Galerkin Methods for PDEs

T.C. Johnson

Engineering Sciences and Applied Mathematics, Northwestern University, Evanston, IL

Abstract

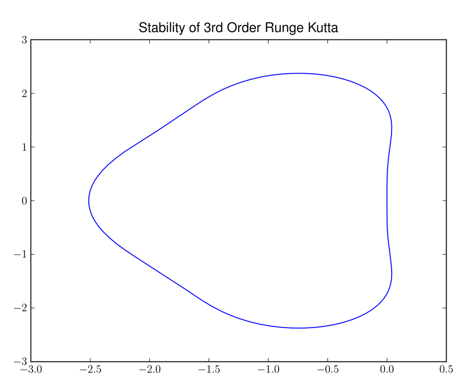
Spectral representations are possible with any number of polynomial approximations of a function. Typically collocation methods are used to minimize programming effort, but more accurate methods known as Galerkin Methods are available. Here we examine Galerkin methods for solving PDEs, comparing them to methods taught in ESAM446-1 and -2.

Goals

1. Explore alternatives to Finite Differences, Fourier, and Chebyshev Differentiation: Galerkin, Nodal Continuous Galerkin, and Nodal Discontinuous Galerkin
2. Build a basis for implementing spectral element methods
3. Write a lot of numerical Python code

Background: Finite Difference Methods and Timestepping Scheme

$$D_x = \begin{pmatrix} 0 & 1 & 0 & \cdots & & 0 \\ -1 & 0 & 1 & 0 & \cdots & \\ & & & & & \\ 0 & \cdots & & -1 & 0 & 1 \\ 0 & \cdots & & & -1 & 0 \end{pmatrix}$$



Williamson's 3^{rd} order, low storage Runge Kutta scheme for wave equation(Stability $\lambda_{\max}\Delta t < -2.51$)

Input: t_n, Δt, u, D
for $i = 1$ to 3 do **do**
  $t \leftarrow t_n + b_m \Delta t$
  $\dot{u} \leftarrow TimeDerivative(u, D)$
  for $j = 0$ to $N-1$ do **do**
    $G_j \leftarrow a_m G_j + \dot{u}$
    $u_j \leftarrow u_j + g_m \Delta t G_j$
  end for
end for
$\dot{u} \leftarrow$ dot(D,F)
return $\dot{u}$

References:

1. Kopriva, Implementing Spectral Methods for Partial Differential Equations
2. Langtangen, Python Scripting for Computational Science
3. LeVeque, Finite Difference Methods for Ordinary and Partial Differential Equations
4. Patera, A spectral element method for fluid dynamics: laminar flow in a channel expansion
5. Recktenwald, Numerical Methods in MATLAB
6. Trefethen, Spectral Methods in MATLAB

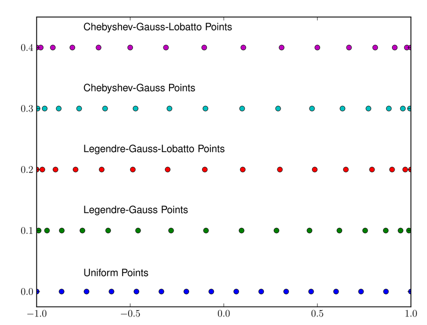
Alternative Basis Sets and Equations

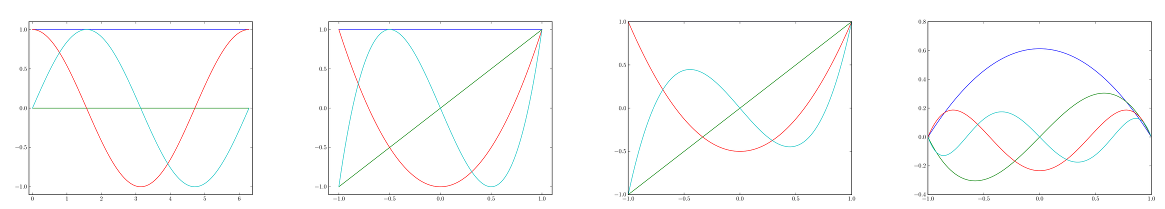
Polynomial Representations

- Fourier:
$$\Phi = \sum \Phi_n e^{inx}$$
- Chebyshev:
$$\Phi = \sum \Phi_n \cos(n \cos^{-1} x)$$
- Legendre:
$$\Phi = \sum \Phi_n \ell(x)$$
- Modified Legendre:
$$\Phi = \sum \Phi_n \frac{L_k(x) - L_{k+2}(x)}{\sqrt{4k+6}}$$

Quadrature nodes and weights

- Legendre-Gauss
- Legendre-Gauss-Lobato
- Chebyshev-Gauss
- Chebyshev-Gauss-Lobato





Weak Forms

Collocation and Galerkin Methods

- $\varphi_t + \varphi_x = \nu\varphi_{xx}$
- $\varphi(x,t) \approx \Phi(x,t) = \sum \Phi_n h_n(x)$
- $\Phi_t + \Phi_x = \nu\Phi_{xx} = 0$
- $\dot{\Phi} + \sum \Phi_n(t) h_n'(x) = \nu \sum \Phi_n(t) h_n''(x)$
- $\dot{\Phi}_j + \mathcal{D}\Phi_j = \nu \mathcal{D}^2 \Phi_j$
- $\dot{\Phi}_j = -\mathcal{D}(\Phi_j - \nu\mathcal{D}\Phi_j)$

- $\varphi_t + \varphi_x = \nu\varphi_{xx}$
- $(\varphi_t, \phi) + (\varphi_x, \phi) = \nu(\varphi_{xx}\phi)$
- $(\varphi_t, \phi) + (\varphi_x, \phi) = \nu\varphi_x\phi|_a^b + \nu(\varphi_x\phi_x)$
- $\varphi = \sum \hat{\Phi}_k \phi_k$

Continuous vs. Noncontinous Galerkin

No need to evaluate inner products analytically; use quadrature

- $\phi = \sum \phi_j \ell(x)$
- $\dot{\Phi}_j w_j + \sum w_k \Phi_k' \ell'(x_k) = 0$
- $\dot{\Phi}_j + \sum \Phi_n \sum w_k \ell'(x_k)\ell'(x_k) = 0$

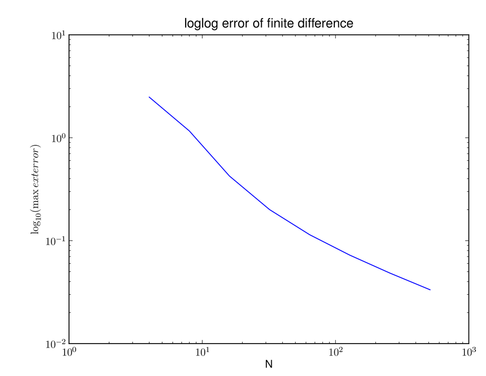
No need to satisfy BCs with test functions

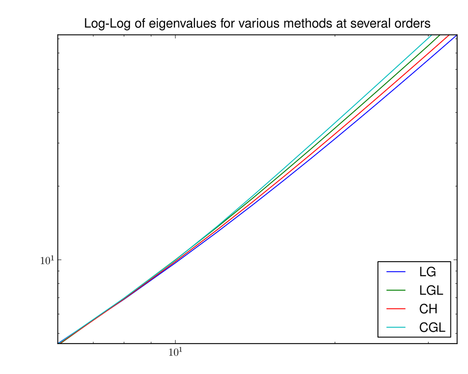
- $(\Phi_t, \ell_j) + \Phi\ell_j|_{-1}^{1} - (\Phi, \ell_j') = 0$
- $(\Phi_t, \ell_j) + \Phi(1)\ell_j(1) - g(t)\ell_j(-1) - (\Phi, \ell_j') = 0$

Contributions to Kopriva Errata

- Pages 98 and 116, Algorithms 42 and 50 have the same name, CollocationStepByRK3
- Page 115, The PDE in Equation 4.77 should read $\phi_t = k^2\phi_{xx}$ instead of $\phi_t = \phi_{xx}$ to be consistent with the solution given in 4.82.
- Page 127, Algorithm 53 should have $\Phi \leftarrow 0$ instead of $U \leftarrow 0$
- Page 133, Algorithm 57's summand in the inner loop should be $D_{k,n}D_{k,j}w_k$ to be consistent with eq 4.123
- Page 141, Algorithm 62; dg.TimeDerivative should be dg.DGTimeDerivative, or the Procedures list in 58 should be changed
- Page 213, Algorithm 91; LegendreGaussNodesAndWeights here takes a parameter N, in definition it takes only x, xj, and w. Same with PolynomialDerivativeMatrix.
- Page 215, Algorithm 93: ExternalState calls in η block are missing commas, muddles 'Q^L-1' vs 'Q^L, -1'

Differentiation Performance



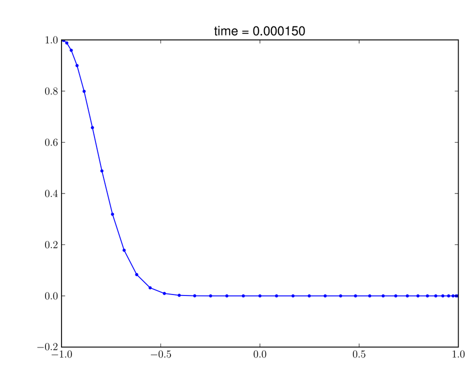


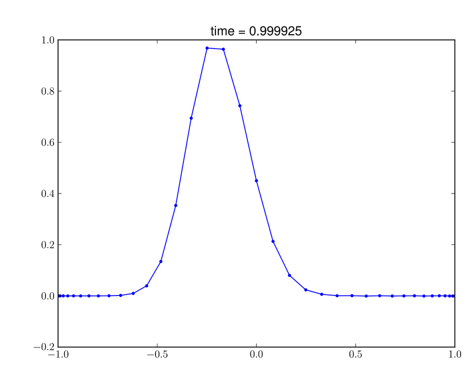
Summary

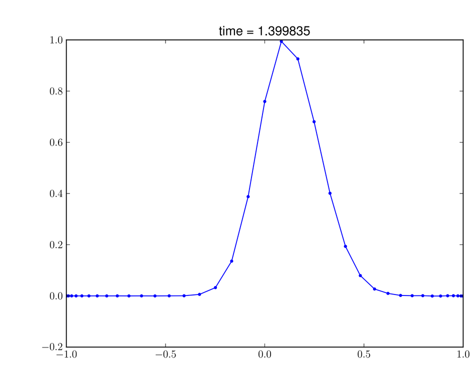
- Large eigenvalues require small Δt
- But great approximations for $N = O(30)$
- Finite differences, not great for $N = O(512)$

PDE Performance

Wave equation Forced at $x = -1$







Methods Implemented

FourierDerivativeMatrix
MxVDerivative
LegendrePolynomial
ChebyshevPolynomial
LegendrePolynomialAndDerivative
LegendreGaussNodesAndWeights
LegendreGaussLobattoNodesAndWeights
ChebyshevGaussNodesAndWeights
ChebyshevGaussLobattoNodesAndWeights
BarycentricWeights
LagrangeInterpolation
PolynomialInterpolationMatrix
InterpolateToNewPoints
PolynomialDerivativeMatrix
mthOrderPolynomialDerivativeMatrix
FourierCollocationTimeDerivative
FourierCollocationStepByRK3
FourierCollocationDriver
AdvectionDiffusionTimeDerivative
FourierGalerkinStep
EvaluateFourierGalerkinSolution
FourierGalerkinDriver
CollocationStepByRK3
LegendreCollocationIntegrator
LagrangeInterpolatingPolynomials
DGConstructor
DGDerivative
InterpolateToBoundary
DGTimeDerivative
DGStepByRK3
DGDriver
initialValues
AlmostEqual
TriDiagonalSolve
CGDerivativeMatrix
GalerkinStepByRK3
CGDerivativeMatrixIntegrator
TDerivative
LegendrePolynomialAndDeriv