D deblurring problem (4.2) with $\lambda = 1e - 4$.

number of unknowns	512	1024	2048	4096	8192	16384	32768
MG (triv)	10	10	8	6	4	4	3
MG (lin)	37	26	17	12	9	7	6
Circ-CG	27	25	27	26	26	26	26

Table 8c. Iteration counts for 1D deblurring problem (4.2) with $\lambda = 1e - 5$.

Tables 8a to 8c show that by using circulant-preconditioned conjugate gradients as a smoother we can obtain the typical convergence behaviour of the multigrid method of the second kind also for the case of very small λ, i.e., iteration numbers even decline for a larger number of unknowns. Hence the idea to employ a semi-iterative smoother can be seen as an extension of the multigrid method of the second kind, in order to handle very small λ. For more details we again refer to the Ph.D. thesis of the second author [34].

Furthermore, we observe that using trivial injection as the restriction operator leads to smaller iteration counts in our W-cycle solvers. We also emphasize that the multigrid structure itself comes out to be sensible and to pay off from the point of view that for large problems and small λ our W-cycle solvers need significantly less PCG smoothing steps on the finest level than circulant-preconditioned CG used as a stand-alone solver.

We would like to admit right now that the comparisons with fixed regularization parameters λ reported in the previous tables may seem slightly questionable from the point of view of solving an inverse problem from signal or image processing. Certainly, the regularization parameter would normally not be picked without looking at the matrix first. However, we are deliberately presenting our numerical results in this way in order to focus on the strong connection of our algorithms and the multigrid method of the second kind.

We would finally like to stress that by obtaining the coarse grid matrices via rediscretization we have improved the algorithms from the paper [7] considerably. Note that it is problematic to compute the optimal T. Chan circulant preconditioner – which is needed within our PCG smoothing – for a general dense matrix (such as, e.g., a coarse grid matrix obtained via the Galerkin approach) with a computational effort less than $O(n^2)$; on the other hand, as long as we construct our coarse level matrices via rediscretizations we will have Toeplitz structure on all levels and can set up our preconditioner with $O(n)$ efforts as explained in [10].

5. Twodimensional case: Block Toeplitz matrices with Toeplitz Blocks.

5.1. Positive definite problems generated by nonnegative functions.

5.1. Positive definite problems generated by nonnegative functions. In the 2D-case we consider Block Toeplitz matrices with Toeplitz blocks (BTTB matrices) related to a function of the form

$$f(x,y) = \sum a_{j,k} e^{ijx} e^{iky},$$

e.g., $f(x,y) = 2 - cos(x) - cos(y)$ for the Laplacian discretized on the unit square by the 5-point stencil. The bad condition numbers of the matrices are again caused by the zeros $x = y = 0$ of $f(x,y)$.
We are in the simple case as long as the function f has only a unique isolated zero $(x_0, y_0) \in]-\pi, \pi]^2$. Then we can try to proceed with multigrid algorithms similar to section 2. For simplicity, let us first take a look at the case of a single isolated zero (x_0, y_0) of order 2. In a multigrid approach we can choose

$$b(x,y) = (cos(x_0) + cos(x)) * (cos(y_0) + cos(y)) \tag{5.1}$$

for prolongation and restriction. Note that this is nothing else than the Kronecker product of the corresponding 1D matrices. According to our heuristics (1.2) the function f_2 associated with the Galerkin coarse grid operator is the reduction of $\tilde{f}(x,y) = b(x,y)f(x,y)b(x,y)$ to every second coefficient relative to x and y. For the matrix this is nothing else than the projection onto every second row/column and row/column block, respectively. Therefore, we have,

$$f_2(x,y) = \frac{1}{4} * \left(\tilde{f}(\frac{x}{2}, \frac{y}{2}) + \tilde{f}(\frac{x}{2} + \pi, \frac{y}{2}) + \tilde{f}(\frac{x}{2}, \frac{y}{2} + \pi) + \tilde{f}(\frac{x}{2} + \pi, \frac{y}{2} + \pi) \right). \tag{5.2}$$

Hence, f_2 will have the isolated zero $(2x_0, 2y_0)$ – and the prolongation $b(x,y)$ needs to have the three zeros $(x_0 + \pi, y_0)$, $(x_0, y_0 + \pi)$ and $(x_0 + \pi, y_0 + \pi)$.
However, for BTTB matrices it is even more important to use a natural coarse grid operator instead of Galerkin coarsening. Again, for Galerkin coarsening and standard transfer operators we can only be sure to preserve BTTB structure on every coarse grid if the matrix size is of the form $n = (2^q - 1)^2$ (with q integer). More importantly, the perturbations introduced via Galerkin operators are no longer of low rank like in the Toeplitz case, but normally grow proportional to the matrix size.

However, the result is as simple as in section 2. For a single zero $(x_0, y_0) \in]-\pi, \pi]^2$ we can scale our linear system first via the matrices

$$I \otimes diag(1, e^{\pm ix_0}, e^{\pm 2ix_0}, ...) \qquad \text{and} \qquad diag(1, e^{\pm iy_0}, e^{\pm 2iy_0}, ...) \otimes I,$$

respectively, and thus shift the zero to the origin.

Then we can proceed as usual just as we would do for the 2D Laplacian. We can carry over virtually everything presented in subsection 2.5 from diagonal scalings to natural coarse grid operators via Kronecker products, as long as we have only a single zero.

Analogously to 2.5 we recall the fact that multigrid algorithms with natural coarse grid operators have been long been known to converge for two-dimensional Laplace problems. As in [36] we carry over our reasoning to employ a natural coarse grid operator to functions $f(x,y)$ satisfying

$$\min_{(x,y)\in[-\pi,\pi]^2} \frac{f(x,y)}{2 - cos(x) - cos(y)} > 0. \tag{5.3}$$

Note that (5.3) certainly includes non-separable generating functions, as e.g., $f(x,y) = 20 - 8 * cos(x) - 8 * cos(y) - 4 * cos(x) * cos(y)$, which corresponds to a 9-point discretization of the Laplacian on the unit square. However, we shall only give numerical results for separable problems in the following table. There we list iteration counts for W-cycle solvers with natural coarse grid operators for separable BTTB problems related to generating functions from Example 3. In all experiments reported in this section the coarsest level representation is a $2^4 \times 2^4$-matrix.

number of unknowns	$16 * 16$	$32 * 32$	$64 * 64$	$128 * 128$	$256 * 256$
$g_1(x,y) = x^2 + y^2$	14	14	14	14	14
$g_2(x,y) = x^2 + (y/4) * sin(y/2)$	23	24	24	24	24
$g_3(x,y) = \|x\| + \|y\|$	7	8	8	8	8
$g_4(x,y) = \|x/\pi\| + \|sin(y/2)\|$	8	9	9	10	10
$g_5(x,y) = x^2 + \|y\|$	15	15	15	15	15

Table 9. Iteration numbers for W-cycle solvers for BTTB systems related to the matrices from Example 3.

Again, our multigrid algorithms give efficient solvers of optimal computational complexity $O(n \log n)$. Note that for g_1 and g_2 we deal with a zero of order 2 and – as in the Laplace case – we scale the residuals with the factor 4, whereas for g_3 and g_4 the appropriate scaling factor is 2 since we are dealing with a zero of order 1. As for g_5 we assign the order 1.5 to the zero and scale the defects with the factor $2^{1.5} = 2\sqrt{2}$.

However, our approach runs into trouble as soon as there is more than a single zero of finite order. According to (5.1) and (3.2) we would need to build prolongations $b(x,y)$ incorporating all the zeros. However, this forces us to build prolongations which are much too dense. For example, for BTTB matrices belonging to the function $f(x,y) = 2 - cos(2x) - cos(2y)$, we would – in view of (5.1) – need to work with prolongations involving 8 ”elementary” factors corresponding to the 4 zeros $(0,0), (0,\pi), (\pi,0), (\pi,\pi)$. This does not lead to computationally feasible algorithms.

5.2. An algorithm for restoring images subject to atmospheric turbulence blur. The algorithms from section 4 carry over simply to practical (i.e. two-dimensional) image deblurring problems. There we are dealing with a Gaussian blur again, i.e., we need to solve an integral equation of the first kind of the form

$$\mathcal{K}u(x,y) = \int_\Omega k(x-x', y-y')u(x',y')dx'dy', \tag{5.4}$$

with a convolution kernel $k(x,y) = exp(-(x^2+y^2)/\sigma^2)$, with $\sigma \in]0,1[$ on the square $\Omega = [-p,p]^2$. This kernel models atmospheric turbulence blur and it is used in practice, e.g., for the restoration of satellite images.

Analogously to section 4, we discretize via midpoint quadrature and end up with a positive definite BTTB matrix K connected to a “single zero of infinite order” at $x_0 = (\pi,\pi)$. Then we can do Tikhonov regularization with respect to the L_2-norm and need to solve a linear system of the form $L = K + \lambda * I$.

In any event, as for the mathematical theory concerning regularization and the multigrid method of the second kind, the treatment of the two-dimensional convolution operator (5.4) does not differ at all from the one-dimensional case. Asin section 4 we can build efficient multigrid algorithms by employing conjugate gradients with either the BCCB preconditioner by T. Chan and J. Olkin [11] or the Block circulant extension preconditioner by Hanke and Nagy [23] as a smoother. Note that this idea can only lead to a practical $O(n \log n)$ image deblurring algorithm if we get our coarse grid operators via rediscretization. In particular, observe that the Block circulant extension preconditioner is only defined for BTTB matrices.

In our numerical tests we are dealing with a discretization of the two-dimensional deblurring operator (5.4) on $\Omega = [-1,1]^2$. Again, our W-cycle solvers employ two PCG steps for pre-smoothing and no post-smoothing.

This time more different variants are to be considered than in the 1D case. Within our multigrid algorithms we wish to test two different restrictions and two different BCCB preconditioners inside our PCG smoother. Like in subsection 6.2.4 “triv” will indicate that we use trivial injection for restriction whereas “lin” identifies the full weighting operator. As the BCCB preconditioner by T. Chan and J. Olkin [11] minimizes the Frobenius norm we indicate it by “FR-CI” whereas “EX-CI” stands for the block circulant extension preconditioner. Analogously, “FR-CI-CG” and “EX-CI-CG” identifty BCCB-preconditioned conjugate gradient methods as stand-alone solvers.

The following tables list the iteration counts for our W-cycle solvers with natural coarse grid operators for the choice $\sigma = 0.05$ in (5.4) and for the two regularization parameters $\lambda = 1e-4$ and $\lambda = 1e-5$. The linear systems are of the form (4.2), i.e. we use L_2-based Tikhonov regularization.

number of unknowns	$64*64$	$128*128$	$256*256$	$512*512$	$1024*1024$
MG (FR-CI, triv)	15	9	6	5	4
MG (FR-CI, lin)	20	12	8	5	5
FR-CI-CG	24	25	25	25	25
MG (EX-CI, triv)	7	6	5	5	4
MG (EX-CI, lin)	8	7	6	4	4
EX-CI-CG	18	17	17	17	17

Table 10a. Iteration counts for 2D deblurring problem (5.4) with $\lambda = 1e-4$.

number of unknowns	$64*64$	$128*128$	$256*256$	$512*512$	$1024*1024$
MG (FR-CI, triv)	42	32	21	14	9
MG (FR-CI, lin)	68	30	20	24	21
FR-CI-CG	39	41	44	43	44
MG (EX-CI, triv)	22	23	18	13	7
MG (EX-CI, lin)	16	16	21	19	17
EX-CI-CG	37	40	41	42	42

Table 10b. Iteration counts for 2D deblurring problem (5.4) with $\lambda = 1e-5$.

From the above tables we can observe the typical convergence behaviour of the multigrid method of the second kind. For fixed regularization parameter λ iteration counts decrease for larger matrix sizes. Again, our multigrid algorithms can also handle very small regularization parameters λ.

Furthermore, we also observe that trivial injection for restriction does a better job than full weighting. However, the differences are a little less striking than in the one-dimensional case. We also confirm the observation by Vogel (see [38], ch. 5) that the Block circulant extension preconditioner usually leads to slightly faster convergence than the optimal BCCB preconditioner from [11]. Due to its faster convergence it also gives the better smoother.

5.3. Indefinite Problems. The situation gets much more complicated if the condition $f(x, y) = 0$ has a whole curve $(x(t), y(t))$ as solution. Certainly, we can no longer "shift" the curve of zeros to the origin by scaling. For the Multigrid prolongation in view of (1.3) we need a function with zeros at $(x(t) + \pi, y(t))$, $(x(t), y(t) + \pi)$, and $(x(t) + \pi, y(t) + \pi)$. We can build such a function by setting

$$b(x, y) = f(x + \pi, y) * f(x, y + \pi) * f(x + \pi, y + \pi) \ .$$

Again, the disadvantage of this approach is that the resulting matrices connected to $f_2(x, y)$ are getting more and more dense – and we cannot expect to obtain a practical algorithm.

Let us take a look at shifted Laplacians with the underlying function of the form

$$f(x, y) = 2 - \alpha - cos(x) - cos(y).$$

For small α the curve described by $f(x(t), y(t)) = 0$ is nearly the circle around the origin with radius $\sqrt{2\alpha}$.

Asymptotically the eigenvalues of the BTTB matrix are given by (see e.g. [30])

$$f(x_j, y_j) = 2 - \alpha - cos(\frac{\pi j}{n+1}) - cos(\frac{\pi k}{n+1}) \approx \frac{\pi^2(j^2 + k^2)}{(n+1)^2} - \alpha \ \ , \ \ j, k = 1, ..., n \ .$$

As we are dealing with shifted 2D-Laplacians our matrices can be diagonalized by the 2D-Sine Transform matrix with $S_1 = \sqrt{\frac{2}{n+1}}(sin(\pi jk/(n+1))_{j,k=1}^n$, $S_2 = S_1 \otimes S_1$, and

$$S_2 \, BT \, S_2 = diag(\lambda_j + \lambda_k - \alpha),$$

where λ_j are the eigenvalues of the 1D-Laplacian. Hence, the eigenvalues are exactly given by

$$f(x_j, y_j) = 2 - \alpha - cos(\frac{\pi j}{n+1}) - cos(\frac{\pi k}{n+1}) \ \ , \ \ j, k = 1, ..., n,$$

and the eigenvectors related to the near-zero eigenvalues are of the form

$$sin(\pi jm/(n+1))_{m=1}^n \otimes sin(\pi km/(n+1))_{m=1}^n,$$

with

$$j^2 + k^2 \approx \alpha(n+1)^2/\pi^2. \tag{5.5}$$

Hence we have to design a method that can deal with the error components in these directions. For the same problem a very sophisticated and highly promising algorithm that is related to this idea has been introduced by Brandt and Livshits based on a totally different approach [3]. There, more than one coarse grid is employed in order to resolve the problematic error components.

Finally, we wish to emphasize that the above indefinite model problem should not be viewed as a Helmholtz problem. Helmholtz equations usually model scattering phenomena

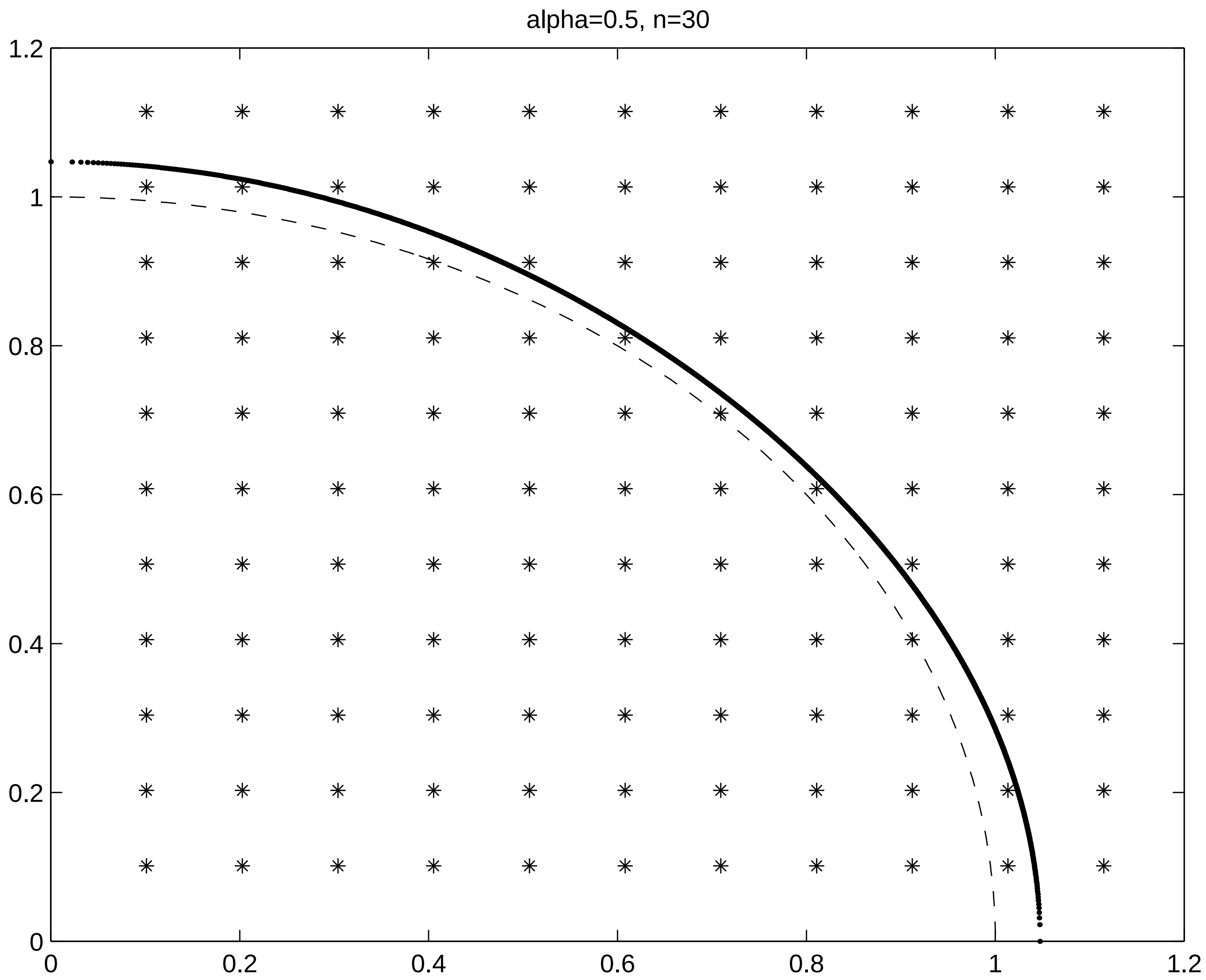

FIG. 5.1. *Curve $f(x, y) = 0$ and approximate circle*

on an exterior domain and the system matrices can never be expected to have Toeplitz structure. Furthermore, absorbing boundary conditions have to be introduced which turn the system complex-symmetric. For a state of the art algorithm for multigrid for Helmholtz problems that is also applicable to the non-constant coefficient case, we refer to recent work by Elman, Ernst and O'Leary [13], [14].

In Figure 5.1 we display the (j, k)-grid (5.5) with the curve $f(x, y) = 0$ and the approximating circle in the (x, y)-plane. Figure 5.2 shows the exact eigenvalues of the matrix on the mesh in the positive (x, y)-quadrant and the curve with $f(x, y) = 0$. The mesh also models the surface described by the function f.

6. Outlook and conclusions.

We have proposed new multigrid algorithms with natural coarse grid operators for Toeplitz matrices generated by nonnegative functions with a finite number of isolated equidistant zeros of finite order in the interval $]-\pi, \pi]$. That way we can

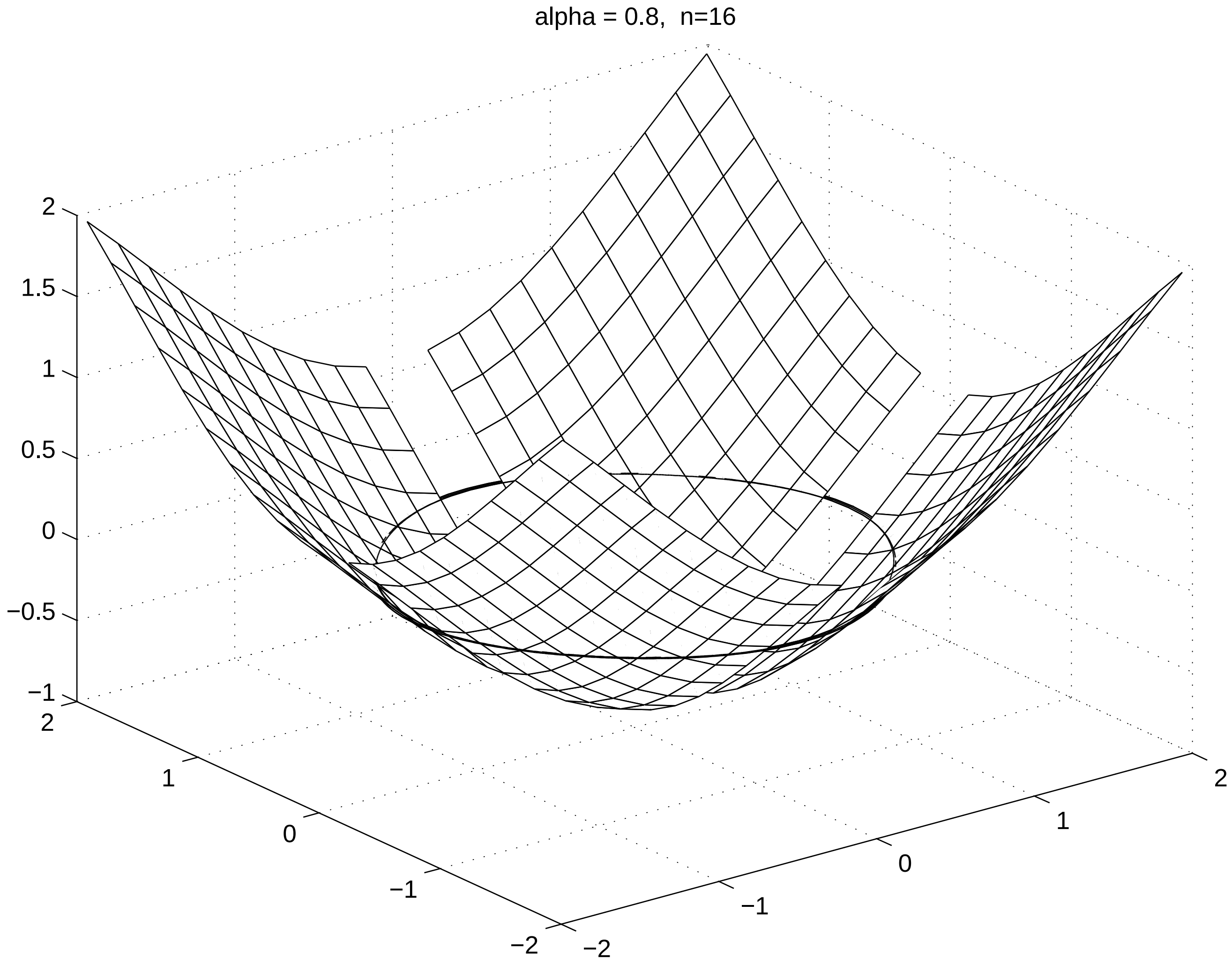

FIG. 5.2. *Function $f(x, y)$, mesh points as eigenvalues*

overcome the problem that usually Toeplitz structure is lost on the coarse levels if Galerkin coarsening is employed. The numerical experiments show clearly that these algorithms lead to optimal $O(n \log n)$ solvers. In the future we will try to give rigorous proofs of convergence for our algorithms. However, this comes out to be very difficult even in the two-grid case, since we are no longer using a Galerkin coarse grid operator, the seminal result of Ruge and Stüben ([28], p. 89) is no longer at our disposal.

In section 4 we have extended our ideas to Fredholm integral equations with a kernel of convolution type. By employing preconditioned conjugate gradients for smoothing, we have been able to extend the multigrid method of the second kind for the case that the parameter λ is very small. This idea certainly deserves further attention and we plan to investigate also on general Fredholm integral equations not necessarily related to convolution kernel.

Finally, we have carried over our ideas to symmetric BTTB matrices. If the matrix is related to a function with a single isolated zero $x_0 \in]-\pi,\pi]^2$, then usually the methods presented here are applicable. In particular, the need to use a natural coarse grid operator is even more prominent. Natural coarse grid operators also help to develop feasible multigrid algorithms with semi-iterative smoothing for image deblurring problems. However, if the function has a nontrivial curve of zeros then more advanced algorithms, possibly employing more than one coarse grid, need to be developed.

Acknowledgments. The authors wish to thank the referees whose suggestions helped to improve the paper significantly.

REFERENCES

[1] J. Bramble, J. Pasciak and J. Xu, *Parallel multilevel preconditioners*, Math. Comp., 55 (1990), pp. 1-22.

[2] A. Brandt, *Multilevel adaptive solution to boundary value problems*, Math. Comp., 51 (1977), pp. 389-414.

[3] A. Brandt and I. Livshits, *Wave-ray Multigrid Method for Standing Wave Equations*, Electron. Trans. Numer. Anal., 6 (1997), pp. 162-181. http://etna.mcs.kent.edu/vol.6.1997/pp162-181.dir/pp162-181.pdf.

[4] W. Briggs, *A Multigrid Tutorial*, SIAM, 1987.

[5] R. Chan and M. Ng, *Conjugate Gradient Methods for Toeplitz Systems*, SIAM Rev., 38 (1996), pp. 427-482.

[6] R. Chan and M. Ng, *Scientific Applications of Iterative Toeplitz Solvers*, Calcolo, 33 (1996), pp. 249-267.

[7] R. Chan, T. Chan and J. Wan, *Multigrid for Differential-Convolution Problems Arising from Image Processing*, in Proceedings of the Workshop on Scientific Computing, G. Golub, S.H. Lui, F. Luk and R. Plemmons, eds., Springer, 1999, pp. 58-72.

[8] R. Chan, T. Chan and C. Wong, *Cosine Transform Based Preconditioners for Total Variation Deblurring*, IEEE Trans. Image Proc., 8 (1999), pp. 1472-1478.

[9] R. Chan, Q. Chang and H. Sun, *Multigrid method for Ill-conditioned Symmetric Toeplitz Systems*, SIAM J. Sci. Comput., 19 (1998), pp. 516-529.

[10] T. Chan, *An Optimal Circulant Preconditioner for Toeplitz Systems*, SIAM J. Sci. Statist. Comput., 9 (1988), pp. 766-771.

[11] T. Chan and J. Olkin, *Circulant preconditioners for Toeplitz-block matrices*, Numer. Algorithms, 6 (1994), pp. 89-101.

[12] T. Chan, E. Chow, Y, Saad, M. Yeung, *Preserving symmetry in preconditioned Krylov subspace methods*, SIAM J. Sci. Comput., 20 (1999), pp. 568-581.

[13] H. Elman and O. Ernst, *Numerical experiences with a Krylov-enhanced multigrid solver for exterior Helmholtz problems*, in Mathematical and numerical aspects of wave propagation (Santiago de Compostela, 2000), SIAM, 2000, pp. 797–801.

[14] H. Elman, O. Ernst and D. O'Leary, *A multigrid method enhanced by Krylov subspace iteration for discrete Helmholtz equations*, SIAM J. Sci. Comput., 23 (2001), pp. 1290-1314.

[15] H. Engl, M. Hanke and A. Neubauer, *Regularization of Inverse Problems*, Kluwer Academic Publishers, 1996.

[16] G. Fiorentino and S. Serra, *Multigrid methods for Toeplitz matrices*, Calcolo, 28 (1992), pp. 283-305.

[17] G. Fiorentino and S. Serra, *Multigrid methods for symmetric positive definite block Toeplitz matrices with nonnegative generating functions*, SIAM J. Sci. Comput., 17 (1996), pp. 1068-1081.

[18] G. Fiorentino and S. Serra, *Multigrid methods for indefinite symmetric Toeplitz matrices*, Calcolo, 33 (1996), pp. 223–236.

[19] R. Gonzalez and R. Woods, *Digital Image Processing*, Addison-Wesley, 1992.

[20] A. Greenbaum, *Iterative Methods for Solving Linear Systems*, SIAM, 1997.

[21] U. Grenander and G. Szegö, *Toeplitz Forms and Their Applications*, Second Edition, Chelsea, 1984.

[22] W. Hackbusch, *Multigrid Methods and Applications*, Springer, 1985.

[23] M. Hanke and J. Nagy, *Toeplitz approximate inverse preconditioner for banded Toeplitz matrices*, Numer. Algorithms, 7 (1994), pp. 183-199.

[24] P. Hemker and H. Schippers, *Multiple grid methods for the solution of Fredholm integral equations of the second kind*, Math. Comp., 36 (1981), pp. 215–232.

[25] T. HUCKLE AND J. STAUDACHER, *Matrix Multilevel Methods and Preconditioning*, BIT, 42 (2002), pp. 541–560.

[26] T. HUCKLE AND J. STAUDACHER, *Multigrid methods for general Block Toeplitz matrices*, preprint, submitted to Numer. Linear Algebra Appl., September 2002.

[27] K. RILEY, *Two-level preconditioners for regularized ill-posed problems*, Ph.D. thesis, Department of Mathematics, University of Montana at Bozeman, 1999.

[28] J. RUGE AND K. STÜBEN, *Algebraic Multigrid*, In Multigrid Methods, S. F. McCormick, ed., SIAM, 1987, pp. 73-130.

[29] Y. SAAD, *A flexible inner-outer preconditioned GMRES algorithm*, SIAM J. Sci. Comput., 14 (1993), pp. 461-469.

[30] S. SERRA, *Conditioning and solution of Hermitian (block) Toeplitz systems by means of preconditioned conjugate gradient methods*, in Proceedings in Advanced Signal Processing Algorithms, Architectures, and Implementations - SPIE conference, F. Luk, ed., SPIE, 1995, pp. 326-337.

[31] S. SERRA CAPIZZANO, *Convergence analysis of two-grid methods for elliptic Toeplitz/PDEs Matrix-sequences*, Numer. Math., Electronic Version, DOI 10.0007/s002110100331 (15/11/2001).

[32] S. SERRA CAPIZZANO AND C. TABLINO POSSIO, *Preliminary Remarks on Multigrid Methods for Circulant Matrices*, in Numerical Analysis and Its Applications, Second International Conference, NAA 2000, Rousse, Bulgaria, L. Vulkov, J. Wasniewski, P. Yalamov, eds., LNCS 1988, Springer, 2000, pp. 152-159.

[33] S. SERRA CAPIZZANO AND C. TABLINO POSSIO, *Multigrid methods for multilevel circulant matrices*, preprint (2001), submitted to SIAM J. Sci. Comput., 16 pages.

[34] J. STAUDACHER, *Multigrid methods for matrices with structure and applications in image processing*, Ph.D. thesis, Department of Computer Science, Technical University Munich, submitted April 2002. To be made available online before September 2002 via
http://www5.informatik.tu-muenchen.de/persons/staudacj.html

[35] H. SUN, R. CHAN AND Q. CHANG, *A note on the Convergence of the Two-grid method for Toeplitz Systems*, Comput. Math. Appl., 34 (1997), pp. 11-18.

[36] H. SUN, X. JIN AND Q. CHANG, *Convergence of the multigrid method for ill-conditioned Block Toeplitz systems*, BIT, 41 (2001), pp. 179-190.

[37] U. TROTTENBERG, C. OOSTERLEE AND K. SCHÜLLER, *Multigrid*, Academic Press, 2001.

[38] C. VOGEL, *Computational Methods for Inverse Problems*, SIAM, 2002.

[39] X. ZHANG, *Multilevel Schwarz methods*, Numer. Math., 63 (1992), pp. 521-539.