Electronic Transactions on Numerical Analysis.
Volume 55, pp. 341–364, 2022.

ISSN 1068–9613.
DOI: 10.1553/etna_vol55s341

ETNA
Kent State University and
Johann Radon Institute (RICAM)

EXPLOITING COMPRESSION IN SOLVING DISCRETIZED LINEAR SYSTEMS*

ERIN CARRIER† AND MICHAEL T. HEATH‡

Abstract. We propose a method for exploiting compression in computing the solution to a system of linear algebraic equations. The method is based on computing an approximate solution in a reduced space, and thus we seek a basis in which the solution has a compressed representation and can consequently be computed more efficiently. Although the method is completely general, it is especially effective for linear systems resulting from discretization of an underlying continuous problem, which will be our main focus. We address three primary issues: (1) how to compute an approximate solution to a given linear system using a given basis, (2) how to choose a basis that will yield significant compression, and (3) how to detect when the chosen basis is of sufficient dimension to provide a satisfactory approximation. While all three aspects have antecedents in previous ideas and methods, we combine, adapt, and extend them in a manner we believe to be novel for the purpose of solving discretized linear systems. We demonstrate that the resulting method can be competitive with—and often substantially outperforms—current standard methods and is effective for efficiently solving linear systems resulting from the discretization of major classes of continuous problems, including both differential equations and integral equations.

Key words. linear systems, compression basis, compressed solution, projection method, discretized linear system, regularization

AMS subject classifications. 65H10, 65N22, 65F10, 65F22

1. Motivation.

Discretization of a continuous problem (e.g., a differential or integral equation) using local (aka nodal) basis functions generally yields a *sparse* linear system but a *dense* discrete solution in that essentially all of the degrees of freedom are nonzero. Spectral discretization using global (aka modal) basis functions, on the other hand, generally yields a dense linear system but a sparse solution in that rapid convergence of the spectral approximation means that relatively few terms are required to provide a sufficiently accurate approximate solution. Is there a way that we can have the best of both worlds: a sparse linear system with a compactly representable approximate solution? To answer this question, we first observe that a discrete solution generally embodies some degree of structure inherited from the underlying continuous problem as well as the specific discretization employed and thus, being highly non-random, is potentially highly *compressible*.

Motivated by this key insight, the solution method that we propose directly computes (i.e., without first computing a dense approximate solution) such a compressed representation of the discrete solution that typically depends on far fewer parameters than the dimension of the linear system. The effectiveness of the compression method that we propose depends critically on the choice of the compression basis and on a suitable criterion for determining when the approximate solution is sufficiently accurate, and we will address both of these issues. With these issues resolved, we can then answer our earlier question by demonstrating that not only can we simultaneously achieve a sparse representation of both the problem and its solution, but we can often achieve a spectral-like convergence rate in the bargain (though not necessarily spectral accuracy due to the limitations of a sparse discretization).

This paper is based largely on the unpublished PhD. thesis [7], which should be consulted for further details and additional background. The current paper is organized as follows: Details of the compression method are presented in Section 2, guidance on choosing and

*Received November 18, 2021. Accepted December 30, 2021. Published online on February 15, 2022. Recommended by L. Reichel.
†School of Computing, Grand Valley State University, Allendale, MI (carrieer@gvsu.edu).
‡Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL (heath@illinois.edu).

ETNA
Kent State University and
Johann Radon Institute (RICAM)

forming a compression basis in Section 3, computational results in Section 4, a practical termination criterion in Section 5, and conclusions follow in Section 6.

2. Compression method. We begin by formalizing the concept of a compressed representation of a solution. Let $\boldsymbol{Ax} = \boldsymbol{b}$ be an $m \times n$ linear system, where the $m \times n$ matrix $\boldsymbol{A}$ and the m-vector $\boldsymbol{b}$ are given and the unknown solution n-vector $\boldsymbol{x}$ is to be determined. The solution $\boldsymbol{x}$ has a *compressed* representation if there is a k-vector $\boldsymbol{z}$ and an $n \times k$ matrix $\boldsymbol{X}$ such that $\boldsymbol{x} \approx \boldsymbol{Xz}$, with $k \ll n$.

We now formulate an algorithm for exploiting such compression when solving a system of linear algebraic equations. Simply stated, for a linear system $\boldsymbol{Ax} = \boldsymbol{b}$, we strategically sample the column space of $\boldsymbol{A}$ and then orthogonally project $\boldsymbol{b}$ onto the subspace spanned by the samples, thereby approximating the right-hand-side vector $\boldsymbol{b}$ by a linear combination of the columns of $\boldsymbol{AX}$. This basic method is stated more formally in Algorithm 1, where k is the number of samples, $\boldsymbol{X}$ is an $n \times k$ matrix (compression basis) to be chosen, and $\boldsymbol{Y}$ is the resulting $m \times k$ matrix of samples. The resulting vector $\boldsymbol{x}$ is an approximate solution because

$$\boldsymbol{Ax} = \boldsymbol{AXz} = \boldsymbol{Yz} \cong \boldsymbol{b}.$$

Indeed, the residual for the original linear system is the same as that of the least-squares subproblem in Algorithm 1, because

$$\|\boldsymbol{b} - \boldsymbol{Yz}\|_2 = \|\boldsymbol{b} - \boldsymbol{AXz}\|_2 = \|\boldsymbol{b} - \boldsymbol{Ax}\|_2.$$

The hope, of course, is that a basis matrix $\boldsymbol{X}$ can be identified that will produce a sufficiently accurate approximate solution with a relatively small value for k, in which case the method may be significantly less costly computationally than conventional methods. As we will see, such a favorable outcome is often possible for linear systems arising from discretization of continuous problems.

Algorithm 1 Basic Compression Method.

Given $m \times n$ matrix $\boldsymbol{A}$ and m-vector $\boldsymbol{b}$;
Choose integer k, $k \leq n$;
Choose $n \times k$ matrix $\boldsymbol{X}$;
$\boldsymbol{Y} = \boldsymbol{AX}$;
Solve $m \times k$ least-squares problem $\boldsymbol{Yz} \cong \boldsymbol{b}$ for $\boldsymbol{z}$;
$\boldsymbol{x} = \boldsymbol{Xz}$;

Algorithm 1 is remarkably general: it is formally applicable whether or not $\boldsymbol{A}$ is square or has full rank, and in each case it computes an appropriate approximate solution. If $\boldsymbol{A}$ has full column rank, then it approximates the conventional solution (this holds for both the square nonsingular and the overdetermined full-column-rank cases). If $\boldsymbol{A}$ lacks full column rank (including the underdetermined case), then it will compute a regularized solution. Here we focus on the square case $m = n$, but the lessons to be learned are applicable to the other cases as well.

Algorithm 1 leaves open two crucial, gaping questions: how to choose k and how to choose $\boldsymbol{X}$. We will begin to address the first question here and then answer it more definitively in Section 5. Answering the second question is the subject of Section 3. To achieve a compact representation of the solution at the least computational cost, our goal is to choose the smallest value for k that yields a solution of acceptable accuracy. Unfortunately, one generally has no foundation *a priori* for choosing such a value for k. A convenient practical alternative is

ETNA
Kent State University and
Johann Radon Institute (RICAM)

given in Algorithm 2, which builds the compression basis $\boldsymbol{X}$ incrementally, one column at a time, stopping when a residual tolerance is met. (Algorithm 2 is stated with a simple residual tolerance for convenience; a more intelligent stopping criterion is presented in Section 5.) The cost per iteration of Algorithm 2 (matrix-vector multiplication and orthogonalization) is similar to that of GMRES [27] plus the cost of generating the basis vectors $\boldsymbol{x}_k$, which varies from trivial to more significant, depending on the chosen basis. Note also that $\boldsymbol{A}$ is used only as an operator (i.e., for computing matrix-vector products) in Algorithm 2, making it easy to exploit sparsity in $\boldsymbol{A}$, which is often the case for nodal discretizations.

Algorithm 2 Incremental Compression Method.

Given $m \times n$ matrix $\boldsymbol{A}$, m-vector $\boldsymbol{b}$, tolerance tol;
$k = 1$;
Choose n-vector $\boldsymbol{x}_k$; $\boldsymbol{X}_k = [\boldsymbol{x}_k]$;
$\boldsymbol{y}_k = \boldsymbol{A}\boldsymbol{x}_k$; $\boldsymbol{Y}_k = [\boldsymbol{y}_k]$;
Solve $m \times 1$ least-squares problem $\boldsymbol{Y}_k\boldsymbol{z} \cong \boldsymbol{b}$ for $\boldsymbol{z}$ by QR factorization;
while $(\|\boldsymbol{b} - \boldsymbol{Y}_k\boldsymbol{z}\|_2 > tol)$
  $k = k + 1$;
  Choose n-vector $\boldsymbol{x}_k$; $\boldsymbol{X}_k = [\boldsymbol{X}_{k-1}, \boldsymbol{x}_k]$;
  $\boldsymbol{y}_k = \boldsymbol{A}\boldsymbol{x}_k$; $\boldsymbol{Y}_k = [\boldsymbol{Y}_{k-1}, \boldsymbol{y}_k]$;
  Update and extend QR factorization of $\boldsymbol{Y}_{k-1}$ to incorporate $\boldsymbol{y}_k$;
  Solve $m \times k$ least-squares problem $\boldsymbol{Y}_k\boldsymbol{z} \cong \boldsymbol{b}$ using updated QR;
$\boldsymbol{x} = \boldsymbol{X}_k\boldsymbol{z}$;

The possibility of a rank-deficient $\boldsymbol{Y}$ in Algorithm 1 due to possible rank deficiency of $\boldsymbol{X}$ or of $\boldsymbol{A}$ can be handled easily. When $\boldsymbol{Y}$ is rank deficient, simply compute a *basic* solution (i.e., a solution having the fewest possible nonzero components) using the QR factorization already computed (see, e.g., [16, p. 136]). This is an appropriate remedy, as we seek a compressed representation of the solution. In the incremental version of the method given in Algorithm 2, potential rank deficiency can be detected by monitoring r_{kk}, the diagonal entry of the upper triangular matrix $\boldsymbol{R}$ in the QR factorization. If the relative magnitude of r_{kk} falls below some tolerance, then simply skip to the next basis vector, if any. Thus, the final $\boldsymbol{Y}_k$ need never be rank deficient.

In summary, the compression method we propose is both simple (it can be fully stated in a single sentence requiring no mathematical notation) and general (applicable regardless of the shape or rank of $\boldsymbol{A}$). Most conventional methods for solving linear systems are neither simple nor general. It is also worth noting that the compression method, even in its incremental implementation, is in principle a *direct* method in that for any nondegenerate basis matrix $\boldsymbol{X}$ it produces the exact solution when $k = n$. In this sense it is somewhat analogous to Krylov subspace methods that are in principle finitely terminating but are implemented iteratively in practice. In both cases, for economy we wish to terminate after the fewest iterations that yield a sufficiently accurate solution. In the remainder of this paper we will address how to accomplish this for the compression method.

Related work. Compression is ubiquitous in today's digital world, with a wide variety of techniques routinely employed to compress all types of signals, images, and data files for more efficient storage and transmission. The present work differs from standard practice, however, in that the object being compressed—the solution to a discretized linear system—is unknown before the compression procedure begins and is revealed only implicitly in the resulting compressed representation.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

Though it differs in many details, the work presented here has a philosophical kinship with a variety of methods—such as model reduction [2], compressed sensing [31], sparse approximation [1], cross approximation [20], and sparse grids [12]—that reduce high-dimensional problems to lower-dimensional ones. That philosophy is also shared by Krylov subspace methods for linear systems [19] in that they typically produce a solution of acceptable accuracy in far fewer iterations than the dimension of the linear system. Our compression method can be viewed as a generalization of these methods in which the compression basis is not required to be a Krylov sequence. For example, the most popular such method for general linear systems, GMRES [27], can be reproduced by our Algorithm 2 if we use a Krylov sequence as the compression basis.

Turning now to prior work that is more directly relevant, if one assumes that $\boldsymbol{A}$ is square and nonsingular and $\boldsymbol{X}$ is of full rank, then Algorithm 1 becomes a special case of the *Prototype Projection Method* [26, Algorithm 5.1]. However, our Algorithm 1 is substantially more general in that we make *none* of those assumptions, and the choices for $\boldsymbol{X}$ that we will employ differ markedly from those considered in [26]. The basis matrix $\boldsymbol{X}$ could be interpreted as a *right preconditioner* [26, Equation (9.2)], but we do not require that $\boldsymbol{X}$ be square and nonsingular nor do we solve the resulting modified system by a standard iterative method.

Related approaches are often used for the numerical solution of ill-posed problems, such as integral equations of the first kind, for which an otherwise uselessly noisy solution can be regularized by restricting the computed solution to a designated subspace. An example is *subspace restricted SVD* [17, 21], in which the truncated singular value decomposition (TSVD) method is applied to a projected version of the system matrix $\boldsymbol{A}$, with the projection based on a user-chosen subspace of fixed size that is specified in advance. Our Algorithm 1 is simpler and more efficient than SVD-based methods, and most importantly, it lends itself readily to an incremental implementation (as in Algorithm 2), thereby enabling k to be determined adaptively on the fly.

Yet another antecedent from the perspective of ill-posed problems is [14, p. 115], where essentially the same idea as Algorithm 1 is suggested as a potential type of regularization for ill-conditioned systems, but it is illustrated using only a single choice of basis, namely the discrete cosine transform (DCT), and it is not developed any further.

Our approach also differs from *augmented* or *enriched* Krylov subspace methods (see [6, 9, 25]) in that we dispense with the Krylov subspace entirely and rely solely on the subspace spanned by $\boldsymbol{X}$, which may bear no direct relation to the system matrix $\boldsymbol{A}$.

3. Choosing a compression basis matrix. The compression method that we propose is applicable for solving any linear system, but its efficiency depends critically on the existence of a basis matrix $\boldsymbol{X}$ (and our ability to find it) in which the solution has a compressed representation. In essence, we are searching for a pattern in the solution that can be characterized by relatively few parameters. If, for example, the sequence of solution components were totally random, then by definition it has no compressed representation, and the compression method would offer no advantage. It is often the case, however, that the solution does exhibit a systematic pattern, especially when it represents a discrete approximation of an underlying continuous function, which may have known properties—smoothness, periodicity, monotonicity, convexity, even or odd parity, etc.—that can potentially be exploited by the compression method. Moreover, when the linear system results from discretization of a continuous problem, such as a differential or integral equation, then we can exploit our knowledge of the discretization itself to help determine a suitably effective compression basis. Consequently, we will focus here on such discretized linear systems and let approximation theory (see, e.g., [3, 22, 23, 29]) guide us in choosing a compression basis.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

Basic guidance. We first adopt the following terminology regarding the *support* of a given function. A function defined on a continuous (discrete) domain is called *global* (*dense*) if it is nonzero or nonnegligible on a relatively large portion of its domain. Conversely, it is called *local* (*sparse*) if it is nonzero or nonnegligible on only a relatively small portion of its domain. A basis is said to be *modal* (*nodal*) if it consists of global (local) basis functions. Some bases, notably hierarchical or multiresolution bases, are combinations of these.

Using this terminology, consider Table 3.1, which can be interpreted as follows. Approximating a global function (column 1) using a nodal basis (column 2) yields a dense coefficient vector (column 3), whereas using a modal basis (column 2) may yield a sparse coefficient vector (column 3), depending on the convergence rate of the functional expansion. For a local function (column 1), on the other hand, using a nodal basis (column 2) may yield a sparse coefficient vector (column 3), whereas a modal basis (column 2) generally yields a dense coefficient vector (column 3). Table 3.1 is also applicable for approximating discrete functions, with global and local interpreted as dense and sparse, respectively.

TABLE 3.1
Guide to choosing a basis.

function to be approximated	basis type	coefficient vector
global	nodal	dense
global	modal	sparse
local	nodal	sparse
local	modal	dense

In solving a discretized linear system using the compression method, we must consider both the basis used for the discretization and the compression basis used by the solver, so we make two passes through Table 3.1 to help guide the choice of a compression basis. For the first pass, we consider the basis used for the discretization, and the table then indicates what to expect for the solution vector to the discretized linear system. (Simple discretizations with no explicit discretization basis, such as finite differences for differential equations or Nystrom (quadrature) method for integral equations, are implicitly nodal.) On the second pass, we consider approximating a discrete function, namely the solution vector $\boldsymbol{x}$ to the discretized linear system. For the compression method to be advantageous, we need to choose a compression basis in the second pass that yields a sparse coefficient vector.

For example, when approximating a global solution function (column 1) discretized using a nodal basis (column 2), we can expect a dense solution vector $\boldsymbol{x}$ to the linear system (column 3). So, to approximate this dense solution vector (back to column 1), in order to have a chance for a sparse $\boldsymbol{z}$ (column 3), we need a modal basis (column 2). On the other hand, when approximating a global solution function discretized with a modal basis, similar reasoning leads us to choose a sparse compression basis. Interestingly, either path indicates that for a global solution function (by far the most common case), the compression basis should be of opposite type from the discretization basis to benefit from the compression method, effectively decoupling the two bases. When approximating a local solution function (a much less common case), however, the table indicates that the two bases should be of the same type.

Table 3.1 is merely a guide, not a guarantee. Comprehensive tests reported in [7] confirm its validity in the vast majority of cases, with occasional exceptions for unusual combinations that are unlikely to occur in practice. Although the table appears symmetric in the four cases, not all the cases are equally useful or "interesting" with respect to the underlying continuous problem. In particular, continuous problems with global solutions are far more common in

ETNA
Kent State University and
Johann Radon Institute (RICAM)

practice than problems with local solutions (e.g., isolated spikes or steep boundary layers). Thus, the first two cases in the table are of greater interest, especially the first case, for which the compression method can realize the goal stated in Section 1, namely the ability to have a sparse linear system yet still obtain a compact representation of the global solution. For the second case on the other hand, the compression method simply reproduces a standard spectral method in the sense that a nodal compression basis can pick out the nonnegligible coefficients in a modal discretization of the continuous solution. In any case, knowing from the table what general category of basis functions to use is obviously helpful, but it still leaves open several issues that will strongly affect the potential effectiveness of the compression method, which we will address in the remainder of this section.

Choice of basis functions. Approximation theory (see, e.g., [3, 22, 23, 29]) is a plentiful source of specific types of bases for approximating continuous functions. Another potential source of bases is digital signal processing (see, e.g., [4, 5, 10]), which seeks efficient representations of discrete signals, often for purposes of compression or filtering. An extensive menagerie of specific families of basis functions are assessed as potential compression bases in [7], including polynomials, piecewise polynomials, sinusoids, square waves, wavelets, Gaussians, and sinc functions. The upshot is that, perhaps unsurprisingly, generating a basis using orthogonal polynomials (specifically Chebyshev polynomials), suitably translated into the given problem domain of interest, is generally the most effective choice when a modal compression basis is needed. Perhaps more surprisingly, another effective modal compression basis is composed of hierarchical Gaussians with moderate overlap. Among nodal compression bases there is less variety than meets the eye as all tend to reduce to the identity matrix when evaluated at the most natural set of points (see next paragraph).

Choice of evaluation points. For use in the compression method, continuous basis functions must be converted into discrete basis vectors by evaluating each continuous function at a discrete set of points. For continuous basis functions $\phi_j(t)$, $j = 1, \ldots, k$, in one dimension, the entries of the corresponding discrete $n \times k$ basis matrix $\boldsymbol{X}$ are given by

$$x_{i,j} = \phi_j(t_i),$$

where $t_i, i = 1, \ldots, n$, are the *evaluation points*. In principle, any set of distinct evaluation points could be chosen, but the resulting compression basis will be most effective if it matches the discretization of the original continuous problem. For example, if the discretization is based on a set of mesh points or collocation points, then the compression method is most effective when those same points are used as the evaluation points in forming the compression basis matrix. For this same reason, discrete transforms from digital signal processing (e.g., the discrete cosine transform) may fail to perform well as compression bases in this context because they are already evaluated at a predetermined, standard set of evaluation points, leaving no flexibility to match the specific discretization of the continuous problem.

Ordering of evaluation points. For a linear system resulting from discretization of a continuous problem, each entry of the solution vector $\boldsymbol{x}$ corresponds to a specific degree of freedom, which may correspond to a specific location in the problem domain or to a specific discretization basis function (depending on the discretization). By virtue of the ordering employed during the discretization and formation of $\boldsymbol{A}$, these degrees of freedom have a specific ordering. As the effectiveness of the compression method relies on detecting a pattern in the solution vector $\boldsymbol{x}$, it is important that the ordering of the evaluation points (i.e., the row ordering of $\boldsymbol{X}$) be compatible with the ordering of the degrees of freedom of the discretization. For one-dimensional problems, when the degrees of freedom correspond to spatial coordinates, there is a common natural ordering for both the coordinates of the degrees of freedom and the evaluation points, so that incompatibility is generally not a concern in one dimension, but it may be a more significant issue in higher dimensions, which we will address shortly.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

Ordering of basis vectors. For the compression method to realize the full benefit of a compressed representation of the solution, the relevant discrete basis vectors (those with non-negligible coefficients) must come early in the incremental process of Algorithm 2, and thus the *ordering* of the basis vectors (i.e., the column ordering of $\boldsymbol{X}$) is crucial to the effectiveness of the method. Fortunately, many of the bases derived from approximation theory have a natural ordering that works well for most applications. For example, consistent with their convergence theory, polynomial bases are naturally ordered from lower to higher degree, and sinusoidal bases are naturally ordered from lower to higher frequency. In practice, the terms of lower degree or frequency usually carry the main signal, while the terms of higher degree or frequency tend to represent unwanted noise. For this reason, the compression method can have a regularizing effect, especially for poorly conditioned problems. Similarly, hierarchical or multiresolution bases are naturally ordered from coarser to finer levels.

Compression bases in higher dimensions. For a system of linear equations resulting from discretization of a continuous problem over a domain of more than one dimension, an appropriate compression basis can be derived by forming a tensor product of one-dimensional basis functions [3, p. 118]. For a rectangular domain in two dimensions, for example, given a set of one-dimensional basis functions $\phi_j(t)$, $j = 1, \ldots, n$, we define a set of basis functions in two dimensions by

$$\psi_{i,j}(s,t) = \phi_i(s)\phi_j(t), \qquad i,j = 1,\ldots,n. \tag{3.1}$$

As in one dimension, for best results with the compression method, the choice and ordering of the evaluation points should match the discretization as closely as possible, which may be straightforward if the discretization is based on a regular grid of mesh points. For an irregular discretization, such as finite elements with an irregular mesh, using the spatial locations corresponding to the degrees of freedom as evaluation points, in the same order as in the assembly of the system matrix, will ensure compatibility of the row ordering of $\boldsymbol{X}$ with the discretization.

It is important to note that the use of tensor product basis functions does *not* limit the compression method to problems having a rectangular domain, as any bounded problem domain of whatever shape can be contained within a suitably chosen rectangular domain. The tensor product basis functions can then be defined on the encompassing rectangular domain, but the evaluation points for the compression basis are chosen only from within the actual problem domain (e.g., at the nodes of a triangular finite element mesh in two dimensions).

The ordering of the basis functions (the column ordering of $\boldsymbol{X}$) is also more complicated in higher dimensions. In two dimensions, for example, the doubly-indexed set of basis functions defined by equation (3.1) has no default linear ordering. Among the many linear orderings that could be defined, the desired overall trend of low-to-high degree or frequency can be realized by ordering the basis functions by the 1-norm, 2-norm, or ∞-norm of their respective index pairs, smallest to largest, with ties broken in favor of the smaller first index. Two of these orderings are illustrated for a small example in Figure 3.1. All three orderings performed similarly in our tests, with the diagonal ordering (based on the 1-norm) having a slight edge.

4. Computational results. We now present computational results for a series of test problems, both to demonstrate the effectiveness of the compression method and to develop intuition for devising a more intelligent stopping criterion for Algorithm 2, to be presented in Section 5. For each example, we start with a continuous problem, first discretize it, and then solve the resulting linear system using the compression method. See [7] for details of the test problems and testing conditions, as well as for a much more extensive collection of computational test results than we have space to present here. To demonstrate the versatility

ETNA
Kent State University and
Johann Radon Institute (RICAM)

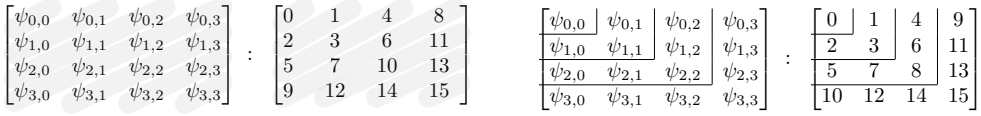


FIG. 3.1. *Illustration of linear orderings of two-dimensional basis functions:* diagonal *ordering (left, based on the* 1*-norm of index pairs), and* chevron *ordering (right, based on the* ∞*-norm of index pairs).*

of the compression method, the test problems include both differential equations and integral equations. We focus here on simple discretizations (finite differences for differential equations and Nystrom (quadrature) for integral equations) and finite element discretizations for boundary value problems in two dimensions. See [7] for additional examples using a wide variety of discretization methods to derive the corresponding linear systems, including both collocation and Galerkin methods for both differential and integral equations, as well as a wide range of compression basis functions. As in [7], all of the examples below are taken from the existing literature except for the two designated as "`designed`," which were designed to illustrate specific features of the compression method rather than as performance tests.

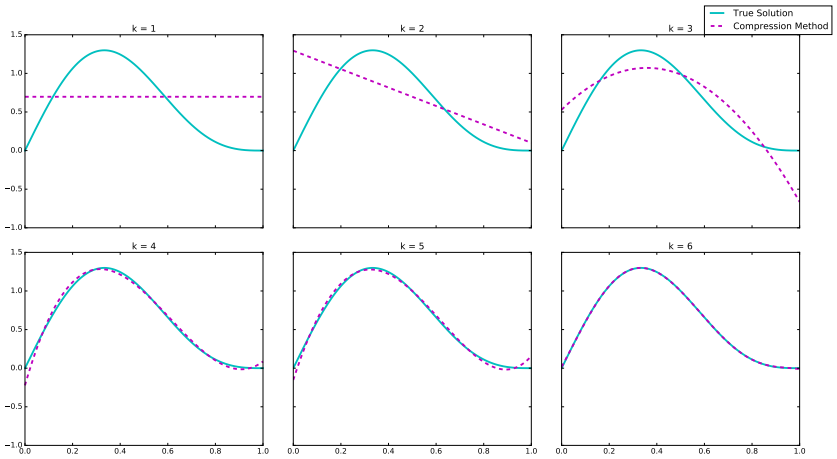
We begin with two examples to illustrate how and why the compression method can work well. For the one-dimensional integral equation `gravity_c1` and the two-point boundary value problem (BVP) `greengard-ex1`, Figure 4.1 displays the sequence of solutions obtained by Algorithm 2 as k increases, along with the true solution of the continuous problem. Local discretizations (Nystrom method with composite trapezoid quadrature and second-order finite differences, respectively), with Chebyshev polynomials evaluated at equally-spaced points as the compression basis, are used for both problems. Although both discretizations yield linear systems with $n = 128$, we see that an accurate solution is already obtained after only six iterations for the integral equation and eleven for the BVP (in Figure 4.1b we have omitted the basis functions for even values of k, which are polynomials of odd degree and therefore make no contribution to the solution, which is of even parity).

It may appear from the plots in Figure 4.1 that we have simply fit an increasingly accurate sequence of polynomials to the true solutions of these problems, but *that is not what happened here*: the compression method has no direct knowledge of the true continuous solution, of course, but nevertheless it is able to fit that solution indirectly by solving the discretized linear system using Algorithm 2, in effect fitting discretized and transformed versions of the polynomial basis functions to the discrete right-hand-side vector $\boldsymbol{b}$ of the linear system.

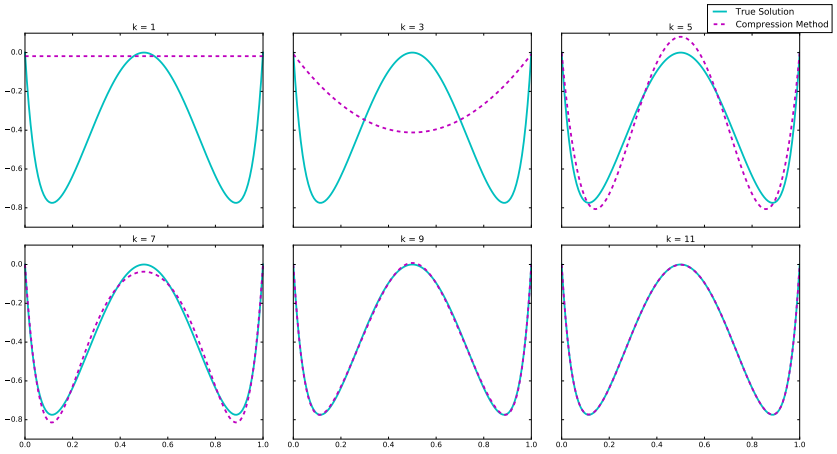
For each of the following performance tests, we plot both the relative residual and the relative error of the approximate solution for successive values of k in Algorithm 2. The relative *residual* that we report is based on the standard residual $\|\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}\|_2$ for the approximate discrete solution $\boldsymbol{x}$ of the discretized linear system. It is important to note that the relative *error* that we report is taken with respect to the known true solution of the continuous problem, and hence it includes discretization error as well as any error in solving the discretized linear system. We include discretization error for three reasons: (1) the continuous problem is ultimately what we really want to solve, whereas the discretized linear system is merely a means to an end; (2) by design, we know the true continuous solution to each test problem, whereas we do not know the true solution to the discretized linear system; (3) accounting for discretization error will enable us to devise a more intelligent stopping criterion in Section 5. The error will be unknown in practice, of course, so our goal here is to determine how the error correlates with the residual, which *is* available in practice.

To compute the relative error, we use the ∞-norm of the difference between the approximate solution and the known continuous solution for a given continuous problem. For a simple

ETNA
Kent State University and
Johann Radon Institute (RICAM)



(a) Integral equation `gravity_c1` discretized by the Nystrom method using composite trapezoid quadrature with $n = 128$.



(b) Boundary value problem `greengard-ex1` discretized using finite differences with $n = 128$.

FIG. 4.1. *Plots of the approximate solution obtained by the compression method for successive values of k, with Chebyshev polynomials evaluated at equally-spaced points as compression basis.*

discretization, the discrete approximate solution vector is compared with the corresponding discrete sample of the continuous true solution function. For a non-simple discretization, the continuous approximate solution is compared with the continuous true solution by sampling both at a set of points throughout the problem domain.

The ground rules for the performance plots are as follows. The horizontal axis shows the iteration counter k, and the vertical axis shows the relative error (left plot) and the relative residual (right plot) for a given problem or method at a given value of k. Both the horizontal and vertical scales are chosen to provide maximum visibility of the most pertinent information rather than for consistency across different plots, so note the individual scales carefully. In particular, in keeping with the way iterative methods are used in practice, we show only enough iterations to capture the behavior prior to nominal convergence and sufficiently beyond to make the longer term behavior reasonably clear, which is typically long before $k = n$, the

ETNA
Kent State University and
Johann Radon Institute (RICAM)

dimension of the linear system being solved.

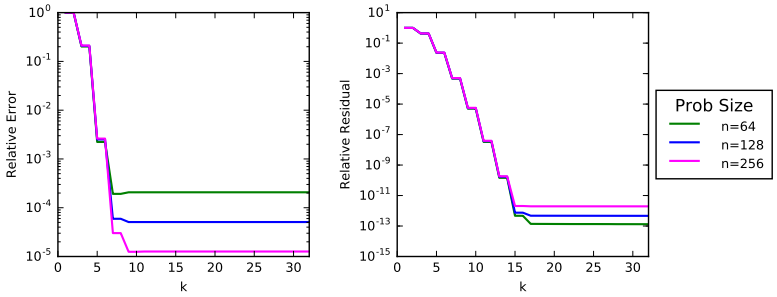
Scaling tests. We first examine how the optimal value for k in Algorithm 2 varies as the dimension n of the linear system grows when the discretization for a given continuous problem is refined. Figure 4.2 displays results using the compression method to solve the one-dimensional BVP `designed_sine_bvp` and the integral equation `designed_sine_ie` discretized using finite differences and composite trapezoid quadrature, respectively, with Chebyshev polynomials evaluated at equally-spaced points as the compression basis. By design, the two problems have identical solutions (a simple sine function). Figure 4.2 reveals a number of interesting observations that are fairly typical of the compression method:

- For both problems, both the error and the residual initially decline sharply as the (discretized and transformed) Chebyshev series rapidly converges to the smooth underlying solution. Once the accuracy of the approximate solution is within the discretization error, however, the error ceases to decline, becoming flat for the BVP and *increasing* for the integral equation.
- For both problems, we see from the error plots that the optimal value for k occurs for $k \ll n$, and it increases little, if any, as n grows.
- For both problems, the relative error, which is ultimately dominated by discretization error, is substantially larger than the relative residual, which is ultimately dominated by rounding error (note the marked difference in the respective vertical scales).
- For both problems, the achievable error level decreases as n increases, as expected, whereas the behavior of the residual as n varies is more complicated.
- For the BVP, the error reaches its minimum well before the residual reaches its minimum, and both become flat thereafter.
- For the integral equation, after reaching its minimum the error then begins to increase, while the residual continues to decrease, though at a slower rate.
- Stairstepping is evident in both the error and residual plots, in this case due to the odd parity of the true solution to the continuous problems, so that alternate basis functions (those of even parity) make no contribution to the approximate solution. Though certainly not universal, such stairstepping is common for a variety of reasons, especially in higher dimensions (as we will see).

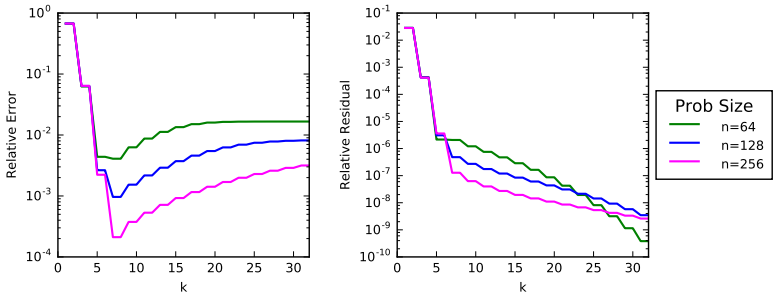
Performance tests. We next present results for a number of test problems to demonstrate the overall behavior and performance of the compression method. In addition to the relative error and the relative residual for Algorithm 2, for comparison we will also plot the corresponding values for two conventional algorithms, GMRES [27] (for all problems) and TSVD [14] (only for integral equations). The former is a popular iterative method for solving general square linear systems, while the latter is commonly used for solving potentially underdetermined or rank-deficient systems, such as those often arising from integral equations. Although Algorithm 2 can reproduce either of these methods (by using a Krylov sequence or the right singular vectors, respectively, as the compression basis), the results we report are for standard, native implementations of these methods. The point of providing comparisons with conventional methods is not so much to compete with them directly but rather to provide familiar benchmarks for assessing the performance potential of the compression method. Recall that the compression method and GMRES have similar costs per iteration (matrix-vector multiplication plus orthogonalization), so comparing them by iteration count rather than by computing time is a fair comparison.

Additionally, for problems having a sufficiently well-conditioned matrix, we draw a horizontal line in each plot indicating the relative error or relative residual when the solution to the linear system is computed using a standard direct solver. Such a line serves not only to set our expectations for the smallest residual achievable, but it also provides an excellent proxy

ETNA
Kent State University and
Johann Radon Institute (RICAM)



(a) Boundary value problem `designed_sine_bvp` discretized using finite differences.



(b) Integral equation `designed_sine_ie` discretized using composite trapezoid quadrature.

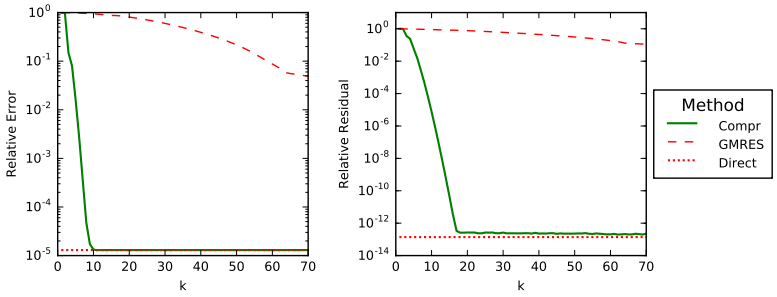
FIG. 4.2. *Scaling study showing the optimal value of k using the compression method for discretized systems of dimension $n = 64, 128, 256$, with Chebyshev polynomials evaluated at equally-spaced points as compression basis.*

for the discretization error, because the rounding error in the direct solution is comparatively small for well-conditioned systems.

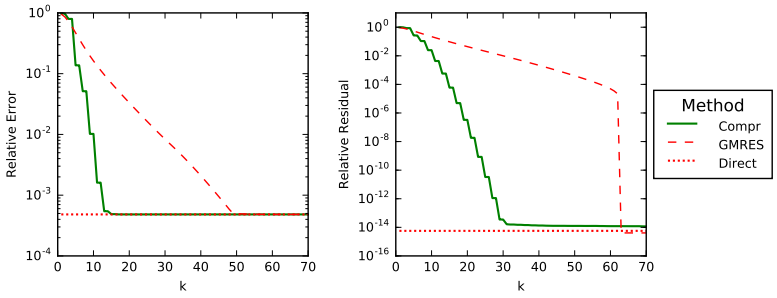
Figure 4.3 displays results for three one-dimensional boundary value problems discretized using finite differences, the first two with $n = 128$ and the third with $n = 512$, which is required to resolve its highly oscillatory solution adequately. We see that for each of these problems, the relative error in the compression method solution levels off (at the level of discretization error) significantly earlier than the residual flattens out, but nevertheless both occur for $k \ll n$. These results suggest that a simple residual tolerance as in Algorithm 2 is a reasonably effective stopping criterion, but it may significantly overshoot the optimal choice for k. Unfortunately, there appears to be no indication in the residual plot (which is all we can observe in practice, of course) of when that earlier optimum occurs, but we will soon see that this is not always the case.

Figure 4.4 shows results for two one-dimensional integral equations, both discretized using composite trapezoid quadrature, with Chebyshev polynomials evaluated at equally-spaced points as compression basis. (To simulate the measurement noise inherent in most empirical integral equation problems, for `gravity_c1` the components of the right-hand-side vector for the discretized problem are randomly perturbed at a level of 10^{-5}.) These problems are typically rank deficient, so we have no direct solution with which to approximate the discretization error, but we do show results for TSVD, which is often used for potentially rank-deficient problems, in addition to GMRES and the compression method. Here we see a rather different picture: not only does the error in the compression method solution reach a minimum for $k \ll n$, but that occurrence is detectable from the residual plot for both problems. Specifically, there is a noticeable change in the slope of the residual plot at the point of

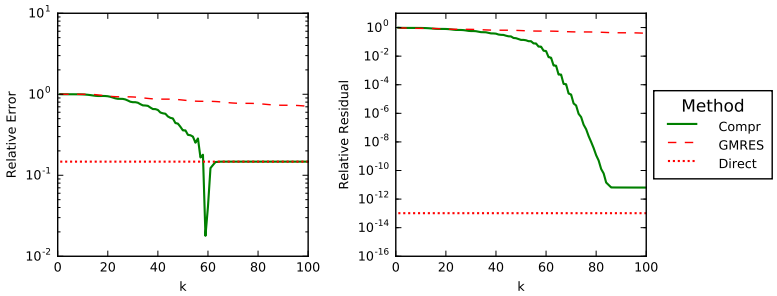
ETNA
Kent State University and
Johann Radon Institute (RICAM)



(a) Boundary value problem `fornberg` discretized using finite differences with $n = 128$.



(b) Boundary value problem `greengard-ex1` discretized using finite differences with $n = 128$.



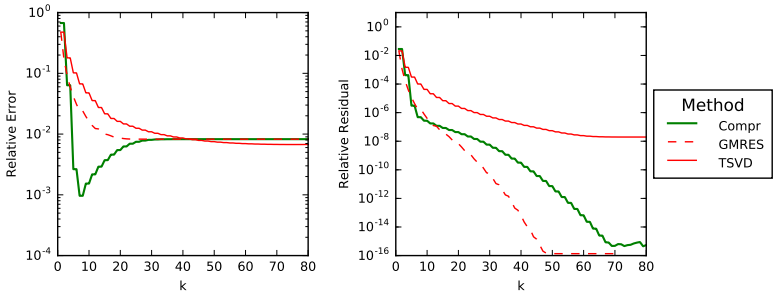
(c) Boundary value problem `greengard-ex3` discretized using finite differences with $n = 512$.

FIG. 4.3. *Performance of the compression method for one-dimensional boundary value problems, with Chebyshev polynomials evaluated at equally-spaced points as compression basis.*

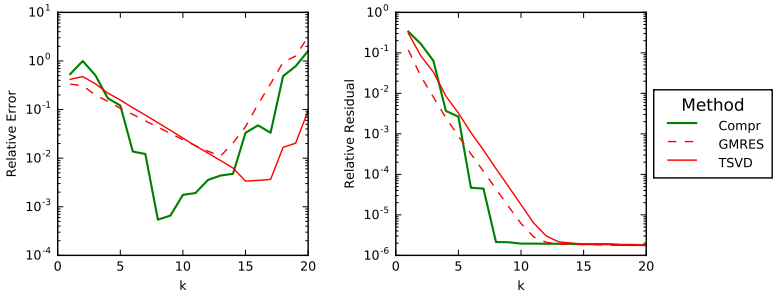
minimum error for both problems. Moreover, the error does not level off but actually *increases* beyond the optimal point, so the penalty for overshooting is to produce a significantly less accurate solution. Finally, we note that the compression method achieves a lower minimum error than either of the other methods, illustrating its usefulness as a regularization method.

To demonstrate the compression method in higher dimensions, we next consider test problems in two dimensions. For all two-dimensional problems, the value of n reported is the dimension of the linear system being solved, i.e., the number of grid points for the finite difference or Nystrom discretizations and the number of degrees of freedom for finite element discretizations. Figure 4.5 displays results for the two-dimensional Poisson problem `ericsson` (see [11]). Discretizations shown include finite differences (Figure 4.5a), finite elements using P1 (linear) basis functions on a uniform triangular mesh (Figure 4.5b),

ETNA
Kent State University and
Johann Radon Institute (RICAM)



(a) Integral equation `designed_sine_ie` discretized using composite trapezoid quadrature with $n = 128$.



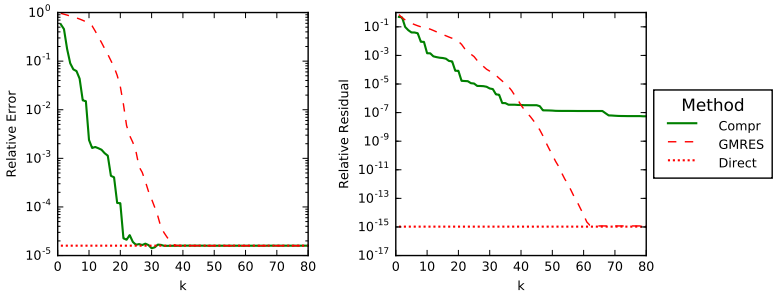
(b) Integral equation `gravity_c1` discretized using composite trapezoid quadrature with $n = 128$.

FIG. 4.4. *Performance of the compression method for one-dimensional integral equations with Chebyshev polynomials evaluated at equally-spaced points as compression basis.*

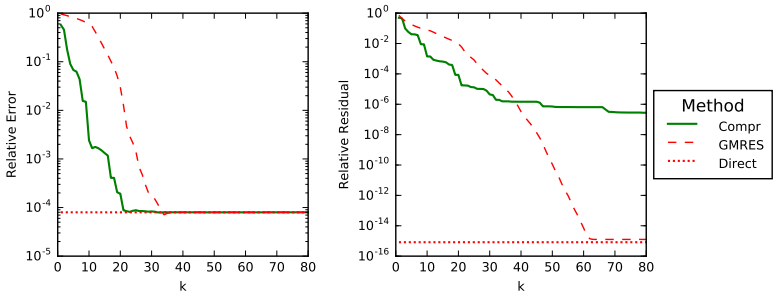
finite elements using P1 basis functions on an irregular triangular mesh (Figure 4.5c), and finite elements on the same irregular triangular mesh but with P2 (quadratic) basis functions (Figure 4.5d). All finite element discretizations were performed using the FEniCS software package [18]. Tensor-product (see equation (3.1)), diagonally-ordered (see Figure 3.1) Chebyshev polynomials evaluated at the mesh points of each discretization form the corresponding compression basis. More specifically, for a finite element discretization using P1 elements, the evaluation points are the vertices of the triangles in the mesh, and for P2 elements the evaluation points include the midpoints of the sides of the triangles as well as the vertices. In all four cases, the compression method quickly reaches the level of discretization error, at which point the residual (as well as the error) levels off, in contrast to GMRES, for which the residual continues to decline but without yielding any additional accuracy.

Recall our earlier observation that despite employing tensor product basis functions for problems in more than one dimension, the compression method is not limited to problems having a rectangular domain. Figure 4.6 demonstrates this fact by showing results for the `ericsson` problem again but this time on two nonrectangular domains, namely the triangular region having vertices at $(0, 0), (0, 1), (1, 0)$ and the circular disc of radius 0.5 centered at $(0.5, 0.5)$. For both domains, the compression basis is composed of tensor-product, diagonally-ordered Chebyshev polynomials defined on the unit square, but the evaluation points are the nodes of an irregular triangular finite element mesh (again generated by FEniCS) for each respective problem domain. Not surprisingly, the behavior for these nonrectangular subdomains is very similar to what we observed in Figure 4.5c and Figure 4.5d for the same problem on the full unit square, with the compression method rapidly converging to within the discretization error, at which point the residual levels off.

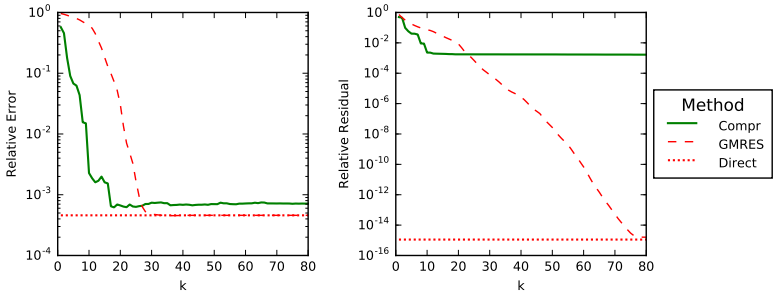
ETNA
Kent State University and
Johann Radon Institute (RICAM)



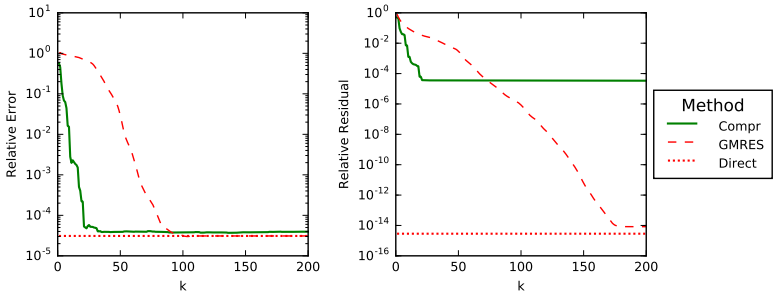
(a) Finite difference discretization with $n = 256$.



(b) Finite element discretization using P1 basis functions on a uniform triangular mesh with $n = 256$.



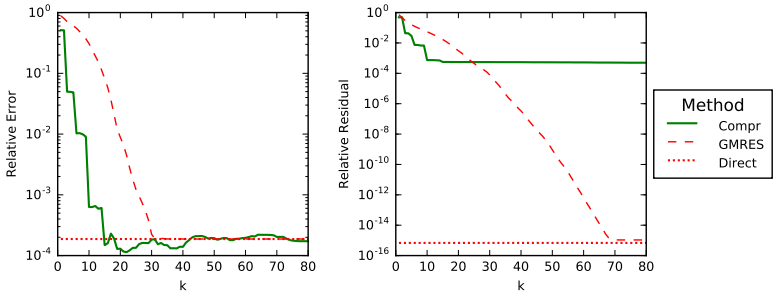
(c) Finite element discretization using P1 basis functions on an irregular triangular mesh with $n = 268$.



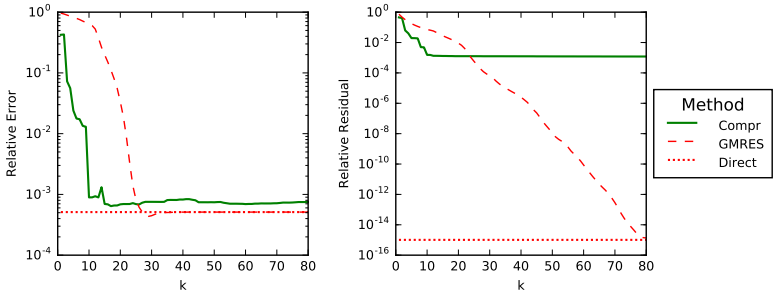
(d) Finite element discretization using P2 basis functions on an irregular triangular mesh with $n = 1005$.

FIG. 4.5. *Performance of the compression method for the two-dimensional Poisson problem* `ericsson` *on the unit square using the indicated discretizations with tensor-product, diagonally-ordered, Chebyshev polynomials evaluated at the mesh points of each discretization as corresponding compression basis.*

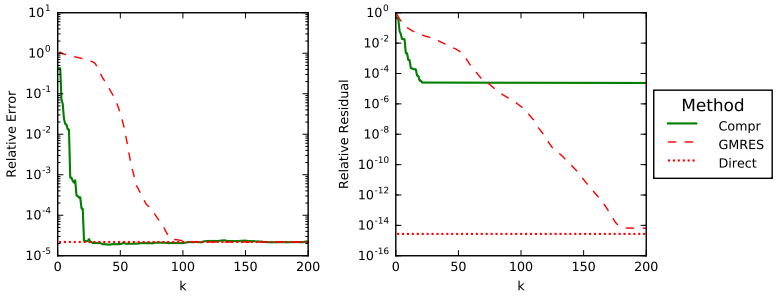
ETNA
Kent State University and
Johann Radon Institute (RICAM)



(a) Finite element discretization on a triangular domain using P1 basis functions on an irregular triangular mesh with $n = 255$.



(b) Finite element discretization on a circular domain using P1 basis functions on an irregular triangular mesh with $n = 272$.



(c) Finite element discretization on a circular domain using P2 basis functions on an irregular triangular mesh with $n = 1021$.

FIG. 4.6. *Performance of the compression method for the two-dimensional Poisson problem* `ericsson` *on the indicated domains using the indicated discretizations with tensor-product, diagonally-ordered Chebyshev polynomials evaluated at the mesh points of each discretization as corresponding compression basis.*

Finally, Figure 4.7 displays results for the two-dimensional integral equation `su-ex3` discretized using composite trapezoid quadrature with tensor-product, diagonally-ordered Chebyshev polynomials evaluated at equally-spaced points as the compression basis. We see that the compression method quickly produces a moderately accurate regularized solution for this ill-conditioned problem before the residual levels off, beyond which point the error becomes dominated by sharply increasing rounding error, amplified by the ill-conditioning. By contrast, the error for GMRES steadily increases almost from the outset even as the residual

ETNA
Kent State University and
Johann Radon Institute (RICAM)

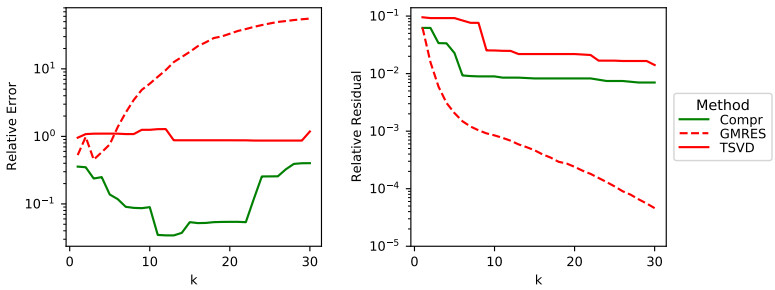


FIG. 4.7. *Performance of the compression method for the two-dimensional integral equation* `su-ex3` *discretized using composite trapezoid quadrature with* $n = 16,384$ *and with tensor-product, diagonally-ordered Chebyshev polynomials evaluated at equally-spaced points as compression basis.*

steadily decreases, and the error for TSVD remains stagnant despite a small initial drop in the residual.

By now the explanation for the foregoing results should be fairly obvious: for the compression method, the overall error initially decreases rapidly as additional basis vectors yield increasingly better approximations to the underlying continuous solution. Once sufficiently many basis vectors have been included for the accuracy of the approximate solution to be within the discretization error, however, then any further iterations will provide no further improvement in accuracy with respect to the continuous solution. Indeed, further iterations may actually increase the error (e.g., when rounding error is amplified by ill-conditioning) even as the residual for the approximate solution to the discretized linear system may continue to decline until it eventually flattens out. Thus, the optimal stopping point for the compression method is at the critical transition between these two phases, which is often—but not always—marked by a noticeable change in the slope of the residual curve, which we will refer to as a "*bend*." Typically, a bend (slope transition) in the residual curve occurs when the error reaches the level of discretization error, or the residual reaches the level of rounding error, or both. Once such a bend is reached, further iterations will yield no improvement in accuracy with respect to the solution of the continuous problem even if the residual continues to decline, so we may as well terminate. Consequently, detecting such a bend will be a major component of the stopping criterion we propose in Section 5.

From the foregoing discussion we see that, unlike conventional methods for solving linear systems, the compression method can somehow "sense" (from the behavior of the residual) the discretization error in the approximation to the underlying continuous problem. How can this be? The key is that the compression basis, by design, reflects the discretization error because the basis functions from which it is composed are evaluated with the *same spatial resolution* as that of the discretization; indeed, ideally the same mesh points or collocation points used for the discretization are also used as the evaluation points in forming the compression basis. Lacking this "sense," conventional iterative methods such as GMRES are often conservatively run well beyond the (unknown) optimal stopping point, at significant additional cost and possibly increasing error. Thus, the compression method often benefits from both faster convergence and more intelligent stopping than conventional methods.

5. Stopping criterion. As we have seen through several examples, the compression method with a suitably chosen compression basis is often capable of efficiently producing an accurate solution to a discretized linear system with $k \ll n$. Although a simple residual tolerance as stated in Algorithm 2 sometimes suffices, in many cases we can potentially do better (i.e., terminate earlier *and* attain maximum accuracy). In practice the error is unknown,

ETNA
Kent State University and
Johann Radon Institute (RICAM)

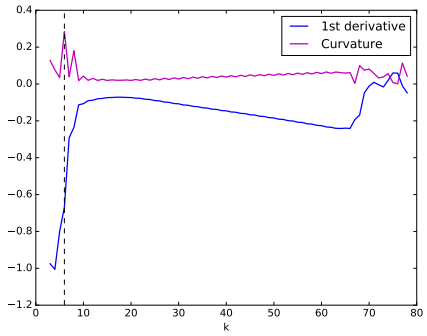


FIG. 5.1. *Slope and curvature of log residual curve for* `designed_sine_ie` *(compare with Figure 4.4a).*

of course, but we can examine the detailed behavior of the residual more closely, seeking to detect the point at which no further useful information will be forthcoming, so that we may as well terminate.

As we saw in Section 4, when the downward slope of the residual curve abruptly slows (exhibits a "bend"), this is an indication either that the accuracy of the approximate solution is within the discretization error or that the residual has reached the level of rounding error. In either case, no further improvement in accuracy is possible (indeed, the accuracy may even deteriorate beyond this point), so we should terminate. Thus, the smarter stopping criterion we seek is based on detecting such a bend in the residual curve on the fly as the iterations proceed. The residual curve may exhibit more than one bend, but for best efficiency and accuracy we want to identify the first one. As we will see, for some problems the residual may exhibit no sufficiently distinct bend before permanently flatlining (within rounding error), so our stopping criterion will need to be able to detect that possibility as well.

Quantitatively, a significant change in the slope (first derivative) of the residual curve corresponds to a peak in its *curvature* (essentially the relative change in the second derivative). A typical example is shown in Figure 5.1, which displays the slope and curvature (how these quantities are computed will be explained below) of the log residual curve for the problem `designed_sine_ie`. We see that the bend in the residual curve and the minimum in the error curve visible in Figure 4.4a both correspond to the peak in the curvature at $k = 6$. We also see the essentially random values of the slope and curvature once the residual reaches the level of rounding error.

In preparation for developing the necessary detection capability, we first make some observations about the residual produced by Algorithm 2 as a function of k, or more specifically $\log(\|\boldsymbol{b} - \boldsymbol{Y}_k \boldsymbol{z}\|_2)$, because we are working in the semilogy scale of the residual plots. First, we note that because Algorithm 2 minimizes over a nondecreasing sequence of subspaces, the residual norm is guaranteed to be monotonically nonincreasing as k increases, absent rounding effects. The residual norm can be flat, however, either temporarily or when rounding error prevents any further decrease.

Although we have informally referred to the residual "curve," the values of the residual norm for successive values of k are obviously discrete, and they can be jagged and noisy, as we have seen, for example, with the stairstepping often caused by parity effects. Thus, smoothing of the discrete data points will be required to define their slope meaningfully and enable reliable detection of a bend. The degree of smoothing presents a delicate choice, however, in that too much smoothing may smear out the abrupt transition we seek to detect, but too little smoothing may yield a useless plethora of abrupt transitions.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

Among the many smoothing techniques available, we have found that an effective choice for the present purpose is the Savitzky-Golay convolution filter [28], which performs a least-squares fit of a low-degree polynomial to a moving window of data points, thereby providing approximate values for the first and second derivatives of the data. Specifically, for one-dimensional problems we use a window of five points and a polynomial of degree two, where the smoothed function value and derivatives are then evaluated at the middle point. Note that a special procedure is necessary initially before five data points are available, and a wider smoothing window (we use size nine) is needed for two-dimensional problems due to the greater jaggedness caused by transitions between tiers in the array of two-dimensional basis functions (see Figure 3.1).

Once we have values for the derivatives, we can identify a bend by detecting a peak in the curvature of the log residual norm. The curvature of a one-dimensional function $y = f(x)$ is defined as

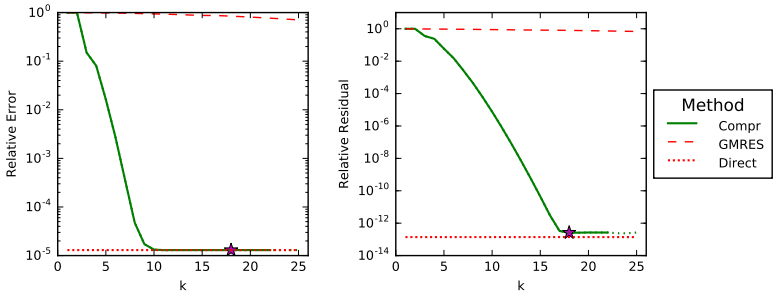
$$\text{curvature} = \frac{|y''|}{(1 + y'^2)^{\frac{3}{2}}}.$$

Substituting the successive smoothed derivative values into this formula, we can now seek a peak in the curvature on the fly as k progresses. The peak detection heuristic that we use is a smoothed z-score algorithm adapted from [30]. Although peak detection forms the core of our stopping criterion, to make it robust, additional logic is required to allow for a number of contingencies that may arise, including the possibility of the residual flatlining (within rounding error) without any bend having been detected. The result is a multi-faceted, heuristic criterion designed to terminate the incremental compression method at or near the optimal value of k (see [7] for further details). The detailed behavior of the automatic termination criterion depends on a number of tolerances and other parameters that can in principle be tuned to a given problem. It is important to note, however, that in the computational tests reported below we have used a fixed set of parameter values for all one-dimensional problems and a slightly different set of fixed values for all two-dimensional problems.

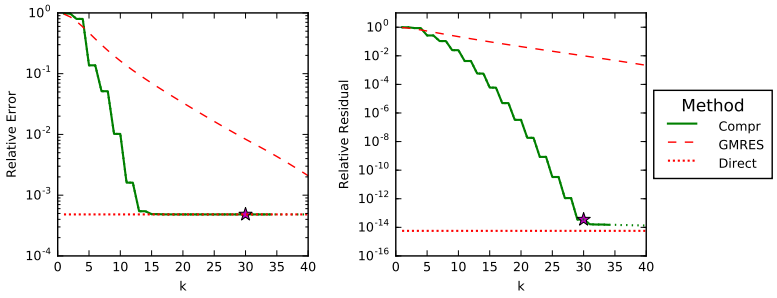
Related work. Detecting what we have called a "bend" in the semilog residual curve for the compression method is analogous to determining the "elbow" or "knee" of a scree plot for principal component analysis [8] or the "corner" or "vertex" of an L-curve for regularizing an ill-posed problem [13, 15]. In their original forms, neither of those approaches is useful in our case, but a more recent version of the L-curve [24] is much closer in spirit to our approach, and though differing significantly in detail, it could possibly provide a plausible alternative to our stopping criterion outlined above.

Computational results. We illustrate the effectiveness of our automatic termination criterion by showing the stopping points that it identifies for some of our previous examples. For each example, a star symbol in the residual plot indicates the stopping point automatically determined based on monitoring the residual from the compression method, and the corresponding point in the error plot is also indicated by a star symbol. Because we use the middle point of a five-point smoothing window (for one-dimensional problems) for peak detection, the compression method will always go at least two iterations beyond a detected peak, and additional checks to confirm that it should stop at the detected peak require two more iterations, so typically the compression method will be run for four iterations beyond the optimal stopping point identified by the automatic termination criterion (or somewhat more for higher-dimensional problems). In the plots, the solid line indicates the iterations necessary to detect the stopping point, and for reference, further behavior of the compression method beyond that point is indicated by a dotted line.

ETNA
Kent State University and
Johann Radon Institute (RICAM)



(a) Boundary value problem `fornberg` discretized using finite differences with $n = 128$.



(b) Boundary value problem `greengard-ex1` discretized using finite differences with $n = 128$.

FIG. 5.2. *Stopping points (indicated by a star symbol) automatically determined for two one-dimensional boundary value problems from Figure 4.3a and Figure 4.3b.*

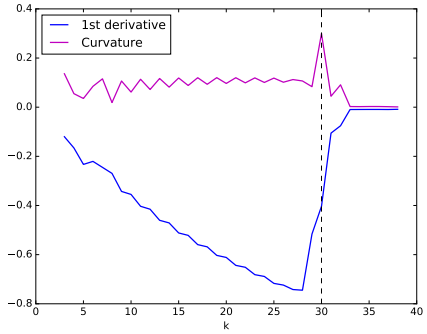
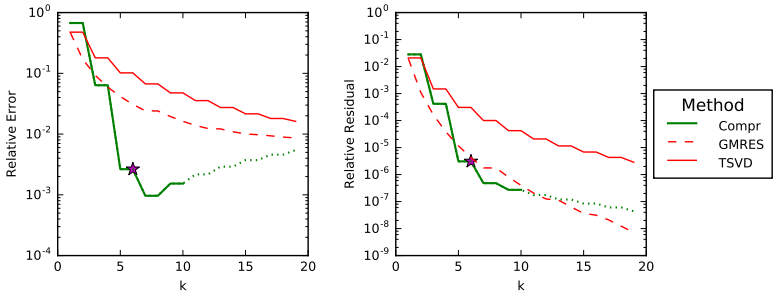


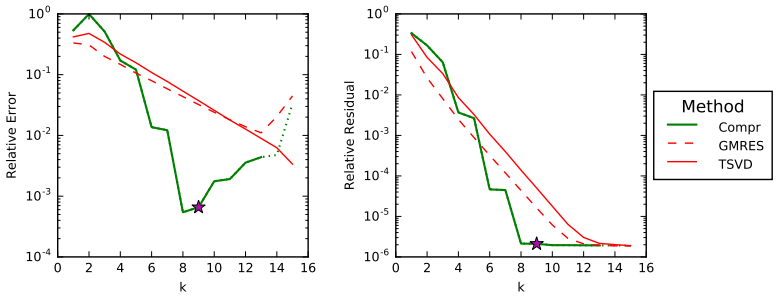
FIG. 5.3. *Slope and curvature of the log residual curve for* `greengard-ex1` *(compare with Figure 5.2b).*

Figure 5.2 displays results for the first two one-dimensional boundary value problems from Figure 4.3. For both of these problems, the only bend in the residual curve occurs when it flattens out at the level of rounding error, which is correctly detected by the termination criterion. Note that the stairstepping in Figure 5.2b (due to the parity of the solution) does not prevent the automatic criterion from correctly identifying the bend from the curvature peak (indicated by the vertical dashed line) clearly visible in Figure 5.3. Although the single bend in the residual curve significantly overshoots the optimal stopping point indicated by the error curve, the automatically determined stopping point is still far earlier than GMRES would permit.

ETNA
Kent State University and
Johann Radon Institute (RICAM)



(a) Integral equation `designed_sine_ie` discretized using composite trapezoid quadrature with $n = 128$.



(b) Integral equation `gravity_c1` discretized using composite trapezoid quadrature with $n = 128$.

FIG. 5.4. *Stopping points (indicated by a star symbol) automatically determined for two one-dimensional integral equations from Figure 4.4.*

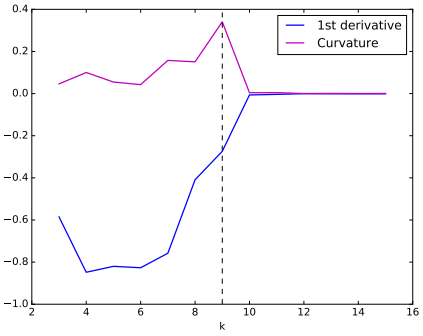
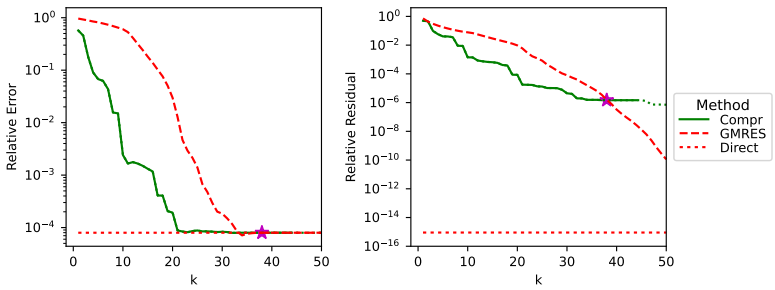


FIG. 5.5. *Slope and curvature of the log residual curve for* `gravity_c1` *(compare with Figure 5.4b).*

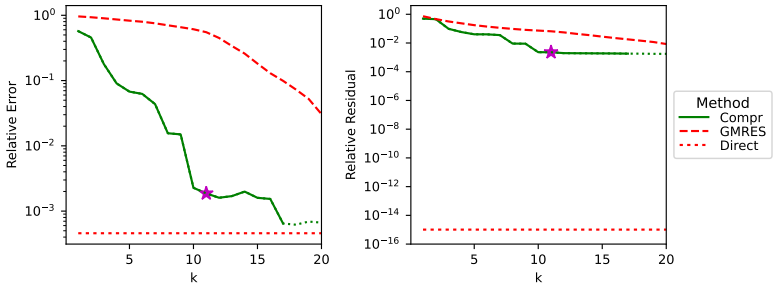
Figure 5.4 displays results for the two one-dimensional integral equations from Figure 4.4. For both of these problems, the residual curve exhibits a bend well before it reaches the level of rounding error. The termination criterion detects this bend at a point that closely approximates the minimum point in the corresponding error curve, yielding a nearly optimal stopping point for these problems that minimizes both cost and error. The identified peaks in the curvature are shown in Figure 5.1 and Figure 5.5, respectively.

Figure 5.6 displays results for two of the discretizations of the two-dimensional Poisson problem `ericsson` shown in Figure 4.5. In Figure 5.6a, for a finite element discretization using a uniform triangular mesh (see Figure 4.5b), the rather gradual, somewhat erratic decline

ETNA
Kent State University and
Johann Radon Institute (RICAM)



(a) Finite element discretization using P1 basis functions on a uniform triangular mesh with $n = 256$.



(b) Finite element discretization using P1 basis functions on an irregular triangular mesh with $n = 268$.

FIG. 5.6. *Stopping points (indicated by a star symbol) automatically determined for two different discretizations of the two-dimensional Poisson problem* `ericsson` *shown in Figure 4.5b and Figure 4.5c, respectively.*

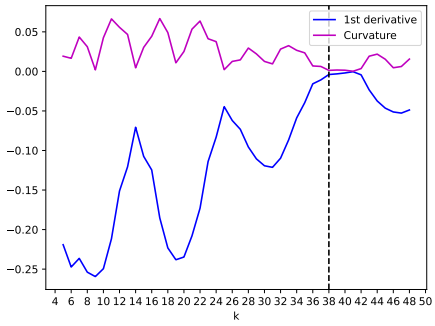
of the residual curve makes a distinct bend difficult to detect. Figure 5.7a shows multiple moderate peaks in the curvature, including one tantalizingly close to the optimal stopping point indicated by the error curve in Figure 4.5b, but none of these peaks is sufficiently pronounced to trigger the stopping criterion, which eventually stops after the residual curve has flattened, well beyond the optimal stopping point. No such ambiguity occurs for the irregular finite element discretization (see Figure 4.5c) shown in Figure 5.6b, for which the automatic criterion detects a peak in curvature at $k = 11$ (see Figure 5.7b) before the residual curve flattens when the level of discretization error is reached, resulting in a nearly optimal stopping point. Such excellent performance is typical for problems with irregular discretizations.

Finally, Figure 5.8 displays results for the two-dimensional integral equation `su-ex3` from Figure 4.7. Here again the termination criterion detects a nearly optimal stopping point when the residual curve flattens (see Figure 5.9) and avoids the subsequent dramatically increasing error caused by ill-conditioning (see Figure 4.7).

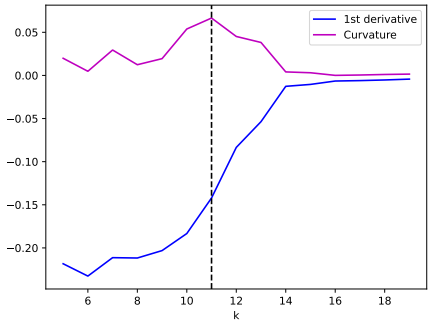
From these computational examples, which are typical of the results we have observed, we conclude that our automatic termination criterion, though certainly not perfect (in common with most other heuristics), often identifies a nearly optimal stopping point for the compression method. Our tentative conjecture is that when a bend is induced by discretization error, it corresponds closely to the optimal stopping point, but when a bend is induced by rounding error, it is typically already well past the optimal stopping point. It is not obvious that anything can be done about the latter case, but even when our automatic criterion significantly overshoots the optimal stopping point, typically it still stops far earlier than conventional methods would permit.

We emphasize that there is nothing sacred about the specific options that we have chosen

ETNA
Kent State University and
Johann Radon Institute (RICAM)



(a) Finite element discretization using a uniform triangular mesh (compare with Figure 5.6a).



(b) Finite element discretization using an irregular triangular mesh (compare with Figure 5.6b).

FIG. 5.7. *Slope and curvature of the log residual curves for* `ericsson` *using finite element discretizations with P1 basis functions on triangular meshes.*

in implementing the termination criterion; others might prefer to substitute their favorite smoothing and peak detection techniques for those we have employed. Finally, we note that while an automatic termination criterion is enabled by the ability of the compression method to "sense" the discretization error from the behavior of the residual, such a criterion is not an inherent part of the compression method itself, which can be used and is often effective in conjunction with a conventional termination criterion such as a simple residual tolerance. An important caveat, however, is that a simple residual tolerance is not suitable for *any* problem (e.g., many integral equations) for which the error may increase even as the residual decreases, which makes a compelling argument for an automatic stopping criterion such as the one we propose, especially for ill-conditioned problems.

6. Concluding summary. We began this paper by asking whether we can have "the best of both worlds" in solving a discretized continuous problem: a sparse linear system with a "sparse" (i.e., compactly represented) solution. For example, can we exploit the geometric flexibility of a finite element method while still enjoying the rapid convergence of a spectral method? We saw that this may indeed be possible provided we employ separate bases for discretizing the continuous problem and for representing the solution to the discretized linear system, and we presented a remarkably simple yet general method for solving a linear system based on this insight.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

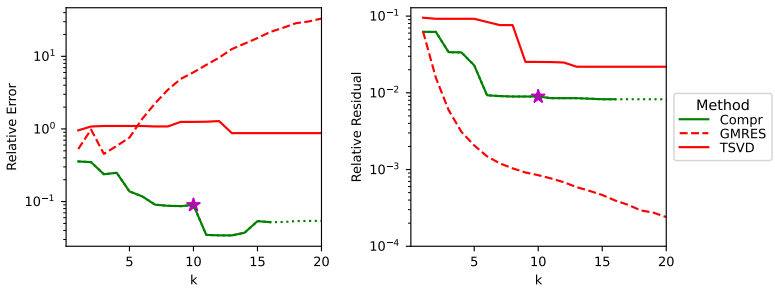


FIG. 5.8. *Stopping point (indicated by a star symbol) automatically determined for the two-dimensional integral equation* `su-ex3` *from Figure 4.7.*

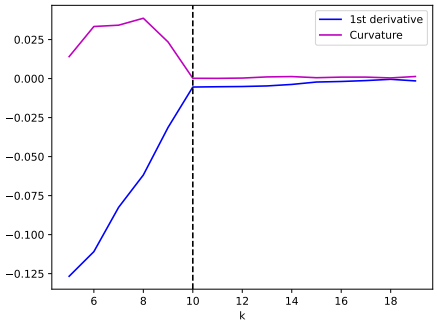


FIG. 5.9. *Slope and curvature of the log residual curve for* `su-ex3` *(compare with Figure 5.8).*

The efficacy of the resulting compression method depends critically on the compression basis selected, and we provided guidance for making an effective choice based on broad knowledge about the continuous solution (e.g., support, smoothness, symmetry), along with detailed knowledge of the specific discretization employed. We also devised an intelligent stopping criterion for the incremental version of the compression method that is often capable of automatically detecting when the accuracy of the approximate solution is within the discretization error, thereby avoiding further iterations that would yield no additional (or even potentially worse) accuracy. Computational results for a variety of test problems, including both differential and integral equations in one and two dimensions, demonstrated the effectiveness of the compression method in practice.

As illustrated by our computational examples, for integral equation problems the compression method often produces a more accurate regularized solution (i.e., having lower error) than conventional methods, and that greater accuracy is then locked in by our automatic stopping criterion rather than potentially being destroyed by continuing with further, unknowingly deleterious iterations, as a simple residual tolerance might do. For boundary value problems, the particular combination of a finite element discretization and a compression basis composed of Chebyshev polynomials yields what might be termed a "discrete spectral method" that can simultaneously exploit the geometric flexibility of finite element methods while enjoying the rapid convergence of spectral methods, with an automatic termination criterion as icing on the cake!

ETNA
Kent State University and
Johann Radon Institute (RICAM)

Acknowledgement. We wish to thank an anonymous referee for helpful suggestions regarding related work.

REFERENCES

[1] B. ADCOCK, S. BRUGIAPAGLIA, AND C. G. WEBSTER, *Sparse Polynomial Approximation of High-Dimensional Functions*, SIAM, Philadelphia, 2021.
[2] P. BENNER, M. OHLBERGER, A. COHEN, AND K. WILLCOX, (eds.), *Model Reduction and Approximation: Theory and Algorithms*, SIAM, Philadelphia, 2017.
[3] J. P. BOYD, *Chebyshev and Fourier Spectral Methods*, 2nd ed., Dover, Mineola, 2000.
[4] W. L. BRIGGS AND V. E. HENSON, *The DFT: An Owner's Manual for the Discrete Fourier Transform*, SIAM, Philadelphia, 1995.
[5] V. BRITANAK, P. C. YIP, AND K. R. RAO, *Discrete Cosine and Sine Transforms*, Elsevier, Amsterdam, 2007.
[6] D. CALVETTI, L. REICHEL, AND A. SHUIBI, *Enriched Krylov subspace methods for ill-posed problems*, Linear Algebra Appl., 362 (2003), pp. 257–273.
[7] E. CARRIER, *Exploiting Compression in Solving Discretized Linear Systems*, PhD. Thesis, Dept. Comp. Sci., University of Illinois at Urbana-Champaign, Champaign, 2019. http://hdl.handle.net/2142/104752
[8] R. B. CATTELL, *The scree test for the number of factors*, Multivar. Behav. Res., 1 (1966), pp. 245–276.
[9] Y. DONG, H. GARDE, AND P. C. HANSEN, *R^3GMRES: including prior information in GMRES-type methods for discrete inverse problems*, Electron. Trans. Numer. Anal., 42 (2014), pp. 136–146. http://etna.ricam.oeaw.ac.at/vol.42.2014/pp136-146.dir/pp136-146.pdf
[10] D. F. ELLIOTT AND K. R. RAO, *Fast Transforms*, Academic Press, New York, 1982.
[11] T. ERICSSON, *Poisson's equation and MPI*, Classroom notes, Chalmers University of Technology, Gothenburg, 2012. http://www.math.chalmers.se/Math/Grundutb/CTH/tma881/1112/Assignments/MPI/poisson.pdf
[12] J. GARCKE, *Sparse grids in a nutshell*, in Sparse Grids and Applications, J. Garcke and M. Griebel, eds., vol. 88 of Lect. Notes Comput. Sci. Eng., Springer, Heidelberg, 2013, pp. 57–80.
[13] P. C. HANSEN, *Analysis of discrete ill-posed problems by means of the L-curve*, SIAM Rev., 34 (1992), pp. 561–580.
[14] P. C. HANSEN, *Discrete Inverse Problems. Insight and Algorithms*, SIAM, Philadelphia, 2010.
[15] P. C. HANSEN AND D. P. O'LEARY, *The use of the L-curve in the regularization of discrete ill-posed problems*, SIAM J. Sci. Comput., 14 (1993), pp. 1487–1503.
[16] M. T. HEATH, *Scientific Computing*, 2nd. ed., SIAM, Philadelphia, 2018.
[17] M. E. HOCHSTENBACH AND L. REICHEL, *Subspace-restricted singular value decompositions for linear discrete ill-posed problems*, J. Comput. Appl. Math., 235 (2010), pp. 1053–1064.
[18] H. P. LANGTANGEN AND A. LOGG, *Solving PDEs in Python*, Springer, Cham, 2016.
[19] J. LIESEN AND Z. STRAKOŠ, *Krylov Subspace Methods*, Oxford University Press, Oxford, 2013.
[20] T. MACH, L. REICHEL, M. VAN BAREL, AND R. VANDEBRIL, *Adaptive cross approximation for ill-posed problems*, J. Comput. Appl. Math., 303 (2016), pp. 206–217.
[21] S. MORIGI, L. REICHEL, AND F. SGALLARI, *A truncated projected SVD method for linear discrete ill-posed problems*, Numer. Algorithms, 43 (2006), pp. 197–213 (2007).
[22] G. M. PHILLIPS, *Interpolation and Approximation by Polynomials*, Springer, New York, 2003.
[23] M. J. D. POWELL, *Approximation Theory and Methods*, Cambridge University Press, Cambridge, 1981.
[24] L. REICHEL AND H. SADOK, *A new L-curve for ill-posed problems*, J. Comput. Appl. Math., 219 (2008), pp. 493–508.
[25] Y. SAAD, *Analysis of augmented Krylov subspace methods*, SIAM J. Matrix Anal. Appl., 18 (1997), pp. 435–449.
[26] ———, *Iterative Methods for Sparse Linear Systems*, 2nd ed., SIAM Philadelphia, 2003.
[27] Y. SAAD AND M. H. SCHULTZ, *GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 856–869.
[28] A. SAVITZKY AND M. J. E. GOLAY, *Smoothing and differentiation of data by simplified least squares procedures*, Anal. Chem., 36 (1964), pp. 1627–1639.
[29] L. N. TREFETHEN, *Approximation Theory and Approximation Practice*, SIAM, Philadelphia, 2013.
[30] J. VAN BRAKEL, *Smoothed z-score algorithm*, Website accessed October 4, 2018. https://stackoverflow.com/questions/22583391/peak-signal-detection-in-realtime-timeseries-data
[31] M. VIDYASAGAR, *An Introduction to Compressed Sensing*, SIAM, Philadelphia, 2020.