LOCAL DISCONTINUOUS GALERKIN METHOD FOR THE STOCHASTIC HEAT EQUATION

MOE EBRAHIMI, MICHAEL HOLST, AND HAYLEY MILES-LEIGHTON

ABSTRACT. In this paper we study the Local Discontinuous Galerkin scheme for solving the stochastic heat equation driven by the space white noise. We begin by giving a brief introduction to stochastic processes, stochastic differential equations, and their importance in the modern mathematical context. From there, using an example stochastic elliptic partial differential equation, we approximate the white noise term using piecewise constant functions and show that it will also hold for the stochastic heat equation. We give an introduction to Local Discontinuous Galerkin method and produce a block matrix equation by separating the stochastic heat equation into two first order partial differential equations. We prove that the stochastic heat equation has a unique solution since its expected value converges to the heat equation without the white noise term. From there, we give a possible numerical way of solving the matrix equation as well as how to handle the stochastic term in this numerical method. After solving the matrix equation, we discuss what the average is of all the equations that result from the matrix equation.

CONTENTS

1. INTRODUCTION AND BACKGROUND

In this paper we study Local Discontinuous Galerkin method for the stochastic heat equation. This problem comes up in mathematical finance often and has many important applications, one of which is the Black-Scholes PDE [2]. This famous equation can be transformed into the heat equation which is the problem we plan to explain and approximate in this paper. We plan to approximate the noise by a piece-wise constant random process to facilitate the convergence of the finite element approximation. We

Date: May 31, 2010.

Key words and phrases. Stochastic PDE, Space-Time elements, Local Discontinuous Galerkin, Adaptive methods, nonlinear equations, approximation theory, nonlinear approximation, convergence, contraction, optimality, weak convergence, weak-* convergence, *a priori* estimates, *a posteriori* estimates, measure spaces.

ME was were supported in part by NSF Award 0715146.

MH was supported in part by NSF Awards 0715146 and 0915220, by DOE Award DE-FG02-04ER25620, and by DOD/DTRA Award HDTRA-09-1-0036.

HM was were supported in part by NSF Award 0715146.

will then formulate the weak form of the stochastic heat equation and use Local Discontinuous Galerkin method to get an approximation for the stochastic heat equation. We will also discuss as well as prove with the matrix formed by the Local Discontinuous Galerkin scheme that the stochastic heat equation has a unique solution. We will discuss the basics of a stochastic term and its uses as well as propose a way to approximate it.

The stochastic heat equation is given by:

$$\begin{cases} \frac{\delta u}{\delta t} - \alpha \nabla^2 u + \dot{W}(x) = 0 \\ u|_{\delta\Omega} = 0 \end{cases} \tag{1.1}$$

To better understand this problem, we need to define other basic terms and concepts first. To begin with, we will define a stochastic process.

A Stochastic Process is one whose behavior is non-deterministic, meaning that its patterns are determined both by predictable actions and random events. Many times there is a time factor within the process, as there is in the problem we will describe in this paper. The goal with a problem with a stochastic term is to come up with the best approximation through time. We find these processes throughout the physical world. There is one famous problem (also mentioned earlier), the Black-Scholes PDE [2], that contains a stochastic process. It helps represent the random events in the stock market, which, as the definition suggests, are both random and somewhat predictable.

To completely define a stochastic process, we need to define a couple of other things first. To begin with, we need to explain what a probability space is. To do this, we need to reach as far back as defining σ-algebra.

Definition 1.1. *Consider a set Ω. We consider a system of subsets of F of Ω. We assume that F satisfies the following axioms:*

- *The set Ω itself and the empty set $\emptyset$ are in F*
- *If the countable (finitely or infinitely) many sets $\{A_i\}_{i=1}$ are in F the the union $\bigcup A_i$ is in F*
- *If $A \in F$ then the complement $A^c = \Omega - A$ is in F*

This such a system of subsets is called a σ-algebra.

Now, Let (Ω, F) be a set and F a σ-algebra of subsets of Ω.

Definition 1.2. *A measure on (Ω, F) is a function $\mu : F \longmapsto \mathbb{R}_{\geqslant 0}$ satisfying the following condition:*

$(\sigma - additivity)$ *If A is the union of finitely or countable infinitely many pairwise disjoint sets $A = \bigcup_{i=1}^{\infty} A_i$ with $A_i \in F$ and $A_i \cap A_j = \emptyset$ when $i \neq j$, then $\mu(A) = \sum_{i=1}^{\infty} \mu(A_i)$.*

Putting all of these different pieces of information together, we come to the definition of a Probability Space:

Definition 1.3. *A measure space (Ω, F, μ) is called a probability space if $\mu(\Omega) = 1$.*

[5]

Stochastic Processes are defined inside their own probability space. It is the probability space that helps define what the expected value is. However, we will leave it here and define a stochastic differential equation.

A stochastic differential equation is a differential equation with a stochastic element. The field of numerical solutions for stochastic differential equations (especially stochastic partial differential equations) is still relatively young. Not much is known about how to approximate the stochastic term and still approximate the differential equation itself. Many times the convergence rate is poor.

The way that Stochastic DE's are solved is to take the stochastic term out and approximate it before approximating the entire DE. In a paper written by Yao & Bo [4], they approximate the stochastic term with constant lines. In the next section, we will describe in detail how exactly they approximated the term and proved its convergence. This approximation will be used to produce the LDG method and the final approximation of the SPDE.

2. Noise Approximation

In this section, we will use an example problem to demonstrate a possible way of approximating the white noise with constant lines. We will show that the expected value of the approximation converges. In addition, the error between the approximation and the actual value of the noise term goes to zero. At the end, we will discuss how this approximation holds for the stochastic heat equation as well.

As the example [4], let us look at:

$$\begin{cases} -\Delta u(x) \;=\; \lambda u(x) + q(x) + \dot{W}(x),\; x \in D \\ u|_{\delta D} \;=\; 0 \end{cases} \tag{2.1}$$

where $D = (0,1)^d$ for $d \in \{2,3\}$, λ is some positive constant, $q(\cdot) : D \to \mathbb{R}$ is an $L^2(D)$-function, and $\{\dot{W}(x); x \in D\}$ denotes the space of white noise on some probability space (Ω, F, P). Formally, $E(\dot{W}(x)\dot{W}(y)) = \delta(|x-y|)$ with $x, y \in D$ and $\delta(\cdot)$ is the Dirac function. $E(\cdot)$ is the expected value. For this problem, the expected value is defined as $\int f(x_1,...,x_n)P(x_1,...,x_n)$ where $P(x_1,...,x_n)$ is the probability density function and is defined above.

From here we can get the weak form:

$$u(x) = \lambda \int_D G(x,y)u(y)dy + \int_D G(x,y)q(y)dy + \int_D G(x,y)W(dy) \tag{2.2}$$

where $G(x,y)$ is Green's function corresponding to the elliptic PDE: $-\Delta v(x) = \phi(x)$ with $v|_{\delta D} = 0$ such that $v(x) = \int_D G(x,y)\phi(y)dy$.

To begin the approximation of the white noise, we need to define several variables. First, let $N \in \mathbb{N}$ and $h = \frac{1}{N}$. Suppose that $\{x_j := jh\}_{j=0}^{N}$ is a partition for $[0,1]$. Set $\mathbf{k} = (k_1, k_2, ..., k_d) \in I_N^d := \{0,1,...,N-1\}^d$, $D = (0,1)^d$ for $d \in \{2,3\}$, and $D_{\mathbf{k}} = \prod_{j=1}^{d}(x_{k_j}, x_{k_j}+h)$. This is all done for $d \geqslant 2$. Then, for $y \in D$, define

$$\dot{\overline{W}} \;:=\; \sum_{\mathbf{k} \in I_N^d} \eta_{\mathbf{k}} \chi_{D_{\mathbf{k}}}(y), \tag{2.3}$$

where $\eta_{\mathbf{k}} := \frac{1}{h^d}\int_{D_{\mathbf{k}}} W(dy) = \frac{1}{h^d}W(D_{\mathbf{k}})$ and $\chi_{\mathbf{A}}(\cdot)$ denotes the indicator of a set A. For each $L^2(D)$-function $f : D \times \Omega \to \mathbb{R}$, define an integral by

$$\int_D f(y)\overline{W}_h(dy) \;:=\; \int_D f(y)\dot{\overline{W}}_h(y)dy. \tag{2.4}$$

Each $\dot{\overline{W}}$ is made by approximating the points inside that specific partition with a line. This all leads to our first Lemma:

Lemma 2.1. *For each $\omega \in \Omega$, $\dot{\overline{W}}_h(\omega,\cdot) \in L^2(D)$, and*

(1) $E(|\int_D \overline{W}_h(dy)|^2) = 1$
(2) $E(||\dot{\overline{W}}_h(\cdot)||^2_{L^2}) = \frac{1}{h^d}$
(3) $E(|\int_D g(y)\overline{W}_h(dy)|^2) \leqslant ||g||^2_{L^2}$
with the deterministic integrand $g \in L^2(D)$.

This Lemma provides us with a better understanding of how our approximation of the noise process acts within our domain. This helps us in proving that the solution for the entire SPDE does converge.

Now, we want to consider the approximated form of the example elliptic stochastic equation:

$$\begin{cases} -\Delta u_h(x) \;=\; \lambda u_h(x) + q(x) + \dot{\overline{W}}_h(x),\; x \in D \\ u_h|_{\delta D} \;=\; 0 \end{cases} \tag{2.5}$$

Note that u_h also satisfies:

$$u_h(x) = \lambda \int_D G(x,y)u_h(y)dy + \int_D G(x,y)d(y)dy + \int_D G(x,y)\overline{W}_h(dy). \tag{2.6}$$

Let $\hat{e} = u(x) - u_h(x)$ and $M(x) = \int_D G(x,y)(W(dy) - \overline{W}_h(dy))$. As the approximation of the example stochastic elliptic PDE converges, $M(x)$ should go to zero. So, by (2.2) and (2.6) we have:

$$\hat{e} = u(x) - u_h(x) = \lambda \int_D G(x,y)u(y)dy + \int_D G(x,y)q(y)dy + \int_D G(x,y)W(dy)$$
$$-\lambda \int_D G(x,y)u_h(y)dy - \int_D G(x,y)d(y)dy - \int_D G(x,y)\overline{W}_h(dy) =$$
$$\lambda \int_D G(x,y)(u(y) - u_h(y))dy + \int_D G(x,y)(W(dy) - \overline{W}_h(dy)) = \lambda \int_D G(x,y)\hat{e}(y)dy + M(x)$$

Using this information, we see that:

$$||\hat{e}||^2_{L^2} = ||u - u_h||^2_{L^2} \leq \lambda^2 \int_{D^2} G^2(x,y)dxdy||\hat{e}||^2_{L^2} + 2\lambda(\int_{D^2} G^2(x,y)dxdy)^{\frac{1}{2}}||\hat{e}||_{L^2}||M||_{L^2} + ||M||^2_{L^2}.$$

If we define $\rho := \lambda(\int_{D^2} G^2(x,y)dydx)^{\frac{1}{2}}$ and let $\rho < 1$ by choosing a small λ. So, the above equation now becomes:

$$(1-\rho^2)E(||\hat{e}||^2_{L^2}) \leq 2\rho(E(||\hat{e}||^2_{L^2}))^{\frac{1}{2}}(E(||M||^2_{L^2}))^{\frac{1}{2}} + E(||M||^2_{L^2}).$$

Below is an estimation of $E(||M||^2_{L^2})$:

Proposition 2.2. *For every* $\epsilon > 0$*, there exists a positive constant* $c_3(\epsilon, d)$ *such that* $E(||M||^2_{L^2}) \leq c_3(\epsilon, d)h^{2-\epsilon}$*, for* $d = 2$.

This proposition states that as h gets very small, the expected value of the error between the approximation of the white noise and the actual white noise term goes to zero as well.

From here, we can refer back to the original problem. As a reminder, this is the original problem:

$$\begin{cases} \frac{\delta u}{\delta t} - \alpha\nabla^2 u + \dot{W}(x) = 0 \\ u|_{\delta\Omega} = 0 \end{cases} \tag{2.7}$$

Referring back up to Lemma 2.1, we see that the approximation of the white noise for the example problem will also work for the stochastic heat equation since the definition is not dependent on the individual problem.

In this section, we used an example problem to approximate the white noise and showed that its expected value does converge. We also explained how this method of approximating the white noise can also be applied to the stochastic heat equation. In the next section, we will apply Local Discontinuous Galerkin (LDG) method to the stochastic heat equation and discuss the uniqueness of the solution.

3. LDG METHOD

Local Discontinuous Galerkin Method is, simply put, the finite element method with discontinuous nodes. For example, if we were to study the one dimensional line from a to b , we would split up the interval into subintervals with a discontinuous point in-between each subinterval. For a visual of this, look at Figure 1 below:

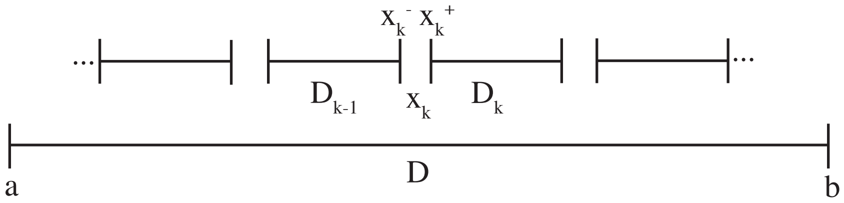

FIGURE 1. An example of a discretized 1-D line in DG method

Each x_k is a node. At each node is the discontinuous point which creates the subintervals. We will describe how to approximate the SPDE over just one of those subintervals. We will then piece together all the different approximations over each node to create the final approximation.

There are many reasons why we chose to use LDG method to approximate the stochastic heat equation. It can easily handle approximations that have polynomials of different degrees in different elements. Since we are approximating the stochastic term over each individual subinterval, LDG method can handle the possible changes from interval to interval. Additionally, the methods are locally conservative which makes them stable and high-order accurate.

To illustrate how to use LDG method, we will go through how it is used with the stochastic heat equation. Here is the original problem:

$$\dot{u} = \nabla^2 u + \dot{W}$$

This equation can be broken into a system of PDE's:

$$\begin{cases} p = \nabla u \\ \dot{u} = \nabla \cdot p + \dot{W} \end{cases}$$

Let,

$$V_k := \{u \in L^2(D_k) : u|_K \in Q_k(K), \forall K \in T\}$$
$$M_k := \{p \in (L^2(D_k))^d : p|_K \in (Q_k(K))^d, \forall K \in T\}$$

From here, the system of equations can be modified into their respective weak formulations:

$$\int_{D_k} p \cdot w + \int_{D_k} u\nabla \cdot w - \int_{\delta D_k} uw \cdot \mathbf{n} = 0 \tag{3.1}$$

$$\int_{D_k} \dot{u}v = \int_{D_k} p \cdot \nabla v - \int_{\delta D_k} vp \cdot \mathbf{n} + \int_{D_k} \dot{W}v \tag{3.2}$$

where v and w are weight functions. Let $u = \sum_{j=1}^{n} u^j\phi_j$ and $p = \sum_{j=1}^{n} p^j\psi_j$, where $w \in M_k$, and $v \in V_k$. T is a triangulation, $\mathbf{n}$ is the normal vector, and $Q_k(K) := \{\text{polynomials of degree at most } k \text{ in each variable K}\}$, for $k \in \mathbb{N}$. We restrict the definitions of w and v to subspaces spanned by $\{\psi_j\}$ and $\{\phi_j\}$, respectively. From here, we can approximate the above equations with the approximations that we have just defined.

If we refer back to Equation 3.2, we see that it now becomes:

$$\int_{D_k} (\sum_{j=1}^{n} \dot{u}^j\phi_j)\phi_i dx = \int_{D_k} (\sum_{j=1}^{n} p^j\psi_j) \cdot \nabla\phi_i dx - \sum_e \int_e \{\phi_j\}[[p^j]]de + \int_{D_k} \dot{W}\phi_j dx$$

where $[[p^j]] = p^+\mathbf{n}^+ + p^-\mathbf{n}^-$ and $\{\phi_j\} = \frac{\phi_j^+ + \phi_j^-}{2}$. The e stands for edges. So, this means that the discontinuous part of the LDG method is approximated in that interval on the edges around the node. As you can see, we are looking at what the functions do at each side of the node inside that interval. This is how the discontinuous node is approximated.

Now, here is the approximation for Equation 3.1. You will see the same kind of approximation involved:

$$\int_{D_k} (\sum_{j=1}^{n} p^j\psi_j)\psi_i dx + \int_{D_k} (\sum_{j=1}^{n} u^j\phi_j) \cdot \nabla\psi_i dx - \sum_e \int_e [[u^j]]\{\psi_j\}de = 0$$

where $[[u^j]] = u^+\mathbf{n}^+ + u^-\mathbf{n}^-$ and $\{\psi_j\} = \frac{\psi_j^+ + \psi_j^-}{2}$, much like before.

From here we can set up a system of equations:

$$\begin{cases} M\dot{u} = B^T p + Z \\ Ap = -Bu \end{cases} \tag{3.3}$$

where $M_{j,i} = \int_{D_k} \phi_j\phi_i dx \in \mathbb{R}^{n\times n}$, $B_{j,i} = \int_{D_k} \phi_j \nabla \cdot \psi_i dx + \sum_e \int_e [[u^j]]\{\psi_j\} \in \mathbb{R}^{n\times dn}$, $Z_j = \int_{D_k} \dot{W}(x)\phi_j dx \in \mathbb{R}^n$, and $A_{j,i} = \int_{D_k} \psi_j\psi_i dx \in \mathbb{R}^{dn\times dn}$. The d stands for what dimension p is in. So, this means that we have two square matrices, one of which is bigger than the other, and two rectangular matrices. These create a block matrix:

$$\begin{bmatrix} M\frac{d}{dt} & -B^T \\ B & A \end{bmatrix} \begin{bmatrix} U \\ P \end{bmatrix} = \begin{bmatrix} Z \\ 0 \end{bmatrix}$$

Since A is a diagonal matrix, we can rewrite this matrix equation as:

$$SU = Z \tag{3.4}$$

where $S = M\frac{d}{dt} + B^T A^{-1} B$.

Now, let $\tilde{u}_{\Delta x}$ be the solution from the LDG approximation of equation 2.7 without the white noise and let $\tilde{U}$ be the corresponding matrix coefficient. Then, we have the following:

Proposition 3.1. *Let $\tilde{u}_{\Delta x}$ be the solution of the LDG approximation to the equation 2.7 without the white noise term, i.e. $\tilde{u}_{\Delta x}$ satisfies*

$$\begin{cases} \dot{\tilde{u}}_{\Delta x} = \alpha \nabla^2 \tilde{u}_{\Delta x} \ in \ D, \\ \tilde{u}_{\Delta x}|_{\delta D} = 0. \end{cases}$$

Then $E(u_{h,\Delta x}(x)) = \tilde{u}_{\Delta x}(x), \forall x \in D.$

Proof. It follows from the definitions of $\dot{\overline{W}}_h$ and Z that $E(Z) = 0$. Hence,

$$\tilde{U} = E(U) = E(S^{-1}Z) = S^{-1}E(Z) = 0.$$

The reason why we can pull S^{-1} out of the expected value computation is because all of the elements of S (and therefore all the elements of S^{-1}) are functions defined throughout Ω. Since the expected value is computed on a probability space, the only function that this affects is the one that contains the noise term.

So, the conclusion follows that $E(u_{h,\Delta x}(x)) = \tilde{u}_{\Delta x}(x), \forall x \in D$ since without the white noise, our matrix equation becomes $S\tilde{U} = 0$. □

This shows us that even though the original equation does contain a stochastic term, it does converge to a solution without a white noise term. So, since $\dot{\tilde{u}}_{\Delta x} = \alpha \nabla^2 \tilde{u}_{\Delta x}$ is a well-posed problem (since it does not have the noise term), and since equation 2.7 has been shown to have an expected value equal to that of the equation above, we can say that equation 2.7 has a unique solution.

Now that we have approximated the white noise using piece-wise constant functions and approximated the solution to the stochastic heat equation using LDG method, we need to use a numerical method to solve the block matrix equation that we have ended up with.

4. Numerical Methods

In the future, we are interested in exploring different numerical methods to solve this problem. In the coming year before my graduation, I hope to be able to explore these different methods. For now, we have given an explanation of a numerical method that could be used to solve the stochastic heat equation. We explain the reason why this is a good method to use as well as discuss the values that result from it.

At this point, we want to look deeper into what makes up each matrix. First, let's look at M. Recall that $M_{j,i} = \int_{D_k} \phi_j \phi_i dx$. Each of the ϕ_i's and ϕ_j's are only defined within their own support. So, if there is no support within the mesh for them, they are equal to zero, just like they would in FEM. This means that M is a banded matrix with elements only in the center bands (how many depends on the definition of the mesh). The same idea applies to B and A. Hence, our block matrix is banded and sparse. This means that we need to be cautious about what method we use to solve this matrix equation because we could possibly increase the complexity while working towards solving it.

There are several different methods that we could use to solve the matrix system. However, there are also some things that we need to consider first before we choose what method to use. First, we already have a sparse matrix, with only three bands of information that we need to work with. When we separate the matrix down, we do not want to make the decomposition dense. We want to be able to take advantage of the fact that it is sparse and keep it that way without greatly increasing the complexity of the problem. The next thing that we will want to examine is the mesh that we are approximating over. If we were to approximate over an even mesh, our matrix would also be symmetric. This makes solving the system even easier. However, in most cases we have an uneven mesh. In those cases, our matrix is most likely not symmetric but remains sparse.

One possible way of solving the matrix system is using sparse Gaussian Elimination. For generality purposes, we will assume that the matrix was created on a non-uniform mesh and hence, the matrix is not symmetric. However, the matrix is symmetric in design. It is tri-banded but still sparse. There are many programs available that can create the LU decomposition for a sparse matrix. We will analyze MATLAB's way of handling a sparse matrix but there are other codes, such as MA28 from Harwell for unsymmetric sparse matrices [1] and SPARSPAK from the University of Waterloo for symmetric sparse matrices [3], that also work with sparse matrices.

In Gaussian Elimination, the idea is to break the matrix up into two separate matrices:

$$PH = LU$$

where H is the block matrix shown earlier, P is the permutation matrix, L is the unit lower triangular with all the entries satisfying $|l_{ji}| \leq 1$, and U is the upper triangular matrix. From here, we have a much easier system to solve. Since P is a permutation matrix, we know that $P^TP = PP^T = 1$, hence it is nonsingular and $P^{-1} = P^T$. This means that we now have $A = P^TLU$. So, instead of solving $Hx = b$, where $x = \begin{bmatrix} U \\ P \end{bmatrix}$ and $b = \begin{bmatrix} Z \\ 0 \end{bmatrix}$, we will solve $P^TLUx = b$. We can do this by successively solving $P^T\hat{b} = b,\ Ly = \hat{b}$, and $Ux = y$.

MATLAB calculates the LU-decomposition using gaussian elimination with partial pivoting. The reason why I chose to use MATLAB as my example is because of the way it handles sparse matrices. It stores the matrix such that it only knows the non-zero values. This way, it is able to save a large amount of space. The function in MATLAB that calculates the LU-decomposition is [L,P,U] = lu(H). Since our H would be stored in MATLAB using a sparse memory allocation, it will preform this operation and save its resulting matrices also in a sparse format. It is because of its ability to save memory that I chose MATLAB as the example.

In MATLAB, we created an example problem in which the structure of the matrix it comes up with is much like what ours would look like if done in one dimension. Below is a snap shot of both matrix H and $L + U$ after the operation:

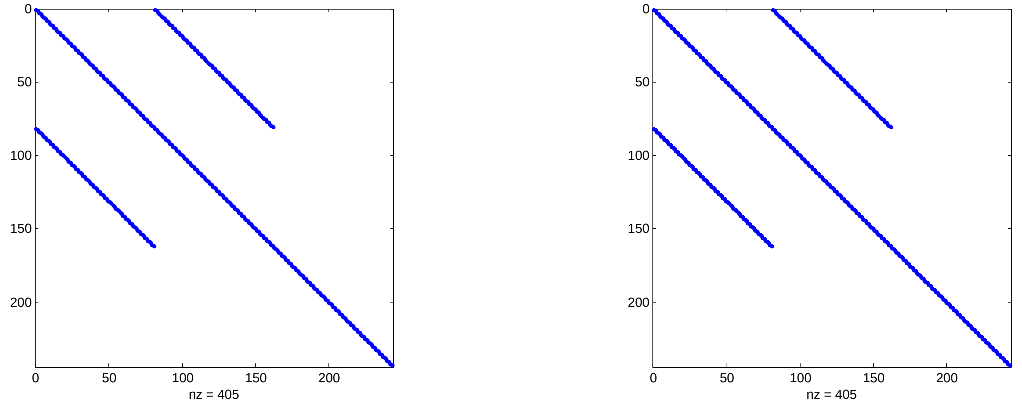

FIGURE 2. The left image is H and the right image is L+U

As you can see, they look exactly the same. The numbers that make up the two are different, but this shows that MATLAB was able to keep the sparseness of the matrix H through the decomposition. This is important because it keeps the complexity of the problem low.

As we have seen earlier, our block matrix is calculated directly through our basis functions. As for the white noise term (the Z vector in the matrix equation), a random number generator would be needed to provide input for the white noise term to calculate the individual integrals. This means that a new random number would need to be created for each node and D_k space. After calculating each of the integrals a set number of times and therefore creating our matrix system, we can begin the calculations that were dictated earlier using MATLAB. After the solution has been made, they all need to be averaged together so as to reach our final approximation of the stochastic heat equation. This approximation will actually be a system of ordinary differential equations (ODE) (one for each D_k ”box” and node) because time is kept as a continuous variable. This is sufficient as long as the solution is not needed for adaptivity in space. This will theoretically create our approximation of the stochastic heat equation and complete the problem.

5. Concluding Remarks

In this paper, we gave a brief introduction to stochastic processes and SDE’s as well as their importance in modern mathematics. We used these definitions to help introduce the stochastic heat equation which was the main focus of the paper. We then gave our approximation of the noise term and proved its convergence in two dimensions using an example stochastic elliptic PDE. After this example, we explained how this approximation works well with the stochastic heat equation as well. From there, we introduced LDG method and used it to approximate our SPDE. After coming up with our block matrix, we were able to show that its expected value did converge to the approximation of the heat equation without a noise term and hence showed that it did have a unique solution. From there, we discussed Sparse Gaussian Elimination as a numerical way to solve our system and showed how it kept the sparse nature of the matrix and therefore kept the complexity of the problem down.

In future work, we are interested in looking into different numerical methods to solve the stochastic heat equation including programming a numerical method and showing that the solution does converge. There is also the possibility of looking into higher order stochastic differential equations, which is part of the doctoral work of Moe Ebrahimi.

6. Acknowledgments

I would like to thank Professor Michael Holst and Moe Ebrahimi for all of their support and guidance through this past year.

MH was supported in part by NSF Awards 0715146 and 0915220, by DOE Award DE-FG02-04ER25620, and by DOD/DTRA Award HDTRA-09-1-0036. ME was supported in part by NSF Award 0715146.

References

[1] Ma28-a set of fortran subroutines for sparse unsymmetric linear equations. *AERE Harwell Laboratories*, (Report No. AERE-R. 8730).

[2] F. Black and M. Scholes. The pricing options and corporate liabilities. *Journal of Political Economy*, 81:637–659, 1973.

[3] A. George and E. Ng. A new release of sparspak: the waterloo sparse matrix package. *ACM SIGNUM Newsletter*, 19(4), 1984.

[4] R. ming Yao and L. jun Bo. Discontinuous galerkin method for elliptic stochastic partial differential equations on two and three dimensional spaces. *Science in China Series A: Mathematics*, 50(11), 2007.

[5] N. Nygaard. Introduction to stochastic process. *University of Chicago*. Lecture notes.

E-mail address: `maebrahi@math.ucsd.edu`

E-mail address: `mholst@math.ucsd.edu`

E-mail address: `hmilesle@ucsd.edu`

DEPARTMENT OF MATHEMATICS, UNIVERSITY OF CALIFORNIA SAN DIEGO, LA JOLLA CA 92093