The Discontinuous Galerkin Method

Joshua Buli
Joint work with Yulong Xing

University of California, Riverside
Grad Student Seminar

November 4, 2016

DG Method for Burgers' Equation

Advantages of the Discontinuous Galerkin (DG) Method

- Discontinuous Galerkin (DG) methods are a class of finite element methods using completely discontinuous piecewise polynomial spaces as the basis
- DG methods are high-order schemes, which allow for a coarse spatial mesh to achieve the same accuracy,
- DG methods achieve local conservativity, easily handle complicated geometries and boundary conditions
- Allow flexibility for h-p adaptivity, efficient parallel implementation, easy coordination with finite volume techniques
- DG methods have attracted attention for high performance computing due to high computational intensity and less data communication

DG Method for Burgers' Equation

Suppose we wish to solve the following IBVP

$$\begin{cases} u_t + \left(\dfrac{u^2}{2}\right)_x = 0 & (x,t) \in (0,L) \times (0,T) \\ u(x,0) = u_0(x) & (x,t) \in (0,L) \times \{0\} \\ u(0,t) = u(L,t) & \end{cases}$$

DG Method for Burgers' Equation

Suppose we wish to solve the following IBVP

$$\begin{cases} u_t + \left(\dfrac{u^2}{2}\right)_x = 0 & (x,t) \in (0,L) \times (0,T) \\ u(x,0) = u_0(x) & (x,t) \in (0,L) \times \{0\} \\ u(0,t) = u(L,t) & \end{cases}$$

- Partition spatial interval (a,b) with nodes $\{x_{j+1/2}\}_{j=0}^{N}$, and set $I_j = (x_{j-1/2}, x_{j+1/2})$
- $\Delta x_j = x_{j-1/2} - x_{j+1/2}$ for $j = 1, \ldots, N$
- For simplicity, take the mesh to be uniform: $\Delta x =$ constant.

DG Method for Burgers' Equation (cont.)

- Denote u_h as the approximate solution.

Multiplication by an arbitrary test function, ϕ_h, and integrating by parts,

$$\int_{I_j} (u_h)_t \phi_h \, dx + \int_{I_j} \left(\frac{u_h^2}{2} \right)_x \phi_h \, dx = 0,$$

$$\int (u_h)_t \phi_h \, dx - \int \left(\frac{u_h^2}{2} \right) (\phi_h)_x - \sum_{j=1}^{N} \left(\left(\frac{1}{2} \widehat{u_h^2} \right) [\phi_h] \right)_{j+1/2} = 0,$$

where $\int = \sum_{j=1}^{N} \int_{I_j}$, $\widehat{u_h^2}$ is the numerical flux, and $[\phi_h] = \phi_h^+ - \phi_h^-$.

DG Method for Burgers' Equation (cont.)

- All that remains is determine how we should define $\widehat{u_h^2}$
- From the method of characteristics, we know how information propagates over time, that is in the direction of the characteristics
- Therefore, we choose what is called the "upwind flux," and take $\widehat{u_h^2} = (u_h^-)^2$, where we take the value at the cell boundary to be from the left side
- In general, the choice of numerical flux is more difficult

Results

For the numerical test, we will solve the problem

$$
\begin{cases}
u_t + \left(\dfrac{u^2}{2}\right)_x = 0 & (x,t) \in (0,1) \times (0,T) \\
u(x,0) = 2 + \sin(2\pi x) & (x,t) \in (0,1) \times \{0\} \\
u(0,t) = u(1,t)
\end{cases}
$$

where T is the final time.

Convergence Rate

Piecewise Linear (p = 1)

N	E_u^1	Order
20	2.7499e-01	
40	6.9351e-02	1.987
80	1.7595e-02	1.978
160	4.4458e-04	1.984

Piecewise Cubic (p = 3)

N	E_u^1	Order
20	2.1803e-05	
40	1.5436e-06	3.820
80	1.0416e-07	3.889
160	6.3546e-09	4.034

Piecewise Quadratic (p = 2)

N	E_u^1	Order
20	9.7444e-04	
40	1.1725e-04	3.054
80	1.4676e-05	2.998
160	1.8413e-06	2.994

Piecewise Quartic (p = 4)

N	E_u^1	Order
20	7.8406e-07	
40	2.6463e-08	4.888
80	8.2327e-10	5.006
160	2.4821e-11	5.051

Results

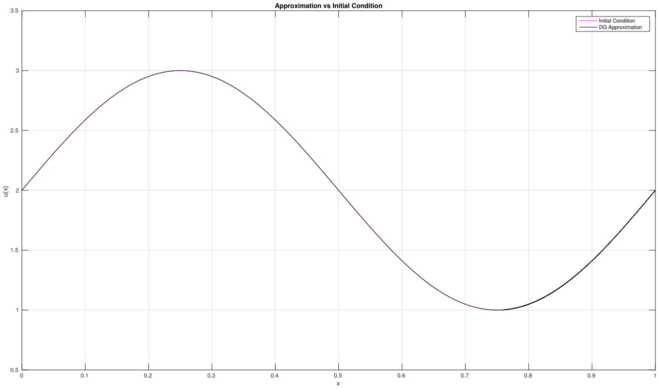

Figure : This is the approximation at time $t = 0$.

Results

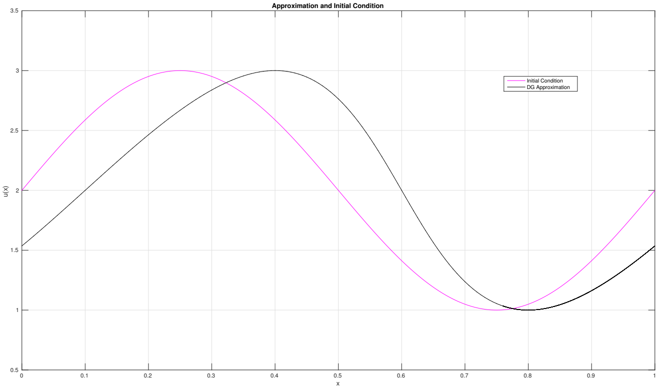

Figure : This is the approximation at time $t = 0.05$.

Results

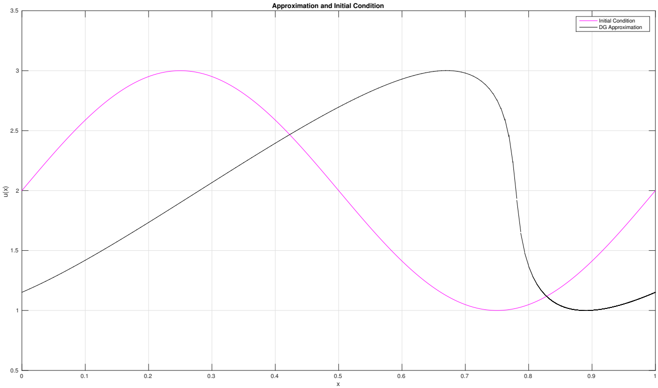

Figure : This is the approximation at time $t = 7.503125$.

DG Method for the BBM-System

Coupled BBM-system in Conservation Form

The coupled BBM-system given by

$$\begin{cases} \eta_t + u_x + (\eta u)_x - \frac{1}{6}\eta_{xxt} = 0, \\ u_t + \eta_x + uu_x - \frac{1}{6}u_{xxt} = 0. \end{cases}$$

Coupled BBM-system in Conservation Form

The coupled BBM-system given by

$$\begin{cases} \eta_t + u_x + (\eta u)_x - \frac{1}{6}\eta_{xxt} = 0, \\ u_t + \eta_x + u u_x - \frac{1}{6}u_{xxt} = 0. \end{cases}$$

We can write the above system in a conservation form

$$\begin{cases} \left(\eta - \frac{1}{6}\eta_{xx}\right)_t + (u + (\eta u))_x = 0, \\ \left(u - \frac{1}{6}u_{xx}\right)_t + \left(\eta + \frac{u^2}{2}\right)_x = 0. \end{cases}$$

Coupled BBM-system as a system of first order equations

We can rewrite the coupled-BBM system into a system of first order equations as the following

$$\begin{aligned}
w_t + (\eta + q)_x &= 0 \\
w &= u - \frac{1}{6} r_x \\
r &= u_x \\
q &= \frac{1}{2} u^2 \\
v_t + (u + p)_x &= 0 \\
v &= \eta - \frac{1}{6} s_x \\
s &= \eta_x \\
p &= \eta u
\end{aligned}$$

DG Formulation

The DG method is formulated as follows: for any test functions $\phi_h, \psi_h, \varphi_h, \zeta_h, \rho_h, \theta_h, \xi_h, \vartheta_h \in V_h^k$, find $w_h, v_h, u_h, \eta_h, r_h, s_h, p_h, q_h \in V_h^k$ such that

$$\int (w_h)_t \phi_h \, dx - \int (\eta_h + q_h)(\phi_h)_x \, dx - \sum_{j=1}^{N} ((\tilde{\eta}_h + \widehat{q_h})[\phi_h])_{j+\frac{1}{2}} = 0$$

$$\int w_h \psi_h \, dx - \int u_h (\psi_h)_x \, dx - \frac{1}{6} \int r_h (\psi_h)_x \, dx - \frac{1}{6} \sum_{j=1}^{N} (\widehat{r_h}[\psi_h])_{j+\frac{1}{2}} = 0$$

$$\int r_h \varphi_h \, dx + \int u_h (\varphi_h)_x \, dx + \sum_{j=1}^{N} (\widehat{u_h}[\varphi_h])_{j+\frac{1}{2}} = 0$$

$$\int q_h \zeta_h \, dx - \int \left(\frac{1}{2}(u_h)^2\right) \zeta_h \, dx = 0$$

DG Formulation (cont.)

$$\int (v_h)_t \rho_h \; dx - \int (u_h + p_h)\,(\rho_h)_x \; dx - \sum_{j=1}^{N} ((\tilde{u}_h + \widehat{p_h})[\rho_h])_{j+\frac{1}{2}} = 0$$

$$\int v_h \theta_h \; dx - \int \eta_h \theta_h \; dx - \frac{1}{6} \int s_h (\theta_h)_x \; dx - \frac{1}{6} \sum_{j=1}^{N} (\widehat{s_h}[\theta_h])_{j+\frac{1}{2}} = 0$$

$$\int s_h \xi_h \; dx + \int \eta_h (\xi_h)_x \; dx + \sum_{j=1}^{N} (\widehat{\eta_h}[\xi_h])_{j+\frac{1}{2}} = 0$$

$$\int p_h \vartheta_h \; dx - \int (\eta_h u_h) \vartheta_h \; dx = 0$$

Choice of Numerical Flux

We investigate two different choices of numerical flux, depending on what properties we wish to preserve. First is the *alternating flux*

$$
\begin{cases}
\widehat{u_h} & = u_h^+, \\
\widehat{\eta_h} & = \eta_h^- .
\end{cases}
\qquad
\begin{cases}
\widetilde{u_h} + \widehat{p_h} & = u_h^+ + p_h^+, \\
\widetilde{\eta_h} + \widehat{q_h} & = \eta_h^- + q_h^-, \\
\widehat{r_h} & = r_h^-, \\
\widehat{s_h} & = s_h^+ .
\end{cases}
$$

- Choice of flux follows from trying cancel out the boundary terms that arise in the DG formulation

Choice of Numerical Flux

We investigate two different choices of numerical flux, depending on what properties we wish to preserve. First is the *alternating flux*

$$\begin{cases} \widehat{u_h} & = u_h^+, \\ \widehat{\eta_h} & = \eta_h^-. \end{cases} \qquad \begin{cases} \widetilde{u_h} + \widehat{p_h} & = u_h^+ + p_h^+, \\ \widetilde{\eta_h} + \widehat{q_h} & = \eta_h^- + q_h^-, \\ \widehat{r_h} & = r_h^-, \\ \widehat{s_h} & = s_h^+. \end{cases}$$

- Choice of flux follows from trying cancel out the boundary terms that arise in the DG formulation
- Choosing u_h, η_h, and p_h, q_h, and r_h, s_h from opposite sides, the summation terms, and some of the integrals cancel out from integration by parts

Choice of Numerical Flux

We investigate two different choices of numerical flux, depending on what properties we wish to preserve. First is the *alternating flux*

$$\begin{cases} \widehat{u_h} &= u_h^+, \\ \widehat{\eta_h} &= \eta_h^-. \end{cases} \qquad \begin{cases} \widetilde{u_h} + \widehat{p_h} &= u_h^+ + p_h^+, \\ \widetilde{\eta_h} + \widehat{q_h} &= \eta_h^- + q_h^-, \\ \widehat{r_h} &= r_h^-, \\ \widehat{s_h} &= s_h^+. \end{cases}$$

- Choice of flux follows from trying cancel out the boundary terms that arise in the DG formulation
- Choosing u_h, η_h, and p_h, q_h, and r_h, s_h from opposite sides, the summation terms, and some of the integrals cancel out from integration by parts
- Remaining terms give the energy which is conserved by the method

Stability Theorem

Theorem (Stability)

For the choice of alternating flux, the (continuous) energy, $\mathcal{E}_h(t)$, is conserved by the DG method, i.e.

$$\frac{d}{dt}\mathcal{E}_h(t) = \frac{d}{dt}\int_I (\eta_h^2 + (1+\eta_h)u_h^2)\ dx = 0$$

for all time.

Stability Theorem

Theorem (Stability)

For the choice of alternating flux, the (continuous) energy, $\mathcal{E}_h(t)$, is conserved by the DG method, i.e.

$$\frac{d}{dt}\mathcal{E}_h(t) = \frac{d}{dt}\int_I (\eta_h^2 + (1+\eta_h)u_h^2)\; dx = 0$$

for all time.

The proof follows from choosing the alternating flux from the previous slides. Boundary terms can be eliminated by integration by parts identities. The proof is similar to that of the energy conservation theorem found in *Chen, Liu (2012)* at the PDE level.

Choice of Numerical Flux

Second, is the upwind flux which introduces numerical dissipation, and has the choices of

$$\begin{cases} \widetilde{u_h} &= \{u_h\} - \frac{1}{2}[\eta_h], \\ \widetilde{\eta_h} &= \{\eta_h\} - \frac{1}{2}[u_h]. \end{cases} \qquad \begin{cases} \widehat{q_h} &= \{q_h\} - \frac{1}{2}[p_h], \\ \widehat{p_h} &= \{p_h\} - \frac{1}{2}[q_h]. \end{cases}$$

$$\begin{cases} \widetilde{(u_h)_t} &= \{(u_h)_t\} + \frac{1}{2}[(\eta_h)_t], \\ \widetilde{(\eta_h)_t} &= \{(\eta_h)_t\} + \frac{1}{2}[(u_h)_t]. \end{cases} \qquad \begin{cases} \widetilde{(r_h)_t} &= \{(r_h)_t\} - \frac{1}{2}[(s_h)_t], \\ \widetilde{(s_h)_t} &= \{(s_h)_t\} - \frac{1}{2}[(r_h)_t]. \end{cases}$$

- Notation: $\{u_h\} = \frac{u_h^+ + u_h^-}{2}$ and $[u_h] = u_h^+ - u_h^-$
- Choice of flux follows from eliminating the third derivative term to get a system of hyperbolic conservation laws
- Upwind flux is the standard choice for this type of system
- Chosen to add numerical dissipation to the system

Energy Dissipation Theorem

Theorem (Energy Dissipation)

For the choice of upwind flux, the energy, $\mathcal{E}_h(t)$, satisfies

$$\frac{d}{dt}\mathcal{E}_h(t) = \frac{d}{dt}\int_I (\eta_h^2 + (1+\eta_h)u_h^2)\ dx \leq 0$$

with the DG method.

Energy Dissipation Theorem

Theorem (Energy Dissipation)

For the choice of upwind flux, the energy, $\mathcal{E}_h(t)$, satisfies

$$\frac{d}{dt}\mathcal{E}_h(t) = \frac{d}{dt}\int_I (\eta_h^2 + (1+\eta_h)u_h^2)\ dx \leq 0$$

with the DG method.

The proof follows similar to the previous stability proof, except we choose the upwind flux choices from previous slides. With this choice, boundary terms from the DG method are still present. These terms can be bounded by a routine application of Young's inequality to get the energy estimate.

Advantages/Disadvantages for Numerical Fluxes

Comparison of Alternating vs. Upwind

- Alternating Flux

Advantages/Disadvantages for Numerical Fluxes

Comparison of Alternating vs. Upwind

- Alternating Flux
    - Method is stable

Advantages/Disadvantages for Numerical Fluxes

Comparison of Alternating vs. Upwind

- Alternating Flux
  - Method is stable
  - Conserves energy exactly

Advantages/Disadvantages for Numerical Fluxes

Comparison of Alternating vs. Upwind

- Alternating Flux
    - Method is stable
    - Conserves energy exactly
    - Good for long time simulations

Advantages/Disadvantages for Numerical Fluxes

Comparison of Alternating vs. Upwind

- Alternating Flux
    - Method is stable
    - Conserves energy exactly
    - Good for long time simulations
- Upwind Flux

Advantages/Disadvantages for Numerical Fluxes

Comparison of Alternating vs. Upwind

- Alternating Flux
  - Method is stable
  - Conserves energy exactly
  - Good for long time simulations
- Upwind Flux
  - Method is stable

Advantages/Disadvantages for Numerical Fluxes

Comparison of Alternating vs. Upwind

- Alternating Flux
    - Method is stable
    - Conserves energy exactly
    - Good for long time simulations
- Upwind Flux
    - Method is stable
    - Dissipates energy over time

Advantages/Disadvantages for Numerical Fluxes

Comparison of Alternating vs. Upwind

- Alternating Flux
    - Method is stable
    - Conserves energy exactly
    - Good for long time simulations
- Upwind Flux
    - Method is stable
    - Dissipates energy over time
    - Not accurate for long time simulations

Advantages/Disadvantages for Numerical Fluxes

Comparison of Alternating vs. Upwind

- Alternating Flux
    - Method is stable
    - Conserves energy exactly
    - Good for long time simulations
- Upwind Flux
    - Method is stable
    - Dissipates energy over time
    - Not accurate for long time simulations
    - Better choice when shocks/discontinuities are present

Time Discretizations

We have used three different types of time discretizations over the course of the project:

- Strong Stability Preserving (SSP) Runge-Kutta (RK) Methods

Time Discretizations

We have used three different types of time discretizations over the course of the project:

- Strong Stability Preserving (SSP) Runge-Kutta (RK) Methods
    1. SSPRK3 and SSPRK4

Time Discretizations

We have used three different types of time discretizations over the course of the project:

- Strong Stability Preserving (SSP) Runge-Kutta (RK) Methods
  1. SSPRK3 and SSPRK4
  2. High order SSP methods maintain the total variation diminishing (TVD) property

Time Discretizations

We have used three different types of time discretizations over the course of the project:

- Strong Stability Preserving (SSP) Runge-Kutta (RK) Methods
  1. SSPRK3 and SSPRK4
  2. High order SSP methods maintain the total variation diminishing (TVD) property
  3. SSP methods are used to control numerical oscillations that occur around discontinuities

Time Discretizations

We have used three different types of time discretizations over the course of the project:

- Strong Stability Preserving (SSP) Runge-Kutta (RK) Methods
    1. SSPRK3 and SSPRK4
    2. High order SSP methods maintain the total variation diminishing (TVD) property
    3. SSP methods are used to control numerical oscillations that occur around discontinuities
- Midpoint Rule Method

Time Discretizations

We have used three different types of time discretizations over the course of the project:

- Strong Stability Preserving (SSP) Runge-Kutta (RK) Methods
  1. SSPRK3 and SSPRK4
  2. High order SSP methods maintain the total variation diminishing (TVD) property
  3. SSP methods are used to control numerical oscillations that occur around discontinuities
- Midpoint Rule Method
  1. Implicit time stepping method

Time Discretizations

We have used three different types of time discretizations over the course of the project:

- Strong Stability Preserving (SSP) Runge-Kutta (RK) Methods
    1. SSPRK3 and SSPRK4
    2. High order SSP methods maintain the total variation diminishing (TVD) property
    3. SSP methods are used to control numerical oscillations that occur around discontinuities
- Midpoint Rule Method
    1. Implicit time stepping method
    2. Conserves the discrete energy equivalent to the continuous case, over longer time than SSPRK4

Time Discretizations

We have used three different types of time discretizations over the course of the project:

- Strong Stability Preserving (SSP) Runge-Kutta (RK) Methods
  1. SSPRK3 and SSPRK4
  2. High order SSP methods maintain the total variation diminishing (TVD) property
  3. SSP methods are used to control numerical oscillations that occur around discontinuities
- Midpoint Rule Method
  1. Implicit time stepping method
  2. Conserves the discrete energy equivalent to the continuous case, over longer time than SSPRK4
  3. Computationally expensive as this is an implicit method

Remaining Theoretical Work

- The error estimate proof of the DG method for the single BBM equation case completed. We have proved the suboptimal error estimate.

Remaining Theoretical Work

- The error estimate proof of the DG method for the single BBM equation case completed. We have proved the suboptimal error estimate.
- For the coupled BBM system, we would also like to establish the sub-optimal error estimate

$$||u - u_h|| \leq Ch^{k+\frac{1}{2}}$$

where u is the true solution, u_h is the DG approximation, and k is the degree of the piecewise polynomial space.

Remaining Theoretical Work

- The error estimate proof of the DG method for the single BBM equation case completed. We have proved the suboptimal error estimate.
- For the coupled BBM system, we would also like to establish the sub-optimal error estimate

$$||u - u_h|| \leq Ch^{k+\frac{1}{2}}$$

  where u is the true solution, u_h is the DG approximation, and k is the degree of the piecewise polynomial space.
- Difficulty arises in this proof due to the nonlinear terms present and the coupled nature of the system.

Solutions to the BBM-system (Exact Traveling Wave Solution)

- *Chen (1998)*, the exact traveling wave solution to the BBM-system is

$$u(x,t) = 3k\ \mathrm{sech}^2\left(\frac{3}{\sqrt{10}}(x-kt-x_0)\right),$$

$$\eta(x,t) =$$

$$\frac{15}{4}\left(-2+\cosh\left(3\sqrt{\frac{2}{5}}(x-kt-x_0)\right)\right)\mathrm{sech}^4\left(\frac{3}{\sqrt{10}}(x-kt-x_0)\right)$$

where $k = \pm\frac{5}{2}$, and x_0 is the x value where the center of the wave is located

Solutions to the BBM-system (Approximate Solitary Wave Solution)

- *Alazman, et. all (2006)*, the coupled BBM-system has solitary wave solutions similar to the single BBM equation given by

$$v_t + v_x + \frac{3}{2}\epsilon v v_x - \frac{1}{6}\epsilon v_{xxt} = 0,$$

Solutions to the BBM-system (Approximate Solitary Wave Solution)

- *Alazman, et. all (2006)*, the coupled BBM-system has solitary wave solutions similar to the single BBM equation given by

$$v_t + v_x + \frac{3}{2}\epsilon v v_x - \frac{1}{6}\epsilon v_{xxt} = 0,$$

where ϵ represents the ratio of the maximum wave amplitude to the undisturbed depth of the liquid.

- The *exact* traveling wave solution to the single BBM equation is

$$v(x,t) = \operatorname{sech}^2\left(\frac{1}{2}\sqrt{\frac{3}{\kappa}}(x - \kappa t - x_0)\right),$$

where $\kappa = 1 + \epsilon/2$.

Solutions to the BBM-system (Approximate Solitary Wave Solution)

- An *approximate* solitary wave can be constructed using the following initial condition with the coupled BBM-system

$$\eta(x,0) = v(x,0),$$
$$u(x,0) = v(x,0) - \frac{1}{4}\epsilon v(x,0)^2,$$

where $v(x,t)$ is the exact traveling solution to the single BBM-equation

Solutions to the BBM-system (Approximate Solitary Wave Solution)

- An *approximate* solitary wave can be constructed using the following initial condition with the coupled BBM-system

$$\eta(x,0) = v(x,0),$$
$$u(x,0) = v(x,0) - \frac{1}{4}\epsilon v(x,0)^2,$$

where $v(x,t)$ is the exact traveling solution to the single BBM-equation

- Compare the single BBM solution to the coupled-BBM system with given initial data

Solutions to the BBM-system (Approximate Solitary Wave Solution)

- An *approximate* solitary wave can be constructed using the following initial condition with the coupled BBM-system

$$\eta(x,0) = v(x,0),$$
$$u(x,0) = v(x,0) - \frac{1}{4}\epsilon v(x,0)^2,$$

  where $v(x,t)$ is the exact traveling solution to the single BBM-equation
- Compare the single BBM solution to the coupled-BBM system with given initial data
- Approximate solitary wave for the coupled-BBM system, $\eta(x,t)$, is accurate to $\mathcal{O}\left(\frac{1}{\epsilon}\right)$ in time

Convergence Test: Alternating Flux, SSPRK4 in Time (Exact Traveling Wave Solution)

Parameters: $k = 2, L = 40, \Delta x = \frac{1}{2^j}$ for $j = 0, \ldots, 4$, $\Delta t = .1\Delta x$, $T = 1$

Nx	j	CPU (s)	E^1_η	Order	E^1_u	Order
40	0	0.163	1.6003e-00		9.3584e-01	
80	1	0.504	1.5717e-01	3.34	6.9160e-02	3.75
160	2	3.505	1.5362e-02	3.35	5.0564e-03	3.77
320	3	25.795	1.7227e-03	3.15	5.2204e-04	3.27
640	4	271.279	2.0514e-04	3.06	6.4118e-05	3.02

Convergence Test: Alternating Flux, and Midpoint Rule in Time (Exact Traveling Wave Solution)

Parameters: $k = 2, L = 40, \Delta x = \frac{1}{2^j}$ for $j = 0, \ldots, 4$, $\Delta t = .1\Delta x^2$, $T = 1$, tolerance $= 10^{-10}$

Nx	j	CPU (s)	E_η^1	Order	E_u^1	Order
40	0	0.710	2.1994e-00		1.5848e-00	
80	1	2.328	1.7709e-01	3.63	1.1434e-01	3.79
160	2	14.034	1.5581e-02	3.50	7.0977e-03	4.00
320	3	214.036	1.6858e-03	3.20	6.0759e-04	3.54
640	4	3327.298	1.9711e-04	3.09	6.7434e-05	3.17

Convergence Test: Dissipative Flux, and SSPRK4 in Time (Approximate Solitary Wave Solution)

Parameters: $k = 2, L = 40, \Delta x = \frac{1}{2^j}$ for $j = 0, \ldots, 4$, $\Delta t = .1\Delta x$, $T = 1$

Nx	j	CPU (s)	E^1_η	Order	E^1_u	Order
40	0	0.166				
80	1	0.422	2.4668e-02		2.4668e-02	
160	2	1.924	2.6662e-03	3.209	2.6685e-03	3.208
320	3	67.544	3.1745e-04	3.070	3.1773e-04	3.070
640	4	530.353	3.9079e-05	3.022	3.9115e-05	3.002

Long Time Test Approximation - Alternating Flux, SSPRK4 (Exact Traveling Wave Solution)

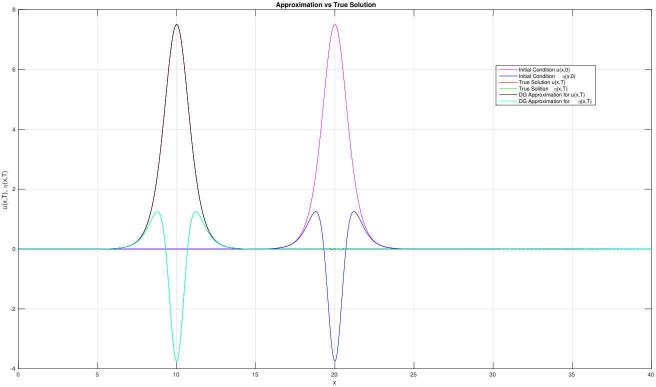

Figure : For the long time test, we run the code up to $T = 60$, and track L^1 errors over time.

Long Time Test L^1 Error - Alternating Flux, SSPRK4 (Exact Traveling Wave Solution)

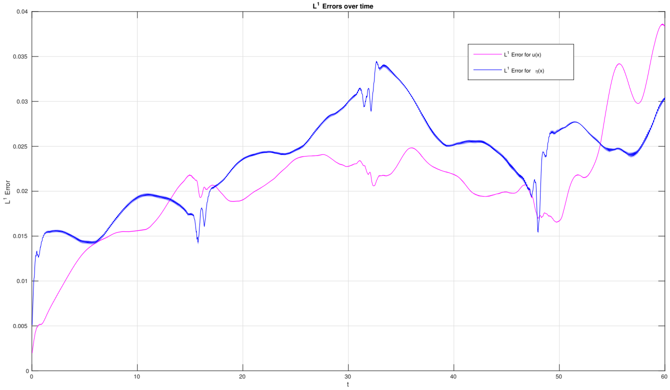

Figure : L^1 errors plotted against time.

Long Time Test Approximation - Alternating Flux, Midpoint in Time (Exact Traveling Wave Solution)

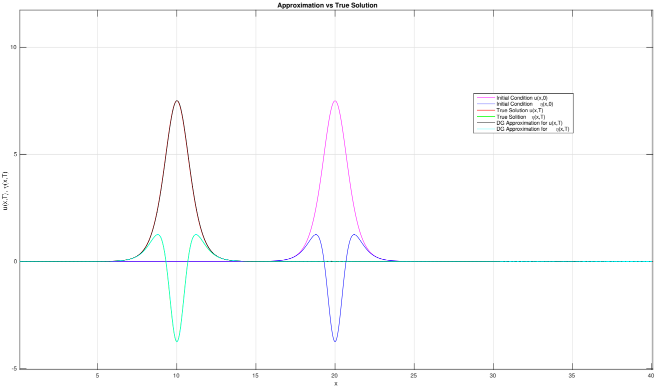

Figure : For the long time test, we run the code up to $T = 60$, and track L^1 errors over time.

Long Time Test L^1 Error - Alternating Flux, Midpoint in Time (Exact Traveling Wave Solution)

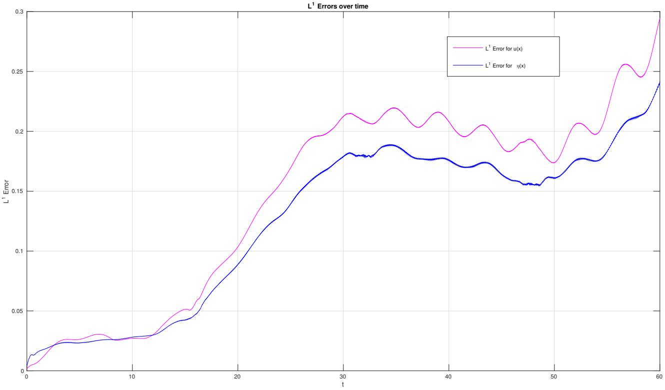

Figure : L^1 errors plotted against time.

Conserved Quantity - Alternating-SSPRK4-Midpoint Comparison (Exact Traveling Wave Solution)

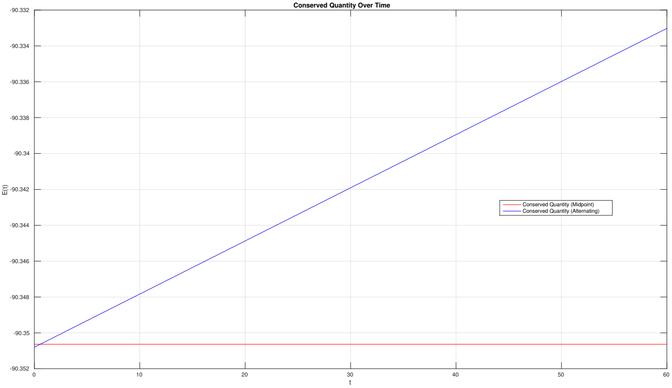

Figure : Comparison of Energy Values of SSPRK4 and Midpoint, with Alternating Flux.

Solitary Wave Generation Test

For the solitary wave generation test, we start with a first order approximation to the traveling wave solution that was used in the mesh refinement, and long time tests. The initial condition is given by

$$\eta(x,0) = \eta_0 \ \mathrm{sech}^2\left(\frac{1}{2}\sqrt{\frac{3\eta_0}{k}}(x-x_0)\right),$$

$$u(x,0) = \eta(x,0) - \frac{1}{4}\eta(x,0)^2,$$

where $\eta_0 = 0.8$ is the peak height for $\eta(x,0)$, and $x_0 = 20$.

Solitary Wave Generation Test

For the solitary wave generation test, we start with a first order approximation to the traveling wave solution that was used in the mesh refinement, and long time tests. The initial condition is given by

$$\eta(x,0) = \eta_0 \ \mathrm{sech}^2\left(\frac{1}{2}\sqrt{\frac{3\eta_0}{k}}(x - x_0)\right),$$

$$u(x,0) = \eta(x,0) - \frac{1}{4}\eta(x,0)^2,$$

where $\eta_0 = 0.8$ is the peak height for $\eta(x,0)$, and $x_0 = 20$.

The wave is evolved over the long domain, then “filtered”, and reset back to the left hand side of the domain. The process is repeated until dispersive tails are “small.”

Solitary Wave Generation Test Initial Condition

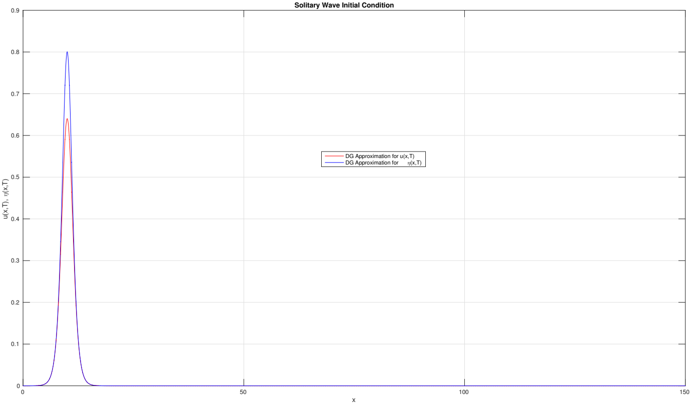

Figure : Solitary wave initial condition profile.

Solitary Wave Generation Test - One Evolution (T = 42)

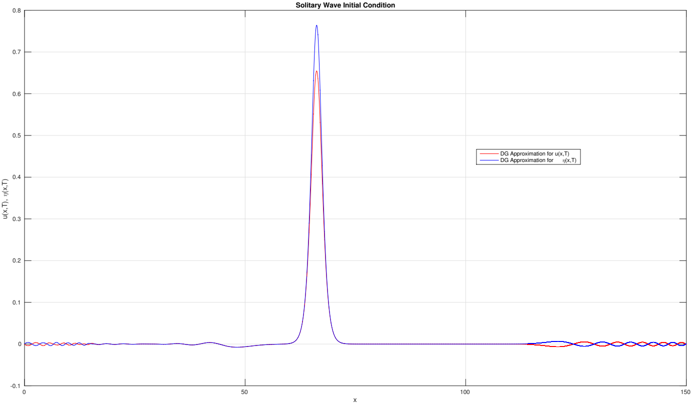

Figure : Solitary wave propagation at $T = 42$.

Solitary Wave Collision Test

Solitary Wave Collision Test

(Loading movie...)

Summary of the BBM-system Project

- Discontinuous Galerkin (DG) method to solve the single BBM equation and BBM-system

Summary of the BBM-system Project

- Discontinuous Galerkin (DG) method to solve the single BBM equation and BBM-system
- Alternating and upwind flux choices that conserve energy and work well for long time simulations

Summary of the BBM-system Project

- Discontinuous Galerkin (DG) method to solve the single BBM equation and BBM-system
- Alternating and upwind flux choices that conserve energy and work well for long time simulations
- Stability results and error estimates for the proposed method

Summary of the BBM-system Project

- Discontinuous Galerkin (DG) method to solve the single BBM equation and BBM-system
- Alternating and upwind flux choices that conserve energy and work well for long time simulations
- Stability results and error estimates for the proposed method
- Numerical experiments validating the usefulness of the method

Burgers' Equation with Stochastic Inputs

Overview of gPC Method for Stochastic PDEs

- Implementation of Generalized Polynomial Chaos (gPC) method

Overview of gPC Method for Stochastic PDEs

- Implementation of Generalized Polynomial Chaos (gPC) method
- Goal is to implement the method, no new results

Overview of gPC Method for Stochastic PDEs

- Implementation of Generalized Polynomial Chaos (gPC) method
- Goal is to implement the method, no new results
- Observe the effect of stochastic initial conditions on solution

Overview of gPC Method for Stochastic PDEs

- Implementation of Generalized Polynomial Chaos (gPC) method
- Goal is to implement the method, no new results
- Observe the effect of stochastic initial conditions on solution
- Test case with Burgers' equation

Burgers' Equation with Stochastic Inputs Problem Statement

Burgers' equation can be stated as

$$\begin{cases} u_t + uu_x = 0 & \text{for } (x,t) \in [a,b] \times [0,T] \\ u(x,0) = u_0(x) & \text{for } (x,t) \in [a,b] \times \{0\} \end{cases}$$

Burgers' Equation with Stochastic Inputs Problem Statement

Burgers' equation can be stated as

$$\begin{cases} u_t + uu_x = 0 & \text{for } (x,t) \in [a,b] \times [0,T] \\ u(x,0) = u_0(x) & \text{for } (x,t) \in [a,b] \times \{0\} \end{cases}$$

We will consider the situation when the initial condition has a stochastic component. The problem is stated as

$$\begin{cases} u_t + uu_x = 0 & \text{for } (x,t,\xi) \in [0,3] \times [0,T^* - \epsilon] \times \mathbb{R} \\ u(x,0,\xi) = u_0(x,\xi) & \text{for } (x,t,\xi) \in [0,3] \times \{0\} \times \mathbb{R} \end{cases}$$

Burgers' Equation with Stochastic Inputs Problem Statement

Burgers' equation can be stated as

$$\begin{cases} u_t + uu_x = 0 & \text{for } (x,t) \in [a,b] \times [0,T] \\ u(x,0) = u_0(x) & \text{for } (x,t) \in [a,b] \times \{0\} \end{cases}$$

We will consider the situation when the initial condition has a stochastic component. The problem is stated as

$$\begin{cases} u_t + uu_x = 0 & \text{for } (x,t,\xi) \in [0,3] \times [0,T^* - \epsilon] \times \mathbb{R} \\ u(x,0,\xi) = u_0(x,\xi) & \text{for } (x,t,\xi) \in [0,3] \times \{0\} \times \mathbb{R} \end{cases}$$

where the solution, $u(x,t,\xi)$ is a function of the random parameter $\xi \sim \mathcal{U}(-1,1)$. We run the code up to time $T^* - \epsilon$ which is the time just before the shock develops.

Problem Statement (cont.)

For the stochastic initial condition, $u_0(x, \xi)$, we have the following formula

$$u_0(x,\xi) = K_0 1_{[0,x_0]}(x - \sigma\xi) + K_1 1_{[x_1,3]}(x - \sigma\xi) + \left(\frac{K_1 - K_0}{x_1 - x_0}(x - \sigma\xi) - \frac{K_0 x_1 - K_1 x_0}{x_1 - x_0} \right) 1_{[x_0,x_1]}(x - \sigma\xi)$$

where 1 is the characteristic function, and σ is the weight for the stochastic component.

Problem Statement (cont.)

For the stochastic initial condition, $u_0(x,\xi)$, we have the following formula

$$u_0(x,\xi) = K_0 1_{[0,x_0]}(x-\sigma\xi) + K_1 1_{[x_1,3]}(x-\sigma\xi) + \left(\frac{K_1-K_0}{x_1-x_0}(x-\sigma\xi) - \frac{K_0x_1-K_1x_0}{x_1-x_0}\right) 1_{[x_0,x_1]}(x-\sigma\xi)$$

where 1 is the characteristic function, and σ is the weight for the stochastic component.
The values of the other components are $K_0 = 1.2$, $K_1 = 0.2$, $x_0 = 0.5$, $x_1 = 1.5$, and $\sigma = 0.01$.

gPC Expansion

To begin, we assume that the solution to the problem can be written as the spectral expansion

$$u(x,t,\xi) = \sum_{i=0}^{\infty} u_i(x,t)\Phi_i(\xi),$$

where the $\Phi_i(\xi)$ are the stochastic basis elements.

gPC Expansion

To begin, we assume that the solution to the problem can be written as the spectral expansion

$$u(x,t,\xi) = \sum_{i=0}^{\infty} u_i(x,t)\Phi_i(\xi),$$

where the $\Phi_i(\xi)$ are the stochastic basis elements.

Denote the inner product over the probability space to be

$$\langle u, v\rangle = \int_{\Omega} uvf(\xi)\, d\xi,$$

where $f(\xi) = \frac{1}{2}$, the probability density function for $\mathcal{U}(-1,1)$.

gPC Expansion of Burgers' Equation

To find the expansion of Burgers' Equation, we substitute the spectral expansion into the PDE to get

$$u_t + uu_x = 0$$

$$\sum_{i=0}^{\infty} \frac{\partial u_i}{\partial t} \Phi_i(\xi) + \left(\sum_{j=0}^{\infty} u_j \Phi_j(\xi) \right) \left(\sum_{i=0}^{\infty} \frac{\partial u_i}{\partial x} \Phi_i(\xi) \right) = 0$$

gPC Expansion of Burgers' Equation

To find the expansion of Burgers' Equation, we substitute the spectral expansion into the PDE to get

$$u_t + uu_x = 0$$

$$\sum_{i=0}^{\infty} \frac{\partial u_i}{\partial t} \Phi_i(\xi) + \left(\sum_{j=0}^{\infty} u_j \Phi_j(\xi) \right) \left(\sum_{i=0}^{\infty} \frac{\partial u_i}{\partial x} \Phi_i(\xi) \right) = 0$$

Now by multiplying through by the basis functions, $\Phi_k(\xi)$ and integrating over the probability space Ω, we have

$$\frac{\partial u_k}{\partial t} \langle \Phi_k, \Phi_k \rangle + \sum_{i=0}^{M} \sum_{j=0}^{M} u_i \frac{\partial u_j}{\partial x} \langle \Phi_i \Phi_j, \Phi_k \rangle = 0 \quad \text{for } k = 0, 1, \ldots, M$$

where we have truncated the expansion to $M + 1$ terms.

gPC Expansion of Burgers' Equation

From the previous slide,

$$\frac{\partial u_k}{\partial t}\langle \Phi_k, \Phi_k \rangle + \sum_{i=0}^{M}\sum_{j=0}^{M} u_i \frac{\partial u_j}{\partial x}\langle \Phi_i \Phi_j, \Phi_k \rangle = 0 \quad \text{for } k = 0, 1, \ldots, M$$

gPC Expansion of Burgers' Equation

From the previous slide,

$$\frac{\partial u_k}{\partial t}\langle \Phi_k, \Phi_k \rangle + \sum_{i=0}^{M}\sum_{j=0}^{M} u_i \frac{\partial u_j}{\partial x}\langle \Phi_i \Phi_j, \Phi_k \rangle = 0 \quad \text{for } k = 0, 1, \ldots, M$$

We can also write the above in conservative form

$$\frac{\partial u_k}{\partial t}\langle \Phi_k, \Phi_k \rangle + \frac{1}{2}\frac{\partial}{\partial x}\sum_{i=0}^{M}\sum_{j=0}^{M} u_i u_j \langle \Phi_i \Phi_j, \Phi_k \rangle = 0 \quad \text{for } k = 0, 1, \ldots, M$$

gPC Solution to Burgers' Equation

The solution then can be written as follows:

$$u(x,t,\xi) = \sum_{i=0}^{N} u_i(x,t)\Phi_i(\xi),$$

$$u_i(x,t) = \sum_{j=0}^{k} a_{i,j}(t)\psi_j(x),$$

where the $\Phi_i(\xi)$ are the Legendre polynomials, $\psi_j(x) \in V_h^k(I_j)$, and the $a_{i,j}$ are coefficient weights at a any time t.

gPC Solution to Burgers' Equation

The solution then can be written as follows:

$$u(x,t,\xi) = \sum_{i=0}^{N} u_i(x,t)\Phi_i(\xi),$$
$$u_i(x,t) = \sum_{j=0}^{k} a_{i,j}(t)\psi_j(x),$$

where the $\Phi_i(\xi)$ are the Legendre polynomials, $\psi_j(x) \in V_h^k(I_j)$, and the $a_{i,j}$ are coefficient weights at a any time t.

Therefore, each $u_i(x,t)$ is a polynomial of degree k from the DG method, with $N+1$ such equations. The numerical solution is thus a finite sum of products of polynomials in x and ξ.

Implementation

In order to solve the stochastic problem, we have to solve the system of conservation laws on the previous slide. First, we have to write the initial condition in terms of the gPC basis

$$u_0(x,\xi) = \sum_{i=0}^{1} \widetilde{u}_i(x)\Phi_i(\xi),$$

where

$$\widetilde{u}_i(x) = \langle \Phi_i(\xi), u_0(x,\xi) \rangle = \frac{1}{\sqrt{2\pi}} \int_\Omega \Phi_i(\xi) u_0(x,\xi) e^{\frac{-\xi^2}{2}} \, d\xi$$

Implementation

In order to compute the integral above, we use Gauss-Hermite quadrature, which approximates

$$\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty} g(\xi)e^{\frac{-\xi^2}{2}}\ d\xi \approx \frac{1}{\sqrt{\pi}}\sum_{i=1}^{n}\omega_i g(\sqrt{2}x_i)$$

where we used a change of variables $x = \frac{\xi}{\sqrt{2}}$, and ω_i and $x_i =$ are the Gauss-Hermite quadrature weights and nodes over $[-\infty, \infty]$.

Implementation

In order to compute the integral above, we use Gauss-Hermite quadrature, which approximates

$$\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty} g(\xi)e^{\frac{-\xi^2}{2}}\, d\xi \approx \frac{1}{\sqrt{\pi}}\sum_{i=1}^{n}\omega_i g(\sqrt{2}x_i)$$

where we used a change of variables $x = \frac{\xi}{\sqrt{2}}$, and ω_i and $x_i =$ are the Gauss-Hermite quadrature weights and nodes over $[-\infty, \infty]$. Another change of variables is required on the initial condition

$$u_0(x,\xi) = K_0 1_{[0,x_0]}(x-\sigma\xi) + K_1 1_{[x_1,3]}(x-\sigma\xi) + \left(\frac{K_1-K_0}{x_1-x_0}(x-\sigma\xi) - \frac{K_0x_1-K_1x_0}{x_1-x_0}\right) 1_{[x_0,x_1]}(x-\sigma\xi)$$

Implementation

This is due to the fact we need to integrate out the stochastic term ξ, so the IC becomes

$$u_0(x,\xi) = K_0 1_{[\frac{x-x_0}{\sigma},\frac{x}{\sigma}]}(\xi) + K_1 1_{[\frac{x-3}{\sigma},\frac{x-x_1}{\sigma}]}(\xi) + \left(\frac{K_1 - K_0}{x_1 - x_0}(x-\sigma\xi) - \frac{K_0 x_1 - K_1 x_0}{x_1 - x_0}\right) 1_{[\frac{x-x_1}{\sigma},\frac{x-x_0}{\sigma}]}(\xi)$$

Implementation

This is due to the fact we need to integrate out the stochastic term ξ, so the IC becomes

$$u_0(x,\xi) = K_0 1_{[\frac{x-x_0}{\sigma},\frac{x}{\sigma}]}(\xi) + K_1 1_{[\frac{x-3}{\sigma},\frac{x-x_1}{\sigma}]}(\xi) + \left(\frac{K_1-K_0}{x_1-x_0}(x-\sigma\xi) - \frac{K_0x_1-K_1x_0}{x_1-x_0}\right) 1_{[\frac{x-x_1}{\sigma},\frac{x-x_0}{\sigma}]}(\xi)$$

In the above, the values of x are taken to be the quadrature points we will use in the next step for the DG method, so the now have the initial condition written in the form

$$u_0(x,\xi) = \sum_{i=0}^{1} \tilde{u}_i(x)\Phi_i(\xi),$$

DG Implementation

We now return to solve the deterministic system

$$u_t + \frac{\partial}{\partial x} f(u) = 0, \quad f(u) = \frac{1}{2} A(u) u$$

where

$$\frac{1}{2} A(u) u = \begin{bmatrix} \frac{1}{2} u_0^2 + \frac{1}{2} u_1^2 \\ u_0 u_1 \end{bmatrix}$$

DG Implementation

We now return to solve the deterministic system

$$u_t + \frac{\partial}{\partial x} f(u) = 0, \quad f(u) = \frac{1}{2} A(u) u$$

where

$$\frac{1}{2} A(u) u = \begin{bmatrix} \frac{1}{2} u_0^2 + \frac{1}{2} u_1^2 \\ u_0 u_1 \end{bmatrix}$$

In system form, we have

$$\begin{aligned} \frac{\partial u_0}{\partial t} + \frac{\partial}{\partial x} \left(\frac{1}{2} u_0^2 + \frac{1}{2} u_1^2 \right) &= 0 \\ \frac{\partial u_1}{\partial t} + \frac{\partial}{\partial x} (u_0 u_1) &= 0 \end{aligned}$$

DG Implementation

The weak formulation of the system is given by the following

$$\int_0^3 (u_0)_t \phi \, dx + \int_0^3 \left(\frac{1}{2} u_0^2 + \frac{1}{2} u_1^2 \right)_x \phi \, dx = 0$$

$$\int_0^3 (u_1)_t \psi \, dx + \int_0^3 (u_0 u_1)_x \psi \, dx = 0,$$

DG Implementation

The weak formulation of the system is given by the following

$$\int_0^3 (u_0)_t \phi \, dx + \int_0^3 \left(\frac{1}{2}u_0^2 + \frac{1}{2}u_1^2\right)_x \phi \, dx = 0$$

$$\int_0^3 (u_1)_t \psi \, dx + \int_0^3 (u_0 u_1)_x \psi \, dx = 0,$$

For the DG formulation, we choose test functions $\phi_h, \psi_h \in V_h^k$, and search for $(u_0)_h, (u_1)_h \in V_h^k$ such that

$$\int ((u_0)_h)_t \phi_h \, dx - \int \left(\frac{1}{2}((u_0)_h)^2 + \frac{1}{2}((u_1)_h)^2\right)(\phi_h)_x \, dx$$
$$- \sum_{j=1}^{N} \left(\left(\frac{1}{2}((\widehat{u_0})_h)^2 + \frac{1}{2}((\widehat{u_1})_h)^2\right)[\phi_h]\right)_{j+\frac{1}{2}} = 0,$$

$$\int ((u_1)_h)_t \psi_h \, dx - \int (p_h)\,(\psi_h))_x \, dx - \sum_{j=1}^{N} (\widehat{p}_h[\psi_h])_{j+\frac{1}{2}} = 0$$

DG Implementation

where we take p_h to be the projection of the non-linear term $(u_0)_h(u_1)_h$ into the DG space. The numerical flux for the hat terms $\widehat{(u_0)_h}, \widehat{(u_1)_h}, \widehat{p}_h$ are all taken to be the upwind flux.

DG Implementation

where we take p_h to be the projection of the non-linear term $(u_0)_h(u_1)_h$ into the DG space. The numerical flux for the hat terms $\widehat{(u_0)}_h, \widehat{(u_1)}_h, \widehat{p}_h$ are all taken to be the upwind flux.

For the implementation, we do a stochastic gPC approximation of order 1, and a DG method that is piecewise linear. The mesh size is taken to be $\Delta x = .1/32$, and $\Delta t = .1\Delta x$. There are 25 stochastic sample initial conditions taken.

A minmod slope limiter is also implemented to reduce oscillations at the edges corners of the piecewise function.

Initial Condition With No Stochastic Component

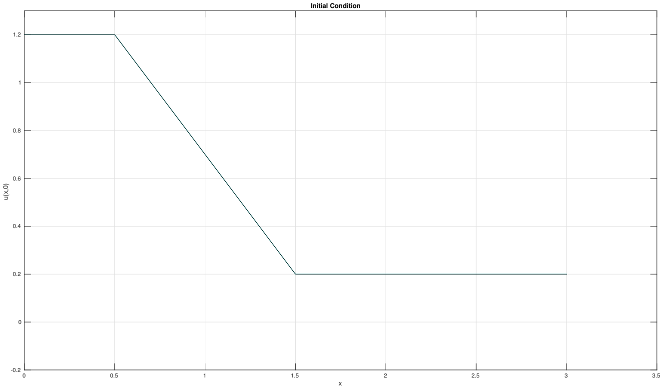

Figure : The initial condition with the stochastic component set to zero.

Order 3 gPC Basis Results

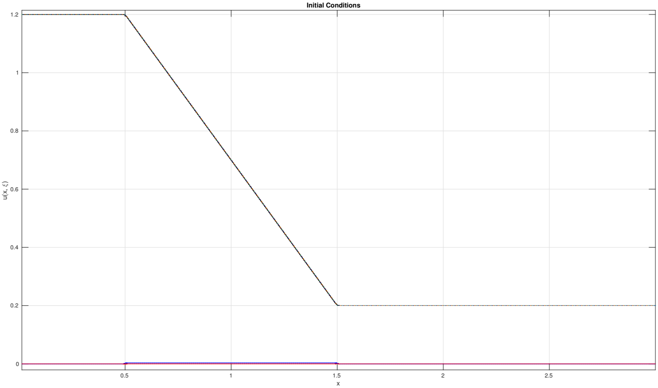

Figure : The light-blue line is the deterministic IC from a previous slide. The multi-color line is the average of the initial conditions. The black, dark blue, and red represent $u_0(x,\xi)$, $u_1(x,\xi)$, and $u_2(x,\xi)$, respectively.

Order 3 gPC Basis Results

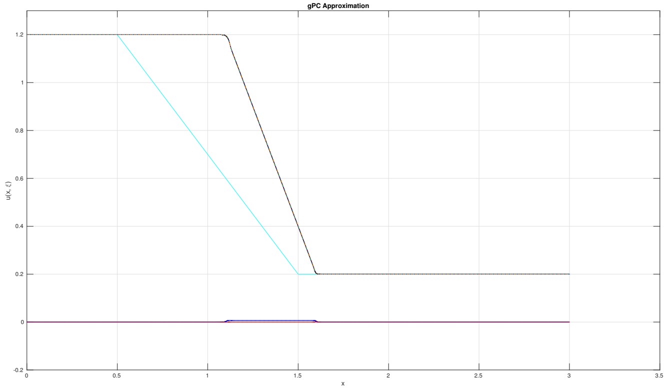

Figure : This is the approximation at time $T = .5$.

Order 3 gPC Basis Results

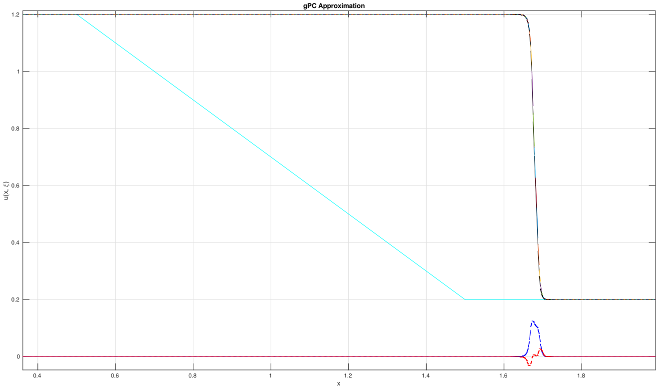

Figure : This is the approximation at time $T = T^* - \epsilon = 0.97$.

Sample Initial Conditions - How Stochasticity Affects Shock Location

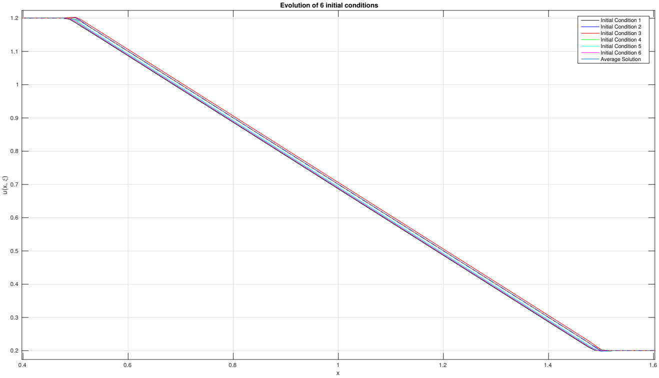

Figure : Six sample initial conditions.

How Stochasticity Affects Shock Location

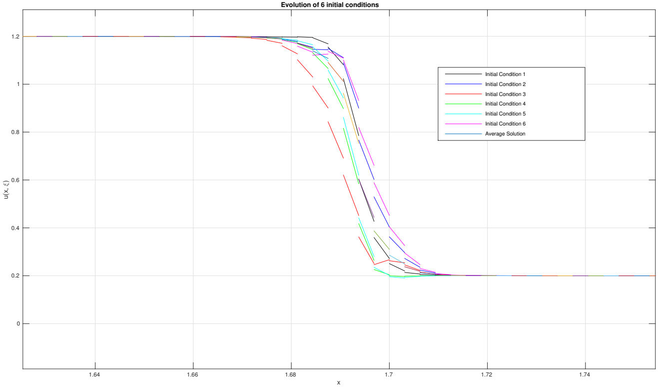

Figure : Six sample initial conditions evolved over time, with the average solution plotted for reference.

References

- Alazman, Albert, Bona, Chen, Wu, Comparisons Between the BBM Equation and a Boussinesq System, Adv. Diff. Eqns. 11, no. 2, pp. 121-166 (2006)
- Bona, Chen, A Boussinesq system for two-way propagation of nonlinear dispersive waves, Physica D 116, pp. 191-224 (1998)
- Chen, Exact Traveling-Wave Solutions to Bidirectional Wave Equations, Int. J. of Theo. Phys. 37, no. 5, pp. 1547-1567 (1998)
- Chen, Liu, On the Well-Posedness of a Weakly Dispersive One-Dimensional Boussinesq System, arXiv: 1203.0365v1 [math.AP] (2012)
- Dougalis, Mitsotakis, Saut, On Initial-Boundary Value Problems for a Boussinesq System of BBM-BBM Type in a Plane Domain, AIMS 23, no. 4, pp. 1191-1204 (2009)
- Dougalis, Mitsotakis, Saut, Boussinesq Systems of Bona-Smith Type on Plane Domains: Theory and Numerical Analysis, J. Sci. Comp. 44, no. 2, pp. 109-135 (2010)