Solution of Time-Dependent PDE Through Component-wise Approximation of Matrix Functions

James V. Lambers *

Abstract—**Block Krylov subspace spectral (KSS) methods are a "best-of-both-worlds" compromise between explicit and implicit time-stepping methods for variable-coefficient PDE, in that they combine the efficiency of explicit methods and the stability of implicit methods, while also achieving spectral accuracy in space and high-order accuracy in time. Block KSS methods compute each Fourier coefficient of the solution using techniques developed by Gene Golub and Gérard Meurant for approximating elements of functions of matrices by block Gaussian quadrature in the spectral, rather than physical, domain. This paper demonstrates the superiority of block KSS methods, in terms of accuracy and efficiency, to other Krylov subspace methods in the literature. It is also described how the ideas behind block KSS methods can be applied to a variety of equations, including problems for which Fourier spectral methods are not normally feasible. In particular, the versatility of the approach behind block KSS methods is demonstrated through application to nonlinear diffusion equations for signal and image processing, and adaptation to finite element discretization.**

Keywords: spectral methods, Gaussian quadrature, block Lanczos, Maxwell's equations, heat equation

1 Introduction

In [17] a class of methods, called block Krylov subspace spectral (KSS) methods, was introduced for the purpose of solving parabolic variable-coefficient PDE. These methods are based on techniques developed by Golub and Meurant in [8] for approximating elements of a function of a matrix by Gaussian quadrature in the *spectral* domain. In [18], these methods were generalized to the second-order wave equation, for which these methods have exhibited even higher-order accuracy.

It has been shown in these references that KSS methods, by employing different approximations of the solution operator for each Fourier coefficient of the solution, achieve higher-order accuracy in time than other Krylov subspace methods (see, for example, [14]) for stiff systems of ODE, and they are also quite stable, considering that they are explicit methods. They are also effective for solving systems of coupled equations, such as Maxwell's equations [23], and elliptic PDE such as Poisson's equation or the Helmholtz equation [21].

In this paper, we review block KSS methods, and compare their performance to other Krylov subspace methods from the literature. It is then shown that block KSS methods are applicable to PDE other than those best suited to Fourier spectral methods. Section 2 presents the approximation of bilinear forms involving functions of matrices by block Gaussian quadrature, as developed by Golub and Meurant. Section 3 describes how block KSS methods are built on this work, as applied to parabolic problems, and summarizes their main properties. Section 4 discusses implementation details, and demonstrates why KSS methods need to explicitly generate only one Krylov subspace, although information from several is used. In Section 5, we discuss modifications to block KSS methods in order to apply them to systems of coupled equations, such as Maxwell's equations. Numerical results are presented in Section 6, in which block KSS methods are compared to other Krylov subspace methods. In Section 7, block KSS methods are applied to nonlinear diffusion equations for signal and image processing. In Section 8, we discuss the adaptation of block KSS methods to bases other than the Fourier basis, which allows application to problems featuring complicated geometries or more general boundary conditions. Conclusions and discussion of future work are in Section 9.

2 Elements of Functions of Matrices

In [8] Golub and Meurant describe a method for computing quantities of the form

$$\mathbf{u}^T f(A)\mathbf{v}, \tag{1}$$

*Submitted September 28, 2010. The University of Southern Mississippi, Department of Mathematics, Hattiesburg, MS 39406-0001 USA Tel/Fax: 601-266-5784/5818 Email: James.Lambers@usm.edu

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.

The basic idea is as follows: since the matrix A is symmetric positive definite, it has real eigenvalues

$$b = \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_N = a > 0, \tag{2}$$

and corresponding orthogonal eigenvectors $\mathbf{q}_j$, $j = 1, \ldots, N$. Therefore, the quantity (1) 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

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

As discussed in [8], the integral $I[f]$ can be approximated using Gaussian 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, $j = 1, \ldots, K$, as well as the weights w_j, $j = 1, \ldots, K$, can be obtained using the symmetric Lanczos algorithm if $\mathbf{u} = \mathbf{v}$, and the unsymmetric Lanczos algorithm if $\mathbf{u} \neq \mathbf{v}$ (see [11]).

In the case $\mathbf{u} \neq \mathbf{v}$, there is a possibility that the weights may not be positive, which destabilizes the quadrature rule (see [2] for details). Instead, we consider

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

which results in the 2×2 matrix

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

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

In [8] Golub and Meurant showed how a block method can be used to generate quadrature formulas. We will describe this process here in more detail. The integral $\int_a^b f(\lambda)\, d\mu(\lambda)$ is now a 2×2 symmetric matrix and the most general K-node quadrature formula is of the form

$$\int_a^b f(\lambda)\, d\mu(\lambda) = \sum_{j=1}^{K} W_j f(T_j) W_j + error, \tag{8}$$

with T_j and W_j being symmetric 2×2 matrices. By diagonalizing each T_j, we obtain the simpler formula

$$\int_a^b f(\lambda)\, d\mu(\lambda) = \sum_{j=1}^{2K} f(\lambda_j)\mathbf{v}_j\mathbf{v}_j^T + error, \tag{9}$$

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

$$\mathcal{T}_K = \begin{bmatrix} M_1 & B_1^T & & & \\ B_1 & M_2 & B_2^T & & \\ & \ddots & \ddots & \ddots & \\ & & B_{K-2} & M_{K-1} & B_{K-1}^T \\ & & & B_{K-1} & M_K \end{bmatrix}, \tag{10}$$

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

3 Krylov Subspace Spectral Methods

We now review block KSS methods, which are easier to describe for parabolic problems. Let $S(t) = \exp[-Lt]$ represent the exact solution operator of the problem

$$u_t + Lu = 0, \quad t > 0, \tag{11}$$

$$u(x, 0) = f(x), \quad 0 < x < 2\pi, \tag{12}$$

$$u(0, t) = u(2\pi, t), \quad t > 0. \tag{13}$$

The operator L is a second-order, self-adjoint, positive definite differential operator of the form

$$Lu = (p(x)u_x)_x + q(x)u, \tag{14}$$

where $p(x) > 0$ and $q(x) \geq 0$ on $[0, 2\pi]$. It follows that L is self-adjoint and positive definite.

Let $\langle \cdot, \cdot \rangle$ denote the standard inner product of functions defined on $[0, 2\pi]$. Block Krylov subspace spectral methods, introduced in [17], use Gaussian quadrature on the spectral domain to compute the Fourier coefficients of the solution. These methods 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 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 $\tilde{u}(x, t_n)$,

$$\hat{u}(\omega, t_{n+1}) = \left\langle \frac{1}{\sqrt{2\pi}} e^{i\omega x}, S(\Delta t)\tilde{u}(x, t_n) \right\rangle. \tag{15}$$

This is accomplished by applying the approach from the previous section for approximating (1), 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, respectively, $\hat{\mathbf{e}}_\omega$ 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 the approximate solution at time t_n, evaluated on an N-point uniform grid.

For each wave number $\omega = -N/2+1, \ldots, N/2$, we define $R_0(\omega) = \begin{bmatrix} \hat{\mathbf{e}}_\omega & \mathbf{u}^n \end{bmatrix}$ and compute the QR factorization $R_0(\omega) = X_1(\omega)B_0(\omega)$. We then carry out block Lanczos iteration, applied to the discretized operator L_N, to obtain a block tridiagonal matrix $\mathcal{T}_K(\omega)$ of the form (10), where each entry is a function of ω.

Then, we can express each Fourier coefficient of the approximate solution at the next time step as

$$[\hat{\mathbf{u}}^{n+1}]_\omega = \left[B_0^H E_{12}^H \exp[-\mathcal{T}_K(\omega)\Delta t] E_{12} B_0\right]_{12} \qquad (16)$$

where $E_{12} = \begin{bmatrix} \mathbf{e}_1 & \mathbf{e}_2 \end{bmatrix}$. The computation of (16) consists of computing the eigenvalues and eigenvectors of $\mathcal{T}_K(\omega)$ in order to obtain the nodes and weights for Gaussian quadrature, as described earlier.

This algorithm has local temporal accuracy $O(\Delta t^{2K-1})$ [17]. Furthermore, block KSS methods are more accurate than the original KSS methods described in [20], even though they have the same order of accuracy, because the solution $\mathbf{u}^n$ plays a greater role in the determination of the quadrature nodes. They are also more effective for problems with oscillatory or discontinuous coefficients [17].

Block KSS methods are even more accurate for the second-order wave equation, for which block Lanczos iteration is used to compute both the solution and its time derivative. In [18, Theorem 6], it is shown that when the leading coefficient is constant and the coefficient $q(x)$ is bandlimited, the 1-node KSS method, which has second-order accuracy in time, is also *unconditionally stable*. In general, as shown in [18], the local temporal error is $O(\Delta t^{4K-2})$ when K block Gaussian nodes are used.

4 Implementation

KSS methods compute a Jacobi matrix corresponding to *each* Fourier coefficient, in contrast to traditional Krylov subspace methods that normally use only a single Krylov subspace generated by the initial data or 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 [19, 22].

With these considerations, the algorithm for a single time step of a 1-node block KSS method for solving (11), where $Lu = -pu_{xx} + q(x)u$, with appropriate initial conditions and periodic boundary conditions, is as follows. We denote the average of a function $f(x)$ on $[0, 2\pi]$ by $\overline{f}$, and the computed solution at time t_n by u^n.

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

It should be noted that for a parabolic problem such as (11), the loop over ω only needs to account for non-negligible Fourier coefficients of the solution, which are relatively few due to the smoothness of solutions to such problems.

5 Application to Maxwell's Equations

We consider Maxwell's equation on the cube $[0, 2\pi]^3$, with periodic boundary conditions. Assuming nonconductive material with no losses, we have

$$\mathrm{div}\,\hat{\mathbf{E}} = 0, \quad \mathrm{div}\,\hat{\mathbf{H}} = 0, \qquad (17)$$

$$\mathrm{curl}\,\hat{\mathbf{E}} = -\mu\frac{\partial \hat{\mathbf{H}}}{\partial t}, \quad \mathrm{curl}\,\hat{\mathbf{H}} = \varepsilon\frac{\partial \hat{\mathbf{E}}}{\partial t}, \qquad (18)$$

where $\hat{\mathbf{E}}$, $\hat{\mathbf{H}}$ are the vectors of the electric and magnetic fields, and ε, μ are the electric permittivity and magnetic permeability, respectively.

Taking the curl of both sides of (18) yields

$$\mu\varepsilon\frac{\partial^2 \hat{\mathbf{E}}}{\partial t^2} = \Delta\hat{\mathbf{E}} + \mu^{-1}\mathrm{curl}\,\hat{\mathbf{E}} \times \nabla\mu, \qquad (19)$$

$$\mu\varepsilon\frac{\partial^2 \hat{\mathbf{H}}}{\partial t^2} = \Delta\hat{\mathbf{H}} + \varepsilon^{-1}\mathrm{curl}\,\hat{\mathbf{H}} \times \nabla\varepsilon. \qquad (20)$$

In this section, we discuss generalizations that must be made to block KSS methods in order to apply them to a

non-self-adjoint system of coupled equations such as (19). Additional details are given in [23].

First, we consider the following 1-D problem,

$$\frac{\partial^2 \mathbf{u}}{\partial t^2} + L\mathbf{u} = 0, \quad t > 0, \tag{21}$$

with appropriate initial conditions, and periodic boundary conditions, where $\mathbf{u} : [0, 2\pi] \times [0, \infty) \to \mathbb{R}^n$ for $n > 1$, and $L(x, D)$ is an $n \times n$ matrix where the (i, j) entry is an a differential operator $L_{ij}(x, D)$ of the form

$$L_{ij}(x, D)u(x) = \sum_{\mu=0}^{m_{ij}} a_\mu^{ij}(x) D^\mu u, \quad D = \frac{d}{dx}, \tag{22}$$

with spatially varying coefficients a_μ^{ij}, $\mu = 0, 1, \ldots, m_{ij}$.

Generalization of KSS methods to a system of the form (21) can proceed as follows. For $i, j = 1, \ldots, n$, let $\overline{L}_{ij}(D)$ be the constant-coefficient operator obtained by averaging the coefficients of $L_{ij}(x, D)$ over $[0, 2\pi]$. Then, for each wave number ω, we define $L(\omega)$ be the matrix with entries $\overline{L}_{ij}(\omega)$, i.e., the symbols of $\overline{L}_{ij}(D)$ evaluated at ω. Next, we compute the spectral decomposition of $L(\omega)$ for each ω. For $j = 1, \ldots, n$, let $\mathbf{q}_j(\omega)$ be the Schur vectors of $L(\omega)$. Then, we define our test and trial functions by $\vec{\phi}_{j,\omega}(x) = \mathbf{q}_j(\omega) \otimes e^{i\omega x}$.

For Maxwell's equations, the matrix A_N that discretizes the operator

$$A\hat{\mathbf{E}} = \frac{1}{\mu\varepsilon}\left(\Delta\hat{\mathbf{E}} + \mu^{-1}\mathrm{curl}\,\hat{\mathbf{E}} \times \nabla\mu\right)$$

is not symmetric, and for each coefficient of the solution, the resulting quadrature nodes λ_j, $j = 1, \ldots, 2K$, from (9) are now complex and must be obtained by a straightforward modification of block Lanczos iteration for unsymmetric matrices.

6 Numerical Results

In this section, we compare the performance of block KSS methods with various methods based on exponential integrators [13, 15, 30].

6.1 Parabolic Problems

We first consider a 1-D parabolic problem of the form (11), where the differential operator L is defined by $Lu(x) = -pu''(x) + q(x)u(x)$, where $p \approx 0.4$ and

$$\begin{aligned} q(x) \quad \approx \quad & -0.44 + 0.03\cos x - 0.02\sin x + 0.005\cos 2x - \\ & 0.004\sin 2x + 0.0005\cos 3x \end{aligned}$$

is constructed so as to have the smoothness of a function with three continuous derivatives, as is the initial data $u(x, 0)$. Periodic boundary conditions are imposed.

We solve this problem using the following methods:

- A 2-node block KSS method. Each time step requires construction of a Krylov subspace of dimension 3 generated by the solution, and the coefficients of L^2 and L^3 are computed during a preprocessing step.
- A preconditioned Lanczos iteration for approximating $e^{-\tau A}\mathbf{v}$, introduced in [24] for approximating the matrix exponential of sectorial operators, and adapted in [30] for efficient application to the solution of parabolic PDE. In this approach, Lanczos iteration is applied to $(I+hA)^{-1}$, where h is a parameter, in order to obtain a restricted rational approximation of the matrix exponential. We use $m = 4$ and $m = 8$ Lanczos iterations, and choose $h = \Delta t/10$, as in [30].
- A method based on exponential integrators, from [13], that is of order 3 when the Jacobian is approximated to within $O(\Delta t)$. We use $m = 8$ Lanczos iterations.

Since the exact solution is not available, the error is estimated by taking the ℓ_2-norm of the relative difference between each solution, and that of a solution computed using a smaller time step $\Delta t = 1/64$ and the maximum number of grid points.

The results are shown in Figure 1. As the number of grid points is doubled, only the block KSS method shows an improvement in accuracy; the preconditioned Lanczos method exhibits a slight degradation in performance, while the explicit fourth-order exponential integrator-based method requires that the time step be reduced by a factor of 4 before it can deliver the expected order of convergence; similar behavior was demonstrated for an explicit 3rd-order method from [14] in [20].

The preconditioned Lanczos method requires 8 Lanczos iterations to match the accuracy of a block KSS method that uses only 2. On the other hand, the block KSS method incurs additional expense due to (1) the computation of the moments of L, for each Fourier coefficient, and (2) the exponentiation of separate Jacobi matrices for each Fourier coefficient. These expenses are mitigated by the fact that the first takes place once, during a preprocessing stage, and both tasks require an amount of work that is proportional not to the number of grid points, but to the number of non-negligible Fourier coefficients of the solution.

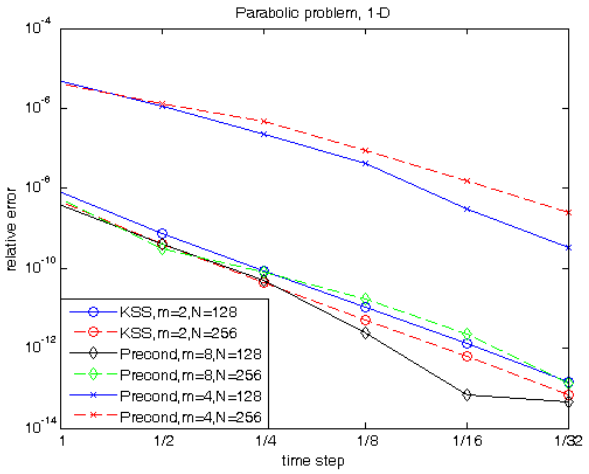

Figure 1: Estimates of relative error at $t = 0.1$ in solutions of (11) computed using preconditioned exponential integrator [30] with 4 and 8 Lanczos iterations, a 4th-order method based on an exponential integrator [15], and a 2-node block KSS method. All methods compute solutions on an N-point grid, with time step Δt, for various values of N and Δt.

6.2 Maxwell's Equations

We now apply a 2-node block KSS method to (19), with initial conditions

$$\hat{\mathbf{E}}(x,y,z,0) = \mathbf{F}(x,y,z), \quad \frac{\partial \hat{\mathbf{E}}}{\partial t}(x,y,z,0) = \mathbf{G}(x,y,z), \tag{23}$$

with periodic boundary conditions. The coefficients μ and ε are given by

$$\begin{aligned}\mu(x,y,z) &= 0.4077 + 0.0039\cos z + 0.0043\cos y - \\ &\quad 0.0012\sin y + 0.0018\cos(y+z) + \\ &\quad 0.0027\cos(y-z) + 0.003\cos x + \\ &\quad 0.0013\cos(x-z) + 0.0012\sin(x-z) + \\ &\quad 0.0017\cos(x+y) + \qquad (24)\\ &\quad 0.0014\cos(x-y), \qquad (25)\\ \varepsilon(x,y,z) &= 0.4065 + 0.0025\cos z + 0.0042\cos y + \\ &\quad 0.001\cos(y+z) + 0.0017\cos x + \\ &\quad 0.0011\cos(x-z) + 0.0018\cos(x+y) + \\ &\quad 0.002\cos(x-y). \qquad (26)\end{aligned}$$

The components of $\mathbf{F}$ and $\mathbf{G}$ are generated in a similar fashion, except that the x- and z-components are zero.

We use a block KSS method that uses $K = 2$ block quadrature nodes per coefficient in the basis described in Section 5, that is 6th-order accurate in time, and a cosine method based on a Gautschi-type exponential integrator [13, 15]. This method is second-order in time, and in these experiments, we use $m = 2$ Lanczos iterations to approximate the Jacobian. It should be noted that when m is increased, even to a substantial degree, the results are negligibly affected.

Figure 2 demonstrates the convergence behavior for both methods. At both spatial resolutions, the block KSS method exhibits approximately 6th-order accuracy in time as Δt decreases, except that for $N = 16$, the spatial error arising from truncation of Fourier series is significant enough that the overall error fails to decrease below the level achieved at $\Delta t = 1/8$. For $N = 32$, the solution is sufficiently resolved in space, and the order of overgence as $\Delta t \to 0$ is approximately 6.1.

We also note that increasing the resolution does not pose any difficulty from a stability point of view. Unlike explicit finite-difference schemes that are constrained by a CFL condition, KSS methods do not require a reduction in the time step to offset a reduction in the spatial step in order to maintain boundedness of the solution, because their domain of dependence includes the entire spatial domain for any Δt.

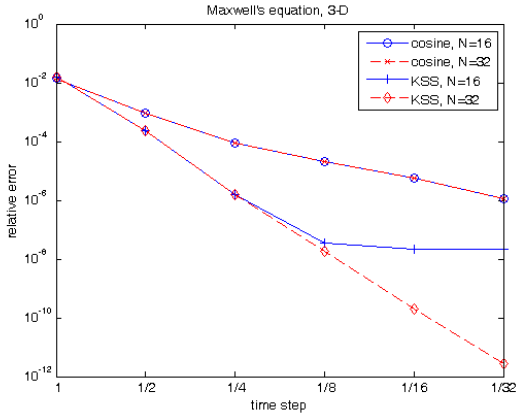

Figure 2: Estimates of relative error at $t = 1$ in solutions of (19), (23) computed using a cosine method based on a Gautschi-type exponential integrator [13, 15] with 2 Lanczos iterations, and a 2-node block KSS method. Both methods compute solutions on an N^3-point grid, with time step Δt, for various values of N and Δt.

The Gautschi-type exponential integrator method is second-order accurate, as expected, and delivers nearly identical results for both spatial resolutions, but even with a Krylov subspace of much higher dimension than that used in the block KSS method, it is only able to achieve at most second-order accuracy, whereas a block

KSS method, using a Krylov subsapce of dimension 3, achieves sixth-order accuracy. This is due to the incorporation of the moments of the spatial differential operator into the computation, and the use of Gaussian quadrature rules specifically tailored to each Fourier coefficient.

7 Other Spatial Discretizations

The main idea behind KSS methods, that higher-order accuracy in time can be obtained by componentwise approximation, 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

$$\mathbf{u}_j^H A^k \mathbf{u}_j, \quad \mathbf{u}_j^H A^k \mathbf{v}, \quad \mathbf{v}^H A^k \mathbf{v} \tag{27}$$

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 (27) can be computed in a preprocessing stage if the operator discretized by A is independent of time, or it 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 (27) are readily obtained from bases for Krylov subspaces generated by $\mathbf{v}$. 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.

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 ODEs 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 [16].

We now present some algorithmic details. Let A be a symmetric positive definite matrix. For $j = 1, \ldots, N$, we approximate $\mathbf{e}_j^T \exp[-At]\mathbf{f}$ using the following algorithm

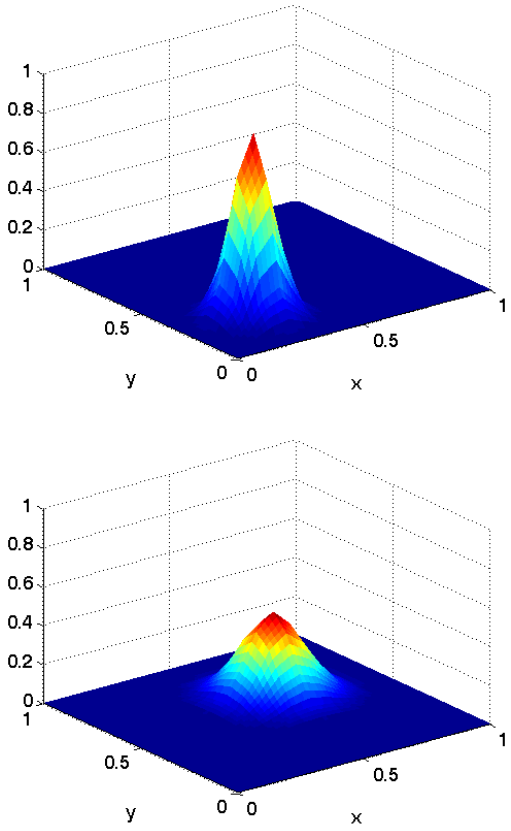

Figure 3: Solution at $t = 0$ (top) and $t = 0.5$ (bottom).

based on block Lanczos iteration. First, we perform the following initializations:

$$X_0 = 0 \tag{28}$$
$$R_0 = \begin{bmatrix} \mathbf{e}_j & \mathbf{f} \end{bmatrix}. \tag{29}$$

Next, we let K denote the number of block Gaussian quadrature nodes to be used for each component. Then, for $k = 1, \ldots, K$, we compute the following:

$$R_{k-1} = X_k \Gamma_{k-1} \quad \text{(QR factorization)} \tag{30}$$
$$M_k = X_k^T A X_k \tag{31}$$
$$R_k = AX_k - X_k M_k - X_{k-1}\Gamma_{k-1} \tag{32}$$

Next, we compute the block tridiagonal matrix $\mathcal{T}_K$ as in (10). Finally, we approximate each component of the solution as in (16):

$$\mathbf{e}_j^T \exp[-At]\mathbf{f} \approx \mathbf{e}_1^T \Gamma_0^T E_{12}^T \exp[-T_K t] E_{12} \Gamma_0 \mathbf{e}_2.$$

The challenge is to efficiently compute the elements of T_K for all $j = 1, \ldots, N$.

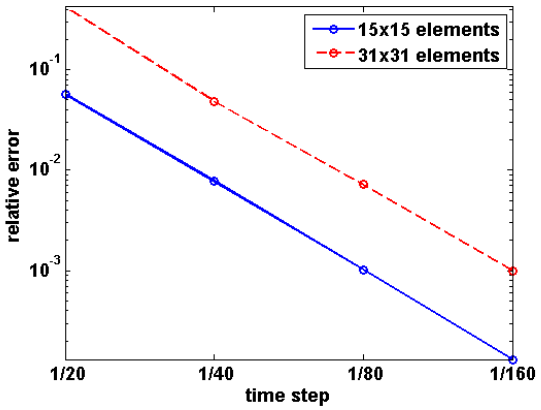

Figure 4: Relative error at $t = 0.5$ at various time steps.

To that end, we now consider an alternative approach to describing block Lanczos iteration, applied to all components simultaneously. We define

$$R_0 = \begin{bmatrix} I & \mathbf{f}\mathbf{e}^T \end{bmatrix}, \tag{33}$$

where $\mathbf{e}$ is an N-vector of all ones. Then, we define

$$R_0 = X_1\Gamma_0 \tag{34}$$

where Γ_0 is a $2N \times 2N$ matrix with block structure

$$\Gamma_0 = \begin{bmatrix} \Gamma_0^{11} & \Gamma_0^{12} \\ 0 & \Gamma_0^{22} \end{bmatrix}. \tag{35}$$

Each block is a diagonal matrix. The entries of these blocks satisfy

$$\begin{bmatrix} [R_0]_j & [R_0]_{j+N} \end{bmatrix} = \begin{bmatrix} [X_1]_j & [X_1]_{j+N} \end{bmatrix} \times \begin{bmatrix} [\Gamma_0]_{jj} & [\Gamma_0]_{j,j+N} \\ [\Gamma_0]_{j+N,j} & [\Gamma_0]_{j+N,j+N} \end{bmatrix} \tag{36}$$

for $j = 1, \ldots, N$. Because this factorization is actually a QR factorization of columns j and $j + N$ of R_0, the $(2, 1)$-block of Γ_0, Γ_i^{21}, is zero. The remaining entries can be computed as follows. We write

$$R_0 = \begin{bmatrix} R_{01} & R_{02} \end{bmatrix}, \quad X_1 = \begin{bmatrix} X_{11} & X_{12} \end{bmatrix}, \tag{37}$$

and then obtain

$$\begin{aligned}
\Gamma_0^{11} &= \mathrm{diag}(\mathbf{e}^T[R_{01} * R_{01}])^{1/2} && (38)\\
X_{11} &= R_{01}[\Gamma_0^{11}]^{-1} && (39)\\
\Gamma_0^{12} &= [\Gamma_0^{11}]^{-1}\mathrm{diag}(\mathbf{e}^T[R_{01} * R_{02}]) && (40)\\
Y_{12} &= R_{02} - R_{01}[\Gamma_0^{11}]^{-1}\Gamma_0^{12} && (41)\\
\Gamma_0^{22} &= \mathrm{diag}(\mathbf{e}^T[Y_{12} * Y_{12}])^{1/2} && (42)\\
X_{12} &= Y_{12}[\Gamma_0^{22}]^{-1} && (43)
\end{aligned}$$

where we denote the componentwise products of matrices A and B by $A * B$.

Next, we define

$$M_1 = (X_1^T A X_1) * (E_2 \otimes I_N), \tag{44}$$

where E_2 is a 2×2 matrix of ones, and I_N is the $N \times N$ identity matrix. In other words, M_1 is a $2N \times 2N$ matrix that is a 2×2 matrix of diagonal blocks, where each block is the diagonal of the corresponding block of $X_1^T A X_1$. It should be noted that M_1 can be computed without forming $X_1^T A X_1$ in its entirety. From

$$[M_1]_{ij} = \delta_{i \bmod N, j \bmod N} \sum_{k,\ell=1}^{N} [X_1]_{ki} A_{k\ell} [X_1]_{\ell j}, \tag{45}$$

and (37), we obtain

$$M_1 = \begin{bmatrix} \mathrm{diag}(\mathbf{e}^T[X_{11} * Y_{11}]) & \mathrm{diag}(\mathbf{e}^T[X_{11} * Y_{12}]) \\ \mathrm{diag}(\mathbf{e}^T[X_{12} * Y_{11}]) & \mathrm{diag}(\mathbf{e}^T[X_{12} * Y_{12}]) \end{bmatrix},$$

$$Y_{11} = AX_{11}, \quad Y_{12} = AX_{12}.$$

The remaining blocks M_k, $k = 2, \ldots, K$, can be computed in a similar fashion. The matrices $R_1, \ldots, R_{K-1}$ can be computed by standard matrix multiplication, taking sparisty into account. The matrices $X_2, \ldots, X_K$ and $\Gamma_2, \ldots, \Gamma_{K-1}$ can be computed by carrying out the QR factorization as in (38)-(43).

Once the matrices M_k and Γ_k are computed, we can compute the matrices $\mathcal{T}_K$, as defined in (10), for each component of the solution. Each of these matrices is $2K \times 2K$. From its eigenvalues and eigenvectors, we obtain the nodes and weights for block Gaussian quadrature that enable us to approximate the appropriate component of the solution.

As an example of the applicability of block Gaussian quadrature to spatial discretizations other than a Fourier spectral discretization, an adaptation of a 2-node block KSS method is used to solve an advection-diffusion problem on the rectangle $(0, 1)^2$, in which the flow is being advected from the origin at a 45-degree angle. The initial data is Gaussian, with homogeneous Dirichlet inflow boundary conditions, and homogeneous Neumann outflow boundary conditions. The Peclet number is 200. A 15×15 uniform mesh and piecewise bilinear basis functions are used. The results are shown in Figures 3 and 4. As in the Fourier spectral case, the adapted KSS method is 3rd-order accurate in time. We see that KSS methods can be just as effective for time-stepping in finite element models, such as that described in [28], as in Fourier spectral methods

8 Nonlinear Diffusion for Signal and Image Processing

In [12], Guidotti and the author introduced the nonlinear diffusion equation

$$u_t - \nabla \cdot (g(u)\nabla u) = 0, \tag{46}$$

$$g(u) = \frac{1}{1 + c^2|\nabla^{1-\varepsilon} u|^2}, \quad 0 < \epsilon < 1. \tag{47}$$

The diffusion coefficient $g(u)$ involves a slight weakening of the nonlinearity featured in the well-known Perona-Malik equation [26] for sharpening and denoising images, in order to overcome its drawbacks of being ill-posed and susceptible to "staircasing" effects [12], without introducing blurring, as some regularizations of Perona-Malik do [1, 5].

For diffusion equations such as (46), (47) it is generally not practical to use explicit time-stepping methods, because of the severe constraints they impose on the time step. A straightforward alternative is to use implicit time-stepping in conjunction with an iterative method such as MINRES [25]. However, this approach has not been found to be efficient, and solutions tend to exhibit staircasing and high-frequency oscillations.

These effects are illustrated by solving (46), (47) in one space dimension, with initial data equal to a characteristic function convolved with a Gaussian kernel. The results are shown in Figure 5. As the signal sharpens, the solution exhibits staircasing and high-frequency oscillations, which are eventually eliminated over time.

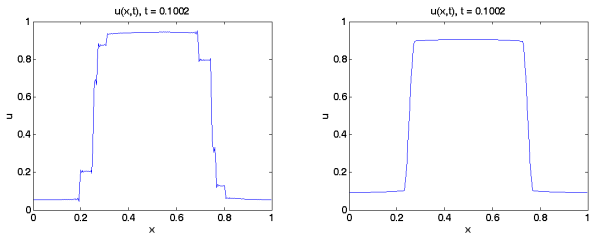

Figure 5: Sharpening of a smooth function, using $c = 1$, $\varepsilon = 0.1$, $\Delta t = 0.0003$. Left plot: backward Euler with MINRES for time-stepping. Right plot: KSS method.

To demonstrate the effectiveness of KSS methods for image processing applications, the same problem is solved with a 1-node KSS method. As seen in Figure 5, the previously observed staircasing and high-frequency oscillations do not occur.

We then illustrate the use of block KSS methods in denoising of images. Here the application (46) is investigated in the context of color images for which $u \in \mathbb{R}^3$. The experiments performed were not limited to a channel-by-channel generalization of (46), but the best results were obtained for such a choice. It would also conceivably be best to choose the contstant c taylored to each channel u^i, $i = 1, 2, 3$. Once more experiments deliver the best results when the constants are chosen equal to give

$$\{\, u^i_t - \nabla \cdot \left(\frac{1}{1 + c|\nabla^{1-\varepsilon} u^i|^2} \nabla u^i \right) = 0\,, u^i(0) = u^i_0\,, \tag{48}$$

for $i = 1, 2, 3$.

In order to make sense of this model, the fractional gradient appearing in the equations needs to be defined. It is convenient (but not necessary) to work with doubly periodic functions (non periodic images can always be made periodic by appropriate reflection across the boundaries as to avoid boundary effects. Then one has

$$\partial_z = \mathcal{F}_z^{-1} \mathrm{diag}\big[2\pi i k_z\big] \mathcal{F}_z\,,\ z = x, y\,,$$

where $\mathcal{F}$ denotes the discrete Fourier transform. The the fractional gradient is defined as $\nabla^\rho = \begin{bmatrix} \partial_x^\rho \\ \partial_y^\rho \end{bmatrix}$, where

$$\partial_z^\rho = \mathcal{F}_z^{-1} \mathrm{diag}\big[(2\pi i k_z)^\rho\big] \mathcal{F}_z\,, z = x, y\,.$$

The exponentiation of $\Lambda_{z;n,n}$ is carried out as follows:

$$(ik)^{1-\varepsilon} = |k|^{1-\varepsilon} e^{i\pi/2(1-\varepsilon)\mathrm{sign}(k)}, \quad k = -n/2{+}1, \ldots, n/2.$$

We now apply this model to a test image that is 256×256. The noise is Gaussian, with standard deviation of 11%. The result is shown in Figure 6. We observe that the noise is removed over a very short interval in time, without any visible artifacts, blurring or loss of contrast.

Figure 6: Denoising of L.A. aerial photo with $\epsilon = 0.1$, $c = 0.5$ and $T = 1.5 \times 10^{-4}$

Block KSS methods also effective as a time-stepping scheme for reaction-diffusion equations that are used to

achieve sharpening of images via deconvolution, such as

$$u_t = \nabla \cdot (g(u)\nabla u)) - \alpha \tilde{h} * (h * u - f), \qquad (49)$$

where h is a blurring kernel and $\tilde{h}$ is the mirror kernel $\tilde{h}(x, y) \equiv h(-x, -y)$. This equation is a modification of an equation introduced in [31], with the diffusion coefficient defined in (47). A 1-node block KSS method is applied to the convolution of a flower-shaped characteristic function,with a Gaussian kernel, resulting in a blurred image.The original and blurred images, as well as the result of the deconvolution, are shown in Figure 7.

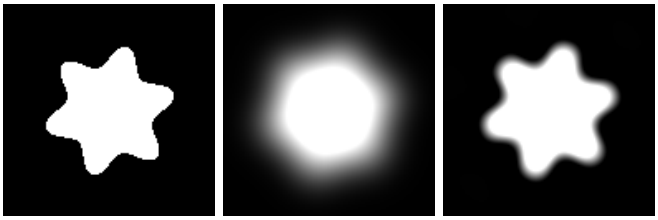

Figure 7: Left plot: Original image defined via flower-shaped characteristic function. Center plot: Blurred flower image obtained by convolution with kernel $h(x, y) = e^{-100(x^2+y^2)}$. Right plot: Deblurred flower image obtained by solving (49) to time $T = 0.3$ with $\epsilon = 0.1$, $c = 0.01$, $\Delta t = 0.0002$, and $\alpha = 10^4$.

It should be noted that the use of iterative methods such as MINRES necessitates that larger images are handled by decomposing them into blocks of a manageable size, such as 128×128, and denoised independently of one another. When using periodic boundary conditions, a "padding" border must be added around each block, reflecting the image across its boundary as needed, in order to prevent artifacts from appearing at the interfaces between blocks or on the boundary of the entire image. Unfortunately, when integrating over longer periods of time, mismatches at the interfaces between blocks can still appear. The block KSS method, on the other hand, is capable of efficiently denoising larger images without such decomposition, thus circumventing these implementation difficulties.

The approach used to model this kind of fractional diffusion equation can readily be applied to other equations involving fractional spatial differentiation, such as in [4].

9 Summary and Future Work

We have demonstrated that block KSS methods can be applied to Maxwell's equations with smoothly varying coefficients, by appropriate generalization of their application to the scalar second-order wave equation, in a way that preserves the order of accuracy achieved for the wave equation. Furthermore, it has been demonstrated that while traditional Krylov subspace methods based on exponential integrators are most effective for parabolic problems, especially when aided by preconditioning as in [30], KSS methods perform best when applied to hyperbolic problems, in view of their much higher order of accuracy.

Future work will extend the approach described in this paper to more realistic applications involving Maxwell's equations, and related models such as the Vlasov-Maxwell-Fokker-Planck system [7], by using symbol modification to efficiently implement perfectly matched layers (see [3]) for simulation on infinite domains, and various techniques (see [6, 29]) to effectively handle discontinuous coefficients. In addition, block KSS methods will be adapted to work with bases of Chebyshev polynomials for problems with non-homogeneous boundary conditions, including nonlocal boundary conditions as in [27].

References

[1] Alvarez, L., Lions, P.-L., Morel, J.-M.: Image selective smoothing and edge-detection by non-linear diffusion. II. *SIAM J. Numer. Anal.* **29**(3) (1991) 845-866.

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

[3] Berenger, J,: A perfectly matched layer for the absorption of electromagnetic waves. *J. Comp. Phys.* **114** (1994) 185-200.

[4] Blackledge, J. M.: Application of the Fractional Diffusion Equation for Predicting Market Behaviour. *IAENG Journal of Applied Mathematics* **40**(3) (2010) 23-51.

[5] Catté, F., Lions, P.-L., Morel, J.-M., Coll, T.: Image selective smoothing and edge-detection by nonlinear diffusion. *SIAM J. Numer. Anal.* **29**(1) (1992) 182-193.

[6] Gelb, A., Tanner, J.: Robust Reprojection Methods for the Resolution of the Gibbs Phenomenon. *Appl. Comput. Harmon. Anal.* **20** (2006) 3-25.

[7] El Ghani, N.: Diffusion Limit for the Vlasov-Maxwell-Fokker-Planck System. *IAENG Journal of Applied Mathematics* **40**(3) (2010) 52-59.

[8] 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)

[9] Golub, G. H., Gutknecht, M. H.: Modified Moments for Indefinite Weight Functions. *Numerische Mathematik* **57** (1989) 607-624.

[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] Guidotti, P., Lambers, J. V.: Two New Nonlinear Nonlocal Diffusions for Noise Reduction. *J. of Math. Imaging Vis.* **33** (2009) 27-35.

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

[14] Hochbruck, M., Lubich, C.: On Krylov Subspace Approximations to the Matrix Exponential Operator. *SIAM Journal of Numerical Analysis* **34** (1997) 1911-1925.

[15] Hochbruck, M., Lubich, C., Selhofer, H.: Exponential Integrators for Large Systems of Differential Equations. *SIAM Journal of Scientific Computing* **19** (1998) 1552-1574.

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

[17] Lambers, J. V.: Enhancement of Krylov Subspace Spectral Methods by Block Lanczos Iteration. *Electronic Transactions on Numerical Analysis* **31** (2008) 86-109.

[18] 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**(4) (2008) 333-348.

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

[20] Lambers, J. V.: Krylov Subspace Spectral Methods for Variable-Coefficient Initial-Boundary Value Problems. *Electronic Transactions on Numerical Analysis* **20** (2005) 212-234.

[21] Lambers, J. V.: A Multigrid Block Krylov Subspace Method for Variable-Coefficient Elliptic PDE. *IAENG Journal of Applied Mathematics* **39**(4) (2009) 236-246.

[22] Lambers, J. V.: Practical Implementation of Krylov Subspace Spectral Methods. *Journal of Scientific Computing* **32** (2007) 449-476.

[23] Lambers, J. V.: A Spectral Time-Domain Method for Computational Electrodynamics. *Adavances in Applied Mathematics and Mechanics* **1** (2009) 781-798.

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

[25] Paige, C. C., Saunders, M. A.: Solution of Sparse Indefinite Systems of Linear Equations. *SIAM J. Numer. Anal.* **12** (1975) 617-629.

[26] Perona, P., Malik, J.: Scale-space and edge detection using anisotropic diffusion. *IEEE Pattern Anal. Mach. Intell.* **12** (1990) 161-192.

[27] Siddique, M.: Numerical Computation of Two-dimensional Diffusion Equation with Nonlocal Boundary Conditions. *IAENG Journal of Applied Mathematics* **40**(1) (2010) 26-31.

[28] Tewari, S. G., Pardasani, K. R.: Finite Element Model to Study Two Dimensional Unsteady State Cytosolic Calcium Diffusion in Presence of Excess Buffers. *IAENG Journal of Applied Mathematics* **40**(3) (2010) 1-5.

[29] Vallius, T., Honkanen, M.: Reformulation of the Fourier nodal method with adaptive spatial resolution: application to multilevel profiles. *Opt. Expr.* **10**(1) (2002) 24-34.

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

[31] Welk, M., Theis, D., Brox, T., Weickert, J.: PDE-Based Deconvolution with Forward-Backward Diffusivities and Diffusion Tensors. *In Scale Space 2005*, Kimmel, R., Sochen, N., Weickert, J. (eds.) Springer (2005) 585-597.