A Well-Balanced Stochastic Galerkin Method for PDEs with Random Forcing

Abhijit Chowdhary* and Andrew Shedlock*

Last Edited: May 3, 2022

1 Introduction

In this past couple of decades, the field of Uncertainty Quantification has grown significantly. Indeed, many of the models we use in the study of numerical methods for them are just that, models derived from some physical process. Most of the time we use idealized examples which ignore some uncertainty in the given parameters under the assumption that they would negligible, but work in the field has proven that to either be untrue, or for those uncertain parameters themselves to the be the quantities of interest. Hence, there is a real need for numerical methods correctly propagating uncertainties.

Of particular interest are PDE-based systems with random forcing, for example bottom topographies in systems describing shallow waters. Taking a simplified 1D model of it, we consider systems of the form

$$u_t + f(u)_x = -b'(x,z)q(u) \tag{1}$$

where z is distributed according to some random variable and is the uncertainty we seek to resolve in the solution. In [XK02; Xiu10], the Stochastic Galerkin method, which is a traditional Galerkin method with orthogonal basis given by a generalized polynomial chaos (gPC) expansion in the random variable space, was constructed and is a viable scheme to solve (1). However, even in the deterministic case, numerical schemes for equations such as (1) need to be built with care should we want them to resolve the steady state solution. This is desirable, as much of the study of such equations revolve around understanding small perturbations of the steady state, and whether or not it is stable to such perturbations. Well-balanced schemes which preserve the steady state have been well studied, and in [Jin01] they constructed what they called the *interface method* which preserved the steady state of the deterministic version of (1).

The goal of this report is to investigate the results of the paper [JXZ15], which took the traditional gPC stochastic Galerkin method and the well balanced interface method and constructed a well balanced stochastic Galerkin (SGWB) method, which, to our knowledge,

*Department of Mathematics, North Carolina State University, Raleigh, NC 27695, USA (achowdh2@ncsu.edu, ajshedlo@ncsu.edu)

was the first such paper to employ such a technique. As the extension of such ideas are of current research interest, the verification of the reproducability of the results is an important task. In section 2 we will lay out the theory for the construction of such a SGWB method, and in section 3 we will reconstruct the figures found in the original paper.

2 Methods

As we described in the introduction, the scheme we will satisfy both a well-balanced condition and a stochastic Galerkin projection condition. In the deterministic case we are working a system of the form

$$u_t + f(u)_x = -b'(x)u \tag{2}$$

At the steady state we have that $u_t = 0$, hence (2) reduces to the steady state condition

$$f(u)_x + b'(x)u = 0 \tag{3}$$

We limit our discussion to when $f(u) = \frac{u^2}{2}$ and $f(u) = \frac{u^4}{4}$ to highlight the central idea. For these two choices of f, we have that $\frac{f'(\xi)}{\xi}$ is a well-defined number for all $\xi \neq 0$. We can now define an operator $D : \mathbb{R} \to \mathbb{R}$ by

$$D(u) = \int_0^u \frac{f'(\xi)}{\xi} d\xi$$

Now if $u(x)$ is a strong solution to (3) and $u > 0$, then by Liebniz's rule

$$D(u(x))_x + b'(x) = \frac{f'(u(x))u_x}{u(x)} + b'(x) = \frac{f(u(x))_x}{u(x)} + b'(x) = 0$$

This shows that if u is a strong solution to (3) and $u > 0$, then $D(u(x)) + q(x)$ is equal to some constant. If we assume that $D(u(x))_x \neq 0$, then we can see that

$$u(x) = \frac{f(u)_x}{D(u(x))_x} \tag{4}$$

Equation (4) will come into play shortly, but first we must introduce our notation for the discrete system. If we adopt the finite volume framework, then we shall have N_x cells with uniform mesh size of Δx. For $j = 1, \ldots, N_x$, we let x_j represent the point at the center of the cell and let $x_{j+1/2}$ represent a point at a cell interface. Since we will have a uniform mesh, then $\Delta x = x_{j+1/2} - x_{j-1/2}$. We also take a uniform discretization with mesh size Δt, and let t^n refer to the nth time step, i.e. $t^n = t^0 + n\Delta t$. The cell average at t^n over $[x_{j-1/2}, x_{j+1/2}]$ is given by

$$u_j^n = \frac{1}{\Delta x} \int_{x_{j-1/2}}^{x_{j+1/2}} u(x, t^n) dx$$

At the cell interfaces we will have

$$D_{j+1/2} = D(u_{j+1/2}) \qquad f_{j+1/2} = f(u_{j+1/2}) \qquad b_{j+1/2} = b(x_{j+1/2})$$

Now the value $u_{j+1/2}$ can be computed at the cell interface by using either a Godunov scheme or a Roe scheme. Under this finite volume framework, (4) can be approximated as

$$u_j^n \approx \frac{\frac{f_{j+1/2}-f_{j-1/2}}{\Delta x}}{\frac{D_{j+1/2}-D_{j-1/2}}{\Delta x}} = \frac{f_{j+1/2}-f_{j-1/2}}{D_{j+1/2}-D_{j-1/2}}$$

This observation leads to the semi-discrete scheme for (2) to be

$$\partial_t u_j + \frac{f_{j+1/2}-f_{j-1/2}}{\Delta x} = -\frac{(b_{j+1/2}-b_{j-1/2})}{\Delta x}\Big(\frac{f_{j+1/2}-f_{j-1/2}}{D_{j+1/2}-D_{j-1/2}}\Big)$$

At the steady state we will get

$$\begin{aligned}
\frac{f_{j+1/2}-f_{j-1/2}}{\Delta x} &= -\frac{(b_{j+1/2}-b_{j-1/2})}{\Delta x}\Big(\frac{f_{j+1/2}-f_{j-1/2}}{D_{j+1/2}-D_{j-1/2}}\Big) \\
D_{j+1/2}-D_{j-1/2} &= -(b_{j+1/2}-b_{j-1/2}) \\
D_{j+1/2}+b_{j+1/2} &= D_{j-1/2}+b_{j-1/2} = \text{constant}
\end{aligned}$$

In particular for the Burgers' equation we have that $f(u) = \frac{u^2}{2}$, so $D(u) = u$ and so

$$\frac{f_{j+1/2}-f_{j-1/2}}{D_{j+1/2}-D_{j-1/2}} = \frac{u_{j+1/2}^2-u_{j-1/2}^2}{2(u_{j+1/2}-u_{j-1/2})} = \frac{u_{j+1/2}+u_{j-1/2}}{2}$$

Hence the following semi-discrete method for Burgers' equation is well-balanced.

$$\partial_t u_j + \Big(\frac{u_{j+1/2}^2-u_{j-1/2}^2}{\Delta x}\Big) = -\Big(\frac{b_{j+1/2}-b_{j-1/2}}{\Delta x}\Big)\Big(\frac{u_{j+1/2}+u_{j-1/2}}{2}\Big) \tag{5}$$

It is described in [Jin01] that for a general f the above scheme is well balanced.

$$\partial_t u_j + \Big(\frac{f_{j+1/2}-f_{j-1/2}}{\Delta x}\Big) = -\Big(\frac{b_{j+1/2}-b_{j-1/2}}{\Delta x}\Big)\Big(\frac{u_{j+1/2}+u_{j-1/2}}{2}\Big) \tag{6}$$

This gives us the formulation for a deterministic well-balanced scheme, in particular this is what we call the cell-interface method. We will incorporate these ideas into the stochastic scenario.

For the stochastic case, we incorporate some uncertainty into our bottom topography via a random variable $z : \Omega \to \mathbb{R}$ with law P. For every instance of our random variable z, let $u(x,t;z)$ be the function $(x,t) \to u(x,t;z)$ which solves

$$u_t(x,t;z) + f(u(x,t;z))_x = -b(x;z)u(x,t;z) \tag{7}$$

If we suppose that we have appropriate integrability conditions so that $u(x,t;z), z \in L^2(P)$ holds for all (x,t), then $u(x,t;z)$ will be $z-$measurable. Consider

$$\mathcal{P} = \{\phi(z) : \phi \text{ is a polynomial of any order}\}$$

Then $\mathcal{P}$ is a dense vector subspace in the space of all $z-$measurable functions. Since $\mathcal{P} \subset L^2(P)$ by our integrability assumption on z and that $L^2(P)$ is a Hilbert space, then we can find an orthonormal basis $\{\Phi_m(z)\}_{m\in\mathbb{N}}$ of $\mathcal{P}$. Thus $\{\Phi_m(z)\}_{m\in\mathbb{N}}$ is dense in the space $z-$measurable functions. Then since $u(x,t;z)$ is $z-$measurable, then we can find coefficients $\hat{u}_m(x,t)$ such that $P-$almost everywhere

$$u(x,t;z) = \sum_{m\in\mathbb{N}} \hat{u}_m(x,t)\Phi_m(z) \qquad \hat{u}_m(x,t) = \mathbb{E}[u(x,t;z)\Phi_m(z)]$$
$$b(x;z) = \sum_{m\in\mathbb{N}} \hat{b}_m(x)\Phi_m(z) \qquad \hat{b}_m(x) = \mathbb{E}[b(x;z)\Phi_m(z)]$$

Also necessary is the associated expected value and variance of $u(x,t;z)$. Using the fact that $\{\Phi_m(z)\}_{m\in\mathbb{N}}$ forms an orthonormal basis for $L^2(P)$ and that $\Phi_0(z) = 1$, one can show [Xiu10, Chapter 5.3] that:

$$\mathbb{E}[u(x,t;\cdot)] = \hat{u}_0(x,t) \qquad \mathbb{V}[u(x,t;\cdot)] = \sum_{m=1}^{\infty} \hat{u}_m^2(x,t)$$

We want (7) to hold for almost every instance of our random variable. Since $\{\Phi_m(z)\}_{m\in\mathbb{N}}$ forms an orthonormal basis, this is equivalent to enforcing the condition

$$\mathbb{E}[(\partial_t u(x,t;z) + f(u(x,t;z))_x)\Phi_m(z)] = \mathbb{E}[-b'(x,z)u(x,t;z)\Phi_m(z)] \tag{8}$$

for all $m \in \mathbb{N}$. In the discrete setting, we can only take a finite number of basis elements, so let N represent the number of nodes that we are considering. Then the galerkin approximation of u and b in terms of z is given by

$$u_N(x,t;z) = \sum_{m=1}^{N+1} \hat{u}_m(x,t)\Phi_m(z)$$
$$b_N(x;z) = \sum_{m=1}^{N+1} \hat{b}(x)\Phi_m(z)$$

If we fix t, then let $u_{N,j} = u_N(x_j,t;z)$ where the right hand side is the cell average over $[x_{j-1/2}, x_{j+1/2}]$. We take $u_{j+1/2} \approx u_j$ and similarly for the other components to receive a scheme identical to (6)

$$\partial_t u_{N,j} + \Big(\frac{f(u_{N,j}) - f(u_{N,j-1})}{\Delta x}\Big) = -\Big(\frac{b_{N,j} - b_{N,j-1}}{\Delta x}\Big)\Big(\frac{u_{N,j} + u_{N,j-1}}{2}\Big)$$

If we then enforce the condition (8) on the above, we have:

$$\mathbb{E}\Big[\Big(\partial_t u_{N,j} + \Big(\frac{f(u_{N,j}) - f(u_{N,j-1})}{\Delta x}\Big)\Big)\Phi_m(z)\Big] = -\mathbb{E}\Big[\Big(\frac{b_{N,j} - b_{N,j-1}}{\Delta x}\Big)\Big(\frac{u_{N,j} + u_{N,j-1}}{2}\Big)\Phi_m(z)\Big] \tag{9}$$

We say that any scheme which is consistent (8) for all $m \in \mathbb{N}$, over all cells and for all time is a Well-Balanced Stochastic Galerkin method for (1). In particular, (9) is an example of such a scheme. If we set $\mathbf{u}_j = (\hat{u}_{1,j}, \ldots, \hat{u}_{N+1,j})^T$, define $\mathbf{E}$ to be the tensor with components $e_{klm} = \mathbb{E}[\Phi_k \Phi_l \Phi_m]$ and $\mathbf{D}$ to be the tensor with components $d_{klmnp} = \mathbb{E}[\Phi_k \Phi_l \Phi_m \Phi_n \Phi_p]$ and where $1 \leq k, l, m, n, p \leq N+1$. We then define matrices $\mathbf{A}_j, \mathbf{S}_j, \mathbf{B}$ to have components

$$\begin{aligned}
\mathbf{A}_{kl,j} &= \sum_{m=1}^{N+1} e_{klm} \hat{u}_{m,j} \\
\mathbf{B}_{kl,j} &= \sum_{m=1}^{N+1} e_{klm} \hat{b}_{m,j} \\
\mathbf{S}_{kl,j} &= \sum_{m,n,p=1}^{N+1} d_{klmnp} \hat{u}_{m,j} \hat{u}_{n,j} \hat{u}_{p,j}
\end{aligned}$$

If $f(u) = \frac{u^2}{2}$, then a scheme which satisfies (9) is

$$\partial_t \mathbf{u}_j + \frac{\mathbf{A}_j \mathbf{u}_j - \mathbf{A}_{j-1} \mathbf{u}_{j-1}}{2\Delta x} = -\frac{(\mathbf{B}_j - \mathbf{B}_{j-1})(\mathbf{u}_j + \mathbf{u}_{j-1})}{2\Delta x} \tag{10}$$

If $f(u) = \frac{u^4}{4}$, then a scheme which satisfies (9) is

$$\partial_t \mathbf{u}_j + \frac{\mathbf{S}_j \mathbf{u}_j - \mathbf{S}_{j-1} \mathbf{u}_{j-1}}{4\Delta x} = -\frac{(\mathbf{B}_j - \mathbf{B}_{j-1})(\mathbf{u}_j + \mathbf{u}_{j-1})}{2\Delta x} \tag{11}$$

3 Results

In order to verify that the proposed methods satisfy the well-balanced claim, in this section we provide several numerical examples of it doing so. Consider the stochastic PDE (7) that we constructed methods for last section; imposing the boundary conditions:

$$\begin{cases} u(x,0) = 0, & \text{for } x > 0 \\ u(0,t) = 2, & \text{for } t > 0 \end{cases}$$

we solve the PDE under both Burger's flux $f(u) = u^2/2$ and the alternate flux given by $f(u) = u^4/4$ over the following choices of bottom topography:

$$b_1(x,z) = \begin{cases} (2+z)\cos(\pi x), & 4.5 \leq x \leq 5.5 \\ 0, & \text{otherwise} \end{cases} \tag{12}$$

$$b_2(x,z) = \begin{cases} 0.1(2+z)\cos(\pi x), & 5 \leq x \leq 6 \\ 0, & \text{otherwise} \end{cases} \tag{13}$$

Bottom topography (12) is continuous whereas (13) is discontinuous, resulting in continuous and discontinuous resulting solutions. In all examples $z \sim \text{Uniform}[-1, 1]$.

For a point of comparison, we compare our method to the so called *cell-average* method:

$$\partial_t \mathbf{u}_j + \frac{\mathbf{A}_j \mathbf{u}_j - \mathbf{A}_{j-1}\mathbf{u}_{j-1}}{2\Delta x} = -\frac{(\mathbf{B}_j - \mathbf{B}_{j-1})}{\Delta x}\mathbf{u}_j \tag{14}$$

$$\partial_t \mathbf{u}_j + \frac{\mathbf{S}_j \mathbf{u}_j - \mathbf{S}_{j-1}\mathbf{u}_{j-1}}{2\Delta x} = -\frac{(\mathbf{B}_j - \mathbf{B}_{j-1})}{\Delta x}\mathbf{u}_j \tag{15}$$

where (14) corresponds to the flux $f(u) = u^2/2$ and (15) corresponds to $f(u) = u^4/4$, and matrices $\mathbf{A}, \mathbf{B}, \mathbf{S}$ are as described above.

For all simulations, we choose number of cells $N_c = 100$, time step fixed to $\Delta t = 3200^{-1}$, and we integrate to final time $T = 10$. In the steady state, assuming $u > 0$, (7) reduces down to $u'(x) = -b'(x, z)$, hence we find the steady state solution $u(x) = 2 - b(x, z)$, from which we can derive the steady state mean and standard deviation $\mathbb{E}[u] = 2 - 2b(x, 0)$ and $\mathbb{V}[u] = \frac{1}{\sqrt{3}}|b(x, 0)|$, which is what we compare the numerically computed solution to in Figure 1 and similar in Figure 2. However, when a true mean and variance aren't available for comparison, for example with flux $f(u) = u^4/4$, one is numerically integrated for instead.

The results of these tests, found in Figures 1, 2 and 3, ultimately verify the fact that our scheme is well-balanced and offers a significant accuracy improvement over a naive cell-average scheme. Furthermore, a convergence study was done varying in both gPC-order and number of cells, where we verified that the method was indeed first-order convergent in space, see Figure 4. It also verified the expected conclusion that the errors in the random space saturate extremely quickly, as in both b_1 and b_2 are of order $\mathcal{O}(z)$.

Finally, one can view table 1 for time estimates for running the given method, varying in both N_c and gPC-order. Indeed, getting these times down to their current level was not a trivial computational task, initial versions of the code would take around 120 seconds with $N_c = 100$ and 4 gPC coefficients. Algorithmic improvements primarily through exploiting symmetry in $\mathbf{A}$ and $\mathbf{S}$ reduced the wall time down to around 50 seconds, but an eventual rewrite from MATLAB into Julia and exploiting its capabilities in avoiding unnecessary allocations and pre-compilation as opposed to a JIT system resulted in the final time of around 4 seconds, with around 4 additional seconds for compilation.

CPU Time	100	200	400
1	2.83	3.62	7.28
2	2.84	5.69	11.22
3	3.14	6.07	12.22
4	3.81	7.64	15.37

Table 1: The estimated CPU time (seconds) for the Well-Balanced Stochastic Galerkin method for bottom topography b_1 and flux $f(u) = u^2/2$ with cell count $N_c \in \{100, 200, 400\}$ and number of gPC coefficients $M \in \{1, 2, 3, 4\}$.

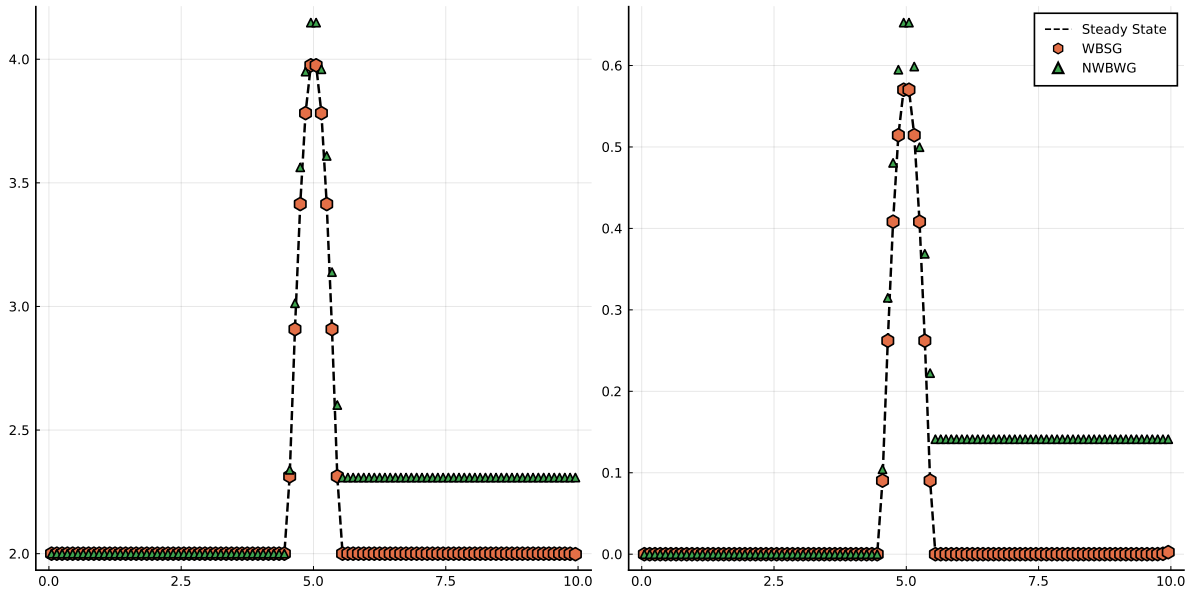

Figure 1: The results of the constructed WBSG (circles) (10) versus the non-WBSG method (triangles) (14) with forcing b_1 and flux $f(u) = u^2/2$. We use $N_c = 100$ cells and limit ourselves to a 4th order gPC expansion. On the left is a graph of the computed mean and on the right is the computed standard deviation. Indeed, we find that the new method resolves the steady state correctly.

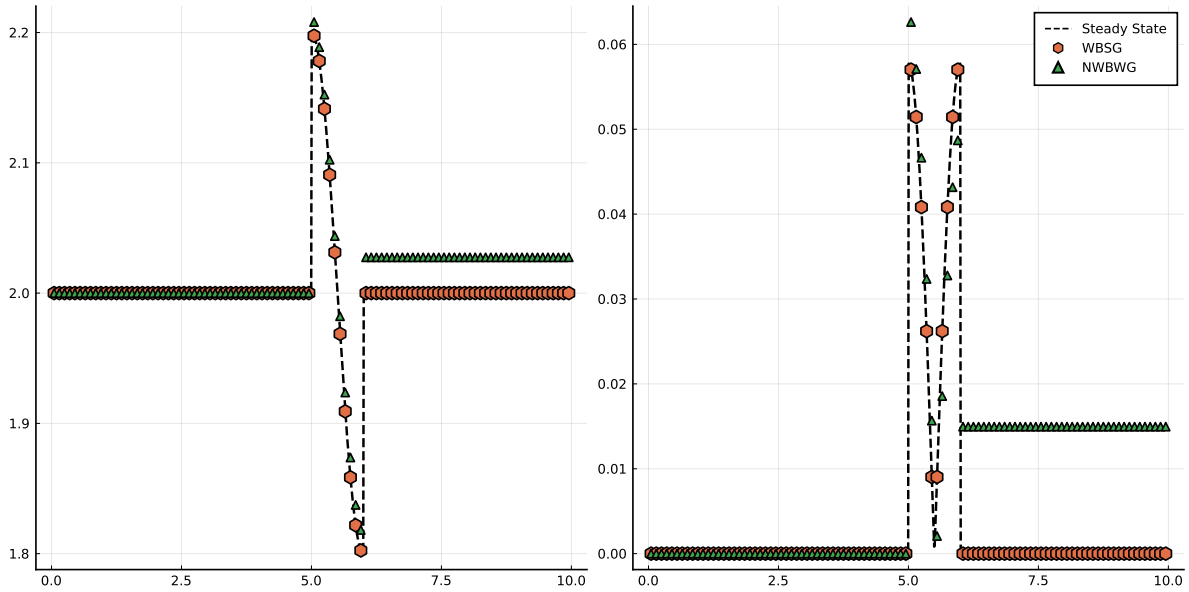

Figure 2: The results of the constructed WBSG (circles) (10) versus the non-WBSG method (triangles) (14) with forcing b_2 and flux $f(u) = u^2/2$. We use $N_c = 100$ finite volume cells and limit ourselves to a 4th order gPC expansion. On the left is a graph of the computed mean and on the right is the computed standard deviation. Indeed, we find that the new method resolves the steady state correctly even in the discontinuous case.

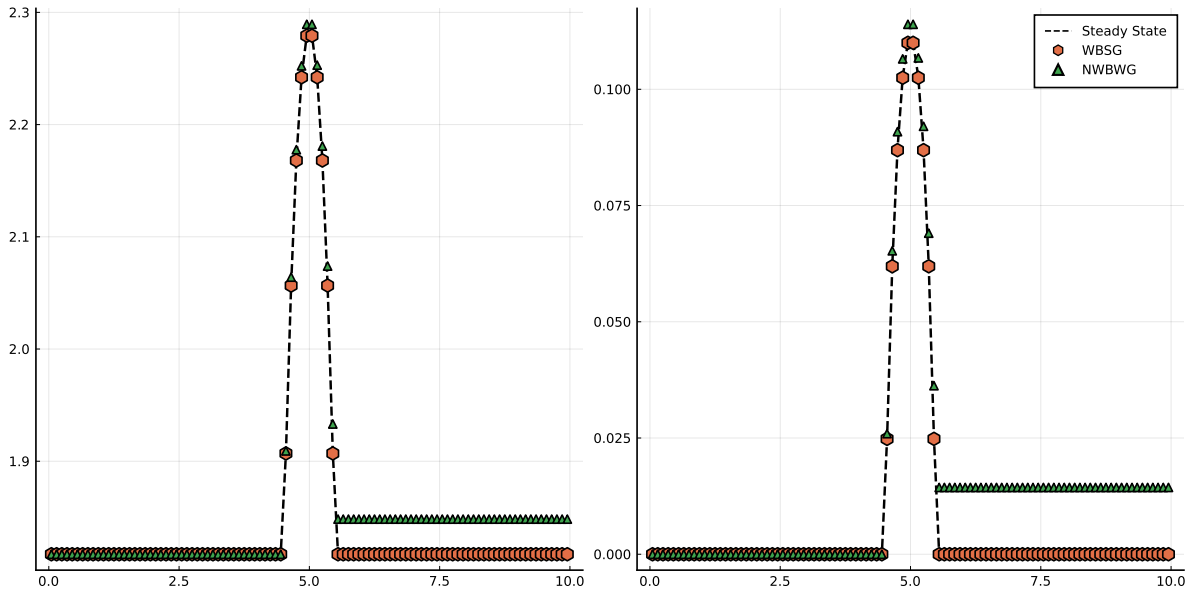

Figure 3: The results of the constructed WBSG (circles) (11) versus the non-WBSG method (triangles) (15) with forcing b_1 and flux $f(u) = u^4/4$. We use $N_c = 100$ finite volume cells and limit ourselves to a 4th order gPC expansion. On the left is a graph of the computed mean and on the right is the computed standard deviation. Indeed, we find that the new method resolves the steady state correctly even for alternative fluxes.

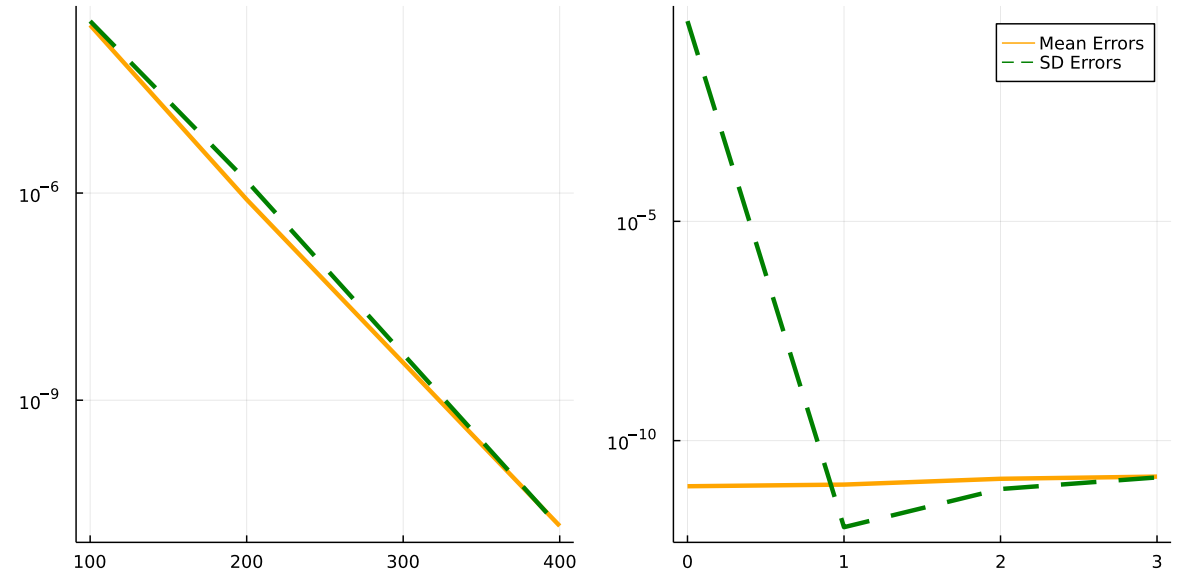

Figure 4: This figure denotes the convergence results for (10) with forcing b_1. The figure on the left denotes convergence in space with a 4th order gPC expansion, and the figure on the right denotes convergence in the order of the gPC expansion holding the spatial discretization at $N_c = 800$. Indeed we verify that our method is spatially of first order, and that it took only 2 gPC coefficient to resolve the uncertainties.

4 Conclusion

In conclusion, we've verified the results presented in [JXZ15], i.e. we've taken the well-balanced scheme constructed in [Jin01] and and extended it to a stochastic setting with the gPC stochastic Galerkin method. This is a proof of concept for taking previously constructed deterministic well-balanced schemes and forming Galerkin via gPC methods from them which are well balanced in a stochastic sense, and in theory this should work in general. This is useful as, in comparison to stochastic collocation methods which can preserve the well-balanced property and the collocation nodes, but struggle to preserve it between them in the interpolation step, these intrusive methods preserve steady state in a weak sense over the whole spatial domain. As an example, 2 years after this paper was published, [WTX17] produced an extension of this idea for quasilinear PDEs whose gPC-system may not be hyperbolic.

However, the difficulties of this method ought to be noted. Outside of the difficulty casting fluxes not of the form $f(u) = u^p/p$ into this method, it is clear, through the construction of $\mathbf{S}$ in the case of $f(u) = u^4/4$, that this method suffers from the curse of dimensionality in gPC-order. At each time iteration, one must iterate the gPC-coefficients of $\mathbf{u}$ at each spatial point, resulting in an $N_c \times M$ system to integrated. This is reasonable for problems with 1 spatial dimension, or with random effects that require few gPC-order to resolve, but as your problem increases in spatial dimension or has increasingly nonlinear random effects the size of coefficients being integrating may grow to be too large. This is ignoring multi-dimensional random variables, who would case the size of the space $\mathcal{P}_z$ to suffer combinatorial explosion. This is, of course, intractable, and methods to resolve these dimensionality troubles remain a further area of research.

References

[Jin01] Shi Jin. "A steady-state capturing method for hyperbolic systems with geometrical source terms". In: ESAIM: Mathematical Modelling and Numerical Analysis 35.4 (July 2001), pp. 631–645. DOI: 10.1051/m2an:2001130.

[JXZ15] Shi Jin, Dongbin Xiu, and Xueyu Zhu. "A Well-Balanced Stochastic Galerkin Method for Scalar Hyperbolic Balance Laws with Random Inputs". In: JSC 67.3 (Nov. 2015), pp. 1198–1218. DOI: 10.1007/s10915-015-0124-2.

[WTX17] Kailiang Wu, Huazhong Tang, and Dongbin Xiu. "A stochastic Galerkin method for first-order quasilinear hyperbolic systems with uncertainty". en. In: J. Comput. Phys. 345 (Sept. 2017), pp. 224–244.

[Xiu10] Dongbin Xiu. Numerical methods for stochastic computations. Princeton, NJ: Princeton University Press, July 2010.

[XK02] Dongbin Xiu and George Em Karniadakis. "The Wiener–Askey Polynomial Chaos for Stochastic Differential Equations". In: SIAM Journal on Scientific Computing 24.2 (2002), pp. 619–644. DOI: 10.1137/S1064827501387826.