High-order Time-stepping for Galerkin and Collocation Methods Based on Component-wise Approximation of Matrix Functions

James V. Lambers

***Abstract*—This paper describes an effort to develop time-stepping methods for partial differential equations that can overcome the difficulties that existing methods have with stiffness of the system of ordinary differential equations that results from spatial discretization. Stiffness is caused by the contrasting behavior of coupled components of the solution, and makes "one-size-fits-all" polynomial and rational approximations of the solution operator ineffective. This fate can be avoided through a component-wise approach, in which each component of the solution in an appropriate basis will be computed using an individualized approximation of the solution operator. This can be accomplished using "matrices, moments and quadrature" to approximate bilinear forms involving functions of matrices.**

Krylov subspace spectral (KSS) methods represent the first application of techniques from "matrices, moments and quadrature" to numerical methods for PDE. This paper describes the next step in the evolution of KSS methods, building upon their success with various PDE, including diffusion equations and wave equations. Adaptation to other spatial discretizations, such as finite elements and meshless methods, extends the applicability of KSS methods from problems defined on n-dimensional boxes to problems defined on domains with complicated geometries. The result is a class of algorithms that provides more effective solvers for PDE from a variety of applications such as structural analysis, heat transfer, and fluid dynamics.

***Index Terms*—spectral methods, finite elements, meshless methods, Gaussian quadrature, block Lanczos**

I. Introduction

The rapid advancement of computing power in recent years has allowed higher resolution models. This has introduced greater stiffness into systems of ordinary differential equations that arise from discretization of time-dependent PDE, which presents difficulties for time-stepping methods due to the coupling of components of the solution. This paper describes a *component-wise* approach to time-stepping in order to circumvent these difficulties. The resulting new methods will provide a more practical alternative to researchers in many branches of science and engineering who rely on the numerical solution of parabolic and hyperbolic PDE.

This goal is to be accomplished by continuing the evolution of Krylov subspace spectral (KSS) methods, which have been developed and advanced over the last five years [17], [18], [21], [23]. These methods feature explicit time-stepping with high-order accuracy, and stability that is characteristic of implicit methods. This "best-of-both-worlds" combination is achieved through a component-wise approach, in which each Fourier coefficient of the solution is computed using an approximation of the solution operator that is, in some sense, optimal for that coefficient. Initial work on KSS methods [17], [18], [19], [20], [21], [22], [23], [24], [26], [27] has yielded promising results, in terms of accuracy and stability. These results encourage their continued development, including enhancement of their accuracy and efficiency, expansion of their applicability, and their adaptation to important applications.

To date, KSS methods have been applied almost exclusively to problems on n-dimensional boxes, for $n = 1, 2, 3$, with either periodic or homogeneous boundary conditions. In this paper, KSS methods will be extended to PDE on domains with complicated geometries by combination with existing methods that are applicable to such problems, including finite element methods and meshless methods. Specifically, the goal is to combine the spatial discretization of these methods with the approach to high-order time-stepping that KSS methods provide.

KSS methods represent the first application to time-dependent PDE of techniques from the emerging research area of "matrices, moments and quadrature": the approximation of bilinear forms involving matrix functions by treating them as Riemann-Stieltjes integrals, and then applying Gaussian quadrature rules that are generated by the Lanczos algorithm. However, as beneficial as KSS methods can be in their own right, the scientific computing community as a whole can benefit far more from an understanding of the body of work from which they arose. As such, the next section reviews key ideas from this broader subject.

II. Matrices, Moments, Quadrature and PDE

This section describes the main ideas behind KSS methods. For simplicity, the one-dimensional case is discussed first, followed by generalization to higher dimensions. Let $S(t) = \exp[-Lt]$ represent the solution operator of a parabolic PDE on $(0, 2\pi)$,

$$u_t + Lu = 0, \quad t > 0, \tag{1}$$

with appropriate initial conditions and periodic boundary conditions. The operator L is a second-order, self-adjoint, positive definite differential operator.

Let $\langle \cdot, \cdot \rangle$ denote the standard inner product of functions defined on $[0, 2\pi]$. KSS methods, introduced in various forms in [17], [18], [21], [23], are time-stepping algorithms that compute the solution at time $t_1, t_2, \ldots$, where $t_n = n\Delta t$ for some choice of Δt. Given the computed solution $\tilde{u}(x, t_n)$ at

Manuscript received March 23, 2011

J. V. Lambers is with the Department of Mathematics, University of Southern Mississippi, Hattiesburg, MS, 39406-0001 USA e-mail: James.Lambers@usm.edu.

time t_n, the solution at time t_{n+1} is computed by approximating the Fourier coefficients that would be obtained by applying the exact solution operator to $\tilde{u}(x, t_n)$,

$$\hat{u}(\omega, t_{n+1}) = \left\langle \frac{1}{\sqrt{2\pi}} e^{i\omega x}, S(\Delta t)\tilde{u}(x, t_n) \right\rangle.$$

Clearly, such an approach requires an effective method for computing bilinear forms.

A. Elements of Functions of Matrices

In [9] Golub and Meurant described a method for computing quantities of the form

$$\mathbf{u}^T f(A)\mathbf{v}, \tag{2}$$

where $\mathbf{u}$ and $\mathbf{v}$ are N-vectors, A is an $N \times N$ symmetric positive definite matrix, and f is a smooth function. Our goal is to apply this method with $A = L_N$, where L_N is a spectral discretization of L, $f(\lambda) = \exp(-\lambda t)$ for some t, and the vectors $\mathbf{u}$ and $\mathbf{v}$ are derived from $\hat{\mathbf{e}}_\omega$ and $\mathbf{u}^n$, where $\hat{\mathbf{e}}_\omega$ is a discretization of $\frac{1}{\sqrt{2\pi}} e^{i\omega x}$, and $\mathbf{u}^n$ is a discretization of the solution at time t_n on an N-point uniform grid. In the following exposition, it is assumed that $\mathbf{u}$ and $\mathbf{v}$ are real; generalization to the complex case is straightforward.

Since the matrix A is symmetric positive definite, it has real eigenvalues $b = \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_N = a > 0$, and corresponding orthonormal eigenvectors $\mathbf{q}_j$, $j = 1, \ldots, N$. Therefore, the quantity (2) can be rewritten as

$$\mathbf{u}^T f(A)\mathbf{v} = \sum_{j=1}^{N} f(\lambda_j)\mathbf{u}^T \mathbf{q}_j \mathbf{q}_j^T \mathbf{v}, \tag{3}$$

which can also be viewed as a Riemann-Stieltjes integral

$$\mathbf{u}^T f(A)\mathbf{v} = I[f] = \int_a^b f(\lambda)\, d\alpha(\lambda), \tag{4}$$

where the measure $d\alpha(\lambda)$ is derived from the coefficients of $\mathbf{u}$ and $\mathbf{v}$ in the basis of eigenvectors.

As discussed in [9], the integral $I[f]$ can be approximated using Gaussian, Gauss-Radau or Gauss-Lobatto quadrature rules, which yields an approximation of the form

$$I[f] = \sum_{j=1}^{K} w_j f(\lambda_j) + R[f], \tag{5}$$

where the nodes λ_j, $j = 1, \ldots, K$, and the weights w_j, $j = 1, \ldots, K$, can be obtained using the Lanczos algorithm and variations [11].

B. Block Gaussian Quadrature

In the case $\mathbf{u} \neq \mathbf{v}$, there is the possibility that the weights may not be positive, which destabilizes the quadrature rule [1]. Instead, one can consider the approximation of

$$\begin{bmatrix} \mathbf{u} & \mathbf{v} \end{bmatrix}^T f(A) \begin{bmatrix} \mathbf{u} & \mathbf{v} \end{bmatrix}, \tag{6}$$

which results in the 2×2 matrix integral

$$\int_a^b f(\lambda)\, d\mu(\lambda) = \begin{bmatrix} \mathbf{u}^T f(A)\mathbf{u} & \mathbf{u}^T f(A)\mathbf{v} \\ \mathbf{v}^T f(A)\mathbf{u} & \mathbf{v}^T f(A)\mathbf{v} \end{bmatrix}, \tag{7}$$

where $\mu(\lambda)$ is a 2×2 matrix, each entry of which is a measure of the form $\alpha(\lambda)$ from (4).

As discussed in [9], the most general K-node quadrature formula is of the form

$$\int_a^b f(\lambda)\, d\mu(\lambda) = \sum_{j=1}^{2K} f(\lambda_j)\mathbf{v}_j \mathbf{v}_j^T + error, \tag{8}$$

where, for each j, λ_j is a scalar and $\mathbf{v}_j$ is a 2-vector. Each node λ_j is an eigenvalue of the matrix

$$\mathcal{T}_K = \begin{bmatrix} M_1 & B_1^T & & & \\ B_1 & M_2 & B_2^T & & \\ & \ddots & \ddots & \ddots & \\ & & B_{K-2} & M_{K-1} & B_{K-1}^T \\ & & & B_{K-1} & M_K \end{bmatrix}, \tag{9}$$

which is a block-tridiagonal matrix of order $2K$. The vector $\mathbf{v}_j$ consists of the first two elements of the corresponding normalized eigenvector. The matrices M_j and B_j are computed using the block Lanczos algorithm, which was proposed by Golub and Underwood in [10].

C. KSS Methods

KSS methods proceed as follows. For each wave number $\omega = -N/2 + 1, \ldots, N/2$, the QR factorization of the block $R_0(\omega) = \begin{bmatrix} \hat{\mathbf{e}}_\omega & \mathbf{u}^n \end{bmatrix}$ is computed, which yields $R_0(\omega) = X_1(\omega)B_0(\omega)$. Then, block Lanczos iteration is applied to the discretized operator L_N with initial block $X_1(\omega)$, producing a block tridiagonal matrix $\mathcal{T}_K(\omega)$ of the form (9), where each entry is a *function* of ω. Then, each Fourier coefficient of the solution at t_{n+1} can be expressed as

$$[\hat{\mathbf{u}}^{n+1}]_\omega = \left[B_0^H E_{12}^H \exp[-\mathcal{T}_K(\omega)\Delta t] E_{12} B_0\right]_{12},$$

where $E_{12} = \begin{bmatrix} \mathbf{e}_1 & \mathbf{e}_2 \end{bmatrix}$.

This algorithm has local temporal accuracy $O(\Delta t^{2K-1})$ for the parabolic problem (1) [18]. By contrast, methods that apply Lanczos iteration only to the solution from the previous time step (see, for example, [13]) achieve $O(\Delta t^{K-1})$ accuracy, where K is the dimension of the Krylov subspace used. Even higher-order accuracy, $O(\Delta t^{4K-2})$, is obtained for the second-order wave equation [19]. Furthermore, in [18], [19], it is shown that under appropriate assumptions on the coefficients of the differential operator L in (1), the 1-node KSS method is *unconditionally stable*.

D. Relation to Other Time-Stepping Methods

Numerical experiments comparing KSS methods to a variety of time-stepping methods, including finite-difference methods, Runge-Kutta methods and backward differentiation formulae, can be found in [19], [20], [23], [24]. Comparisons with a number of Krylov subspace methods based on exponential integrators [12], [13], [14], including preconditioned Lanczos iteration [32], [37], are made in [26]. Here, we discuss the differences in design between established time-stepping methods and KSS methods, in order to highlight the advantages in, and potential of, the latter approach.

Consider a system of ODE of the form

$$\mathbf{u}'(t) = A\mathbf{u}, \quad \mathbf{u}(t_0) = \mathbf{u}_0.$$

One-step methods such as Runge-Kutta methods yield an approximate solution of the form

$$\mathbf{u}(t_{n+1}) = f(A; \Delta t_n)\mathbf{u}(t_n), \quad \Delta t_n = t_{n+1} - t_n,$$

where $f(A;\Delta t_n)$ is a polynomial of A that agrees with leading terms of the Taylor series expansion of $\exp[A\Delta t_n]$ if the method is explicit, or a rational function of A if the method is implicit.

Similarly, Krylov subspace methods based on exponential integrators, such as those described in [13], [12] yield approximate solutions that are equal to a polynomial of A times a vector. The polynomial is obtained by computing the exponential of a much smaller matrix that is generated by a process such as Lanczos iteration or Arnoldi iteration. Modifications described in [32], [37] produce rational approximations.

All such methods have difficulty with stiff systems, such as those that arise from the spatial discretization of time-dependent PDE. The solution of a stiff system, even one that represents a typically smooth solution of a parabolic PDE, includes rapidly-changing components that are coupled with components that evolve more slowly. Because of this phenomenon, explicit methods for such systems require small time steps, while implicit methods typically require the solution of ill-conditioned systems of linear equations.

Notable exceptions are very simple model problems such as one of the form

$$u_t(\mathbf{x},t) = k^2\Delta u(\mathbf{x},t), \quad \mathbf{x}\in(0,2\pi)^n, \quad t>0,$$

with periodic boundary conditions. The eigenfunctions of the differential operator $L = k^2\Delta$ are $e^{i\vec{\omega}\cdot\mathbf{x}}$ for each $\vec{\omega}\in\mathbb{Z}^n$. Therefore, the Fourier components of the solution can be decoupled from one another, and each one can then be computed individually using an appropriate approximation of $e^{-k^2\|\vec{\omega}\|_2^2 t}$.

The central idea behind KSS methods is to compute each component of the solution, in some orthonormal basis, using an approximation that is specifically tailored to that component, even though all components are coupled. It follows that each component uses a different polynomial approximation of $S(A)$, where the function S is based on the solution operator of the PDE, and A is the discretization of the spatial differential operator. Taking all components together reveals that the computed solution has the form

$$\mathbf{u}(t_{n+1}) = \tilde{f}(A;\Delta t_n)\mathbf{u}(t_n) = \sum_{j=0}^{K} D_j(\Delta t_n)A^j\mathbf{u}(t_n)$$

where $D_j(\Delta t_n)$ is a diagonal matrix in the chosen basis. In other words, KSS methods are explicit methods that are inherently more flexible than explicit time-stepping methods that use a polynomial approximation of the solution operator. The intent of the work described in the next section is to exploit this flexibility.

E. Implementation

KSS methods compute a Jacobi matrix corresponding to *each* Fourier coefficient, in contrast to traditional Krylov subspace methods (see, for example, [12], [13], [14], [32], [37]) that use only a single Krylov subspace generated by the solution from the previous time step. While it would appear that KSS methods incur a substantial amount of additional computational expense, that is not actually the case, because nearly all of the Krylov subspaces that they compute are closely related by the wave number ω, in the 1-D case, or $\vec{\omega} = (\omega_1,\omega_2,\ldots,\omega_n)$ in the n-D case.

In fact, the only Krylov subspace that is explicitly computed is the one generated by the solution from the previous time step, of dimension $(K+1)$, where K is the number of block Gaussian quadrature nodes. In addition, the averages of the coefficients of L^j, for $j = 0,1,2,\ldots,2K-1$, are required, where L is the spatial differential operator. When the coefficients of L are independent of time, these can be computed once, during a preprocessing step. This computation can be carried out in $O(N\log N)$ operations using symbolic calculus [22], [24].

With these considerations, the algorithm for a single time step of a 1-node KSS method for solving (1), where $Lu = -pu_{xx} + q(x)u$, with appropriate initial conditions and periodic boundary conditions, is as follows. The average of a function $f(x)$ on $[0,2\pi]$ is denoted by $\overline{f}$.

$\hat{u}^n = \mathbf{fft}(u^n)$, $v = Lu^n$, $\hat{v} = \mathbf{fft}(v)$
for each ω **do**
  $\alpha_1 = -p\omega^2 + \overline{q}$ (in preprocessing step)
  $\beta_1 = \hat{v}(\omega) - \alpha_1\hat{u}^n(\omega)$
  $\alpha_2 = \langle u^n, v\rangle - 2\,\mathrm{Re}\,[\hat{u}^n(\omega)\overline{\hat{v}(\omega)}] + \alpha_1|u^n(\omega)|^2$
  $e_\omega = [\langle u^n, u^n\rangle - |\hat{u}^n(\omega)|^2]^{1/2}$
  $T_\omega = \begin{bmatrix} \alpha_1 & \beta_1/e_\omega \\ \beta_1/e_\omega & \alpha_2/e_\omega^2 \end{bmatrix}$
  $\hat{u}^{n+1}(\omega) = [e^{-T_\omega\Delta t}]_{11}\hat{u}^n(\omega) + [e^{-T_\omega\Delta t}]_{12}e_\omega$
end
$u^{n+1} = \mathbf{ifft}(\hat{u}^{n+1})$

As with the abovementioned preprocessing step, the amount of work per time step is $O(N\log N)$, assuming that the FFT is used for differentiation in the computation of v. This same complexity applies to KSS methods with a higher number of quadrature nodes [21]. Furthermore, KSS methods allow substantial parallelism, as each component of the solution is obtained from its associated block Jacobi matrix, independently of other components. Coefficients of powers of L can also be computed in parallel.

Generalization of this algorithm to higher dimensions and other PDE is straightforward. For example, on the domain $[0,2\pi]^n$, if $L = -p\Delta + q(\mathbf{x})$, the **for** loop would iterate over all n-tuples $\vec{\omega} = (\omega_1,\omega_2,\ldots,\omega_n)$, and for each such $\vec{\omega}$, $\alpha_1 = -p\|\vec{\omega}\|_2^2 + \overline{q}$, where $\overline{q}$ is the average of $q(\mathbf{x})$ on the domain. No additional modification to the algorithm is necessary. For the second-order wave equation, two matrices of the form T_ω are computed for each $\vec{\omega}$, corresponding to the solution from the previous time step and its time derivative.

KSS methods have also been generalized to systems of coupled equations; the details are presented in [20]. In particular, it is shown in [27] that KSS methods are effective for systems of three equations in three spatial variables, through application to Maxwell's equations.

III. Adaptation of KSS Methods

This section describes proposed modifications to KSS methods that will expand their applicability to a much greater variety of problems.

A. Other Spatial Discretizations

The main idea behind KSS methods, that higher-order accuracy in time can be obtained by component-wise ap-

proximation, is not limited to the enhancement of spectral methods that employ Fourier basis functions. Let A be an $N \times N$ matrix and f be an analytic function. Then $f(A)\mathbf{v}$ can be computed efficiently by approximation of each component, with respect to an orthonormal basis $\{\mathbf{u}_j\}_{j=1}^N$, by a K-node Gaussian quadrature rule if expressions of the form

$$\mathbf{u}_j^H A^k \mathbf{u}_j, \quad \mathbf{u}_j^H A^k \mathbf{v}, \quad \mathbf{v}^H A^k \mathbf{v} \tag{10}$$

can be computed efficiently for $j = 1, \ldots, N$ and $k = 0, \ldots, 2K-1$, and transformation between the basis $\{\mathbf{u}_j\}_{j=1}^N$ and the standard basis can be performed efficiently.

The first expression in (10) can be computed analytically if the members of the basis $\{\mathbf{u}_j\}_{j=1}^N$ can be simply expressed in terms of j, as in Fourier spectral methods. The other two expressions in (10) are readily obtained from bases for Krylov subspaces generated by $\mathbf{v}$. If A is sparse, then each block recursion coefficient, across *all* components, can be represented as the sum of a sparse matrix and a low-rank matrix [25]. Thus it is worthwhile to explore the adaptation of KSS methods to other spatial discretizations for which the recursion coefficients can be computed efficiently. The temporal order of accuracy achieved in the case of Fourier spectral methods is expected to apply to such discretizations, as only the measures in the Riemann-Stieltjes integrals are changing, not the integrands.

B. Finite Element Methods

Since their introduction nearly seventy years ago, finite element methods have been used for a variety of applications, including structural problems such as stress analysis, buckling, vibrational analysis, and non-structural problems such as heat transfer, fluid flow, and distribution of electric or magnetic potential [30]. The ease with which they can be applied to problems featuring irregularly shaped domains and heterogeneity in the coefficients and boundary conditions makes finite element methods an ideal class of methods to combine with KSS methods in order to expand the applicability of their time-stepping advantages.

Consider a general PDE of the form $u_t = Lu$ on a domain Ω, with appropriate initial and boundary conditions. A finite element discretization results in a system of ODE of the form $M\mathbf{u}_t = K\mathbf{u} + \mathbf{F}$, where M is the mass matrix, K is the stiffness matrix, $\mathbf{F}$ is the load vector, and $\mathbf{u}$ is a vector of coefficients of the approximate solution in the basis of trial functions. Because M and K are sparse, our approach can be used to compute bilinear forms involving functions of $M^{-1}K$, where the basis vectors $\mathbf{u}_j$ are simply the standard basis vectors. Through mass lumping, M can be replaced by a diagonal matrix in a way that preserves spatial accuracy [15].

As an example, an adaptation of a 2-node KSS method is used to solve a diffusion equation to a domain with a non-rectangular boundary and holes. The initial data is Gaussian, with homogeneous Dirichlet boundary conditions on the portion of the boundary for which $y < 0.3$, and homogeneous Neumann boundary conditions on the remainder of the boundary. A triangular mesh with 545 nodes is used, in conjunction with piecewise bilinear basis functions. The results are shown in Figure 1, along with comparison to various other time-stepping methods. As in the Fourier spectral case, the adapted KSS method is 3rd-order accurate in time.

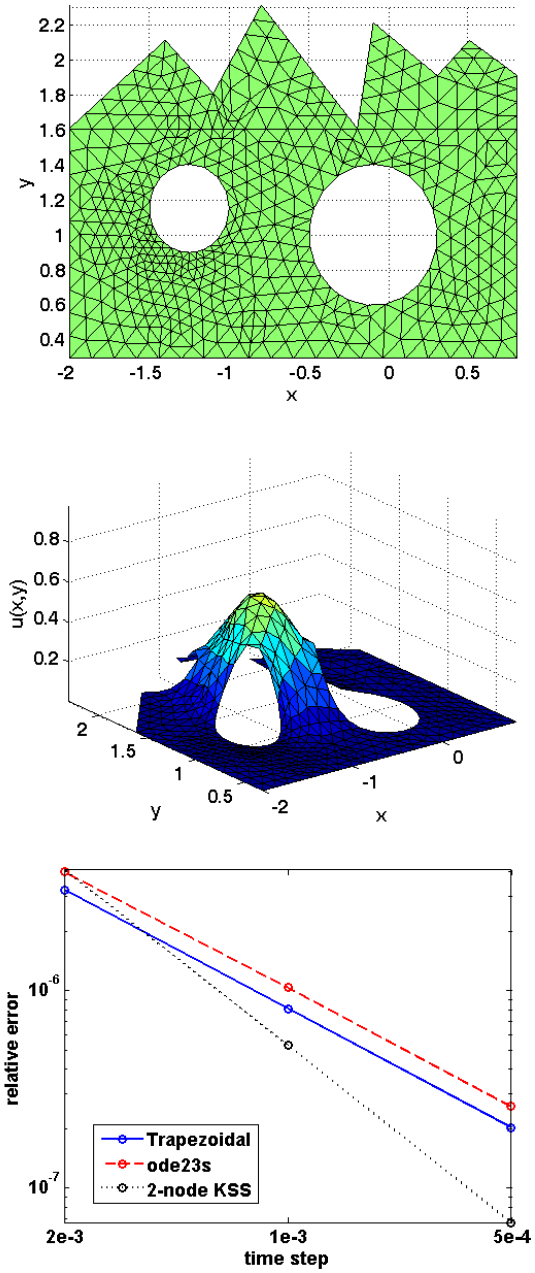

Fig. 1. Left plot: triangular mesh on region with polygonal boundary and holes. Center plot: solution to heat equation at $t = 0.1$ with Gaussian initial data. Right plot: estimates of temporal relative error in trapezoidal rule (blue solid curve), MATLAB solver `ode23s` (red dashed curve), and 2-node KSS method (black dotted curve).

Explicit time-stepping methods can be unstable if the time step is large relative to the eigenvalues of the stiffness matrix. This is illustrated in Figure 2, in which forward Euler exhibits instability, while a solution computed using a 2-node KSS method, with the same time step, does not, even though it too is an explicit method. It is worth noting that if the mesh is refined, then the explicit MATLAB solver `ode23` is unstable for the same time steps as those shown in Figure 1, whereas the explicit 2-node KSS method is not.

C. Meshless Methods

In view of the computational expense of constructing meshes for finite element discretization, meshless methods

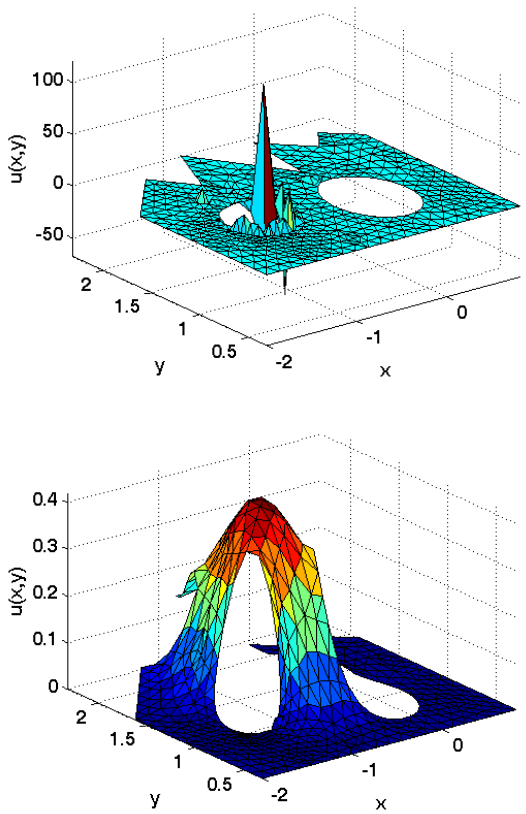

Fig. 2. Left plot: solution computed using forward Euler at $t = 0.2$ with time step $\Delta t = 0.04$. Right plot: solution computed using 2-node KSS method with same final time and time step.

[2], [3], [29] have emerged as an alternative approach for PDE arising from a variety of applications, such as fluid dynamics and solid mechanics [4], [7], [31]. A variety of meshless methods make use of radial basis functions (RBFs) [5] due to their usefulness for PDE [16], such as Kansa's RBF collocation method [6], [8], [33]. However, these methods require the solution of dense systems of linear equations. This shortcoming has inspired several localized meshless methods [28], [34], [36] that yield sparse matrix representations of differential operators.

To illustrate how ideas from KSS methods can benefit meshless methods, we consider a local meshless method known as the Explicit Local Method of Approximate Particular Solutions (ELMAPS) [38], applied to the heat equation

$$\frac{\partial u}{\partial t}(\mathbf{x}, t) = k^2 \Delta u(\mathbf{x}, t) + F(\mathbf{x}, t), \quad \mathbf{x} \in \Omega, \quad t > 0, \quad (11)$$

with Dirichlet boundary conditions. Let $\mathbf{x}_i$, $i = 1, 2, \ldots, N$, be collocation points in $\overline{\Omega}$. For each $\mathbf{x}_i$, let $\Omega_i \subset \Omega$ be a local domain that contains the n nearest neighbors to $\mathbf{x}_i$, which we denote by $\{\mathbf{x}_k^{[i]}\}_{k=1}^n$. Then, on each Ω_i, the solution at time t is approximated by a linear combination of basis functions $\Phi_{k,i}(\mathbf{x}) = \Phi(\|\mathbf{x} - \mathbf{x}_k^{[i]}\|)$, $k = 1, 2, \ldots, n$, that satisfy

$$\Delta \Phi(r) = \phi(r),$$

where $\phi(r)$ is an RBF. The coefficients of the linear combination are obtained by solving an $n \times n$ system of equations, that is time-independent so that LU factors can be stored and reused.

Because $k^2 \Delta u(\mathbf{x}, t_m)$ is approximated by a linear combination of only n basis functions chosen from the set $\{\Phi(\mathbf{x} - \mathbf{x}_j)\}_{j=1}^N$, the problem (11) is now approximated by a system of ODE

$$\mathbf{u}_N'(t) = A_N \mathbf{u}_N(t) + \mathbf{g}_N(t), \quad (12)$$

where $\mathbf{g}_N(t)$ arises from the source term and boundary conditions, A is a sparse $N \times N$ matrix with at most n nonzero elements per row, and

$$[\mathbf{u}_N(t)]_j = u(\mathbf{x}_j, t), \quad j = 1, 2, \ldots, N.$$

KSS methods can be adapted to this system in the same way as for the finite element case. During each time step, the value of the solution at each collocation point $\mathbf{x}_j$ is obtained by approximating bilinear forms of the form $\mathbf{u}^T f(A_N)\mathbf{v}$, where $\mathbf{u}$ is a standard basis vector and $\mathbf{v}$ is derived from the solution from the previous time step and the boundary conditions.

The process is illustrated through an example in which the problem (11), with $k = \pi^{-1}$, is solved on the star-shaped domain defined by the polar inequality $r \leq 1 + \cos^2(4\theta)$, $0 \leq \theta \leq 2\pi$, with Dirichlet boundary conditions $u(x, y, t) = e^{-t-\pi(x+y+4)} + x + y$ and source term $F(x, y, t) = -3e^{-t-\pi(x+y+4)}$. The chosen radial basis functions are $\Phi(r) = \sqrt{r^2 + c^2}$, with $c = 0.5$. The initial data is $f(x, y) = e^{-\pi(x+y+4)}$. The 272 collocation points are shown in Figure 3, along with error estimates for ELMAPS, `ode23s` and a 2-node KSS method, which exhibit the expected linear, quadratic and cubic orders of accuracy, respectively.

IV. Summary

The overarching goal of the research directions described in this paper is to bridge a fundamental, and worsening, disconnect between the time-stepping methods that are most often used in applications, and the rapidly evolving computing environments in which they are used. Many of these methods, some of which have been in existence for over one hundred years, are not designed with stiff systems in mind. They fall into two broad categories:

- explicit methods, that require very small time steps in order to maintain stability, and
- implicit methods, that can use larger step sizes than explicit methods, but usually require the solution of a large and ill-conditioned system of equations during each time step.

There are also implicit time-stepping methods, called stiff solvers, that are designed with stiff systems in mind (see, for example, [35]), but they usually require the computation and inversion of the Jacobian, which can be quite expensive, especially for large-scale systems.

In many cases, the solution of such systems of ODE can be expressed in terms of the approximate evaluation of a matrix function such as the exponential. Over the last few decades, much advancement has been made in the approximation of quantities of the form $f(A)\mathbf{v}$ and $\mathbf{u}^T f(A)\mathbf{v}$. Techniques for computing $f(A)\mathbf{v}$, such as those described in [12], [13], have been used for several years to solve systems of ODE, but can encounter the same difficulties with stiffness.

On the other hand, techniques for computing the bilinear form $\mathbf{u}^T f(A)\mathbf{v}$, normally used for computing selected

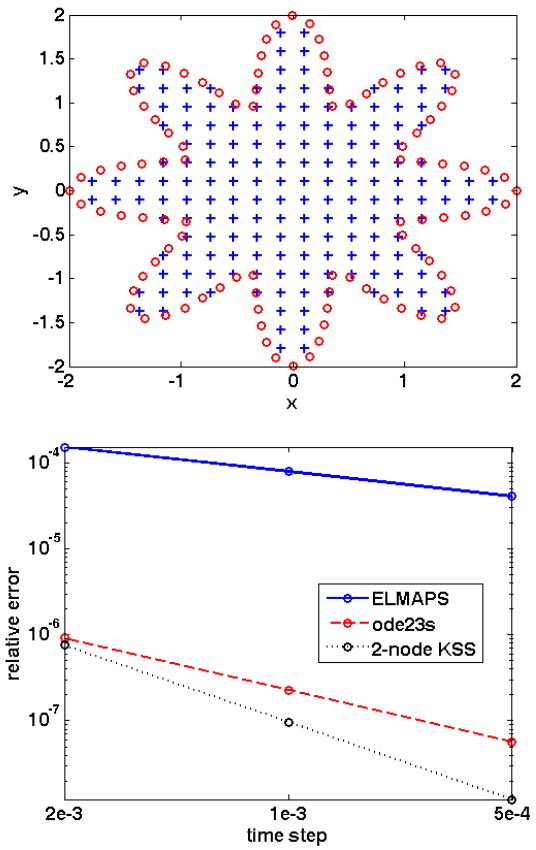

Fig. 3. Left plot: collocation points chosen from the domain $r \leq 1 + \cos^2(4\theta)$, $0 \leq \theta \leq 2\pi$. Right plot: temporal error estimates of ELMAPS, `ode23s` and 2-node KSS method applied to (11), at $t = 0.1$ and various time steps. Details of the problem are given in the text.

components of the solution of a system of equations, open the door to rethinking how time-stepping is carried out. Because they allow individual attention to be paid to each component of $f(A)\mathbf{v}$, they can be used to construct new time-stepping methods, based on KSS methods, that are more effective for large-scale problems. As an ingredient in these new methods, an established time-stepping method can be used more effectively, in that it is only applied to small-scale systems, for which any stiffness is far more easily handled.

REFERENCES

[1] Atkinson, K.: *An Introduction to Numerical Analysis, 2nd Ed.* Wiley (1989)

[2] Atluri, S. N.: *The Meshless Method (MLPG) for Domain and BIE Discretization*. Tech Science Press, Forsyth (2004).

[3] Atluri, S. N., Shen, S.: *The Meshless Local Petrov-Galerkin (MLPG) Method*. Tech Science Press, Encino (2002).

[4] Atluri, S. N., Shen, S.: *The Meshless Method*. Tech Science Press, Forsyth (2002).

[5] Buhmann, M. D.: *Radial Basis Functions, Theory and Implementations*. Cambridge University Press (2003).

[6] Chen, W.: New rbf collocation schemes and kernel rbfs with applications. *Lecture Notes in Computational Science and Engineering* **26** (2002) 7586.

[7] Fasshauer, G. E., Liu, G.: *Mesh free methods: moving beyond the finite element method*. CR Press, Boca Raton (2007).

[8] Franke, R.: Scattered data interpolation: tests of some methods. *Mathematics of Computation* **48** (1982) 181200.

[9] Golub, G. H., Meurant, G.: Matrices, Moments and Quadrature. *Proceedings of the 15th Dundee Conference*, June-July 1993, Griffiths, D. F., Watson, G. A. (eds.), Longman Scientific & Technical (1994)

[10] Golub, G. H., Underwood, R.: The block Lanczos method for computing eigenvalues. *Mathematical Software III*, J. Rice Ed., (1977) 361-377.

[11] Golub, G. H., Welsch, J.: Calculation of Gauss Quadrature Rules. *Math. Comp.* **23** (1969) 221-230.

[12] Hochbruck, M., Lubich, C.: A Gautschi-type method for oscillatory second-order differential equations, *Numer. Math.* **83** (1999) 403-426.

[13] Hochbruck, M., Lubich, C.: On Krylov Subspace Approximations to the Matrix Exponential Operator. *SIAM J. Numer. Anal.* **34** (1996) 1911-1925.

[14] Hochbruck, M., Lubich, C., Selhofer, H.: Exponential Integrators for Large Systems of Differential Equations. *SIAM J. Sci. Comput.* **19** (1998) 1552-1574.

[15] Hughes, T. J. R.: *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. Dover (2000).

[16] Kansa, E. J.: Multiquadrics - a scattered data approximation scheme with applications to computational fluid dynamics - II. *Comput. Math. Applic.* **19**(8/9) (1990) 147161.

[17] Lambers, J. V.: Derivation of High-Order Spectral Methods for Time-dependent PDE using Modified Moments. *Electron. T. Numer. Ana.* **28** (2008) 114-135.

[18] Lambers, J. V.: Enhancement of Krylov Subspace Spectral Methods by Block Lanczos Iteration. *Electron. T. Numer. Ana.* **31** (2008) 86-109.

[19] Lambers, J. V.: An Explicit, Stable, High-Order Spectral Method for the Wave Equation Based on Block Gaussian Quadrature. *IAENG Journal of Applied Mathematics* **38** (2008) 333-348.

[20] Lambers, J. V.: Implicitly Defined High-Order Operator Splittings for Parabolic and Hyperbolic Variable-Coefficient PDE Using Modified Moments. *International Journal of Computational Science* **2** (2008) 376-401.

[21] Lambers, J. V.: Krylov Subspace Methods for Variable-Coefficient Initial-Boundary Value Problems. Ph.D. Thesis, Stanford University (2003).

[22] Lambers, J. V.: Krylov Subspace Spectral Methods for the Time-Dependent Schrödinger Equation with Non-Smooth Potentials. *Numer. Algorithms* **51** (2009) 239-280.

[23] Lambers, J. V.: Krylov Subspace Spectral Methods for Variable-Coefficient Initial-Boundary Value Problems. *Electron. T. Numer. Ana.* **20** (2005) 212-234.

[24] Lambers, J. V.: Practical Implementation of Krylov Subspace Spectral Methods. *J. Sci. Comput.* **32** (2007) 449-476.

[25] Lambers, J. V.: Solution of Time-Dependent PDE Through Component-wise Approximation of Matrix Functions. *IAENG Journal of Applied Mathematics* **41**(1) (2011) 1-10.

[26] Lambers, J. V.: Spectral Methods for Time-dependent Variable-coefficient PDE Based on Block Gaussian Quadrature. *Proceedings of the 2009 International Conference on Spectral and High-Order Methods* (2010) in press.

[27] Lambers, J. V.: A Spectral Time-Domain Method for Computational Electrodynamics. *Adv. Appl. Math. Mech.* **1**(6) (2009) 781-798.

[28] Lee, C. K., Liu, X., Fan, S. C.: Local multiquadric approximation for solving boundary value problems. *Computational Mechanics* **30** (2003) 396409.

[29] Liu, G. R., Dai, K. Y., Lim, Y. T., Gu, K. M.: A point interpolation mesh free method for static and frequency analysis of two-dimensional piezoelectric structures. *Computational Mechanics* **29** (2002) 510519.

[30] Logan, D. L.: *A first course in the finite element method*. Cengage Learning (2007).

[31] Mai-Duy, N., Tran-Cong, T.: Mesh-free radial basis function network methods with domain decomposition for approximation of functions and numerical solution of poissons equation. *Engineering Analysis with Boundary Elements* **26** (2002) 133156.

[32] Moret, I., Novati, P.: RD-rational approximation of the matrix exponential operator. *BIT* **44** (2004) 595-615.

[33] Šarler, B.: From global to local radial basis function collocation method for transport phenomena. *Advances in Meshfree Techniques, Computational Methods in Applied Sciences* **5** (2007) 257282.

[34] Šarler, B., Vertnik, R.: Meshfree explicit local radial basis function collocation method for diffusion problems. *Computers and Mathematics with Applications* **51** (2006) 12691282.

[35] Shampine, L. F., Reichelt, M. W.: The MATLAB ODE suite. *SIAM Journal of Scientific Computing* **18** (1997) 1-22.

[36] Shu, C., Ding, H., Yeo, K. S.: Local radial basis function-based differential quadrature method and its application to solve two dimensional incompressible navier-stokes equations. *Computer Methods and Applied Mechanics Engineering* **192** (2003) 941954.

[37] van den Eshof, J., Hochbruck, M.: Preconditioning Lanczos approximations to the matrix exponential. *SIAM J. of Sci. Comput.* **27** (2006) 1438-1457.

[38] Yao, G.: Local Radial Basis Function Methods for Solving Partial Differential Equations. Ph.D. Thesis, University of Southern Mississippi (2010).