The Spectral Signal Processing Suite

SCOTT A. SARRA
Marshall University

A software suite written in the Java programming language for the postprocessing of Chebyshev approximations to discontinuous functions is presented. It is demonstrated how to use the package to remove the effects of the Gibbs-Wilbraham phenomenon from Chebyshev approximations of discontinuous functions. Additionally, the package is used to postprocess Chebyshev collocation and Chebyshev super spectral viscosity approximations of hyperbolic partial differential equations. The postprocessing method is the Gegenbauer reconstruction procedure. The Spectral Signal Processing Suite is the first publicly available package that implements the procedure. State-of-the-art techniques are used to implement the algorithms with efficiency while reducing round-off error.

Categories and Subject Descriptors: G.1 [**Numerical Analysis**]; G.1.2 [**Numerical Analysis**]: Approximation; G.1.8 [**Numerical Analysis**]: Partial Differential Equations—*Spectral methods*

General Terms: Algorithms

Additional Key Words and Phrases: Gegenbauer postprocessing, Gibbs-Wilbraham phenomenon, Java, edge detection, chebyshev

1. INTRODUCTION

Spectral approximations based on Chebyshev polynomials are exponentially accurate for analytic functions. However, for discontinuous but piecewise analytic functions, the spectral partial sum approximates the function poorly throughout the domain. Away from the discontinuities, only first-order accuracy is achieved. Near the discontinuity there are $O(1)$ oscillations which do not decrease with N, the number of terms retained in the spectral partial sum. This is known as the *Gibbs-Wilbraham phenomenon*. The problem is reduced to one of signal processing in order to recover spectral accuracy.

Several methods exist for postprocessing spectral approximations. One class of postprocessing methods consists of variations of the spectral mollification (SM) idea which was originally developed in Gottlieb and Tadmor [1985]. Spectral mollification involves applying a two-parameter family of filters. The method can recover spectral accuracy up to within a neighborhood of each discontinuity. The Gibbs phenomenon can be removed, but some smearing at the discontinuity locations will occur. This idea is discussed in more detail in Kaber and Mahmoud [1994] and examples of using the method to postprocess

Author's address: Department of Mathematics, Marshall University, One John Marshall Drive, Huntington, WV, 25755-2560; email: scott@scottSarra.org.
Permission to make digital/hard copy of part or all of this work for personal or classroom use is granted without fee provided that the copies are not made or distributed for profit or commercial advantage, the copyright notice, the title of the publication, and its date appear, and notice is given that copying is by permission of ACM, Inc. To copy otherwise, to republish, to post on servers, or to redistribute to lists requires prior specific permission and/or a fee.
© 2003 ACM 0098-3500/03/0600-0195 $5.00

partial differential equation (PDE) solutions are contained in Kaber [1996]. The method of Gottlieb and Tadmor [1985] can be improved upon if the locations of the edges are known [Gelb 2000; Tadmor and Tanner 2002]. This allows one of the two parameters to be optimized, which leads to increased accuracy away from the discontinuities and less smearing at the discontinuities. Further optimization of the method is considered in Tadmor and Tanner [2002]. While spectral mollification may be applied with or without the knowledge of the location of edges (discontinuities) in the function, better results are achieved when the locations of the edges are known.

Another way to postprocess spectral approximations is the Padé-based algorithm for removing the Gibbs phenomenon from Fourier approximations [Driscoll and Fornberg 2001]. The algorithm is stated in the context of Fourier approximations, but could possibly be extended to nonperiodic Chebyshev approximations.

The postprocessing method that is the focus of the current version of the *Spectral Signal Processing Suite* (SSPS) is the Gegenbauer Reconstruction Procedure (GRP) [Gottlieb et al. 1992; Gottlieb and Shu 1995a, 1995b, 1996, 1997]. The GRP is capable of recovering spectral accuracy at every point, even at the locations of the discontinuities. However, despite showing more potential than spectral mollification, the GRP is not as robust as the SM postprocessing methods due to two unoptimized parameters used by the method. The GRP will need to know the location of edges in the function. The purpose of this paper and software package is to describe and implement the state-of-the-art Gegenbauer Reconstruction Procedure and edge detection algorithms. Prior to release 1.0 of the Spectral Signal Processing Suite, publicly available software that implemented the algorithms was not available.

The GRP may also be used to recover spectral accuracy from approximations of PDE solutions arising from Chebyshev pseudospectral methods [Gottlieb and Shu 1995b]. In the context of postprocessing PDE solutions, the term *edges* refers to discontinuities and shocks. If the spectral approximation is to a nonlinear hyperbolic conservation law, spectral viscosity will need to be added to the approximation in order to obtain a stable approximation which converges to the exact entropy solution [Tadmor 1989]. While the spectral viscosity solution is a highly accurate approximation to the collocation solution, only partial theoretical justification can be found for using the postprocessing method on the spectral viscosity solution. Nevertheless, numerical results indicate that exponential accuracy can be achieved by applying the Gegenbauer postprocessing procedure to the spectral viscosity solution [Gelb and Tadmor 2000b]. The same can be said about the edge detection method. The theoretical results are limited to locating the jump discontinuities of a piecewise smooth function, but numerical evidence advocates applying the edge detection procedure to the spectral viscosity solution. In the examples, we have used the super spectral viscosity method (SSV) of Ma [1998].

This paper is organized a follows. Section 2 reviews approximation by a Chebyshev partial sum. Section 3 summarizes a method developed in Gelb and Tadmor [2000a] to locate edges in a function. Section 4 summarizes the GRP postprocessing method. Section 5 describes the software suite. Section 6

presents examples of using the software suite to reconstruct piecewise analytic functions and examples of using the methods to postprocess the numerical solution of PDEs by Chebyshev pseudospectral methods. The numerical examples will emphasize both a local approach and a global approach for selecting the reconstruction parameters. Section 7 contains concluding remarks.

2. CHEBYSHEV APPROXIMATIONS

By a Chebyshev approximation of a function, we mean the Chebyshev partial sum

$$u_N(x) = \sum_{n=0}^{N} a_n T_n(x). \tag{1}$$

The discrete Chebyshev coefficients, a_n, are defined by

$$a_n = \frac{2}{N}\frac{1}{c_n}\sum_{n=0}^{N}\frac{u(x_j)T_n(x_j)}{c_j}, \quad where \quad c_j = \begin{cases} 2 & \text{when } j = 0, N \\ 1 & \text{otherwise.} \end{cases}, \tag{2}$$

The Chebyshev polynomials are known in closed form as

$$T_k(x) = \cos(k \arccos(x)). \tag{3}$$

The Chebyshev pseudospectral method is based on assuming that an unknown PDE solution, u, can be represented by a global, interpolating, Chebyshev partial sum of the form (1). More detailed information on pseudospectral methods may be found in the standard references [Canuto et al. 1988; Fornberg 1996; Funaro 1992; Gottlieb and Orszag 1977; Gottlieb et al. 1984; Trefethen 2000].

If the PDE solution contains shocks, the Chebyshev pseudospectral method will not converge to the correct entropy solution [Tadmor 1989]. In this case, a spectrally small viscosity term must be added in order to stabilize the approximation and ensure convergence to the entropy solution. This can be done without sacrificing spectral accuracy and can be accomplished in several different ways, with each way being labeled a particular type of spectral viscosity method. In the examples, we have used the SSV method of Ma [1998]. The reader is referred to Sarra [2003a] for details on the implementation of the SSV method in regard to the example problems.

3. EDGE DETECTION

If the exact location of discontinuities, or edges, in a piecewise analytic function are known, the Gegenbauer Reconstruction Procedure recovers spectral accuracy at all points, including the discontinuity points. If a PDE solution is being postprocessed and the solution contains rarefaction waves, discontinuities in the first derivative of the function will exist and will need to be located. The method used to find the edges originated in Gelb and Tadmor [2000a] for periodic and nonperiodic functions. The method is specialized to approximations of functions by Chebyshev methods and is summarized below.

Denote the location of discontinuities as α_j. Let

$$[f](x) := f(x^+) - f(x^-)$$

denote a local jump in the function and define

$$ue(x) = \frac{\pi\sqrt{1-x^2}}{N}\sum_{k=0}^{N} a_k \frac{d}{dx} T_k(x), \tag{4}$$

where

$$\frac{d}{dx} T_k(x) = \frac{k\sin(k\arccos(x))}{\sqrt{1-x^2}}.$$

Essentially, we are looking at the derivative of the spectral projection of the numerical approximation to determine the location of the discontinuities. The series $ue(x)$ has the convergence properties

$$ue(x) \to \begin{cases} O(\frac{1}{N}) & \text{when } x \neq \alpha_j, \\ [f](\alpha_j) & \text{when } x = \alpha_j. \end{cases}$$

The series converges to both the height and direction of the jump at the location of a discontinuity. However, the GRP does not need the direction; it only needs the magnitude and location of the jumps. While a graphical examination of the series $ue(x)$ verifies that the series does have the desired convergence properties, an additional step is needed to numerically pinpoint the location of the discontinuities. For that purpose, make a nonlinear enhancement to the edge series as

$$un(x) = N^{\frac{Q}{2}}[ue(x)]^Q.$$

The values, $un(x)$, will serve to amplify the separation of scales which has taken place in Equation (4). The series has the convergence properties

$$un(x) \to \begin{cases} O(N^{\frac{-Q}{2}}) & \text{when } x \neq \alpha_j, \\ N^{\frac{Q}{2}}[[f](\alpha_j)]^Q & \text{when } x = \alpha_j. \end{cases}$$

By choosing $Q > 1$ we enhance the separation between the $O([\frac{1}{N}]^{\frac{Q}{2}})$ points of smoothness and the $O(N^{\frac{Q}{2}})$ points of discontinuity. The parameter J, whose value will be problem dependent, is a critical threshold value. Finally, redefine $ue(x)$ as

$$ue(x) = \begin{cases} ue(x) & \text{if } un(x) > J, \\ 0 & \text{otherwise.} \end{cases}$$

With Q large enough, one ends up with an edge detector $ue(x) = 0$ at all but $O(\frac{1}{N})$ neighborhoods of the discontinuities $x = \alpha_j$. Only those edges with amplitude larger than $J^{1/Q}\sqrt{1/N}$ will be detected.

Often the series ue is slow to converge in the area of a discontinuity and the nonlinear enhancement has difficulty pinpointing the exact location of the edge. If an additional parameter, η, is added to the procedure, this problem can be overcome in a simple manner. The parameter specifies that only one edge may be located in the interval $(x[i-\eta], x[i+\eta])$, $i = 0, \ldots, N$, with appropriate one-sided intervals being considered near boundaries. The correct edge will be the maximum of ue in this subinterval. The value of η is problem dependent and is best chosen after the edge detection procedure has been applied once.

The edge detection parameters J, Q, and η, are all problem dependent. Various combinations of the parameters may be used to successfully locate edges represented by jumps of magnitude in a certain range.

As mentioned previously, if a PDE solution is being postprocessed and the solution contains rarefaction waves, the first derivative of the solution will also have discontinuities and the edge detection procedure will have to be used to examine the first derivative of the solution in each piecewise smooth subinterval. After the shock locations are determined, the numerical solution can be differentiated in each C^0 smooth interval. Then, the locations of the discontinuities in the function and its derivatives are arranged in increasing order.

In most situations, it is sufficient to consider a more naïve approach which is easier to implement. If the numerical solution is differentiated over the entire computational domain, neighborhoods of noise will exist around the points where the edges in the solution were found. However, if the rarefaction waves are far enough away from the shock locations, edges in the derivative of the function can be successfully located by searching only a subinterval of the computational domain which is clear of any influence from the shocks.

4. GEGENBAUER RECONSTRUCTION PROCEDURE

The GRP works by expanding the function in another basis, the *Gibbs complementary basis*, via knowledge of the known Chebyshev coefficients and the location of discontinuities. The Chebyshev partial sums are projected onto a space spanned by the Gegenbauer polynomials. The approximation converges exponentially in the new basis even though it only converged very slowly in the original basis due to the Gibbs-Wilbraham phenomenon. The choice of a Gibbs complementary basis is the ultraspherical or Gegenbauer polynomials, C_n^λ. The Gegenbauer polynomials are orthogonal polynomials of order n which satisfy

$$\int_{-1}^{1}(1-x^2)^{\lambda-1/2}C_k^\lambda(x)C_n^\lambda(x)\,dx = \begin{cases} h_n^\lambda & k=n, \\ 0 & k\neq n, \end{cases}$$

where (for $\lambda \geqslant 0$)

$$h_n^\lambda = \pi^{\frac{1}{2}}C_n^\lambda(1)\frac{\Gamma\left(\lambda+\dfrac{1}{2}\right)}{\Gamma(\lambda)(n+\lambda)}$$

with

$$C_n^\lambda(1) = \frac{\Gamma(n+2\lambda)}{n!\Gamma(2\lambda)}.$$

Methods to implement the Gegenbauer polynomials of degree (λ, m) are in the class *gegenbauerPolynomial* of the SSPS.

The Gegenbauer expansion of a function $u(x), x \in [-1, 1]$ is

$$u(x) = \sum_{l=0}^{\infty}\widehat{f}_l^\lambda C_l^\lambda(x),$$

where the continuous Gegenbauer coefficients, $\widehat{f}_l^\lambda$, of $u(x)$ are

$$\widehat{f}_l^\lambda = \frac{1}{h_l^\lambda}\int_{-1}^{1}(1-x^2)^{\lambda-1/2}C_l^\lambda(x)u(x)\,dx. \tag{5}$$

Since we do not know the function $u(x)$, implementing the GRP requires obtaining an exponentially accurate approximation, $\widehat{g}_l^\lambda$, to the first m coefficients $\widehat{f}_l^\lambda$ in the Gegenbauer expansion from the first $N+1$ Chebyshev coefficients of $u(x)$. The approximate Gegenbauer coefficients are defined as the integral

$$\widehat{g}_l^\lambda = \frac{1}{h_l^\lambda} \int_{-1}^{1} (1-x^2)^{\lambda-1/2} C_l^\lambda(x) u_N(x)\, dx, \tag{6}$$

where u_N is the Chebyshev partial sum (1). The integral should be evaluated by Gauss-Lobatto quadrature in order to insure sufficient accuracy. The coefficients $\widehat{g}_l^\lambda$ are now used in the Gegenbauer partial sum to approximate the original function as

$$u(x) \approx u_m^\lambda(x) = \sum_{l=0}^{m} \widehat{g}_l^\lambda C_l^\lambda(x).$$

In practice, there will be discontinuities in the interval $[-1, 1]$ and the reconstruction must be done on each subinterval $[a, b]$ in which the solution remains smooth. To accomplish the reconstruction on each subinterval, define a local variable for each subinterval as $x(\xi) = \epsilon\xi + \delta$, where $\epsilon = (b-a)/2$, $\delta = (b+a)/2$, and $\xi_j = \cos(\pi j/N)$. The reconstruction in each subinterval is then accomplished by

$$u_m^{\lambda,\epsilon}(\epsilon\xi + \delta) = \sum_{l=0}^{m} \widehat{g}_\epsilon^\lambda(l) C_l^\lambda(\xi),$$

where

$$\widehat{g}_\epsilon^\lambda(l) = \frac{1}{h_l^\lambda} \int_{1}^{-1} (1-\xi^2)^{\lambda-1/2} C_l^\lambda(\xi) u_N(\epsilon\xi + \delta)\, d\xi.$$

Notice that we have used collocation points on the entire interval $[-1, 1]$ to build the approximation in $[a, b]$. This is referred to as a *global-local* approach [Gottlieb and Shu 1995b]. The global-local approach seems to be best when postprocessing PDE solutions where u_N is obtained from the time evolution of the problem . The point values $u(x_i)$ may not be accurate, but the global interpolating polynomial $u_N(x)$ is accurate.

In order to show that the GRP yields uniform exponential accuracy for the approximation, it is necessary to select λ and m such that $\lambda = m = \beta\epsilon N$, where $\beta < 2e/(27(1+1/2p))$, and p is the distance from $[-1, 1]$ to the nearest singularity in the complex plane, in each subinterval where the function being reconstructed is assumed to be analytic [Gottlieb and Shu 1997]. The choice of $\lambda = m$ is necessary to make the proof work for the exponential convergence of the method, but in practice it is not necessary and usually not advisable to choose $\lambda = m$. We are often more concerned with obtaining results for a fixed N, rather than maintaining an exponential convergence rate.

If the function to be postprocessed consists of homogeneous features, the reconstruction parameters can be successfully chosen as $\lambda = k_\lambda \epsilon N$ and $m = k_m \epsilon N$ for each subinterval where k_λ and k_m are user chosen, globally applied parameters. We refer to this strategy as the *global approach*. However, in problems with solutions containing varying detail throughout the computational

domain, the reconstruction parameters may need to be chosen independently in each subinterval [Sarra 2002]. We refer to this strategy as the *local approach*. To date there is no known method to choose optimal values of the reconstruction parameters m and λ. The parameters remain very problem dependent.

4.1 Computational Expense

At first examination, the GRP seems to require a triple summation for each grid point value of a function that is reconstructed. The method can be very computationally expensive as N and m grow. However, the use of the Christoffel-Darboux [Davis 1975] formula allows one of the sums to be eliminated [Gelb 2000]. The summation of the Chebyshev series can be done in an efficient manner with Clenshaw's [1962] recurrence formula. These two modifications result in a much more efficient method and are implemented in the software.

4.2 Round-Off Error

The Gegenbauer polynomials grow very rapidly with λ and m, which leads to a round-off error that may completely ruin the approximation. Round-off error is especially problematic for functions with a lot of variation and that require large values of m and/or λ. While the use of the Christoffel-Darboux formula reduces the computational effort, it adds to the round-off error problem as now two Gegenbauer polynomials are multiplied together. To lessen this problem, Gelb [2000] has suggested that the computation be rearranged in a way such that the two large and increasing Gegenbauer polynomial terms are first multiplied by quantities that are small and decreasing with respect to m, λ, and N. This rearrangement of the computation leads to a much more robust method and has been implemented in the software.

4.3 A Hybrid Approach

Even with the computational savings made via the Christoffel-Darboux formula, Gegenbauer Reconstruction may still be very computationally expensive in higher dimensions for large values of N. A hybrid approach was suggested in Gelb [2000] that uses an exponential filter, which may be applied very cheaply, in smooth regions and the GRP in the neighborhood of discontinuities. The exponential filter is

$$\sigma\left(\frac{k}{N}\right) = \exp\left(-\alpha\left|\frac{k}{N}\right|^{\beta}\right), \tag{7}$$

where α is the strength of the filter and β is the order of the filter. The exponential filter is an example of a spectral filter [Vandeven 1991] and can be used to recover a high order of accuracy away from points of discontinuity. The filtered partial sum takes the form

$$u_N(x) = \sum_{n=0}^{N} \sigma\left(\frac{n}{N}\right) a_n T_n(x).$$

Spectral filters do not completely remove the Gibbs-Wilbraham phenomenon, as oscillations in the neighborhood of discontinuities will not be removed.

The hybrid Gegenbauer postprocessing method is implemented for one-dimensional functions in the method *hybrid.postProcess()*.

5. SOFTWARE

Source code for the Spectral Signal Processing suite, a demonstration Applet containing all examples in this paper, and documentation of the graphical user interface (GUI) may be found at `www.scottsarra.org/signal/signal.html`.

5.1 Supported Collocation Grids

The standard collocation points for a Chebyshev collocation method are usually defined by

$$x_j = -\cos\left(\frac{\pi j}{N}\right), \qquad j = 0, 1, \ldots, N. \tag{8}$$

These points are extrema of the kth order Chebyshev polynomial $T_k(x)$ (Equation (3)). The points are often labeled the Chebyshev-Gauss-Lobatto (CGL) points, a name which alludes to the points role in certain quadrature formulas. The CGL points cluster quadratically around the endpoints and are less densely distributed in the interior of the domain. The CGL grid is denoted as grid 0 in the software. In the practical application of Chebyshev collocation methods for PDEs, a change of variable is often used to redistribute the collocation points. Three coordinate maps are supported by the software and are used in the examples.

The first map (grid 1 in the software) is the Kosloff/Tal-Ezer map [Kosloff and Tal-Ezer 1993]:

$$x = g(\xi, \gamma) = \frac{\arcsin(\gamma\xi)}{\arcsin(\gamma)}, \tag{9}$$

with $\gamma \in (0, 1)$. As γ approaches one, the grid points become nearly evenly spaced, and as γ approaches zero, the CGL grid is approached. The mapping also relaxes the $O(N^{-2})$ time-stepping restriction that is present when advancing Chebyshev methods with explicit time-stepping algorithms using the CGL grid.

The second map (grid 2) [Basdevant et al. 1986] is

$$x = g(\xi, \gamma) = (1.0 - \gamma)\xi^3 + \gamma\xi, \tag{10}$$

with $\gamma \in (0, 1)$. Smaller values of γ cluster grid points around the center of the computational interval while still maintaining a dense grid point distribution near boundary points. As $\gamma \to 1$ the grid approaches the CGL grid. The map can be used to resolve regions of rapid variation in the center of a computational domain.

The two-parameter tangent map (grid 3) [Bayliss and Turkel 1992] is

$$x = g(\xi, \gamma, \mu) = x_0 + \frac{\tan(\delta\xi + \omega)}{\gamma}, \tag{11}$$

where $\kappa = \arctan(\gamma(1 - \mu))$, $\gamma = \arctan(\gamma(1 + \mu))$, $\delta = 0.5(\kappa + \gamma)$, $\omega = 0.5(\kappa - \gamma)$, and $x_0 = -1 + 2(\mu - a)/(b - a)$. The map can be used to resolve solutions with

either a region of rapid variation in the interior or at boundaries. The map can be used to cluster grid points around the point μ in the interval $[a, b]$. The parameter $\gamma > 0$ determines the degree to which the clustering takes place.

5.2 Summary of Signal Processing Classes

Two classes implement versions of the edge detection procedure. Both classes have one public method, *findEdges*. The findEdges method in class *edgeDetect* locates edges in a function on its entire domain. The method takes input: f, an array of function values, and edge detection parameters, J, Q, and η. The method outputs an array containing the edge series, *ue*, and nonlinear enhancement, *nle*. The location of the edges are returned in the Vector d. The Vector of edges can be used as input to the postprocessing methods.

```
edgeDetect.findEdges(double[] f, double J, int Q,
     double[] ue, double[] nle, Vector d, int eta)
```

The findEdges method in class *edgeDetectAB* is used to find edges in the derivative of a function in the interval $[A, B]$. The method is identical to the findEdges method in class *edgeDetect* except that it requires as additional input the endpoints of the interval searched as well as an array, *xm*, containing the grid on which the function f is known.

```
edgeDetectAB.findEdges(double[] f, double J, int Q, double[] ue,
double[] nle, Vector d, int eta, double A, double B, double[] xm)
```

Two versions of the Gegenbauer reconstruction procedure are implemented. Both classes implement one public method, *postProcess*. The *postProcess* method in class *gegenbauerReconstruction* implements the procedure by using global reconstruction parameters LK and MK, which are input into the method. The method also takes as inputs: an array of function values, f, and a vector d, containing the location of the edges. As output, the method returns an array, *fG*, containing the values of the postprocessed function. The method also returns Vectors *mv* and *Lv*, which contain the value of the reconstruction parameters, m and λ in each smooth subinterval. The vectors *mv* and *Lv* are used as input to the *postProcess* method of class *gegenbauerReconstructionB*.

```
gegenbauerReconstruction.postProcess(double f[], double d[],
     double LK, double MK, double fG[], Vector mv, Vector Lv)
```

The *postProcess* method of class *gegenbauerReconstructionB* implements the local approach to specifying reconstruction parameters.

```
gegenbauerReconstructionB.postProcess(double f[], double d[],
                              double fG[], Vector mv, Vector Lv)
```

The methods takes as inputs: an array of function values, f, and a vector d, containing the location of the edges. Additionally, Vectors *mv* and *Lv* specifying the reconstruction parameters, m and λ, in each smooth subinterval need to be input. The method returns the array *fG* containing the postprocessed function values.

The classes *hybrid* and *hybridB* implement *postProcess* methods similar to those of *gegenbauerReconstruction* and *gegenbauerReconstructionB* except that an exponential filter may be used to postprocess the input function in neighborhoods away from the edge locations. The methods in both classes take additional inputs, alpha and gamma, which, respectively, determine the strength and order of the exponential filter (7). The *postProcess* method of class hybrid also takes as an input the parameter *EPS*, which specifies that the GRP is to be applied in intervals $(d[j] - EPS, d[j] + EPS)$ around each edge $d[j]$.

```
hybrid.postProcess(double f[], double d[], double LK,
  double MK, double fG[], Vector mv, Vector Lv, double alpha,
  double gamma, double EPS)
```

The *postProcess* method of class *hybridB* allows different reconstruction parameters to be applied in each smooth subinterval through the input Vectors *mv* and *Lv*, in the same way that *gegenbauerReconstructionB.postProcess* does. Additionally, the value of *EPS* around each edge may be specified separately by values input in the Vector *ev*.

```
hybridB.postProcess(double f[], double d[], double fG[],
Vector mv, Vector Lv, Vector ev, double alpha, double gamma)
```

6. EXAMPLES

The example functions are included as part of the software package and are available from the examples menu in the graphical user interface (GUI). The first four examples use the software suite to reconstruct functions approximated by a Chebyshev partial sum (1). The examples use two parameters, *Nex*, which sets the number of terms in the partial sum (1) to $Nex + 1$, and M, which has various meanings depending on the example. The two parameters may be set through the GUI. The remaining examples demonstrate using the software as a postprocessing method for the Chebyshev collocation method and Chebyshev SSV method. All PDE example problems can be stated as a system conservation laws with a source term as

$$u_t + f(u)_x = \psi(u). \tag{12}$$

6.1 Step Function

The first example consists of the step function on $[-1, 1]$ defined as $f(x) = -1$ if $x \leqslant 0$ and $f(x) = 1$ if $x > 0$. The approximation of the step Function by a Chebyshev series is shown in Figure 1.

With $Nex = 60$ and on the CGL grid (grid $= 0$), the edge at $x = 0$, which has a jump of magnitude 2, can be located with $J = 10$, $Q = 1$, and $\eta = 2$. This choice of edge detection parameters results in jumps of $J^{1/Q}\sqrt{1/N} \approx 0.9$ and larger being located. In this case, we just need $J^{1/Q}\sqrt{1/N} \geqslant 0.85$ to locate the correct edge location.

The function has a homogeneous structure throughout its domain, which indicates that the reconstruction parameters can be chosen globally. The reconstruction parameters can be specified by setting $k_\lambda = 0.1$ and $k_m = 0.025$,

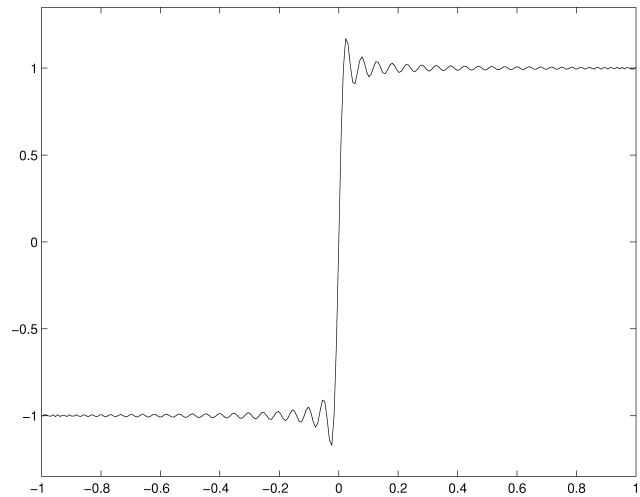

Fig. 1. Step function approximation.

which results in $m = 1$ and $L = 3$ in each smooth subinterval. A slightly better result (check the error from the options menu) can be obtained with $k_\lambda = 0.5$ and $k_m = 0.025$, which results in $m = 1$ and $L = 15$ in each smooth subinterval.

6.2 Sine Wave

The function $f(x) = Sin(M\pi x)$ will test the resolution properties of the Gegenbauer expansion. This smooth function contains M complete wavelengths in the interval $[-1, 1]$. It was shown in Gottlieb and Shu [1994] that the Gegenbauer expansions needs a minimum of π points per wavelength to completely resolve a wave.

Choose $Nex = 350$, use the CGL grid (grid = 0), and set $M = 20$. The function is smooth so there are no edges in the interval. If the global Gegenbauer parameters are set by specifying $k_\lambda = 0.003$ and $k_m = 0.25$, (which results in $m = 88$ and $L = 1.05$), the function is well resolved.

6.3 Combination

The previous two examples demonstrate that if the function has homogeneous structure throughout its domain, then the reconstruction parameters can be chosen globally. However, if the function consists of subintervals of varying detail, a local approach to choosing the reconstruction parameters may be necessary. For example, consider function (13) below. Experimentally, we were unable to chose the reconstruction parameters globally and obtain good results. Instead, a local approach seems necessary, where independent values of m and λ are specified separately in each smooth subinterval.

For example, take $Nex = 400$, $grid = 0$, and $M = 0.015$ to specify the width of the feature in the interval [0, 1]. The resulting Chebyshev approximation is shown in Figure 2. The edges can be located with $J = 20$, $Q = 1$, and $\eta = 2$. In the regions where the function is piecewise constant, reconstruction parameters of

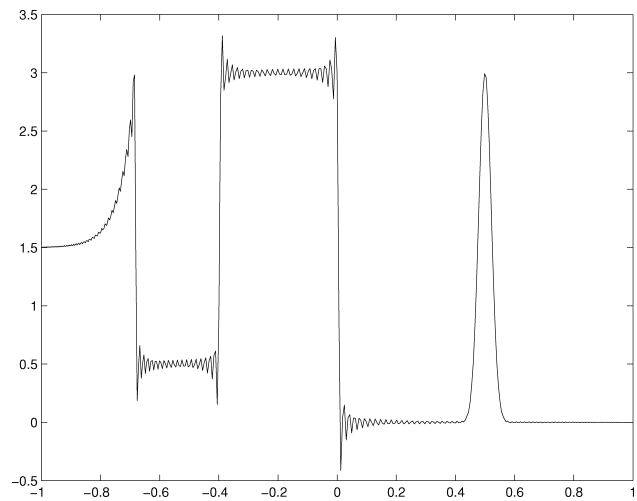

Fig. 2. Chebyshev approximation of function (13).

$\lambda = 2$ and $m = 2$ provide good results. In the region $[-1, -0.68]$, the function is of moderate detail, and reconstruction can be accomplished with moderate values of $m = 9$ and $\lambda = 9$. In the interval $[0, 1]$, which consists of a narrow exponential spike, the function contains small-scale structures which will require a large value of m, and small value of λ, similar to those in the sine wave example, for example, $\lambda = 0.1$ and $m = 70$.

$$f(x) = \begin{cases} 3\exp(\frac{-(x-0.5)^2}{4M^2}) & \text{if } 0 \leqslant x \leqslant 1, \\ 3 & \text{if } -0.4 \leqslant x < 0, \\ 0.5 & \text{if } -0.68 \leqslant x < -0.4, \\ 1.5 + 1.5\exp(20(x - 0.68)) & \text{if } -1 \leqslant x < -0.64. \end{cases} \tag{13}$$

6.4 Center Step

Theoretical proofs [Gottlieb and Shu 1995b] exist showing exponential convergence properties of the GRP for functions known on the CGL grid. Numerical evidence indicates that method may also be applied on grids arising from mappings of the Chebyshev grid. The determining factor in the accuracy of the reconstruction is how well the chosen grid can capture the function [Sarra 2002].

For example, consider the piecewise analytic function (14). Set $Nex = 160$, $M = 0.15$, and form the grid with map (9) with $\gamma = 0.9999$. The grid is nearly evenly spaced. The Chebyshev approximation of the function is shown in Figure 3. The global Gegenbauer parameters can be set by specifying $k_\lambda = 0.2$ and $k_m = 0.025$ to achieve a successful reconstruction of the function on the

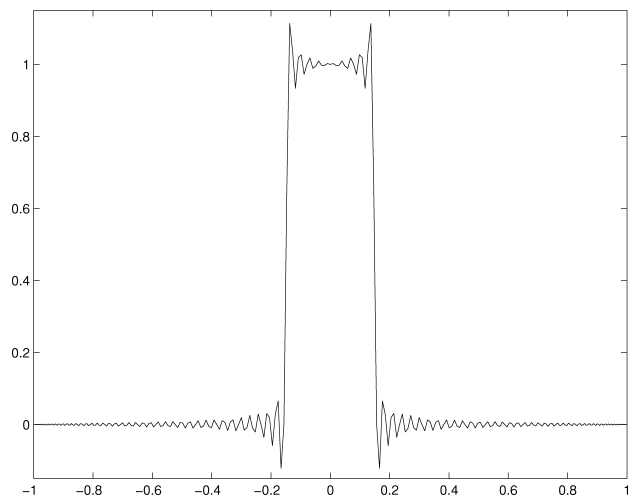

Fig. 3. Chebyshev approximation of function (14).

mapped grid.

$$f(x) = \begin{cases} 0 & \text{if } x \leqslant M - 1 \text{ or } x \geqslant 1 - M, \\ 1 & \text{otherwise.} \end{cases} \tag{14}$$

6.5 Hyperbolic Heat Transfer

Our first example of using the GRP as a postprocessing method for the Chebyshev collocation method for PDEs is to the equations of hyperbolic heat transfer. The governing equations of hyperbolic heat transfer can be written in form (12), with $u = [T, Q]^T$, $f(u) = [Q, T]^T$, $\psi = [S/2, -2Q]^T$. $T(x, t)$ is the temperature, $Q(x, t)$ is the heat flux, and $S(x, t)$ is the energy generation rate. Both examples used the initial conditions $T(x, 0) = 0$ and $Q(x, t) = 0$ on [0, 1]. The problem is linear and the Chebyshev collocation method is stable without the addition of any spectral viscosity. The system was advanced in time with a fourth-order explicit Runge-Kutta method. More detailed information about the spectral solution of this problem may be found in Sarra [2003b].

For the first hyperbolic heat transfer example, system (12) is solved with boundary conditions of $Q(0, t) = 1$, $Q(1, t) = 0$, $T_t(0, t) = -Q_x(0, t)$, and $T_x(1, t) = 0$. The energy generation rate, S, is set to zero.

In the Figure 4, the temperature solution, T, is shown at time $t = 0.5$ with $N = 33$ on the CGL grid (8). Strong oscillations are noticeable at the boundary $x = 0$, due to the jump in the heat flux, Q ,which is felt by the temperature.

An edge is found to be at $x = 0.476$ with the parameters $J = 200$, $Q = 4$, and $\eta = 2$. This choice of edge detection parameters results in jumps of 0.65 and larger being located. The exact jump is 0.65 in magnitude. By specifying $\eta = 2$, the oscillation near $x = 0$ is not falsely determined to be a jump in the function. With only 33 grid points, the convergence of the edge series, Figure 5, is not yet readily apparent. However, if the edge detection parameters are chosen appropriately, the correct edge locations will be found.

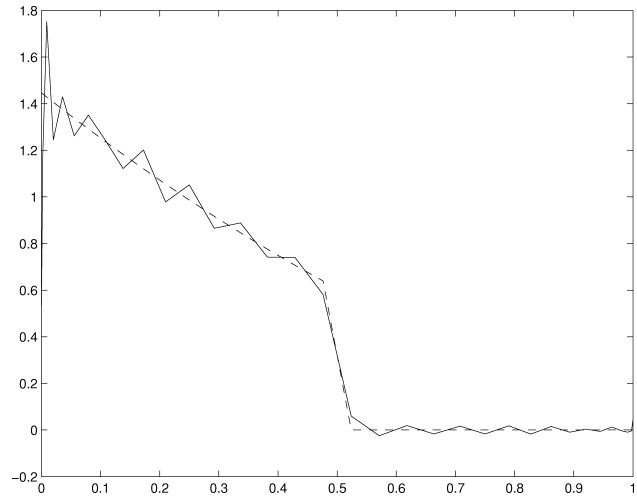

Fig. 4. Numerical (solid) versus exact.

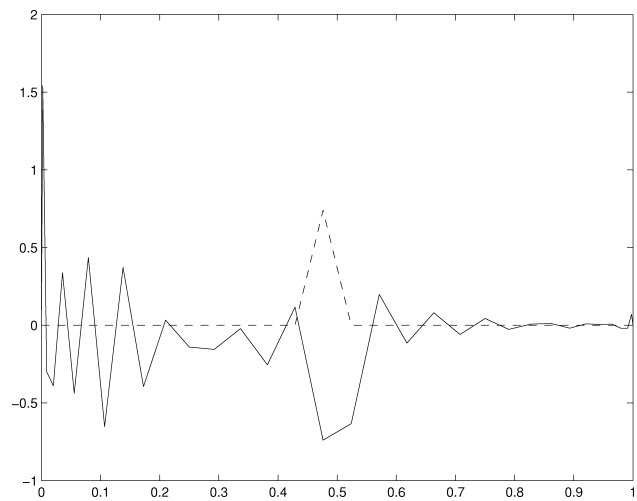

Fig. 5. Edge series (solid) and enhancement.

After the edges have been located, the GRP is applied in each smooth subinterval by using global parameters chosen as $k_\lambda = 0.3$ and $k_m = 0.1$, which results in $m = 2$ and $\lambda = 4.7$ in subinterval $(0, 0.476)$ and $m = 2$ and $\lambda = 5.2$ in subinterval $(0.476, 1)$. For the homogeneous solution in this example, choosing global reconstruction parameters results in a successful application of the GRP. The postprocessed solution is shown in Figure 6.

For the second hyperbolic heat transfer example, system (12) is solved with boundary conditions of $Q(0,t) = 0$, $Q(1,t) = 0$, $T_x(0,t) = 0$, and $T_x(1,t) = 0$. The energy generation rate is specified as $S(x,t) = \frac{1}{dn}$, if $0 \le x \le dn$, and zero otherwise. The energy generation rate, S, represents a pulsed energy source released instantaneously at time $t = 0$.

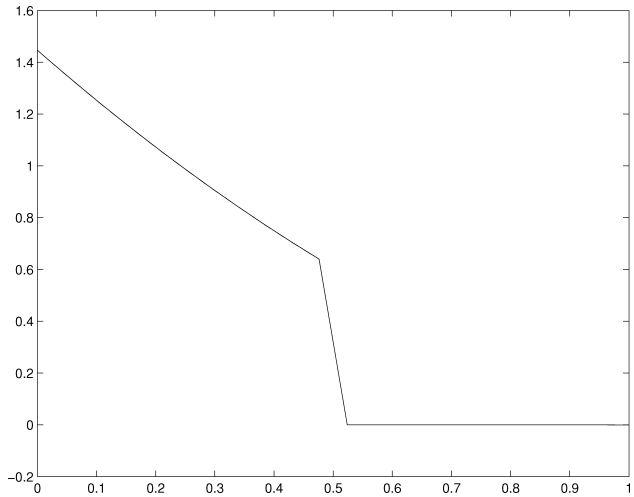

Fig. 6. Postprocessed (solid) versus exact.

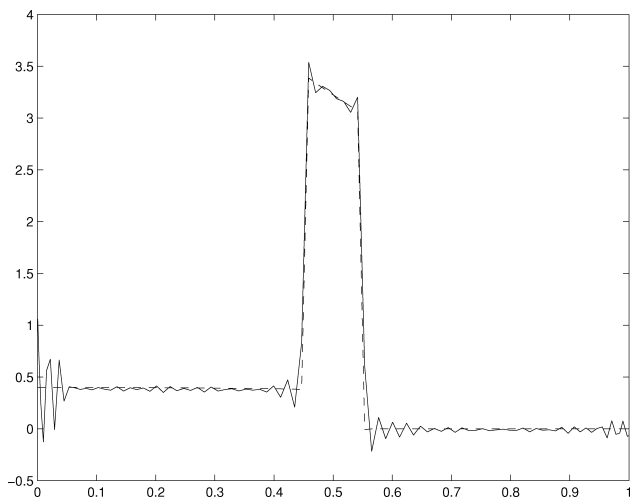

Fig. 7. Numerical (solid) and exact (dashed).

The temperature solution (Figure 7) with $dn = 0.05$ is shown at time $t = 0.5$ with $N = 99$ collocation points distributed with map (9) with $\gamma = 0.96$. By taking the map parameter as $\gamma = 0.96$, the grid becomes closer to evenly being spaced and better resolution is realized in the center of the domain.

Edges (Figure 8) are found to be at $x = 0.447$ and $x = 0.541$ with the parameters $J = 5000$, $Q = 3$, and $NE = 1$. With these choices of the edge detection parameters, only jumps of magnitude greater than 1.72 are found. Other combinations of J and Q could work equally as well.

After the edges have been located, the GRP is applied in each smooth subinterval by using the global parameters $k_\lambda = 0.2$ and $k_m = 0.02$. The results are shown in Figure 9.

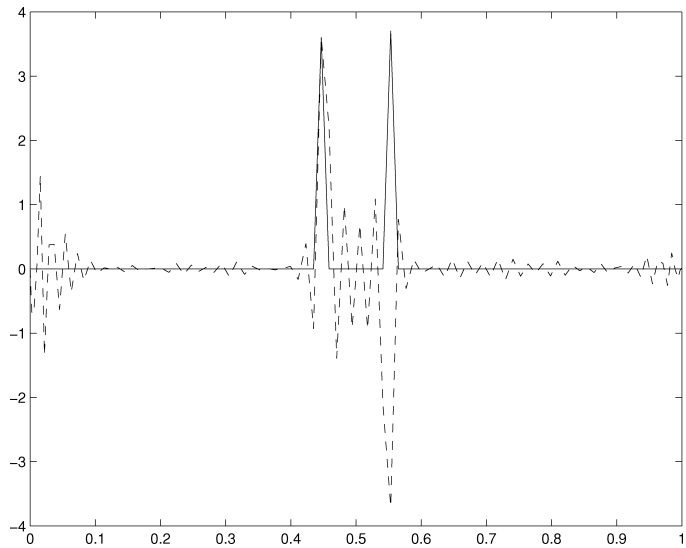

Fig. 8. Enhancement (solid) and edge series.

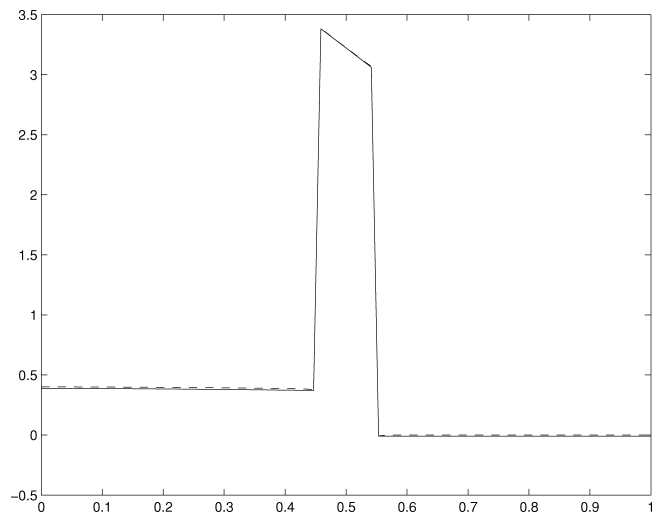

Fig. 9. Postprocessed (solid) and exact.

6.6 Shallow Water Equations

Cast in form (12), the shallow water equations are $u = [v, h]^T$, $f(u) = [v^2h + 0.5gh^2, vh]^T$, and $\psi(u) = 0$. The variable $h(x, t)$ is the height of the free upper surface, $v(x, t)$ is the depth averaged fluid velocity, and g is the acceleration due to gravity.

Our example is the dam break problem consisting of an initial condition of $h(x, 0) = h_0$ if $x < 0$, $h(x, 0) = h_1$ if $x \geqslant 0$, and $v(x, 0) = 0$. The solution consists of a right moving-shock and a left-moving rarefaction. The Chebyshev SSV solution (Figure 10) was calculated with $N = 128$ on a grid formed with map

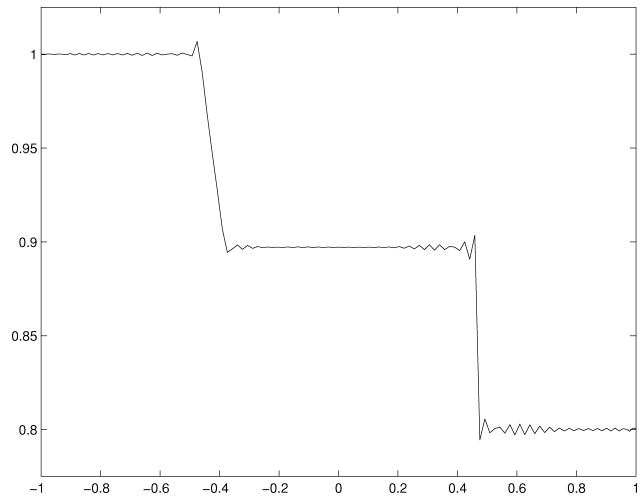

Fig. 10. SSV solution, h(x,t).

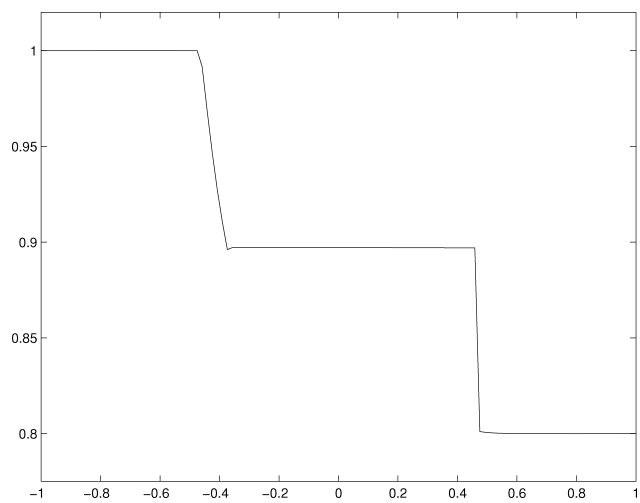

Fig. 11. Postprocessed, h(x,t).

(9) with $\gamma = 0.99$. The system was advanced in time with a fourth-order explicit Runge-Kutta method.

The edge detection procedure finds edges in the first derivative of the height solution in the interval $[-1, 0]$ at $x = -0.475$ and $x = -0.374$ with $J = 70$, $Q = 2$, and $\eta = 1$. A shock is found in the solution at $x = 0.458$ with $J = 1$, $Q = 1$, and $\eta = 1$.

The homogeneous features of the solution allow the reconstruction parameters to be chosen globally through the parameters $k_\lambda = 0.6$ and $k_m = 0.15$. The postprocessed solution is shown in Figure 11.

An exact solution to the problem exists. However, for convenience, the reference solution in the software was computed with the second-order Nessayhu-Tadmor scheme [Nessyahu and Tadmor 1990] on a very fine grid ($N = 4000$) and interpolated with third-order accuracy to the spectral grid.

6.7 Fluidized Bed Equations

Fluidized beds are used in the chemical and fossil fuel processing industries to mix particulate solids and fluids (gases or liquids). A typical fluidized bed consists of a vertically oriented chamber, a bed of particulate solids, and a fluid flow distributor at the bottom the chamber. The fluid flows upward through the particles creating a force that counteracts gravity, at which time a state of minimum fluidization is reached. Stronger gas inflows (more than is necessary to maintain minimum fluidization) lead to pockets of gas, or equivalently low particle concentrations, resembling bubbles in a liquid traveling upward through the particles. Each rising bubble pushes a large amount of mass in front of it. Particles move downward through and around the rising bubble until it reaches the top of the bed. A settled bed is reestablished, and the cycle repeats. Each set of upward moving particles is referred to as a *slug*.

A model that represents a one-dimensional simplification of two- and three-dimensional fluidized bed models can be cast in the form on a nonlinear system of conservation laws in form (12) with $u = [\alpha, m]^T$ and $f(u) = [m, m^2/\alpha + F(\alpha)]^T$. The variable $\alpha(x, t)$ denotes the concentration of particles by volume, $m(x, t) = \alpha v$ represents the particle momentum, and $v(x, t)$ the particle velocity.

A more detailed description of the model can be found in Christie et al. [1991] and Christie and Palencia [1991]. If the source term is neglected, an exact solution to the Riemann problem for the homogeneous system can be found. The exact solution is developed in Christie and Palencia [1991]. A detailed description of the Chebyshev SSV solution of the model may be found in Sarra [2002, 2003a].

The appearance of the slugging behavior in the solution will create a solution with nonhomogeneous structure and detail throughout the domain of the problem. A local approach to specifying the reconstruction parameters will be needed.

6.7.1 *Homogeneous System, Shock-Rarefaction.* In our first example using the fluidized bed equations, we considered the Riemann problem for the homogeneous system for which an exact solution exists. This example consisted of a left-moving shock and a right-moving rarefaction wave. The initial conditions consisted of $\alpha(x, 0) = 0.3$ if $x < 0$ and $\alpha(x, 0) = 0.55$ if $x \geq 0$. The initial velocity was $v(x, 0) = 0$ for all x. The computation was done on a domain of $[-0.2, 0.2]$. Figure 12 shows the SSV solution at $t = 0.5$. The grid consists of 64 points distributed by map (10) with $\gamma = 0.25$. The use of the coordinate map had the effect of placing more points in the center of the domain. The system was advanced in time with a fourth-order explicit Runge-Kutta method.

The rarefaction wave is characterized by the solution having a discontinuous first derivative; thus edge detection must be applied to the first derivative

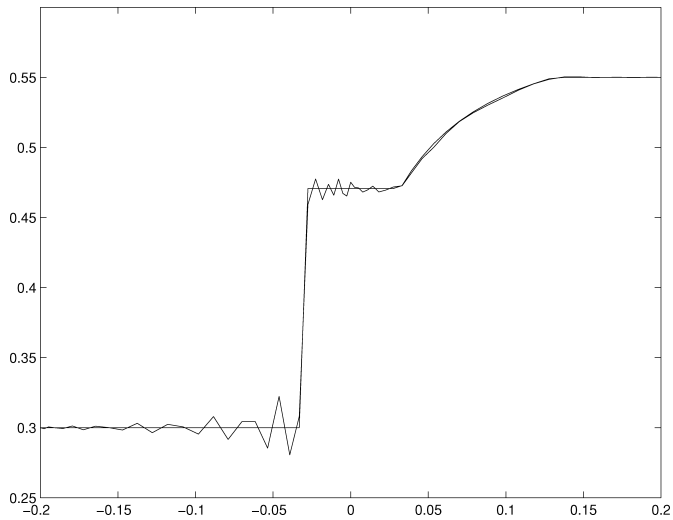

Fig. 12. $\alpha(x, t = 0.5)$, SSV approximation (oscillatory) versus exact.

Table I. Local Reconstruction Parameters

Subinterval	m	λ
(−0.2,−0.033)	1	2
(−0.033,0.033)	1	3
(0.033,0.1374)	4	1
(0.1374,0.2)	1	2

of the solution in addition to the solution itself. The edge detection procedure with $Q = 1$ and $J = 1$ located jumps of magnitude greater than 0.125. With these settings, the edge detection procedure located edges in the function and the first derivative of the function at $x = -0.0331$, $x = 0.0331$, and $x = 0.1374$.

We were unable to get good postprocessed results by specifying the reconstruction parameters globally through the parameters k_λ and k_m. Global parameter specification failed due to the solution containing three intervals of piecewise constant values and a fourth interval, $(0.033, 0.1374)$, consisting of a function requiring different reconstruction parameters. Good results were obtained by specifying the GRP parameters locally in each smooth subinterval, as listed in Table I. The postprocessed solution is shown in Figure 13.

6.7.2 *Slugging Problem.* The initial conditions were taken as the state of minimum fluidization, which is the state of the system at the minimum gas flow necessary for the particle phase to be balanced by the upward force of the gas flow. Details of the exact determination of minimum fluidization for this problem may be found in Sarra [2003a]. The nonhomogeneous system was advanced in time with Strang splitting Strang [1968] and a second-order explicit Runge-Kutta method.

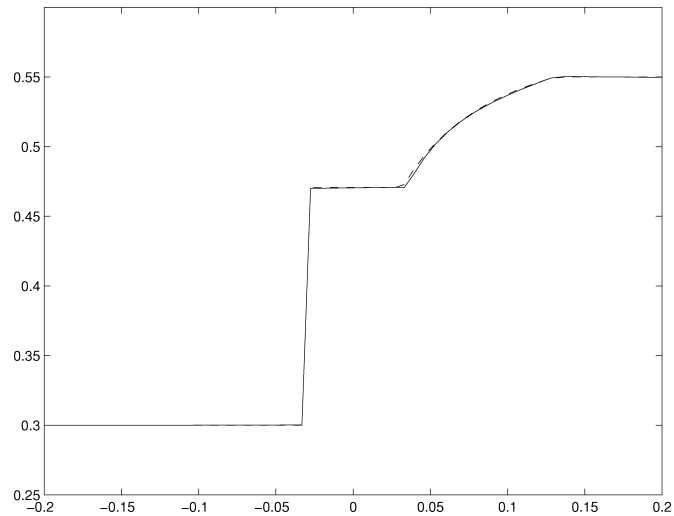

Fig. 13. Postprocessed (solid) versus exact (dashed).

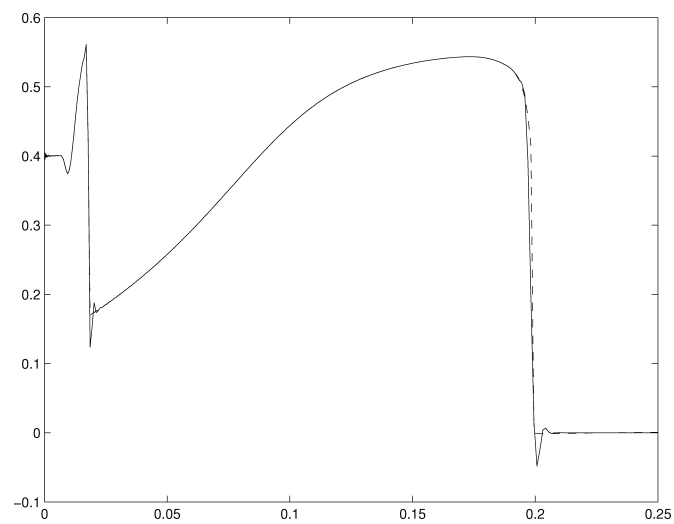

Fig. 14. SSV (solid) versus postprocessed, $t = 0.5$.

The SSV solution of the slugging problem was examined at $t = 0.5$, when the slugging behavior was first becoming apparent. The collocation grid consisted of 256 CGL (8) grid points. The edge detection procedure, Figure 15, with $J = 1$, $Q = 1$, and $\eta = 2$ located shocks at $x = 0.01778$ and $x = 0.19822$. The postprocessed solution, Figure 14, was obtained by locally specifying the reconstruction parameters in each smooth subinterval, as listed in Table II.

An exact solution to the problem is not known. The reference solution included in the software was computed by Roe's method [Roe 1981] with $N = 1024$ and interpolated with third-order accuracy to the spectral grid. In Figure 16,

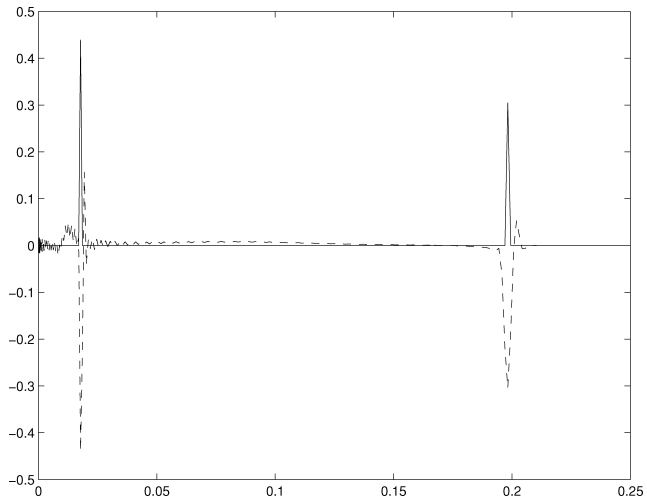

Fig. 15. Edge series and enhancement(solid).

Table II. Local Reconstruction Parameters

Subinterval	m	λ
(0,0.01778)	15	2
(0.01778,0.19822)	14	4
(0.19822,0.25)	1	1

the Roe's method solution is shown with the postprocessed spectral solution from Figure 14. There is a good agreement between the two solutions. The slight variation in the two solution is largely due to the fact that Roe's method is calculated on a uniform grid while the spectral method uses a nonuniform grid, which led to the initial conditions being slightly different. The solution of the system is very dependent on the initial conditions. The slightest variation of the initial conditions results in a noticeably different concentration profile at later times. A more detailed description of applying the GRP to this problem can be found in Sarra [2002, 2003a].

7. CONCLUDING COMMENTS

In this paper, Version 1.0 of the Spectral Signal Processing Suite was described. In the current version methods are available to locate edges in and to post-process Chebyshev approximations of discontinuous functions. The algorithms have been implemented using the most efficient known methods, as a straightforward implementation of the GRP algorithm is extremely costly. Additionally, steps have been taken to reduce round-off errors which plague the GRP algorithm.

Work is underway on version 2.0 of the SSPS. Version 2.0 will implement edge detection, the GRP, spectral mollification, spectral filtering, and Padé-based algorithms for both Chebyshev and Fourier approximations. SSPS 2.0 will contain a large number of the known methods that may be used to remove or reduce the Gibbs-Wilbraham phenomenon in spectral approximations

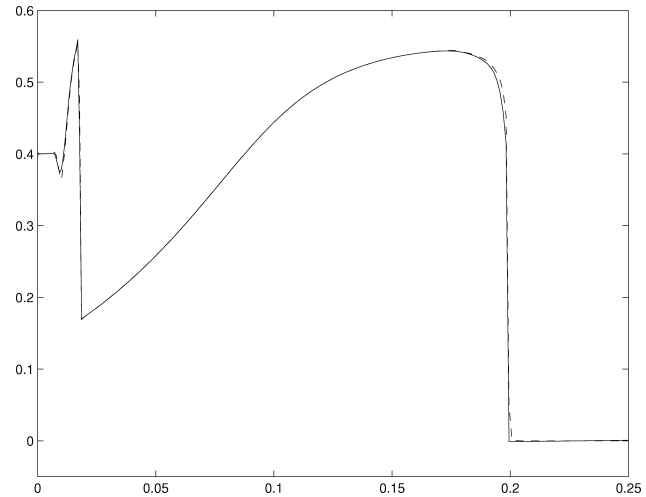

Fig. 16. Postprocessed versus reference (dashed), $t = 0.5$.

of discontinuous functions. The package will allow the user to compare both the results and the efficiency of the various methods.

REFERENCES

BASDEVANT, C., DEVILLE, M., HALDENVVANG, P., LACROIX, J., ORLANDI, D., PATERA, A., PEYRET, R., AND QUAZZANI, J. 1986. Spectral and finite difference solutions of Burgers' equation. *Computat. Fluids 14*, 1, 23–41.

BAYLISS, A. AND TURKEL, E. 1992. Mappings and accuracy for Chebyshev pseudospectral approximations. *J. Computat. Phys. 101*, 349–359.

CANUTO, C., HUSSAINI, M. Y., QUARTERONI, A., AND ZANG, T. A. 1988. *Spectral Methods for Fluid Dynamics*. Springer-Verlag, New York, NY.

CHRISTIE, I., GANSER, G., AND SANZ-SERNA, J. 1991. Numerical solution of a hyperbolic system of conservation laws with source term arising in a fluidized bed model. *J. Computat. Phys. 93,* 2, 297–311.

CHRISTIE, I. AND PALENCIA, C. 1991. An exact Riemann solver for a fluidized bed model. *IMA J. Numer. Anal. 11*, 493–508.

CLENSHAW, C. 1962. *Mathematical Tables, Volume 5*. National Physics Laboratory, London, U.K.

DAVIS, P. J. 1975. *Interpolation and Approximation*. Dover Publications, New York, NY.

DRISCOLL, T. AND FORNBERG, B. 2001. A Padé-based algorithm for overcoming the Gibbs phenomenon. *Numer. Alg. 26*, 77–92.

FORNBERG, B. 1996. *A Practical Guide to Pseudospectral Methods*. Cambridge University Press, New York, NY.

FUNARO, D. 1992. *Polynomial Approximation of Differential Equations*. Springer-Verlag, New York, NY.

GELB, A. 2000. A hybrid approach to spectral reconstruction of piecewise smooth functions. Preprint.

GELB, A. AND TADMOR, E. 2000a. Detection of edges in spectral data II: Nonlinear enhancement. *SIAM J. Numer. Analy. 38,* 4, 1389–1408.

GELB, A. AND TADMOR, E. 2000b. Enhanced spectral viscosity approximations for conservation laws. *Appl. Numer. Math. 33*, 3–21.

GOTTLIEB, D., HUSSAINI, M. Y., AND ORSZAG, S. A. 1984. Theory and application of spectral methods. In *Spectral Methods for Partial Differential Equations*, R. G. Voigt, D. Gottlieb, and M. Y. Hussaini, Eds. SIAM, Philadelphia, PA, 1–54.

GOTTLIEB, D. AND ORSZAG, S. A. 1977. *Numerical Analysis of Spectral Methods*. SIAM, Philadelphia, PA.

GOTTLIEB, D. AND SHU, C.-W. 1994. Resoulution properties of the Fourier method for discontinuous waves. *Computat. Methods Appl. Mech. Engrg. 116*, 27–37.

GOTTLIEB, D. AND SHU, C.-W. 1995a. On the Gibbs phenomenon IV: Recovering exponential accuracy in a subinterval from a Gegenbauer partial sum of a piecewise analytic function. *Math. Computat. 64*, 1081–1095.

GOTTLIEB, D. AND SHU, C.-W. 1995b. On the Gibbs phenomenon V: Recovering exponential accuracy from collocation point values of a piecewise analytic function. *Numerische Math. 71*, 511–526.

GOTTLIEB, D. AND SHU, C.-W. 1996. On the Gibbs phenomenon III: Recovering exponential accuracy in a subinterval from a partial sum of a piecewise analytic function. *SIAM J. Numer. Anal. 33*, 280–290.

GOTTLIEB, D. AND SHU, C.-W. 1997. On the Gibbs phenomenon and its resolution. *SIAM Rev. 39,* 4, 644–668.

GOTTLIEB, D., SHU, C.-W., SOLOMONOFF, A., AND VANDEVEN, H. 1992. On the Gibbs phenomenon I: Recovering exponential accuracy from the Fourier partial sum of a nonperiodic analytic function. *J. Computat. Appl. Math. 43*, 81–98.

GOTTLIEB, D. AND TADMOR, E. 1985. Recovering pointwise values of discontinuous data within spectral accuracy. In *Progress and Supercomputing in Computational Fluid Dynamics*, E. M. Murman and S. S. Abarbanel, Eds. Birkhäuser, Boston, MA, 357–375.

KABER, O. 1996. A Legendre pseudospectal viscosity method. *J. Computat. Phys. 128*, 165–180.

KABER, O. AND MAHMOUD, S. 1994. Filtering non-periodic function. *Computat. Meth. Appl. Mech. Eng. 116*, 123–130.

KOSLOFF, R. AND TAL-EZER, H. 1993. A modified Chebyshev pseudospectral method with an $O(1/N)$ time step restriction. *J. Computat. Phys. 104*, 457–469.

MA, H. 1998. Chebyshev-Legendre super spectral viscosity method for nonlinear conservation laws. *SIAM J. Numer. Anal. 35*, 3, 893–908.

NESSYAHU, H. AND TADMOR, E. 1990. Non-oscillatory central differencing for hyperbolic conservation laws. *J. Computat. Phys. 87*, 2, 408–463.

ROE, P. 1981. Approximate Riemann solvers, parameter vectors, and difference schemes. *J. Computat. Phys. 43*, 357–372.

SARRA, S. A. 2002. Chebyshev pseudospectral methods for conservation laws with source terms and applications to multiphase flow. Ph.D. dissertation, West Virginia University, Morgantown, WV.

SARRA, S. A. 2003a. Chebyshev super spectral viscosity method for a fluidized bed model. *J. Computat. Phys. 186*, 2, 630–651.

SARRA, S. A. 2003b. Spectral methods with postprocessing for numerical hyperbolic heat transfer. *Numer. Heat Trans. A 43,* 7, 717–730.

STRANG, G. 1968. On the construction and comparison of difference schemes. *SIAM J. Numer. Anal. 5,* 3, 506–517.

TADMOR, E. 1989. Convergence of spectral methods for nonlinear conservation laws. *SIAM J. Numer. Anal. 26*, 30–44.

TADMOR, E. AND TANNER, J. 2002. Adaptive mollifers—high resolution recovery of piecewise smooth data from its spectral information. *Found. Computat. Math. 2*, 155–189.

TREFETHEN, L. N. 2000. *Spectral Methods in Matlab*. SIAM, Philadelphia, PA.

VANDEVEN, H. 1991. Family of spectral filters for discontinuous problems. *SIAM J. Sci. Computat. 6*, 2, 159–192.

Received May 2002; revised February 2003; accepted March 2003