Modeling particle diffusion in laminar tube flow with spectral collocation

C. M. Thibeault[1,2,3,*], F. C. Harris Jr.[3], and P. A. Tebbe[1]

[1]*Department of Civil and Mechanical Engineering, Minnesota State University, Mankato. Mankato MN.*
[2]*Department of Electrical and Biomedical Engineering, University of Nevada, Reno. Reno, NV.*
[3]*Department of Computer Science and Engineering, University of Nevada, Reno. Reno, NV.*

Abstract

The spectral collocation method is a numerical approximation technique that seeks the solution of a differential equation using a finite series of infinitely differentiable basis functions. This inherently global technique enjoys an exponential rate of convergence and has proven to be extremely effective in computational fluid dynamics. Despite the initial complexity of understanding spectral collocation, the use of the method is relatively straight forward. Here we present a complete example of applying this method to modeling particle diffusion in laminar tube flow. The included code, written for Octave, highlights the reduction of a partial differential equation into a matrix interpolation using spectral collocation. The results are compared to an analytical solution for accuracy and a finite difference method for performance.

Key Words: *Spectral Collocation, Computational Fluid Dynamics, Laminar Tube Flow, Particle Diffusion*

1 Introduction

1.1 Spectral collocation

The spectral method is a numerical modeling technique for approximating the solution of partial and ordinary differential equations. Related to the method of weighted residuals, spectral methods employ infinitely differentiable functions as trial functions. The result is a global method with an exponential rate of convergence for problems with smooth solutions. Since the discovery of fast Fourier transforms the use and practicality of spectral methods has steadily increased. There has been broad success in several areas including Weather, Turbulence, Seismic and Quantum Modeling [1–5].

There are several features that set spectral collocation methods apart from other numerical solutions to partial differential equations. The first is the computational domain. In Finite Element (FE) modeling for example, the overall physical domain is broken up into a number of sub-domains (elements) and a local basis function is chosen to be nonzero over a small number of sub-intervals within that domain. Conversely, spectral methods chose a basis function that is global to the entire computational domain and is nonzero except at isolated points. It is this reason that spectral methods are often referred to as a global numerical method.

The second distinguishing feature of spectral methods is the choice of basis functions. Spectral methods select basis functions that are high degree polynomials or trigonometric polynomials that are infinitely differentiable. The basis functions for FE methods are generally low-order polynomials and as stated before, local in nature. Because of this, FE methods are better suited for complex geometries but suffer from lower accuracy as compared to spectral methods.

1.2 Particle diffusion in laminar tube flow

When aerosol particulates are exposed to a concentration gradient they diffuse from from high to low concentrations by Brownian motion. Similarly, when those particles are dispersed in a fluid that is traveling down a tube or between parallel plates, the walls of the vessel will act like a sink to those particles—by the same Brownian motion. For most models of this process the concentration at the wall is taken to be zero and the particles will not only diffuse radially to the wall but will also deposit there. This diffusion process is important in a number of applications in both desired—such as air cleaning, hot-gas filtration, optical fiber manufacturing, and thin film production—and undesired—pipe fouling, micro-contamination and corrosion—applications [6, 7].

The modeling of aerosol deposition in a tube is a well characterized problem and there have been several analytical solutions for the deposition efficiency. The first was developed by Gormley and Kennedy in 1949 [8, 9]. This is the benchmark solution referred to by most of the subsequent studies. C. N. Davies presented an analytical solution that differed from the Gormley results, however, Ingham[10]] later showed that there was a mistake in that evaluation. In addition, a solution for small values of the diffusion coefficient was developed in that. These analytical treatments make this problem ideal for demonstrating the accuracy of

*Corresponding Author. cmthibeault@hrl.com

numerical methods and the application of spectral collocation to this problem is unique to this paper.

1.3 Navier-Stokes equations

The motion of fluid in space is described by the Navier-Stokes equations. Emerging from Newton's second law, the Navier-Stokes equations relate the gradients of the dependent variables to form a system of nonlinear partial differential equations. For Newtonian fluids these are
conservation of momentum

$$\rho\left(\frac{\partial \vec{u}}{\partial t}+\vec{u}\cdot\bigtriangledown\vec{u}\right)=-\bigtriangledown P+\nu\bigtriangledown^2\vec{u}+f, \tag{1}$$

conservation of energy

$$\begin{aligned}\frac{\partial}{\partial t}\left(\frac{1}{2}\rho u^2+\rho\hat{U}\right)=&-\left(\bigtriangledown\cdot\left(\frac{1}{2}\rho u^2+\rho\hat{U}\right)\vec{u}\right)\\&-\left(\bigtriangledown\cdot\vec{q}\right)-\left(\bigtriangledown\cdot P\vec{u}\right)\\&-\left(\bigtriangledown\cdot\left[\vec{\tau}\cdot\vec{u}\right]\right)\\&+\rho\left(\vec{u}\cdot\vec{g}\right),\end{aligned} \tag{2}$$

and conservation of mass species

$$\frac{\partial\rho_\alpha}{\partial t}=-\left(\bigtriangledown\cdot\rho_\alpha\vec{u}\right)-\left(\bigtriangledown\cdot\vec{j}_\alpha\right)+r. \tag{3}$$

Where ρ is the fluid density ($\frac{kg}{m^3}$), $\vec{u}$ is the flow velocity ($\frac{m}{s}$), t is time (s), P is the pressure (Pa), ν is the kinematic viscosity (m^2/s), f represent other body forces (i.e. gravity), $\hat{U}$ is the internal energy ($\frac{J}{kg}$), $\vec{q}$ is the heat flux ($\frac{J}{sm^2}$), $\vec{\tau}$ is the surface stress (Pa), $\vec{g}$ is the gravitational force ($N\frac{m}{kg}^2$), ρ_α is the concentration of species α (M), and $\vec{j}_\alpha$ is the flux of species α ($\frac{M}{sm^2}$).

Finally, in order to fully describe the fluid motion the continuity equation describing the conservation of mass is required,

$$\frac{\partial\rho}{\partial t}+\bigtriangledown\cdot\bar{u}=0. \tag{4}$$

For the simple diffusion problem problem it is assumed that there is no formulation of aerosols and the mass transfer along the direction of flow due to diffusion can be neglected. These are both valid as long as the Peclet number is significantly greater than 1 — meaning the movement due to advection is significantly higher than the movement due to diffusion. With these assumptions Equation 3 becomes

$$\begin{aligned}\frac{1}{r}\frac{\partial}{\partial r}\left[\left(r\rho\left(u_r+V_{Tr}\right)n\right)\right.&+\left.\frac{\partial}{\partial z}\left[\rho\left(u_z+V_{Tz}\right)n\right]\right.\\&=\frac{1}{r}\frac{\partial}{\partial r}\left(r\rho D\frac{\partial n}{\partial r}\right)+\frac{\partial}{\partial z}\left(\rho D\frac{\partial n}{\partial z}\right).\end{aligned}$$

Where r is the distance along the tube radius (m), V_{Tr} is the thermophoretic velocity along r ($\frac{m}{s}$), V_{Tz} is the thermophoretic velocity along the tube length z ($\frac{m}{s}$), n is the dimensionless particle concentration, and D is the particle diffusion coefficient ($\frac{m^2}{s}$).

With the assumption of laminar tube flow a parabolic velocity profile — $v_z(r)=V_{max}\left(1-(r/R)^2\right)$, where R is the overall tube radius and V_{max} is the velocity of the fluid along the centerline—which can be substituted directly into Equation 3. This satisfies Equations 1 and 4. Additionally, the flow is irrotational and the viscosity of the fluid can be neglected along with particle slip at the boundary. Axial diffusion ($\frac{\partial}{\partial z}\left(\rho D\frac{\partial n}{\partial z}\right)$), thermophoresis V_{Tz} and V_{Tr}, as well as velocity in r, u_r, can also be neglected. Finally, the diffusivity is assumed to be independent of temperature. The equation describing the movement of particles in the fluid can then be reduced to

$$v_{z,max}\left(1-\left(\frac{r}{R}\right)^2\right)\frac{\partial n}{\partial z}=D\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial n}{\partial r}\right). \tag{5}$$

1.3.1 Analytical Solution

The analytical solution for Equation 5 was first developed by Gormley and Kennedy and slightly reformed by Hinds [8]. This was found to closely approximate experimental results and is used here as a way to estimate the accuracy of the spectral method. The solution is defined as

$$P(\mu)=\begin{cases}1-5.5\cdot\mu^{(2/3)}+3.77\cdot\mu & \mu<0.009\\ -0.819\cdot e^{(-11.5\mu)} & \mu\geq 0.009\\ \quad +0.0975\cdot e^{(-70.1\mu)} & \end{cases} \tag{6}$$

Where μ is the dimensionless diffusion coefficient defined as $\frac{D\cdot L}{Q}$, $Q=\frac{V_{max}\pi R^2}{2}$, D is the tube diameter (m), and L is the tube length (M).

2 Methods

2.1 Numerical methods

To generalize, the spectral collocation, or pseudospectral, method interpolates all of the discrete data and then approximates the derivative of the interpolant along that data globally [11]. In this case, the global space is a discrete grid of Gauss-Lobatto-Chebyshev points and interpolants are Chebyshev polynomials. The domain of the points lie along the domain $[-1,1]$ where Chebyshev polynomials remain analytic and convergence of the spectral series can be ensured.

Spectral collocation is based on derivative matrices described by [4, 11, 12]. The differentiation matrices specific to spectral collocation develop from polynomial interpolation and can loosely be defined in two steps [12].

Listing 1: Octave code for generating a spectral differentiation matrix.

```
function [D,x] = cheb_derivative(N)
 if N==0, D=0;
     x=1;
 return, end
 % Collocation Points
 x = cos(pi*(0:N)/N)';
 X = repmat(x,1,N+1);
 dX = X-X';
 % Derivative Matrix
 D=zeros(N+1,N+1);
 % Top and Bottom Rows
 top = (2*(-1).^(1:N-1))./(1-reshape(x(2:N),1,N-1));
 bottom=-top(N-1:-1:1);
 % Left and Right
 right = -reshape(bottom./4,(N-1),1);
 left= -right(N-1:-1:1);
 %top
 D(1,2:N) = top;
 D(N+1,2:N) = bottom;
 D(2:N,1)=left;
 D(2:N,N+1) =right;
 % Middle
 ij = repmat([0:N],N+1,1)+
      repmat(reshape([0:N],N+1,1),1,N+1);
 mid = ((-1).^ij)./dX;
 D(2:N,2:N) = mid(2:N,2:N);
 % Diagonal
 dia=(1:N+1:(N+1)*(N+1)) + (0:N);
 dia = dia(2:N);
 D(dia) = -(x(2:N)./(2*(1-x(2:N).^2)));
 % Corners
 D(1) = (2*N^2+1)/6;
 D((N+1)*(N+1))=-D(1);
 D(N+1) = -0.5*(-1)^N;
 D((N+1)*(N+1)-(N)) = -D(N+1);
```

1. Given a vector v, interpolate it by a polynomial q that can be defined as

$$q(x) = q_N(x). \tag{7}$$

2. At the points x_j the derivative of the interpolant is found by

$$\frac{dq}{dx}_j = (D_N v)_j = w_j. \tag{8}$$

The differentiation matrix, D_N, approximates the derivative of a vector with points at the Chebyshev extreme points. Higher-order derivatives can be found by raising the first-order differentiation matrix by an exponent corresponding to the order (i.e. squaring the matrix for the second derivative). Finally, two-dimensional differentiation matrices can be constructed by taking the Kronecker product of two one-dimensional differentiation matrices [11]. Figure 1 illustrates how these points are constructed for an $(N-1)\times(N-1)$ grid. Constructing this matrix can be condensed using vectorized functions available in MATLAB and Octave, however, expanding this formulation better illustrates the matrix construction and improves performance. This is presented in Listing 1.

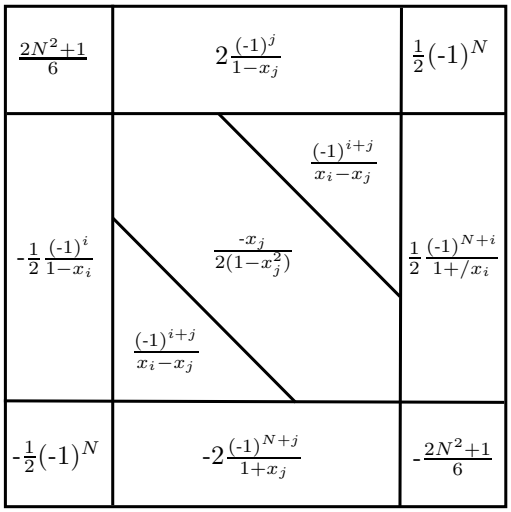

Figure 1: Chebyshev differentiation matrix [11]. The $(N-1)\times(N-1)$ matrix is indexed by j for columns and i for rows.

2.2 Discretization

Before discretizing the governing differential equations, the coordinates of the solution domain must be projected onto the computational domain of $[-1,1]$. In the axial direction z is transformed by

$$z = \frac{L}{2}\eta + \frac{L}{2}. \tag{9}$$

Solving for η and taking the first derivative results in $\partial\eta/\partial z = 2/L$. The first partial derivative with respect to z is then

$$\frac{\partial}{\partial z} = \frac{\partial}{\partial \eta}\frac{\partial \eta}{\partial z} = \frac{\partial}{\partial \eta}\frac{2}{L}. \tag{10}$$

Likewise in radial direction, $r = R\xi$, we can solve for ξ and $\partial\xi/\partial r$, which will result in the first partial derivative

$$\frac{\partial}{\partial r} = \frac{\partial}{\partial \xi}\frac{\partial \xi}{\partial r} = \frac{\partial}{\partial \xi}\frac{1}{R}. \tag{11}$$

The second partial derivative is then

$$\frac{\partial^2}{\partial r^2} = \frac{\partial^2}{\partial \xi^2}\frac{\partial \xi^2}{\partial r^2} = \frac{\partial^2}{\partial \xi^2}\frac{1}{R^2}. \tag{12}$$

Substituting these coordinate transformations in Equation 5 results in

$$\begin{aligned}&\frac{2}{L}\frac{\partial n}{\partial \eta}\left(V_{max}\left(1-\xi^2\right)\right) = \\ &D\left(\frac{1}{R^2\xi}\frac{\partial n}{\partial \xi} + \frac{1}{R^2}\frac{\partial^2 n}{\partial \xi^2}\right).\end{aligned} \tag{13}$$

The discretization follows as:

$$\left[\left(\mathbf{A} \otimes \mathbf{D}_{\eta}^{\mathbf{1}}\right) - \left(\mathbf{\Omega} \cdot \frac{1}{R^2} \cdot \mathbf{D}_{\xi}^{\mathbf{1}} \otimes \mathbf{I}_{\mathbf{M}_{\xi}}\right) D + \left(\frac{1}{R^2} \cdot \mathbf{D}_{\xi}^{\mathbf{2}} \otimes \mathbf{I}_{\mathbf{M}_{\eta}}\right) D \right] \{\Theta\} = \{f\}.$$

Where the matrices are defined as

$$\mathbf{A} = diag\left(\frac{2}{R} \cdot V_{max} \cdot \left(1 - \xi^2\right)\right) \tag{14}$$

and

$$\mathbf{\Omega} = diag\left(\frac{1}{\xi}\right). \tag{15}$$

Finally, to make a comparison with the analytical solution, the particle concentration at a given cross-section along the axis of the tube can be found using

$$P = \frac{\int_0^R n\left(r, Z\right)\left(r - \frac{r^3}{R^3}\right) dr}{\frac{n_{in} R^2}{4}}. \tag{16}$$

2.3 Code development

Once the differentiation matrices are generated —Listing 1— their application follows from basic linear algebra. The solution for this problem, including initialization, boundary-condition enforcement, solving, and post-condition, is presented in Listing 2—notice that the bulk of this code is taken up by comments. The use of built-in MATLAB/Octave functions eases the programming burden however, converting this code to a more efficient language would be reasonable. More importantly, would be generalizing this technique into a library for solving generic partial differential equations.

3 Results

The output from Listing 2 is presented in Figure 2 for different diffusion coefficients. The affect the diffusion coefficient has is clear from these plots. For this a grid of $41x41$ was used and plotted directly. However, a cubic spline interpolation can be used to further improve the estimation and create smoother solutions. This is also one strategy for improving the interpretation of results from smaller grid sizes.

The spectral solution for the simple diffusion case was validated against the analytical solution—Figure 3. The right axis of Figure 3 is the penetration along the dimensionless value μ as the grid size is varied. This was calculated using Equation 16. The difference between the two methods is illustrated in Figure 3 (bottom). As can be seen the

Listing 2: Solving for particle deposition using spectral collocation. This computes the particle number concentration of fluid flow in a circular pipe. It assumes a constant particle diffusion coefficient D and does not account for thermophoresis. The input variables are D, particle diffusion coefficient (m^2/s), V_{max}, maximum velocity along the tube axis (m/s), R, radius of the tube, (m), Len, length of the tube (m),N,number of points in the radial direction (Must be odd), and M, number of points in the axial direction (Must be odd).

```
function [u,uu,r,z] = spec_tube_diff(D,Vmax,R,Len,N,M)
% ------------------------------------------------------
    Boundary_in = 1;
    Boundary_Side = 0;
% ------------------------------------------------------
%   Create the Chebyshev differentiation Matrices
%   * Note that in the accompanying literature r and z
%     are Xi and Eta respectively.
    [Dr,r] = cheb_derivative(N);      % Radial Direction
    [Dz,z] = cheb_derivative(M);      % Axial Direction
%   Compute the second order Chebyshev differentiation
%   matrix directly
    Dr2 = Dr^2;
% ------------------------------------------------------
%   Compute the coefficients and resulting diagonal
%   matrices
% ------------------------------------------------------
%   Calculate the first coefficient of the energy
%   equation as a column matrix.
    a = ((2/Len)*Vmax)*(1 - (r.^2));
    A = diag(a(1:N+1));
%   Create identity matrix.
    I = eye(M+1);
%   Compute Omega.
    O = (1/(R^2)).*diag(1./r);
% ------------------------------------------------------
%   Compute the tensor product spectral grid.
% ------------------------------------------------------
    L = kron(A,Dz) -
        kron((1/(R^2)).*Dr2,I).*D -
        kron(O*Dr,I).*D;
% ------------------------------------------------------
%   Create a column matrix to hold the RHS of the energy
%   equation
% ------------------------------------------------------
    mSize = ((N+1)*(M+1));
    rhs = zeros(mSize,1);
%   Create two matrices containing the coordinate points
%   inside the solution plane.
    [rr,zz] = meshgrid(r,z);
%   Create one column vector that consists of the two
%   coordinate vectors stacked on top of each other.
    rr = rr(:);
    zz = zz(:);
% ------------------------------------------------------
%   Apply the Boundary Conditions
% ------------------------------------------------------
%   Find the bounday points of the coordinate vectors:
    b = find(zz==-1 | abs(rr)==1);
%   Impose the boundary conditions on the rhs vector.
    rhs(b) = (zz(b)==-1).*Boundary_in + ...
             (rr(b)==1).*Boundary_Side + ...
             (rr(b)==-1).*Boundary_Side;
%   Correct the entry boundary conditions at the top and
%   bottom corners.
    rhs(1) = Boundary_Side;
    rhs(size - M) = Boundary_Side;
%   Apply the boundary conditions to the laplacian
    L(b,:) = zeros(length(b),length(rhs));% Zero the rows
    L(b,b) = eye(length(b));  % put in 1 for known points
% ------------------------------------------------------
%   Solve the Matrix equation
% ------------------------------------------------------
    u = L\rhs;
    uu = reshape(u,(M+1),(N+1));
% ------------------------------------------------------
```

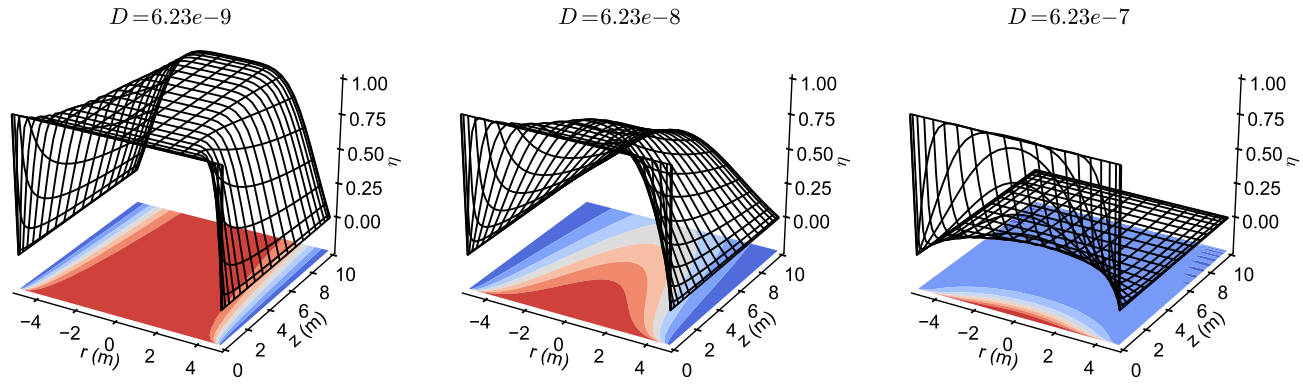

Figure 2: Particle diffusion along an identical tube for different diffusion coefficients. There are a number of parameters that affect the efficiency of the particle deposition. In this case as the diffusion coefficient is increased for an identical pipe, a larger number of particles will deposit on the pipe walls. For this a square grid of size $41x41$ was used.

spectral solution is in excellent agreement with the analytical solution.

To illustrate the both the accuracy and the efficiency of the spectral method a simple finite difference solver was developed. This solved the dimensionless form of the governing equation presented in Equation 5 on a rectangular grid with equally spaced points using a Jacobi method solver. Although this is not the most efficient finite difference solver it is one that closely resembles the programming complexity of the spectral method. Figure 4 presents the results of this comparison. The speedup —finite difference time divided by the spectral method time—decreases as the grid size is increased. However, the spectral method stays above 50 times faster than the finite difference method. For this comparison the timings were recorded using the built-in MATLAB/Octave timers and the average of 3 runs is presented.

Although the speedup decreases with grid size, the accuracy when compared to the analytical method continues to improve for the spectral method—Figure 4 (bottom). The finite difference accuracy also improves with the grid size, however it is not as accurate as the spectral method.

4 Discussion

When applied to a classic fluid dynamics problems the spectral collocation method is in excellent agreement with previously published results. Its computational efficiency coupled with its high accuracy make it ideally suited to this class of problems. Although this is a well classified example, this work contributes a method for approximating their solutions that is more accurate and computationally tractable.

Despite only presenting a specific example, this paper lays the foundation for extending this work to a generic partial differential equation library. In addition, migrating to a more efficient programming language is also something that will be done in the future. Finally, although a steady-state problem was presented here, the transition into solving a dynamic version of this equation would simply involve using spectral collocation in space and a discrete (i.e. a leap frog method) in time.

References

[1] B. Fornberg, *A Practical Guide to Pseudospectral Methods*. New York, NY, USA: Cambridge University Press, 1996.

[2] C. Canuto, M. Hussaini, A. Quarteroni, and T. Zang, *Spectral Methods in Fluid Dynamics*. New York, NY, USA: SpringerVerlag, 1987.

[3] C. Canuto, M. Hussaini, A. Quarteroni, and T. Zang, *Spectral Methods Evolution to Complex Geometries and Applications to fluid Dynamics*. New York: Springer, 2007.

[4] J. P. Boyd, *Chebyshev and Fourier Spectral Methods Second Edition*. Mineola, NY: Dover Publications, 2001.

[5] G. BenYu, *Spectral Methods and Their Applications*. New Jersey, USA: World Scientific, 1998.

[6] F. Stratmann and H. Fissan, "Nondimensional investigation of submicron particle transport on cooled laminar tube flow," *J. Aerosol Sci.*, vol. 22, Suppl. 1, pp. S211–S214, 1991.

[7] X. Luo and S. Yu, "Deposition of aerosol in a laminar pipe flow," *Science in China Series E: Technological Sciences*, vol. 51, no. 8, pp. 1242–1254, 2008.

[8] W. Hinds, *Aerosol Technology, 2nd Ed.* New York: John Wiley & Sons, 1999.

[9] P. Baron and K. Willeke, *Aerosol Measurement Principles, Techniques, and Applications*. New York: John Wiley & Sons, 2001.

[10] D. Ingham, "Diffusion of aerosols from a stream flowing through a cylindrical tube," *Journal of Aerosol Science*, vol. 6, no. 2, pp. 125 – 132, 1975.

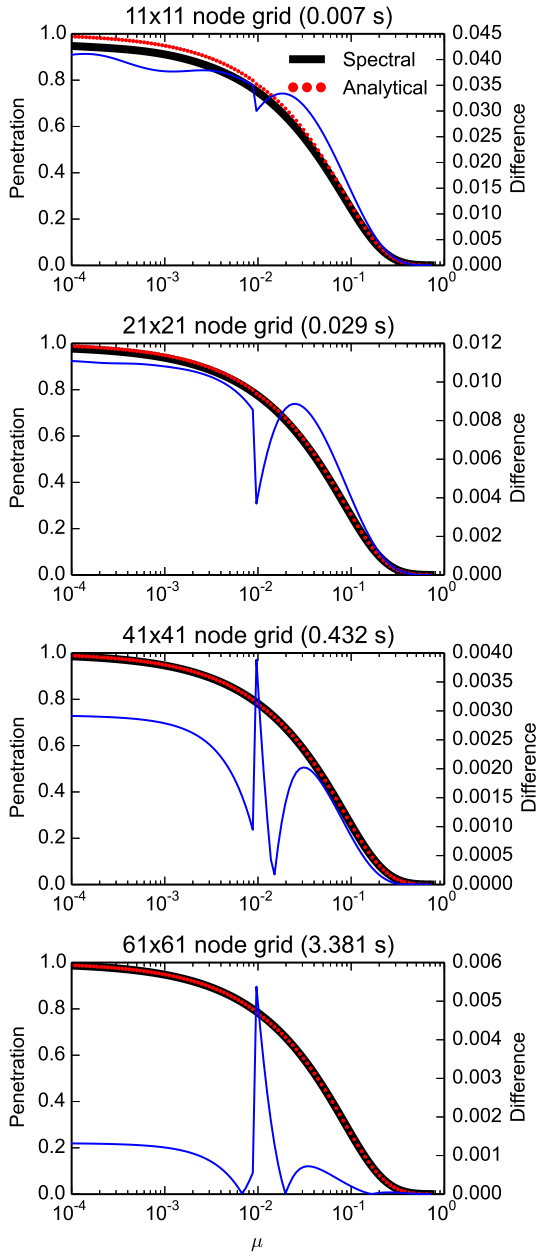

Figure 3: Accuracy of spectral method compared to analytical solution. As the grid size is increased the accuracy of the spectral method increases as compared to the analytical solution. The left hand axis is the penetration as calculated by Equation 16 as the dimensionless diffusion coefficient μ is changed. The right hand axis is the difference between the two methods at each data point (blue line).

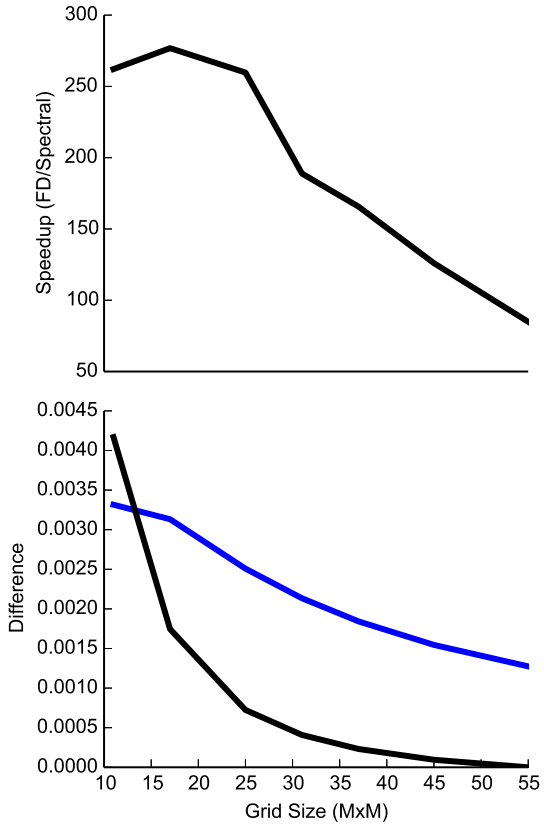

Figure 4: Spectral collocation compared to a naive finite difference method. Speedup between the methods (top). As the grid size is changed equally for both methods the speedup for the spectral method decreases. Accuracy compared to the analytical solution (bottom). Initially the finite difference method (blue line) is slightly more accurate. However, as the grid size is increased the spectral method's (black line) accuracy continues to improve—whereas the finite difference method does not improve at as fast a rate.

[11] L. N. Trefethen, *Spectral Methods in Matlab*. Philadelphia, PA.: SIAM, 2000.

[12] L. N. Trefethen, *Finite Difference and Spectral Methods for Ordinary and Partial Differential Equations*. available at http://www.comlab.ox.ac.uk/nick.trefethen/pdetext.html: unpublished text, 1996.