The Matlab Postprocessing Toolkit[*]

S. A. Sarra
Marshall University
Huntington, WV

July 10, 2008

Abstract

Global polynomial approximation methods applied to piecewise continuous functions exhibit the well-known Gibbs phenomenon. We summarize known methods to remove the Gibbs oscillations and present a collection of Matlab programs that implement the methods. The software features a Graphical User Interface that allows easy access to the postprocessing algorithms for benchmarking and educational purposes.

1 Introduction

Spectral methods approximate functions by projection onto a space $\mathbb{P}_N$ of orthogonal polynomials of degree $\leq N$. When the underlying function is periodic trigonometric (Fourier) polynomials are employed while a popular choice for non-periodic functions are the Chebyshev polynomials. Legendre polynomials are another option in the non-periodic case but are not as popular in applications due to the lack of a fast transform method. However spectral methods based on other, non-classical orthogonal polynomials are possible as well. For example see references [6] and [44]. Spectral methods yield extremely accurate approximations of smooth functions. Due to their excellent approximation properties, spectral methods have become popular in applications such as numerical partial differential equations. However, when functions are only piecewise smooth the well-known Gibbs phenomenon appears as an accuracy reduction to first order away from discontinuities and $\mathcal{O}(1)$ oscillations in the neighborhoods of jumps. The Gibbs phenomenon

[*] This work was partially supported by NSF grant DMS-0609747

is illustrated in figure 1 for both the Fourier and Chebyshev approximation of the function

$$f(x) = \chi_{[-0.5,0.5]} * \sin[\cos(x)] \tag{1}$$

that will be used throughout to demonstrate the software.

A number of methods have been suggested for the purpose of reducing or eliminating the Gibbs phenomenon. They include: spectral filtering [43], physical space filtering using mollifiers [19, 39, 40, 38, 41], digital total variation filtering [33], rational reconstruction [27, 21], and a variety of direct [17, 12] and indirect [36, 23] reprojection methods. The most powerful methods need to know the exact location of all discontinuities.

The purpose of this paper is to describe a Matlab software package, the Matlab Postprocessing Toolkit (MPT), that implements edge detection and postprocessing algorithms for Chebyshev and Fourier spectral methods in one and two space dimensions. The software is intended for applications, algorithm benchmarking, and educational purposes. The MPT is a significant extension and translation of the Spectral Signal Processing Suite (SSPS) [32]. The SSPS was implemented in the Java programming language which limited its usefulness. The SSPS only implemented edge detection, spectral filtering, and Gegenbauer Reprojection, for one dimensional Chebyshev approximations. The MPT is implemented in a language known by a large number of scientists and engineers and is broader in the scope of algorithms implemented.

2 Global Polynomial Approximation Methods

The software package is based on interpolation, rather than expansion, methods incorporating Chebyshev and trigonometric polynomials. Interpolation and expansion methods have the same excellent approximation properties but we have chosen interpolation since pseudospectral methods for PDEs are based on interpolation. The interpolating approximation

$$\mathcal{I}_N f(x) = \sum_k a_k \, \phi_k(x) \tag{2}$$

with expansion coefficients a_k and basis functions $\phi_k(x)$ on interval $\Omega = [-1, 1]$, satisfies $\mathcal{I}_N f(x_i) = f(x_i)$ at $N + 1$ interpolation sites x_i. *Interpolation* means that $f(x)$, the function that is approximated, is a known function (at least at the interpolation sites) while the terms *collocation* and *pseudospectral* are applied to global polynomial interpolatory methods for solving differential equations for an unknown function $f(x)$. We refer to

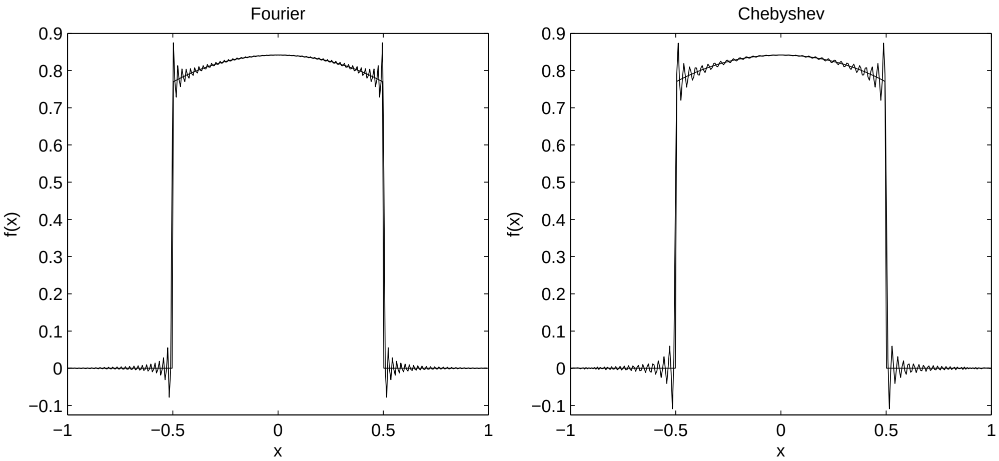


Figure 1: Spectral approximation of function (1) vs. the exact function. The function is known at $N = 200$ interpolation sites and the interpolant is evaluated at $M = 298$ evenly spaced points. Left: Fourier. Right: Chebyshev.

both situations as spectral approximation or spectral methods. Detailed information on spectral methods may be found in the standard references [3, 4, 20, 29, 42].

2.1 Chebyshev Interpolation

In (2) the index k runs over $k = 0, 1, \dots, N$ and the basis functions are the Chebyshev polynomials [26]

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

The expansion coefficients are efficiently calculated via the FFT (chebyshevCoefficients.m). The interpolation sites are the Chebyshev-Gauss-Lobatto (CGL) points

$$x_k = -\cos\left(\frac{k\pi}{N}\right) \quad k = 0, 1, \dots, N. \tag{4}$$

The CGL points are the locations of the $n - 1$ extrema of $T_n(x)$ plus the endpoints of the interval $[-1, 1]$. (chebyshevInterpolation.m, chebyshevInterpolation2d.m)

2.2 Fourier (Trigonometric) Interpolation

The degree $2N$ Fourier approximation (fourierInterpolation.m, fourierInterpolation2d.m) method uses evenly spaced interpolation sites

$$x_k = -1 + \frac{2}{N} k, \quad k = 0, 1, \ldots, 2N - 1 \tag{5}$$

on $[-1, 1]$. In (2) the index k runs over $k = -N, -N + 1, \ldots, N$ and the basis functions are the trigonometric polynomials

$$\phi_k(x) = e^{ik\pi x}. \tag{6}$$

The expansion coefficients are efficiently calculated via the FFT.

3 Edge Detection

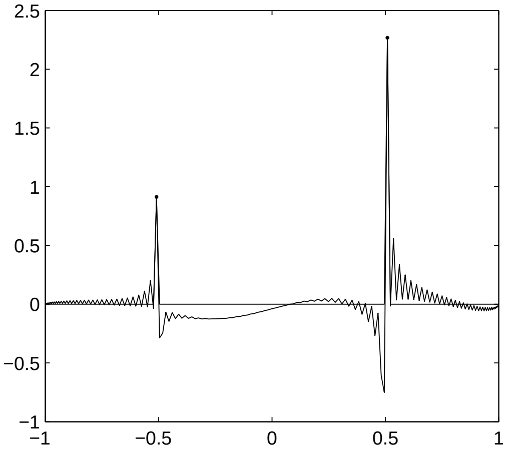


Figure 2: Un-enhanced and enhanced edge detection data produced by edgeDetectChebyshev_example.m using the exponential concentration factor. The following edge detection parameters were used: $Q = 2$, $J = 30$, and $\eta = 2$.

The majority of postprocessing algorithms either require or may incorporate the exact location of discontinuities, or edges, in the function. Edge detection methods have been developed in references [10, 11]. Two choices of concentration factors are available in the edge detection routines, a linear concentration factor of $\sigma(\xi) = \xi$ and an exponential concentration factor

$\sigma(\xi) = \xi \exp(\frac{1}{6\xi(\xi-1)})$. Details on concentration factors can be found in references [10] and [11].

The edges are located by examining a weighted derivative of the spectral interpolant

$$ue(x) = w\frac{d}{dx}\mathcal{I}_N f(x) \tag{7}$$

where the weight is $w = 1/N$ in the Fourier case and $w = \pi\sqrt{1-x^2}/N$ for the Chebyshev case. Denoting the location of discontinuities as α_j and defining jumps as

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

the convergence of $ue(x)$ to the location of the discontinuities may be described as

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

While a graphical examination of $ue(x)$ verifies that it does have the desired convergence properties, an additional step is needed to numerically pinpoint the location of the discontinuities. For that purpose, a non-linear enhancement [11] is made to $ue(x)$ as

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

The enhanced sum has the convergence properties

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

By choosing $Q > 1$, the separation is enhanced between the $O([\frac{1}{N}]^{\frac{Q}{2}})$ points of smoothness and the $O(N^{\frac{Q}{2}})$ points of discontinuity. The problem dependent threshold parameter J is then used to pinpoint the location of all jumps and the edges are located by redefining $ue(x)$ as

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

Computational experience [32] has lead to the inclusion of an additional parameter η which controls the number of edges that can be found in the neighborhood of a local maximum of $ue(x)$. If the maximum occurs at $x(i)$, then the parameter allows only one edge to be found in the interval $(x[i-\eta], x[i+\eta])$, $i = 0, ..., N$.

Matlab files: (Note: the inputs and outputs of all the MPT functions are documented in the m-files)

1. edgeDetectChebyshev.m
2. edgeDetectFourier.m
3. edgeDetectChebyshev_example.m - The example applies the edge detection routine to the Chebyshev approximation of the function

   $$f(x) = \begin{cases} \cos(\pi x/2) & x < -0.5 \\ x^3 - \sin(3\pi x/2) & -0.5 \leq x \leq 0.5 \\ x^2 + 4x^3 - 5x & x > 0.5 \end{cases} \tag{8}$$

   The output is shown in figure 2.

4 Postprocessing Methods

4.1 Spectral Filters

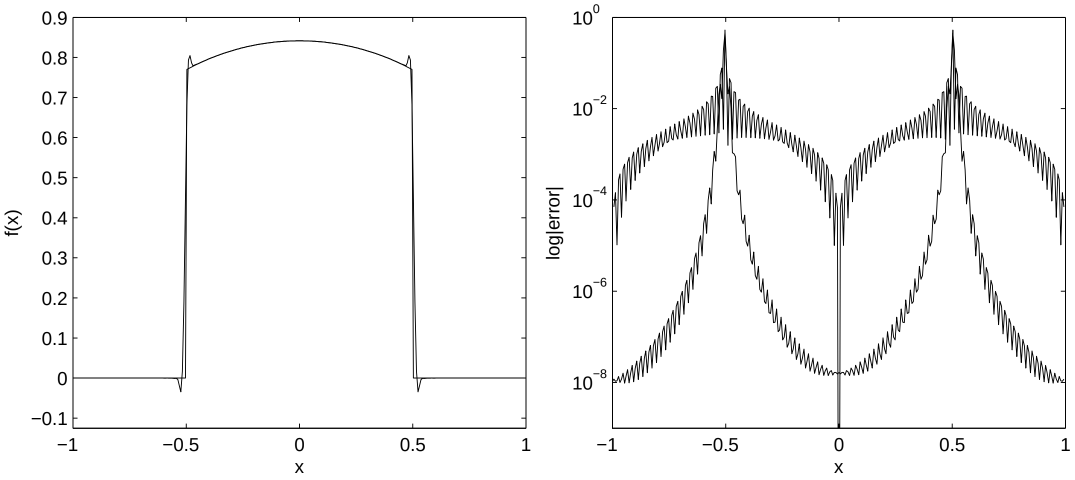


Figure 3: Output produced by spectralFilter_example.m. Left: Function (1), and the filtered Fourier approximation of the function. Right: Error of the Fourier approximation and the reduced error of the filtered approximation.

Spectral filters [43] lessen the effects of the Gibbs phenomenon by working in transform space as

$$\mathcal{F}_N f(x) = \sum_k \sigma(k/N)\, a_k\, \phi_k(x) \tag{9}$$

The convergence rate of the filtered approximation is determined solely by the order, $\rho > 1$, of the filter and the regularity of the function away from the point of discontinuity. If the filter order, ρ, is chosen increasing with N, the filtered expansion recovers exponential accuracy away from a discontinuity. Assuming that $f(x)$ has a discontinuity at x_0 and setting $d(x) = x - x_0$, the estimate

$$|f(x) - \mathcal{F}_N(x)| \leq \frac{K}{d(x)^{\rho-1} N^{\rho-1}} \tag{10}$$

holds where K is a constant. If ρ is sufficiently large, and $d(x)$ is not too small, the error goes to zero faster than any finite power of N, i.e. spectral accuracy is recovered. When x is close to a discontinuity the error increases. If $d(x) = \mathcal{O}(1/N)$ then the error estimate is $\mathcal{O}(1)$.

The following ρ^{th} order spectral filters are implemented in the MPT:

1. exponential filter
$$\sigma_1(\omega) = e^{(\ln \varepsilon_m)\, \omega^\rho}, \tag{11}$$
where ρ even and ε_m represents machine zero.

2. Erfc-Log filter [2]
$$\sigma_2(\omega) = \frac{1}{2}\,\mathrm{erfc}\left(2\sqrt{\rho}\,[|\omega| - 1/2]\sqrt{\frac{-\ln\left(1 - 4\left[|\omega| - 1/2\right]^2\right)}{4\left[|\omega| - 1/2\right]^2}}\right) \tag{12}$$

3. Vandeven filter [43]
$$\sigma_3(\omega) = 1 - \frac{(2\rho - 1)!}{(p-1)!}\int_0^{|\omega|} t^{\rho-1}(1-t)^{\rho-1} dt \tag{13}$$

Matlab files:

1. spectralFilter.m

2. spectralFilter_example.m - The example calculates the Fourier approximation of the function (1) and the filtered approximation using the fourth-order Vandeven filter. The output is shown in figure 3. The right image shows the increased accuracy of the filtered approximation away from the discontinuity. The oscillatory Fourier approximation is in the left image of figure 1.

3. filterFourier.m

4. filterFourier2d.m

5. filterChebyshev.m
6. filterChebyshev2d.m

4.2 Digital Total Variation Filtering

The Rudin, Osher, and Fatemi (ROF) Total Variation (TV) denoising model is a popular image processing method to remove noise from a digital image. The model formulates a minimization problem which leads to a nonlinear Euler-Lagrange PDE to be solved by numerical PDE methods. In [5, 28] the authors develop a discrete version of the TV model on a graph - Digital Total Variation (DTV) filtering. Viewing an oscillatory function as an image with noise, the DTV method was used to postprocess spectral approximations in [33] and Radial Basis Function approximations in [34]. The method works with point values in physical space and not with the spectral expansion coefficients. The DTV method does not need to know the location of edges. The point values may be located at scattered, non-structured sites, in complex geometries. The DTV method is very computationally efficient. While the method does mitigate the effects of the Gibbs phenomenon it does not make any claims of restoring spectral accuracy.

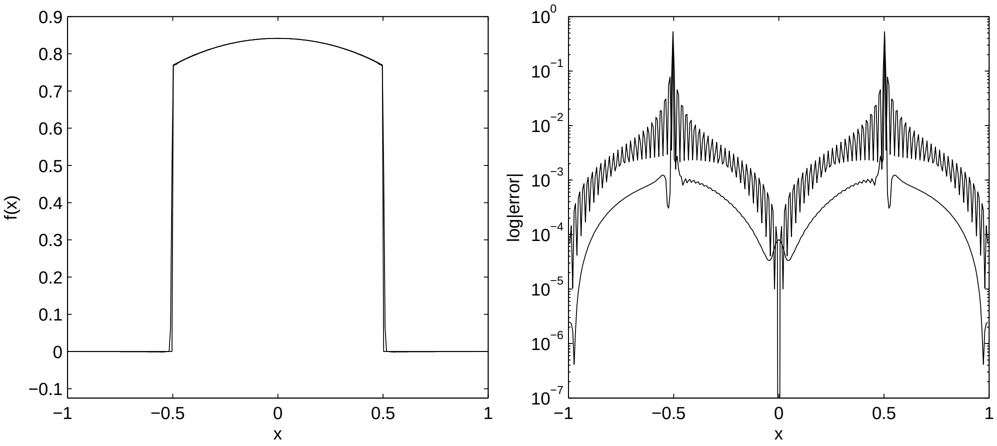


Figure 4: Output produced by dtvFilter_example.m. Left: Function (1) and the DTV postprocessed Fourier approximation. Right: Error of the Fourier approximation and the reduced error of the DTV postprocessed approximation.

To summarize the method, let $[\Omega, G]$ be a finite set Ω of nodes and a dictionary of edges G connecting the nodes. General vertices are denoted by $\alpha, \beta, \cdots$. The notation $\alpha \sim \beta$ indicates that α and β are linked by an edge. All the neighbors of α are denoted by

$$N_\alpha = \{\beta \in \Omega \,|\, \beta \sim \alpha\}. \tag{14}$$

The graph variational problem is to minimize the fitted TV energy

$$E_\lambda^{TV}(u) = \sum_{\alpha \in \Omega} |\nabla_\alpha u|_a + \frac{\lambda}{2} \sum_{\alpha \in \Omega} (u_\alpha - u_\alpha^0)^2 \tag{15}$$

where u^0 is the spectral approximation containing the Gibbs oscillations and λ the user specified fitting parameter. The unique solution to this problem is the solution of the nonlinear restoration equation

$$\sum_{\beta \sim \alpha} (u_\alpha - u_\beta) \left(\frac{1}{|\nabla_\alpha u|_a} + \frac{1}{|\nabla_\beta u|_a} \right) + \lambda (u_\alpha - u_\alpha^0) = 0 \tag{16}$$

where the regularized location variation or strength function at any node α is defined as

$$|\nabla_\alpha u|_a = \left[\sum_{\beta \in N_\alpha} (u_\beta - u_\alpha)^2 + a^2 \right]^{1/2}. \tag{17}$$

The regularization parameter a is a small (the default in the software is $a = 0.0001$) value used to prevent a zero local variation and division by zero.

The nonlinear system can be solved by a linearized Jacobi iteration as was done in [5, 33, 34]. Alternatively, we can work with the nonlinear restoration equation (16) and use time marching to reach a steady state

$$\frac{du_\alpha}{dt} = \sum_{\beta \sim \alpha} (u_\alpha - u_\beta) \left(\frac{1}{|\nabla_\alpha u|_a} + \frac{1}{|\nabla_\beta u|_a} \right) + \lambda (u_\alpha - u_\alpha^0). \tag{18}$$

Preconditioning equation (18)

$$\frac{du_\alpha}{dt} = \sum_{\beta \sim \alpha} (u_\alpha - u_\beta) \left(1 + \frac{|\nabla_\alpha u|_a}{|\nabla_\beta u|_a} \right) + \lambda \, |\nabla_\alpha u|_a \, (u_\alpha - u_\alpha^0). \tag{19}$$

yields a faster convergence to the steady state [28]. The software uses time marching with the explicit Euler's method. Typically about 100 time steps are required to approach a steady state. An optimal value of the fitting parameter λ is not known. However, a large range of values for the fitting parameter results in a "good" postprocessing. In general, stronger oscillations are best handled with a small fitting parameter (< 10) while weaker oscillations require a larger value of the fitting parameter. More details on selecting the value of the shape parameter can be found in references [33] and [35].

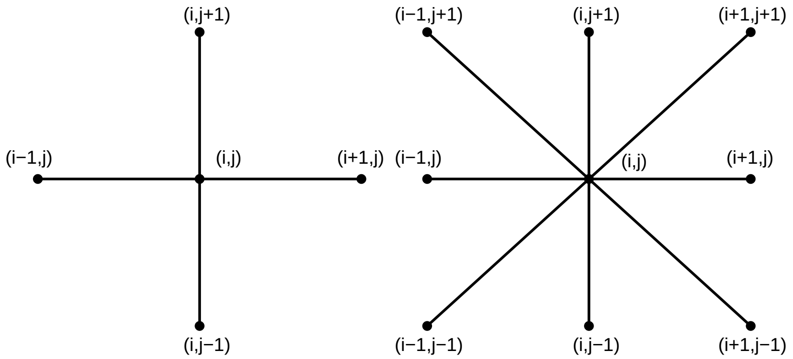


Figure 5: 2d DTV neighborhoods: Left, N_α^4. Right: N_α^8

In two space dimensions there is more than one way to define N_α (figure 5). One is to consider at a node $\alpha_{i,j}$ four neighboring points, $N_\alpha^4 = \{\alpha_{i,j+1}, \alpha_{i+1,j}, \alpha_{i,j-1}, \alpha_{i-1,j}\}$ and another is an eight point neighborhood, $N_\alpha^8 = \{\alpha_{i,j+1}, \alpha_{i+1,j+1}, \alpha_{i+1,j}, \alpha_{i+1,j-1}, \alpha_{i,j-1}, \alpha_{i-1,j-1}, \alpha_{i-1,j}, \alpha_{i-1,j+1}\}$.

Matlab files:

1. digitalTotalVariationFilter.m - 1d non-periodic
2. digitalTotalVariationFilterPeriodic.m - 1d periodic
3. digitalTotalVariationFilter_2d.m - 2d non-periodic using N_α^4
4. digitalTotalVariationFilter_2d_8.m - 2d non-periodic using N_α^8
5. digitalTotalVariationFilterPeriodic_2d.m - 2d periodic using N_α^4
6. digitalTotalVariationFilterPeriodic_2d_8.m - 2d periodic using N_α^4
7. fourierDTV_example.m - The example postprocesses the Fourier approximation of function (1) using the DTV filter (digitalTotalVariationFilterPeriodic.m) with $\lambda = 10$ and 100 time steps. The output is shown in figure 4. The oscillatory Fourier approximation is in the left image of figure 1.

8. periodic_dtv_example_2d.m - The example postprocesses the Fourier approximation of the function
$$f(x) = \begin{cases} 1 & x^2 + y^2 \leq \frac{1}{4} \\ 0 & \text{otherwise} \end{cases} \tag{20}$$
using the DTV filter (digitalTotalVariationFilterPeriodic.m) with $\lambda = 10$ and 100 time steps. The output is shown in figure 6. The function is sampled on a 64×64 grid and evaluated on a uniformly spaced 98×98 grid.

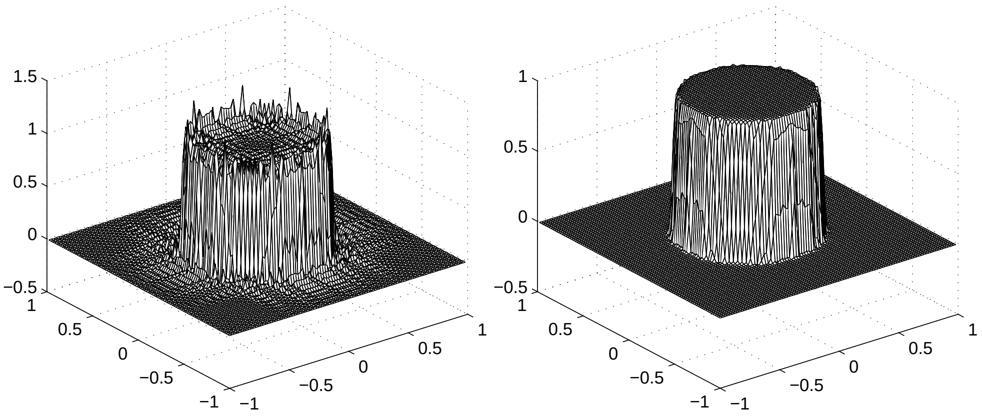


Figure 6: Output produced by periodic_dtv_example_2d.m. Left: Fourier approximation of function (20). Right: The DTV postprocessed Fourier approximation using digitalTotalVariationFilterPeriodic_2d.m.

4.3 Rational Reconstruction

Rational functions have been used in several different forms to reduce the Gibbs phenomenon [7, 8, 21, 27]. Rational functions are more complex than simple polynomials and often do better in approximation discontinuous functions or functions with steep gradients.

A Padé approximant is of the form
$$R_{K,M} = \frac{P_K}{Q_M} = \frac{\sum_{k=0}^{K} p_k \phi_k(x)}{\sum_{m=0}^{M} q_m \phi_m(x)} \tag{21}$$
The linear Padé approximation of a function u is determined by imposing the orthogonality relations
$$\langle Qu - P, \phi \rangle = 0, \qquad \forall \phi \in \mathbb{P}_N \tag{22}$$

This results in a linear system with $M+1$ unknowns and $K-N$ equations. After the degree of the denominator M is chosen the degree of the numerator is set as $K=(N-N_c)-M$ where $N_c <= N$ is a cutoff value that specifies how many of the high order expansion coefficients are not used to form the Padé approximation. This is necessary since a significant portion of the high modes are polluted when discontinuous functions are approximated [21, 27]. This results in a linear system with one more unknown than equation. In the Chebyshev case (chebyshevPade.m based on [7]) this is resolved by setting $q_0 = 1$ which leaves a $M \times M$ linear system to be solved for $q_1, \ldots, q_M$. In the Fourier case (fourierPade.m) we have followed the method outlined in [27].

Matlab files:

1. fourierPade.m

2. chebyshevPade.m

3. chebyshevPade_example.m - Figure 7 shows the output of the chebyshevPade_example.m which applies chebyshevPade.m with $M = 14$ to postprocess function (1). The oscillatory Chebyshev approximation is in the right image of figure 1.

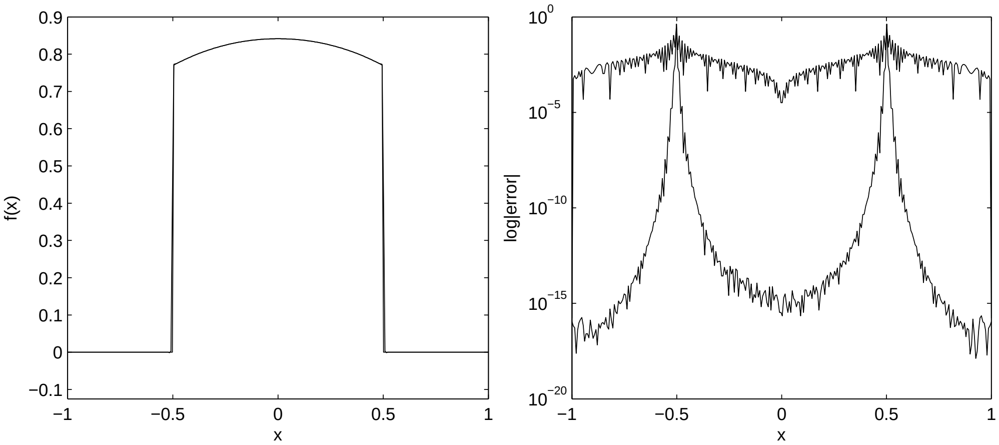


Figure 7: Output produced by chebyshevPade_example.m. Left: Function (1), and the postprocessed approximation of the function. Right: Error of the Chebyshev approximation and the reduced error of the postprocessed approximation.

The rational reconstruction code of the MPT does not incorporate edge detection. However, if we are to recover spectral accuracy at the point of a discontinuity [8] or to accurately postprocess computational data from a PDE problem, it will be necessary to use the locations of discontinuities. A discussion of how to incorporate edge detection into the rational reconstruction algorithms and numerical examples can be found in reference [21].

4.4 Reprojection Methods

Reprojection methods take the spectral projection and project in onto another basis. In the new basis, spectral accuracy is recovered. Pinpointing the exact location of discontinuities is crucial to the success of the methods. Let $\xi(x)$ be the map that takes $x \in [a, b]$ to $\xi \in [-1, 1]$ and let $x(\xi)$ be the inverse of the map. In each of the i smooth subintervals $[a, b]$ the function is reprojected as

$$f_P^i(x) = \sum_{\ell=0}^{m_i} g_\ell^i \Psi_\ell[\xi(x)] \tag{23}$$

onto a basis $\Psi_\ell(x)$ of polynomials, the reprojection basis, which are orthogonal on $[-1, 1]$ with respect to a weight function $w(x)$ under the weighted inner product $(\Psi_k(\xi), \Psi_k(\xi))_w$ which satisfies

$$(\Psi_k(\xi), \Psi_\ell(\xi))_w = \int_{-1}^{1} \Psi_k(\xi)\Psi_\ell(\xi)w(\xi)d\xi = \gamma_\ell \delta_{k\ell} \tag{24}$$

where γ_ℓ is a normalization factor. The expansion coefficients g_ℓ^i are evaluated via a Gaussian quadrature formula. If the function being approximated is known at the quadrature points, g_ℓ^i are referred to as the exact reprojection coefficients. Otherwise the spectral interpolant (2) is used to approximate the function at the quadrature points and the coefficients are referred to as the approximate reprojection coefficients $\hat{g}_\ell^i$.

4.4.1 Gegenbauer Reprojection

The Gegenbauer Reprojection Procedure (GRP) uses the Gegenbauer or Ultraspherical polynomials C_ℓ^λ as the reprojection basis. The GRP was developed in the series of papers [18, 13, 16, 14, 15, 17]. Further analysis and application of the GRP can be found in references [1, 22, 30, 31]. The Gegenbauer polynomials satisfy the conditions of a Gibbs complementary basis [20] which allows for a spectrally accurate reprojection. The weight function associated with the Gegenbauer polynomials is $w(x) = (1 - \xi^2)^{\lambda - 1/2}$. The

Gegenbauer polynomials (gegenbauerPolynomial.m) are calculated via the three term recurrence relation

$$C_{k+1}^{\lambda}(\xi) = \frac{2(k+\lambda)\xi}{k+1} C_k^{\lambda}(\xi) - \frac{k+2\lambda-1}{k+1} C_{k-1}^{\lambda}(\xi), \qquad k = 2, 3, \ldots \tag{25}$$

with $C_0^{\lambda} = 1$ and $C_1^{\lambda} = 2\lambda\xi$.

In smooth subinterval i the GRP postprocessed approximation is

$$f_P^i(x) = \sum_{\ell=0}^{m_i} g_\ell^i C_\ell^{\lambda}[\xi(x)] \tag{26}$$

where the exact Gegenbauer expansion coefficients are

$$g_\ell^i = \frac{1}{\gamma_\ell^{\lambda}} \int_{-1}^{1} (1-\xi^2)^{\lambda-1/2} C_\ell^{\lambda}(\xi) f[x(\xi)] d\xi \tag{27}$$

and

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

If only the expansion coefficients a_k are known and not the underlying function, as in a pseudospectral PDE approximation, the Gegenbauer coefficients are replaced with the approximate Gegenbauer coefficients

$$\hat{g}_\ell^i = \frac{1}{\gamma_\ell^{\lambda}} \int_{-1}^{1} (1-\xi^2)^{\lambda-1/2} C_\ell^{\lambda}(\xi) \mathcal{I}_N f[x(\xi)] d\xi. \tag{28}$$

The integrals in (27) and (28) are evaluated via Chebyshev-Gauss-Lobatto quadrature.

Matlab files:

1. grp.m

2. fourierGRP_example.m - see figure 8. The example has three smooth subintervals $\Omega_1 = [-1, -0.5)$, $\Omega_2 = (-0.5, 0.5)$, and $\Omega_3 = (0.5, 1]$. The example used the following GRP parameters: $m_1 = m_2 = m_3 = 13$, $\lambda_1 = 4$, $\lambda_2 = 7$, and $\lambda_3 = 4$. The oscillatory Fourier approximation is in the left image of figure 1.

3. chebyshevSV_eulerDensity_grp.m - Applies the GRP to postprocess the $N = 128$ Chebyshev spectral viscosity approximation of the density profile of Sod's problem [37] for the Euler equations of gas dynamics. The output is in figure 9. The example has five smooth intervals:

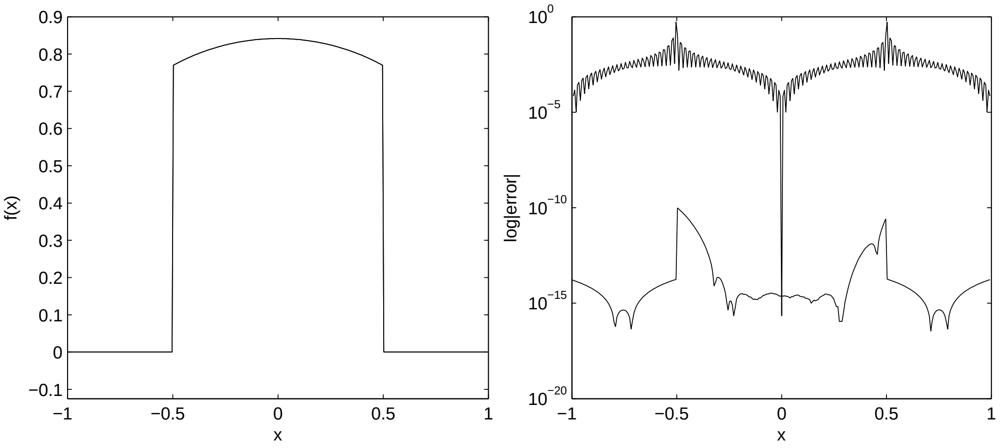


Figure 8: Output produced by fourierGRP_example.m. Left: Function (1) and the GRP postprocessed Fourier approximation (visually indistinguishable). Right: Error of the Fourier approximation and the reduced error of the postprocessed GRP approximation.

$\Omega_1 = [-1, -0.4456)$, $\Omega_2 = (-0.4456, -0.0222)$, $\Omega_3 = (-0.0222, 0.3461)$, $\Omega_4 = (0.3461, 0.6606)$, and $\Omega_5 = (0.6606, 1]$. The following GRP parameters were used in the example: $m_1 = 4$, $m_2 = 12$, $m_3 = 4$, $m_4 = 4$, $m_5 = 4$, $\lambda_1 = 5$, $\lambda_2 = 6$, $\lambda_3 = 6$, $\lambda_4 = 4$, and $\lambda_5 = 4$.

4.4.2 Freud Reprojection

The Freud Reprojection Procedure (FRP) [9, 12] uses the Freud polynomials ψ as the reprojection basis and the weight function is $w(\xi) = e^{-c\xi^{2\lambda}}$ where $\lambda = \alpha N$, $0 < \alpha < 1$, and $c = \ln \epsilon_M$ where ϵ_M is machine epsilon. We have used $\lambda = \text{round}\left(\sqrt{N(b-a)/2} - 2\sqrt{(2)}\right)$ which was suggested in [12].

In [12] an additional condition is added to the three conditions that a Gibbs complementary basis must satisfy. A basis that satisfies the four conditions is called a Robust Gibbs complement. The Freud polynomial basis is an example of a Robust Gibbs complement. Freud reprojection does not suffer from the numerical roundoff errors and the Runge phenomenon that the GRP does [1]. The Freud polynomials are not known explicitly but can be computed recursively as

$$\psi_{k+1}(\xi) = \xi\psi_k(\xi) - \frac{\gamma_k}{\gamma_{k-1}}\psi_{k-1}(\xi)$$

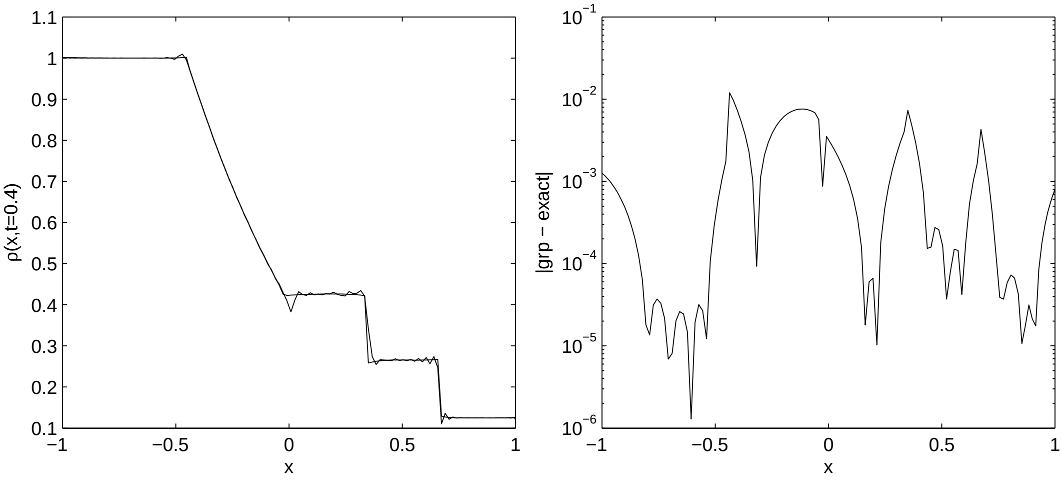


Figure 9: Output of chebyshevSV_eulerDensity_grp.m. Left: Chebyshev spectral viscosity and GRP postprocessed Euler density solution at time t = 0.4. Right: Error of the postprocessed solution.

where $\psi_0(\xi) = 1$ and $\psi_1(\xi) = \xi$. The recursion coefficients are

$$\gamma_k = (\psi_k(\xi), \psi_k(\xi))_w = \int_{-1}^{1} \psi_k(\xi)\psi_k(\xi) e^{-c\xi^{2\lambda}} d\xi. \tag{29}$$

The exact Freud coefficients are

$$g_\ell^i = \frac{1}{\gamma_\ell} \int_{1}^{-1} e^{-c\xi^{2\lambda}} \psi_\ell(\xi) f[x(\xi)] d\xi. \tag{30}$$

and the approximate Freud coefficients are

$$\hat{g}_\ell^i = \frac{1}{\gamma_\ell} \int_{1}^{-1} e^{-c\xi^{2\lambda}} \psi_\ell(\xi) \mathcal{I}_N f[x(\xi)] d\xi. \tag{31}$$

Integrals (29), (30), and (31) are evaluated very accurately using the trapezoid rule which is exponentially accurate for smooth periodic functions. In smooth subinterval i the FRP approximation is

$$f_P^i(x) = \sum_{\ell=0}^{m_i} g_\ell^i \psi_\ell[\xi(x)]. \tag{32}$$

In each subinterval of smoothness m_i is set $m_i = N(b-a)/8$. However, as N increases the number of terms in the reprojection basis m_i is more than is necessary to numerically resolve the function and the higher numbered reprojection coefficients become very close to machine epsilon which leads to round-off errors. As described on p. 15 of [12], the round-off errors can be avoided by resetting m_i to a value that prevents the average of three consecutive reprojection coefficients from being larger than a specified tolerance. Experimentally we have found a tolerance of $10e-12$ to work well in the Fourier case and $10e-8$ to work well with Chebyshev approximations.

For the FRP, the specification of M is not function-dependent as is the case for the GRP. The FRP does not have any function-dependent parameters to be supplied by the user. This is in contrast to the GRP which depends heavily on the proper specification of both a weight parameter λ and reprojection order M.

Matlab files:

1. frp.m
2. freudPolynomials.m
3. fourierFreud_example.m - The script uses the FRP to postprocess the oscillatory Fourier approximation in the left image of figure 1. The results are shown in figure 10. The following FRP parameters were used: $m_1 = 2$, $m_2 = 12$, $m_3 = 2$, $\lambda_1 = 7$, $\lambda_2 = 11$, and $\lambda_3 = 7$.

4.4.3 Inverse Reprojection

The Gegenbauer and Freud reprojection methods are referred to as direct methods as they compute the reprojection coefficients directly from the spectral expansion coefficients a_k (or function values). In contrast, inverse methods compute the reprojection coefficients by solving a linear system of equations

$$Wg = a.$$

The Inverse Reprojection method was developed in [23, 24, 36]. Originally [36], the Gegenbauer polynomials were used as the reprojection basis, but later [23, 24] the method was generalized to yield a unique reconstruction using any set of basis functions. The generalized method is referred to as the inverse polynomial reconstruction method (IPRM). We have implemented the method using the Gegenbauer polynomials as the reconstruction basis. Note that since the IPRM uniquely determines the reconstruction for any reconstruction basis, the Gegenbauer parameter λ does not play a role in the method as it does in the GRP method. The matrix W may be very

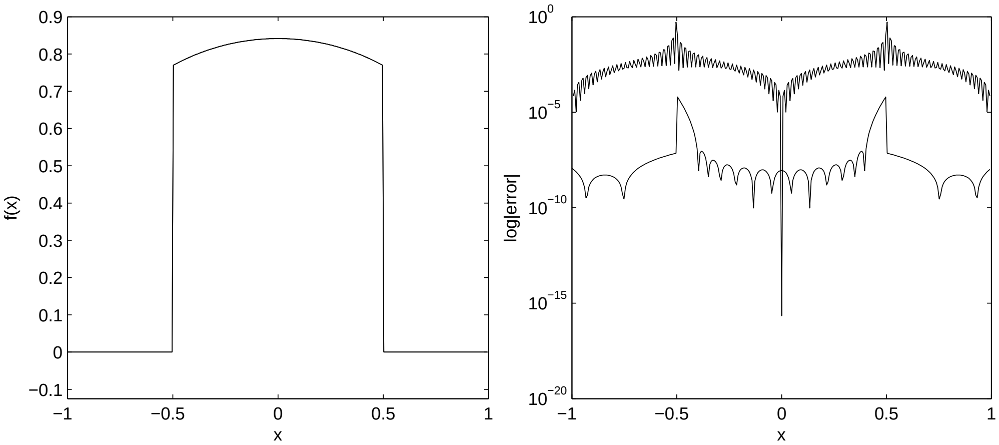


Figure 10: Output produced by fourierFreud_example.m. Left: Function (1) and the FRP postprocessed Fourier approximation (visually indistinguishable). Right: Error of the Fourier approximation and the reduced error of the postprocessed FRP approximation.

ill-conditioned. The conditioning of W is best for small $\lambda > 0$ [36]. In the computer code the default is $\lambda = 1/2$ which corresponds to the Legendre basis. Care has been taken to consistently evaluate the spectral expansion coefficients and the integrals that determine the elements of the matrix W as is discussed in [23]. Both are evaluated using Gaussian quadrature.

The main weakness of the IPRM is that an ill-conditioned linear system has to be solved. This problem is addressed in reference [25]. To date, the IPRM method has not been successfully applied to spectral viscosity solutions of hyperbolic conservation laws.

Matlab files:

1. inverseReprojection.m

2. fourierIPRM_example.m - This script uses the IPRM to postprocess the oscillatory Fourier approximation in the left image of figure 1. The output is shown in figure 11. The following IPRM parameters were used: $m_1 = 2$, $m_2 = 6$, and $m_3 = 2$.

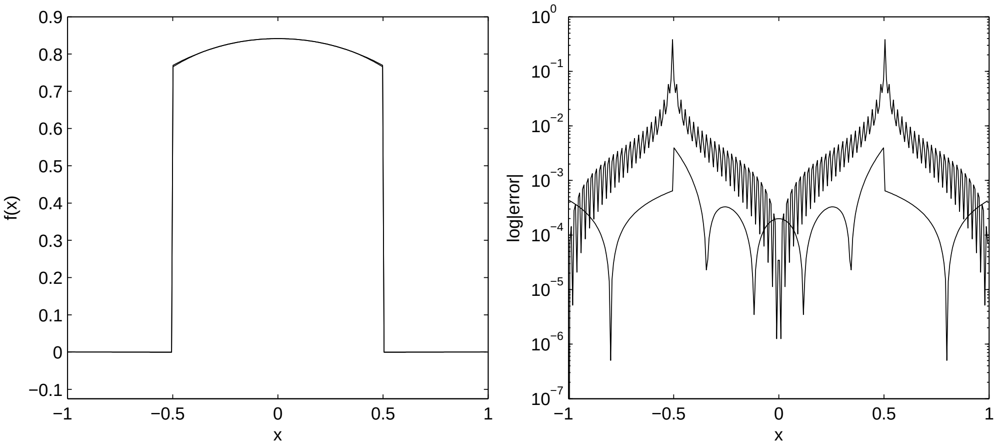


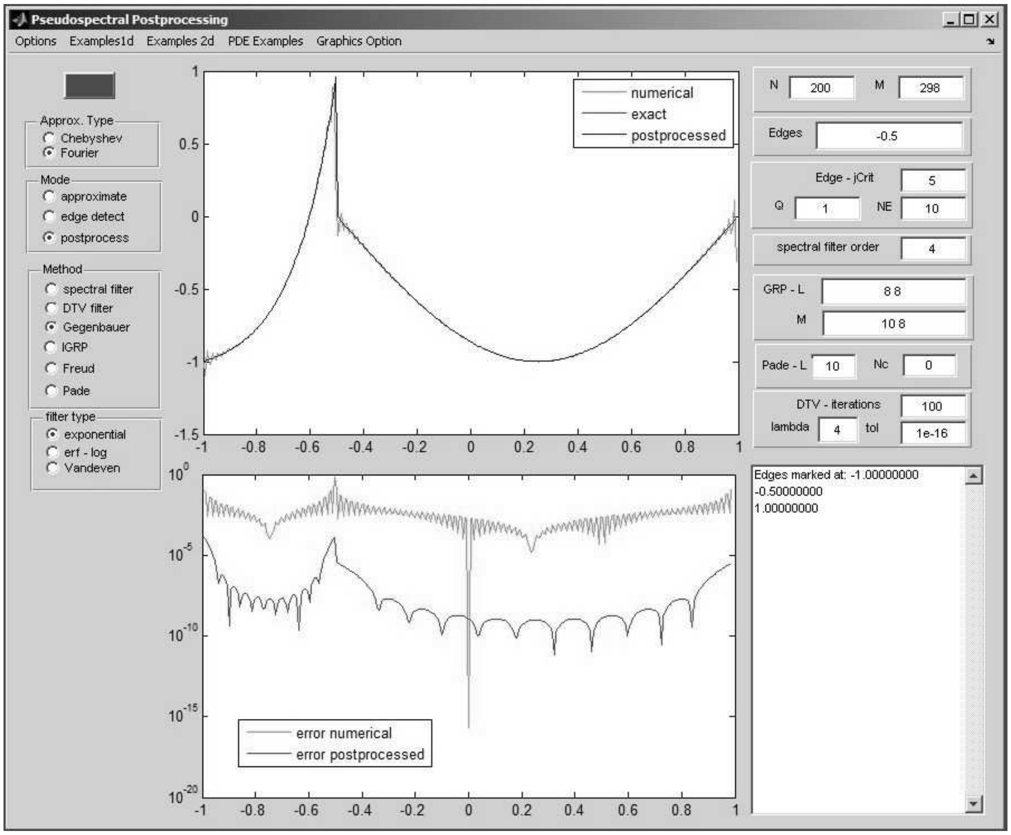
Figure 11: Output produced by fourierIPRM_example.m. Left: Function (1) and the IPRM postprocessed Fourier approximation. Right: Error of the Fourier approximation and the reduced error of the postprocessed IPRM approximation.

5 Graphical User Interface

The Matlab functions of the MPT may be called directly as we have done in the included examples. Additionally, to make the MPT functions more accessible to non-Matlab users a graphical user interface (GUI) has been developed. The GUI has built-in functions and pseudospectral PDE solutions that can be used to demonstrate and benchmark the algorithms. The Fourier PDE examples include linear advection and inviscid Burger's equation and the Chebyshev examples include linear advection and the Euler equations of gas dynamics. It is beyond the scope of this work to document the GUI. A brief users guide to the GUI is in the file mptGuiGuide.pdf. Updates and more details on the GUI will appear at www.scottsarra.org/mpt/mpt.html as they are available. A screen shot of the GUI is shown in figure 12.

6 Concluding Comments

We have described a suite of Matlab programs that implement state-of-the-art postprocessing and edge detection algorithms for Fourier and Chebyshev spectral approximations of piecewise smooth functions in one and two space dimensions. Postprocessing methods that require one or more user defined parameters be specified in each smooth subregion are difficult to implement



Approx. Type
○ Chebyshev
● Fourier

Mode
○ approximate
○ edge detect
● postprocess

Method
○ spectral filter
○ DTV filter
● Gegenbauer
○ IGRP
○ Freud
○ Pade

filter type
● exponential
○ erf - log
○ Vandeven

N 200 M 298

Edges -0.5

Edge - jCrit 5
Q 1 NE 10

spectral filter order 4

GRP - L 8 8
M 10 8

Pade - L 10 Nc 0

DTV - iterations 100
lambda 4 tol 1e-16

```
Edges marked at: -1.00000000
-0.50000000
1.00000000
```

Figure 12: Graphical user interface

in two space dimensions. For this reason, the MPT only implements spectral filtering and DTV filtering in two dimensions. Although not the most powerful one dimensional methods, they are very computationally efficient and the closest to “black box” algorithms in two dimensions. In two dimensional applications, many of the one dimensional methods can been applied to one dimensional slices in the x or y direction. This approach has been taken in references [24] and [27].

The MPT functions may be used as in our examples by calling them from a Matlab script. Alternatively, the routines may be accessed from a GUI. The postprocessing functions and accompanying GUI with built-in example functions and PDE solutions provide users the opportunity to benchmarch and demonstrate the postprocessing algorithms. Experienced Matlab users will find it easy to modify the GUI to incorporate their own algorithms or example problems. The home of the MPT on the web is www.scottsarra.org/mpt/mpt.html. The development of the MPT is ongoing and modifications and extensions will be made as new algorithms are developed.

We conclude with table 6 that summarizes the basic feature of the postprocessing algorithms.

	edge		spectral	large	
method	detection	parameters	accuracy	$\kappa(A)$	noise
spectral filter		ρ	$\sim$		$\surd$
DTV		λ			$\surd$
Padé	$\sim$	M, N_c		$\surd$	$\surd$
GRP	$\surd$	λ, m_i	$\surd$		$\surd$
FRP	$\surd$		$\surd$		$\surd$
IPRM	$\surd$	m_i	$\surd$	$\surd$	

A $\surd$ in the edge detection column indicates the method must know the exact location of the discontinuities while a $\sim$ indicates that the edge location may be incorporated to improve the method. The parameters column lists any user specified parameters. If the method incorporates edge detection the parameters must be specified in each subinterval of smoothness. A $\surd$ in the spectral accuracy column indicates the method is able to recover spectral accuracy over the entire interval while a $\sim$ indicates that spectral accuracy may be recovered over a portion of the interval sufficiently away from the edge locations. A $\surd$ in the large $\kappa(A)$ column indicates that a linear system must be solved to implement the method and that the matrix may have a large condition number. Finally, a $\surd$ in the noise column indicates that the method can use noisy spectral coefficients. The inexact, noisy spectral coefficients may result from a spectral viscosity approximation to a time-dependent nonlinear hyperbolic PDE problem in which time integration errors and the addition of spectral viscosity have altered the solution.

References

[1] J. P. Boyd. Trouble with Gegenbauer reconstruction for defeating Gibbs' phenomenon: Runge phenomenon in the diagonal limit of Gegenbauer polynomial approximations. *Journal of Computational Physics*. 4.4.1, 4.4.2

[2] J. P. Boyd. The Erfc-Log filter and the asymptotics of the Vandeven and Euler sequence accelerations. *Houston Journal of Mathematics*, pages 267–275, 1996. 2

[3] J. P. Boyd. *Chebyshev and Fourier Spectral Methods.* Dover, second edition, 2000. 2

[4] C. Canuto, M. Hussaini, A. Quarteroni, and T. Zang. *Spectral Methods: Fundamentals in Single Domains.* Springer, 2006. 2

[5] T. Chan, S. Osher, and J. Shen. The digital TV filter and nonlinear denoising. *IEEE Transactions on Image Processing*, 10(2), 2001. 4.2, 4.2

[6] Q. Chen, D. Gottlieb, and J. Hesthaven. Spectral methods based on prolate spheroidal wave functions for hyperbolic PDEs. *SIAM Journal on Numerical Analysis*, 43(5):1912–1933, 2005. 1

[7] C. W. Clenshaw and K. Lord. Rational approximations from chebyshev series. In B.K.P. Scaife, editor, *Studies in Numerical Analysis*, page 95113. Academic Press, 1974. 4.3, 4.3

[8] T. Driscoll and B. Fornberg. A Padé-based algorithm for overcoming the Gibbs phenomenon. *Numerical Algorithms*, 26:77–92, 2001. 4.3, 4.3

[9] A. Gelb. Reconstruction of piecewise smooth functions from non-uniform grid data. *Journal os Scientific Computing*, 30(3):409–440, 2007. 4.4.2

[10] A. Gelb and E. Tadmor. Detection of edges in spectral data. *Applied and Computational Harmonic Analysis*, 7:101–135, 1999. 3

[11] A. Gelb and E. Tadmor. Detection of edges in spectral data II: Nonlinear enhancement. *SIAM Journal of Numerical Analysis*, 38(4):1389–1408, 2000. 3, 3

[12] A. Gelb and J. Tanner. Robust reprojection methods for the resolution of the Gibbs phenomenon. *Applied and Computational Harmonic Analysis*, 20:3–25, 2006. 1, 4.4.2, 4.4.2

[13] D. Gottlieb and C.-W. Shu. Resoulution properties of the Fourier method for discontinuous waves. *Comput. Methods Appl. Mech. Engrg.*, 116:27–37, 1994. 4.4.1

[14] D. Gottlieb and C.-W. Shu. On the Gibbs phenomenon IV: Recovering exponential accuracy in a subinterval from a Gegenbauer partial sum of a piecewise analytic function. *Mathematics of Computation*, 64:1081–1095, 1995. 4.4.1

[15] D. Gottlieb and C.-W. Shu. On the Gibbs phenomenon V: Recovering exponential accuracy from collocation point values of a piecewise analytic function. *Numerische Mathematik*, 71:511–526, 1995. 4.4.1

[16] D. Gottlieb and C.-W. Shu. On the Gibbs phenomenon III: Recovering exponential accuracy in a subinterval from a partial sum of a piecewise analytic function. *SIAM Journal of Numerical Analysis*, 33:280–290, 1996. 4.4.1

[17] D. Gottlieb and C.-W. Shu. On the Gibbs phenomenon and its resolution. *SIAM Review*, 39(4):644–668, 1997. 1, 4.4.1

[18] D. Gottlieb, C.-W. Shu, A. Solomonoff, and H. Vandeven. On the Gibbs phenomenon I: recovering exponential accuracy from the Fourier partial sum of a nonperiodic analytic function. *Journal of Computational and Applied Mathematics*, 43:81–98, 1992. 4.4.1

[19] D. Gottlieb and E. Tadmor. Recovering pointwise values of discontinuous data within spectral accuracy. In E. M. Murman and S. S. Abarbanel, editors, *Progress and Supercomputing in Computational Fluid Dynamics*, pages 357–375. Birkhäuser, Boston, 1985. 1

[20] J. Hesthaven, S. Gottlieb, and D. Gottlieb. *Spectral Methods for Time-Dependent Problems*. Cambridge University Press, 2007. 2, 4.4.1

[21] J. Hesthaven, S. Kaber, and L. Lurati. Padé-Legendre interpolants for Gibbs reconstruction. *Journal of Scientific Computing*, 28(2-3):337–359, 2006. 1, 4.3, 4.3, 4.3

[22] Z. Jackiewicz. Determination of optimal parameters for the Chebyshev-Gegenbauer reconstruction method. *SIAM Journal of Scientific Computing*, 25(4), 2003. 4.4.1

[23] J.-H. Jung and B. Shizgal. Generalization of the inverse polynomial reconstruction method in the resolution of the Gibbs phenomenon. *Journal of Computational and Applied Mathematics*, 172:131–151, 2004. 1, 4.4.3

[24] J.-H. Jung and B. Shizgal. Inverse Polynomial Reconstruction of Two Dimensional Fourier Images. *Journal of Scientific Computing*, 25:367–399, 2005. 4.4.3, 6

[25] J.-H. Jung and B. Shizgal. On the numerical convergece with the inverse polynomial reconstruction method for the resolution of the Gibbs phenomenon. *Journal of Computational Physics*, 224:477–488, 2007. 4.4.3

[26] J. Mason and D. Handscomb. *Chebyshev Polynomials*. CRC, 2003. 2.1

[27] M. S. Min, S. M. Kaber, and W. S. Don. Fourier-Padé approximations and filtering for spectral simulations of an incompressible Boussinesq convection problem. *Mathematics of Computation*, 76:1275–1290, 2007. 1, 4.3, 4.3, 6

[28] S. Osher and J. Shen. Digitized PDE method for data restoration. In G. Anastassiou, editor, *Analytic-Computational Methods in Applied Mathematics*, chapter 16, pages 751–771. Chapman and Hall/CRC, 2000. 4.2, 4.2

[29] R. Reyret. *Spectral Methods for Incompressible Viscous Flow*. Springer, 2002. 2

[30] S. A. Sarra. Chebyshev super spectral viscosity method for a fluidized bed model. *Journal of Computation Physics*, 186(2):630–651, 2003. 4.4.1

[31] S. A. Sarra. Spectral methods with postprocessing for numerical hyperbolic heat transfer. *Numerical Heat Transfer*, 43(7):717–730, 2003. 4.4.1

[32] S. A. Sarra. The spectral signal processing suite. *ACM Transactions on Mathematical Software*, 29(2), 2003. www.scottsarra.org/signal/signal.html. 1, 3

[33] S. A. Sarra. Digital Total Variation filtering as postprocessing for Chebyshev pseudospectral methods for conservation laws. *Numerical Algorithms*, 41:17–33, 2006. 1, 4.2, 4.2, 4.2

[34] S. A. Sarra. Digital Total Variation filtering as postprocessing for Radial Basis Function Approximation Methods. *Computers and Mathematics with Applications*, 52:1119–1130, 2006. 4.2, 4.2

[35] S. A. Sarra. Edge detection free postprocessing for pseudospectral approximations. *Submitted to the Journal of Scientific Computing*, 2008. 4.2

[36] B. Shizgal and J.-H. Jung. Towards the resolution of the Gibbs phenomena. *Journal of Computational and Applied Mathematics*, 161:41–65, 2003. 1, 4.4.3

[37] G. Sod. A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws. *Journal of Computational Physics*, 27:1–31, 1978. 3

[38] E. Tadmor. Filters, mollifiers and the computation of the Gibbs phenomenon. *Acta Numerica*, 16:305–378, 2007. 1

[39] E. Tadmor and J. Tanner. Adaptive mollifers - high resolution recovery of piecewise smooth data from its spectral information. *Foundations of Computational Mathematics*, 2:155–189, 2002. 1

[40] E. Tadmor and J. Tanner. Adaptive filters for piecewise smooth spectral data. *IMA Journal of Numerical Analysis*, 25(4), 2005. 1

[41] J. Tanner. Optimal filter and mollifier for piecewise smooth spectral data. *Mathematics of Computation*, 75(254), 2006. 1

[42] L. N. Trefethen. *Spectral Methods in Matlab*. SIAM, Philadelphia, 2000. 2

[43] H. Vandeven. Family of spectral filters for discontinuous problems. *SIAM Journal of Scientific Computing*, 6:159–192, 1991. 1, 4.1, 3

[44] J.A.C. Weideman. Spectral methods based on nonclassical orthogonal polynomials. *International Series of Numerical Mathematics*, 31:239–251, 1999. 1