Indian Institute of
Technology,Roorkee

Department of Mathematics

Project report on

Numerical Approximation for Stochastic Burger's Equation Using Finite Difference Scheme

Submitted By:
Aditya Ganeshan

Under the Guidance of:
Dr. Ankik K. Giri

Acknowledgment

I would like to extend my sincere thanks to **Dr. Ankik K. Giri** , Department of Mathematics, IIT Roorkee for giving me the opportunity to work under his guidance. I am highly indebted to him for being my mentor and for all the support and guidance he provided me throughout the semester.

Aditya Ganeshan

Contents

1 Introduction

1.1 Burger's Equation

Burgers' equation is a fundamental partial differential equation from fluid mechanics. It occurs in various areas of applied mathematics, such as modeling of gas dynamics and traffic flow. It is named for Johannes Martinus Burgers (1895–1981). It relates to the Navier-Stokes equation for incompressible flow with the pressure term removed. We will first see its derivation from Navier-Stokes equation.

1.1.1 Background

The Navier-Stokes equations describes the motion of fluid substances. They are non-linear partial differential equations that can be used to model a range of systems from water owing in pipes to hot dense astrophysical plasma. Unfortunately, the nonlinearity makes most problems difficult or impossible to solve and leads to phenomena such as turbulence. The Navier-Stokes equations are given by –

$$\nabla \cdot \nu = 0 \tag{1}$$

$$(\rho\nu)_t + \nabla \cdot (\rho\nu\nu) + \nabla p - \mu\nabla^2\nu = 0 \tag{2}$$

Where μ is the kinematic viscosity, ν is the velocity of the fluid parcel, p is

the pressure, and ρ isthe fluid density.

Simplification in above equation of the x-component of the velocity vector, which we will call ν^x, gives

$$\rho\frac{\partial\nu^x}{\partial t} + \rho\nu^x\frac{\partial\nu^x}{\partial x} + \rho\nu^y\frac{\partial\nu^x}{\partial y} + \rho\nu^z\frac{\partial\nu^x}{\partial z} + \frac{\partial p}{\partial x} - \mu\left(\frac{\partial^2\nu^x}{\partial x^2} + \frac{\partial^2\nu^x}{\partial y^2} + \frac{\partial^2\nu^x}{\partial z^2}\right) = 0 \tag{3}$$

If we consider a 1D problem with no pressure gradient, the above equation reduces to

$$\rho\frac{\partial\nu^x}{\partial t}+\rho\nu^x\frac{\partial\nu^x}{\partial x}-\mu\frac{\partial^2\nu^x}{\partial x^2}=0 \tag{4}$$

Now, if we now use the traditional variable $u(x,t)$ rather than ν^x and take the kinematic viscosity v, we get the Burger's equation

$$\frac{\partial u(x,t)}{\partial t}+u(x,t)\frac{\partial x(x,t)}{\partial x}=v\frac{\partial^2 u(x,t)}{\partial x^2} \tag{5}$$

1.1.2 Basic Derivation

Consider a mass element of a continuous substance. We assume now that the dynamics is only taking place in one dimension. Following Newton we have then for the mass element, dm, with cross-section area A, moving in the x -direction with velocity $u(x,t)$:

$$\begin{aligned}F_x&=dma_x\\F_x&=dm\frac{du_x}{dt}\\F_x&=\rho\frac{du_x}{dt}Adx.\end{aligned}$$

The velocity can change with time in two ways. First, the velocity varies in time at a certain point in space, and in addition the mass element changes its position with time and thereby, if the velocity is space-dependent, the velocity changes:

$$\begin{aligned}F_x&=\rho\frac{du_x}{dt}Adx\\F_x&=\Big(\rho\frac{\partial u_x}{\partial t}+\rho\frac{\partial x}{\partial t}\frac{\partial u_x}{\partial x}\Big)Adx\\F_x&=\Big(\rho\frac{\partial u_x}{\partial t}+\rho u_x\frac{\partial u_x}{\partial x}\Big)Adx\end{aligned}$$

To apply Newton's equation we need to know the total force that works on the mass element. We can in principle have three different types of forces. First, there might be an external force (like the gravitational force). Second, a pressure gradient over the mass element gives a net force, and last there are intermolecular forces within the medium that can be approximated with the help of the concept of viscosity. Based on different forces, we obtain different forms of Burger's equation.

We first assume that all these forces are zero and obtain the result (skipping the index x on the velocity $u(x;t)$).

$$\frac{\partial u(x,t)}{\partial t} + u(x,t)\frac{\partial u(x,t)}{\partial x} = 0$$

Which is the inviscid (i.e. without viscosity) Burgers' equation. This equation is non-linear and it is the simplest equation for which the solutions can develop shock waves. Alternatively the equation can be written.

$$\frac{\partial u(x,t)}{\partial t} + \frac{1}{2}\frac{\partial}{\partial x}\left(u(x,t)^2\right) = 0$$

This is an equation of the general form

$$\frac{\partial u(x,t)}{\partial t} + \frac{\partial f(u)}{\partial x} = 0 \tag{6}$$

With viscosity, Burgers' equation is

$$\frac{\partial u(x,t)}{\partial t} + u(x,t)\frac{\partial u(x,t)}{\partial x} = v\frac{\partial^2 u(x,t)}{\partial x^2} \tag{7}$$

where the v is the kinematic viscosity. The kinematic viscosity is the (more common) dynamic viscosity divided by the density. The dynamic viscosity has dimension $ML^{-1}T^{-1}(Nsm^{-2}$ in SI-units),while the kinematic viscosity has dimension L^2T^{-1}.

1.1.3 Burger's Equation in Traffic Flow

Consider the flow of cars on a highway and let $\rho(x,t)$ denote the density of cars and $f(x,t)$ the traffic flow. We will also consider ρ^* to be the restriction of ρ to a certain range, $0 \leq \rho^* \leq \rho_{max}$, where ρ_{max} is the value at which cars are bumper to bumper. Since cars are conserved, the density of cars and the flow must be related by the continuity equation

$$\frac{\partial \rho^*}{\partial t^*} + \frac{\partial f}{\partial x^*} = 0 \tag{8}$$

Obviously, the first expression in which one thinks for the flow is $f = v\rho^*$ where v is the velocity. However, it turns out that in order to reflect the fact that drivers will reduce their speed to account for an increasing density ahead we should suppose that f is a function of the density gradient as well. A simple assumption is to take

$$f(\rho^*) = \rho^* v(\rho^*) - D\frac{\partial \rho^*}{\partial x^*}, \tag{9}$$

Where, D is a constant.
We are assuming also that the velocity v is a given function of ρ^*: On a highway we would optimally like to drive at some speed v_{max} (the speed limit perhaps) but with heavy traffic we slow down, with velocity decreasing as density increases. The simplest relation that is aware of this is

$$v(\rho^*) = \frac{v_{max}}{\rho_{max}}(\rho_{max} - \rho^*) \tag{10}$$

Putting the above two equations in the first one leads to

$$\frac{\partial \rho^*}{\partial t^*} + \frac{d}{dx^*}\left[\frac{v_{max}}{\rho_{max}}(\rho_{max} - \rho^*)\rho^*\right] = D\frac{\partial^2 \rho^*}{\partial x^{*2}}$$

Scaling through $v_{max} = x_0/t_0, \rho = \rho_{max}\rho^*, x = x_0 x^* and dt = t_0 t^*$

$$\rho_t + [(1-\rho)\rho]_x = \epsilon\rho_{xx} \quad with \quad \epsilon = \frac{D}{v_{max}x_0} \quad and \quad 0 \leq \rho \leq 1$$

The transformation $u = 2\rho - 1$ leads to the viscid Burgers with the conditions $-1 \leq u \leq 1$.

1.1.4 More Examples of Burger's Equation

Burger's equation is now being used in many fields where the problem can be modeled as flow of particles in a fluid.

"Burger's equation as model for electricity spot price behavior" by Lukyanova Ksenia studies the usability of one-dimensional fluid model, Burger's equation for simulating the behavior of electricity spot prices.

"Sales rank, Burgers-like equation, least-recently-used caching" by Kumiko Hattori and Tetsuya Hattori summarizes about the stochastic ranking process (move-to-front rule) and the sales ranks of online stores, as well as a system of Burgers-like partial differential equations and least-recently-used caching, in order to give an overview of relations among various topics in different fields.

1.2 Stochastic Processes

1.2.1 Introduction

A **stochastic process** is a process whose value changes over time in an uncertain way, and thus we only know the distribution of the possible values of the process at any time point. This is the probabilistic counterpart to a deterministic process. Instead of describing a process which can only evolve in one way (as in the case, for example, of solutions of an ordinary differential equation), in a stochastic or random process there is some indeterminacy: even if the initial condition (or starting point) is known, there are several (often infinitely many) directions in which the process may evolve.

Definition

A stochastic process is a family of random variables $\xi(t)$ parameterized by $t \in T$, where $T \subset R$. When $T = \{1, 2, \ldots\}$, we shall say that $\xi(t)$ is a stochastic process in discreet time (i.e. a sequence of random variables). When T is an interval in R, we shall say that $\xi(t)$ is a stochastic process in continuous time.

Why do we need to study stochastic processes?

Mathematics is also the study of real-life problems. While making a mathematical model for any given problem, it is essential that the model is as close to the reality as possible. It has been seen that practical observations show some "noise" deviations from the mathematical solution of models(which do not incorporate any stochastics). To take these deviations under consideration for making the model more realistic, we introduce stochastic processes to the model.

Many processes have no deterministic solution (not yet !), as they incorporate randomness in themselves. To study such processes we need stochastic modeling.

Hence, to make better models, and to understand random processes we need to study stochastic processes.

1.2.2 Introduction to Wiener's Process (Brownian Motion)

Wiener process is a continuous-time stochastic process named in honor of Norbert Wiener. It is often called standard **Brownian motion**, after Robert Brown. It occurs frequently in pure and applied mathematics, economics, quantitative finance, and physics.

History

R. Brown in 1826–27 observed the irregular motion of pollen particles suspended in water. He and others noted that the path of a given particle is very irregular, having a tangent at no point, And the motions of two distinct particles appear to be independent.

In 1900 L. Bachelier attempted to describe fluctuations in stock prices mathematically and essentially discovered first certain results later rederived and extended by A. Einstein in 1905. N. Wiener in the 1920's (and later) put the theory on a firm mathematical basis.

Defination

A real valued stochastic process $W(.)$ is called a Brownian Motion or Weiner Process if

1. $W(0) = 0$
2. $W(t) - W(s)$ is $\sim N(0, t-s)$ for all $t \geq s \geq 0$
3. For all $0 < t_1 < t_2 < ... < t_n$, the random variables $W(t_1), W(t_2) - W(t_1), ..., W(t_n) - W(t_{n-1})$ are independent.

The derivative of a Weiner process $W(t)$ with respect to time is known as a " 1-dimensional white noise ", $\xi(t)$.

$$\dot{W}(t) = \frac{dW(t)}{dt} = \xi(t) \tag{11}$$

1.2.3 Examples of Stochastic processes

Stochastic Heat Equation

In its simplest form, the stochastic heat equation is a heat equation driven by Additive white noise -

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + \dot{W}, \tag{12}$$

Where $\dot{W}$ is a standard space-time white noise. White noise is a distribution, not a function, so this has to be interpreted in the distribution sense. With proper initial and boundary conditions, the associated initial-boundary-value problem has a unique solution. This solution is continuous, but very rough, in fact nowhere differentiable.

Stochastic Burger's Equation

We consider the Burger's equation

$$\frac{\partial u(t,x)}{\partial t} = \frac{\partial^2 u(t,x)}{\partial x^2} + u(t,x)\frac{\partial u(t,x)}{\partial x} + f(u(t,x)) + \frac{\partial W(t,x)}{\partial t \partial x} \tag{13}$$

for $t \in [0,T], x \in [0,1]$, with Dirichlet boundary condition

$$u(t,0) = u(t,1) = 0, \quad t > 0$$

and Initial condition

$$u(0,x) = u_0(x) = 0, \quad x \in [0,1]$$

Here, $\partial W(t,x)/\partial t \partial x$ is a space-time white noise.
In this project we have taken $f(u(t,x)) = 0$ for all $u(t,x)$

1.3 Numerical Analysis

Numerical analysis is the study of algorithms that use numerical approximation (as opposed to general symbolic manipulations) for the problems of mathematical analysis (as distinguished from discrete mathematics).

1.3.1 Finite Difference Method

The finite difference method is one of several techniques for obtaining numerical solutions to Equation. In all numerical solutions the continuous partial differential equation (PDE) is replaced with a discrete approximation. In this context the word 'discrete' means that the numerical solution is known only at a finite number of points in the physical domain. The number of those points can be selected by the user of the numerical method. In general, increasing the number of points not only increases the resolution (i.e., detail), but also the accuracy of the numerical solution.

First, assuming the function whose derivatives are to be approximated is properly-behaved, by Taylor's theorem, we can create a Taylor Series expansion

$$f(x_0+h) = f(x_0) + \frac{f'(x_0)}{1!}h + \frac{f^{(2)}(x_0)}{2!}h^2 + ... + \frac{f^{(n)}(x_0)}{n!}h^n + R_n(x) \quad (14)$$

where, $R_n(x)$ is a remainder term, denoting the difference between the Taylor polynomial of degree n and the original function. We will derive an approximation for the first derivative of the function f by first truncating the Taylor polynomial

$$f(x_0+h) = f(x_0) + f'(x_0)h + R_1(x),$$

Setting, $x_0 = a$ we have,

$$f(a+h) = f(a) + f'(a)h + R_1(x),$$

Dividing across by h gives

$$\frac{f(a+h)}{h} = \frac{f(a)}{h} + f'(a) + \frac{R_1(h)}{h}$$

Solving for $f'(a)$:

$$f'(a) = \frac{f(a+h) - f(h)}{h} - \frac{R_1(h)}{h}$$

Assuming that $R_1(h)$ is sufficiently small, the approximation of the first derivative of f is

$$f'(a) \approx \frac{f(a+h) - f(h)}{h} \tag{15}$$

1.3.2 Finite Difference for Heat Equation

For the one dimensional heat equation

$$\frac{\partial \phi}{\partial t} = \alpha \frac{\partial^2 \phi}{\partial x^2}, \quad 0 \leq x \leq L, \quad t \geq 0 \tag{16}$$

where $\phi = \phi(x;t)$ is the dependent variable, and α is a constant coefficient. The above equation is a model of transient heat conduction in a slab of material with thickness L. The domain of the solution is a semi-infinite strip of width L that continues indefinitely in time. The material property α is the thermal diffusivity. In a practical computation, the solution is obtained only for a finite time, say t_{max}.

Solution to the above equation requires specification of boundary conditions at $x = 0$ and $x = L$, and initial conditions at $t = 0$. Simple boundary and initial conditions are

$$\begin{aligned} \phi(0,t) &= \phi_0 \\ \phi(L,t) &= \phi_L \\ \phi(x,0) &= f_0(x). \end{aligned}$$

The finite difference method obtains an approximate solution for $\phi(x,t)$ at a finite set of x and t. The discreet x are uniformly spaced in the interval $0 \leq x \leq L$ such that

$$x_i = (i-1)\Delta x, \quad i = 1, 2, ..., N$$

Where N is the total number of spatial nodes, including those on the boundary. Given L and N, the spacing between the x_i is computed with

$$\Delta x = \frac{L}{N-1}$$

Similarly, the discrete tare uniformly spaced in $0 \leq t \leq t_{max}$:

$$t_m = (m-1)\Delta t, \quad m = 1, 2, ..., M$$

Where M is the number of time steps and Δt is the size of a time step

$$\Delta t = \frac{t_{max}}{M-1}$$

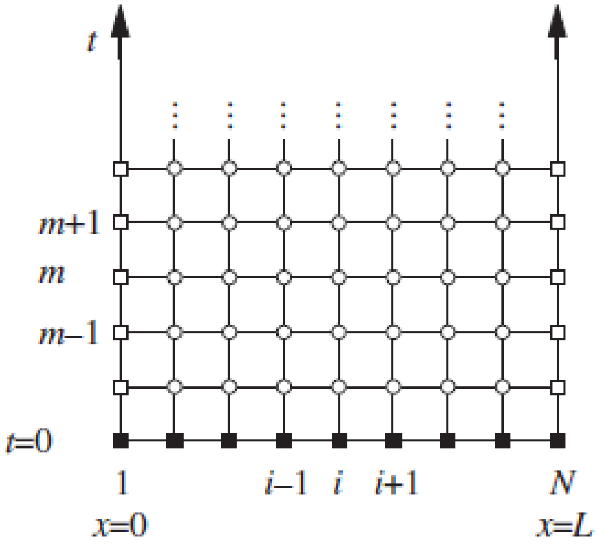

Figure 1: FDM Grid

Mesh on a semi-infinite strip used for solution to the one-dimensional heat equation. The solid squares indicate the location of the (known) initial values. The open squares indicate the location of the (known) boundary values. The open circles indicate the position of the interior points where the finite Difference approximation is computed.

First Order Forward Difference

$$\frac{\partial \phi}{\partial x}\bigg|_{x_i} = \frac{\phi_{i+1} - \phi_i}{\Delta x} + O(\Delta x) \tag{17}$$

First Order Backward Difference

$$\frac{\partial \phi}{\partial x}\bigg|_{x_i} = \frac{\phi_i - \phi_{i-1}}{\Delta x} + O(\Delta x) \tag{18}$$

First Order Central Difference

$$\frac{\partial \phi}{\partial x}\bigg|_{x_i} = \frac{\phi_{i+1} - \phi_{i-1}}{2\Delta x} + O(\Delta x^2) \tag{19}$$

Second Order Central Difference

$$\frac{\partial^2 \phi}{\partial x^2}\bigg|_{x_i} = \frac{\phi_{i+1} - 2\phi_i + \phi_{i-1}}{\Delta x^2} + O(\Delta x^2) \tag{20}$$

1.3.3 Schemes for applying FDM(Finite-Difference Method)

Forward Time, Centered Space (FTCS)

$$\frac{\phi_i^{m+1} - \phi_i^m}{\Delta t} = \alpha \frac{\phi_{i+1}^m - 2\phi_i^m + \phi_{i-1}^m}{\Delta x^2} + O(\Delta t) + O(\Delta x^2)$$

This Gives us

$$\phi_i^{m+1} = \phi_i^m + \frac{\alpha \Delta t}{\Delta x^2}(\phi_{i+1}^m - 2\phi_i^m + \phi_{i-1}^m) \tag{21}$$

Backward Time, Centered Space (FTCS)

$$\frac{\phi_i^m - \phi_i^{m-1}}{\Delta t} = \alpha \frac{\phi_{i+1}^m - 2\phi_i^m + \phi_{i-1}^m}{\Delta x^2} + O(\Delta t) + O(\Delta x^2)$$

This Gives us

$$-\frac{\alpha}{\Delta x^2}\phi_{i-1}^m + \left(\frac{1}{\Delta t} + \frac{2\alpha}{\Delta x^2}\right)\phi_i^m - \frac{\alpha}{\Delta x^2}\phi_{i+1}^m = \frac{1}{\Delta t}\phi_i^{m-1} \tag{22}$$

Crank-Nicolson Scheme (CNS)

$$\frac{\phi_i^m - \phi_i^{m-1}}{\Delta t} = \frac{\alpha}{2}\Big[\frac{\phi_{i+1}^m - 2\phi_i^m + \phi_{i-1}^m}{\Delta x^2} + \frac{\phi_{i+1}^{m-1} - 2\phi_i^{m-1} + \phi_{i-1}^{m-1}}{\Delta x^2}\Big]$$

This Gives us

$$\begin{aligned}-\frac{\alpha}{2\Delta x^2}\phi_{i-1}^m + \Big(\frac{1}{\Delta t} + \frac{\alpha}{\Delta x^2}\Big)\phi_i^m - \frac{\alpha}{2\Delta x^2}\phi_{i+1}^m = \\ -\frac{\alpha}{2\Delta x^2}\phi_{i-1}^{m-1} + \Big(\frac{1}{\Delta t} + \frac{\alpha}{\Delta x^2}\Big)\phi_i^{m-1} - \frac{\alpha}{2\Delta x^2}\phi_{i+1}^{m-1}\end{aligned} \tag{23}$$

The FTCS can yield unstable solutions that oscillate and grow if Δt is too large. The Crank- Nicolson scheme has a truncation error of $O(\Delta t^2) + O(\Delta x^2)$, i.e., the temporal truncation error is significantly smaller than the temporal truncation error of the BTCS scheme. Crank-Nicolson and BTCS give us a system of equations of N variables at each time step. Both BTCS and Crank- Nicolson are unconditionally stable.

Due to decrease in error and stability, we have used Crank-Nicolson Scheme in this project.

2 Finite-Difference Method on Burger's Equation

2.1 Non-Stochastic Burger's Equation

Burger's equation has just one additional term when compared with the standard heat equation, i.e. the non-linear term. Hence, we can use the scheme used for heat equation for all the linear terms in the Burger's equation. The equation is given by,

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \nu\frac{\partial^2 u}{\partial x^2} \tag{24}$$

A mesh with time step and length step is made in the same way as it was made for the heat equation.

Forward time Centered Space

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} + \frac{u_j^n(u_{j+1}^n - u_{j-1}^n)}{2\Delta x} - \nu\frac{u_{j-1}^n - 2u_j^n + u_{j+1}^n}{\Delta x^2} = 0 \tag{25}$$

However, FTCS does not give the correct solution when viscosity coefficient is very low. To get stable solution we need to apply Crank-Nicolson Scheme.

Crank-Nicolson Scheme

First, we write the burgers equation in the following form-

$$\frac{\partial u}{\partial t} + \frac{\partial F}{\partial x} = \nu\frac{\partial^2 u}{\partial x^2} \tag{26}$$

Where $F = u^2/2$.

Now, Applying CNS -

$$\frac{\Delta u_j^{n+1}}{\Delta t} = \frac{L_x\big(F_j^n + F_j^{n+1}\big)}{2} + v\frac{L_{xx}\big(u_j^n + F_j^{n+1}\big)}{2} \tag{27}$$

Where,

$$\Delta u_j^{n+1} = u_j^{n+1} - u_j^n$$

And,

$$L_x = \frac{(-1,0,1)}{2\Delta x} \quad and \quad L_{xx} = \frac{(1,-2,1)}{\Delta x^2}$$

To solve the non-linear term, we apply Taylor series expansion on it –

$$F_j^{n+1} = F_j^n + \Delta t\left[\frac{\partial F}{\partial t}\right]_j^n + \frac{\Delta t^2}{2}\left[\frac{\partial^2 F}{\partial t^2}\right]_j^n + ... \tag{28}$$

$$F_j^{n+1} = F_j^n + A\Delta u_j^{n+1} + O(\Delta t^2) \quad where \quad A = \left[\frac{\partial F}{\partial u}\right]_j^n = u_j^n$$

On applying these expansions and canceling the F_j^n n term, we get the following equation

$$u_j^{n+1} + \frac{1}{2}\Delta t\big[L_x(u_j^n u_j^{n+1}) - \nu L_{xx} u_j^{n+1}\big] = u_j^n + \frac{1}{2}\nu \Delta t L_{xx} u_j^n \tag{29}$$

This can be written as

$$a_j^n u_{j-1}^{n+1} + b_j^n u_j^{n+1} + c_j^n u_{j-1}^{n+1} = d_j^n \tag{30}$$

Where,

$$\begin{aligned}
&a_j^n = -\frac{\Delta t}{4\Delta x} u_{j-1}^n - \frac{s}{2} \\
&b_j^n = 1 + s \\
&c_j^n = -\frac{\Delta t}{4\Delta x} u_{j+1}^n - \frac{s}{2} \\
&d_j^n = 0.5 s u_{j-1}^n + (1-s) u_j^n + \frac{s}{2} u_{j+1}^n \\
&s = \frac{\nu \Delta t}{\Delta x^2}
\end{aligned} \tag{31}$$

Hence the problem has been reduced to a system of equation at each time step. The problem of non-linearity has been solved.

2.2 Stochastic Burger's equation

The Stochastic Burger's equation is given by –

$$\frac{\partial u(t,x)}{\partial t} = \frac{\partial^2 u(t,x)}{\partial x^2} + \frac{1}{2}\frac{\partial}{\partial x}\big(u^2(x,t)\big) + \frac{\partial^2 \widetilde{W}}{\partial t \partial x} \tag{32}$$

In the previous sections we have dealt with the discretization of all of the terms in the above equation other than the additive white Gaussian noise term.

Now, we will apply FDM on that term.

$$\frac{\partial^2 \widetilde{W}}{\partial t \partial x}$$

This term denotes the mixed second-order derivative of the Brownian sheet.
$W(x,t)$ = Time-space weiner process or Brownian sheet.
We approximate the white noise by the formula

$$\frac{\partial^2 \widetilde{W}}{\partial t \partial x} \approx \frac{\partial^2 \widehat{W}}{\partial t \partial x} \tag{33}$$

Consider a grid of Brownian sheet as below :-

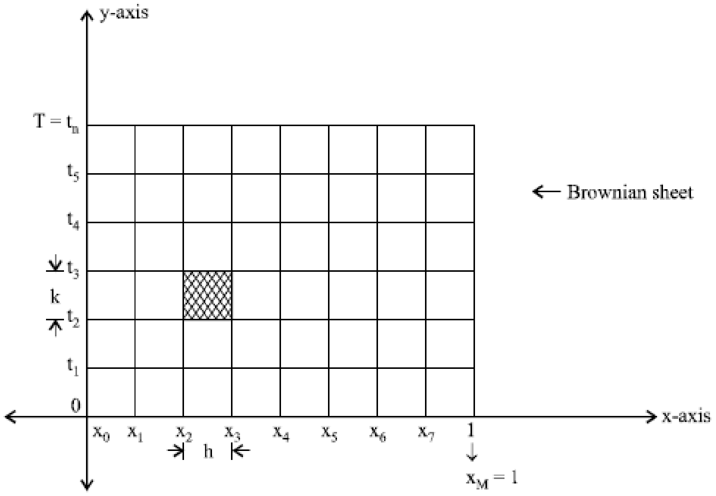

Figure 2: Grid for Brownian Sheet

where h =space-step size,
k = time-step size
We approximate $\partial^2 W(t,x)/\partial t\partial x$ in each small rectangular block by

$$\eta_{i,j} \sim N(0,1) \quad \text{Normal distribution} \tag{34}$$

Now consider two dimensional Gaussian white noise $\partial^2 W(t,x)/\partial t\partial x$ where $W(t,x)$ is Brownian motion on a plane half-strip or a Brownian sheet. Properties of the Brownian sheet are described by Walsh. Several important properties of the Brownian sheet useful –

1. If $s = (t,x) : a \leq t \leq b, c \leq x \leq d$ is a rectangle, then

$$\int_c^d \int_a^b dW(t,x) = \int_c^d \int_a^b \frac{\partial^2 W(t,x)}{\partial t \partial x} dt \cdot dx \tag{35}$$
$$= W(s) \tag{36}$$
$$= W(b,d) - W(a,d) - W(b,c) + W(a,c) \tag{37}$$

   Where $W(s)$ is Gaussian with zero mean and variance $|s|$ and $|s|$ is the area of s.

2. If χ_s is the characteristic function of rectangle s, then

$$\int_0^T \int_a^b \chi_s dW(t,x) = W(s), \quad \text{for} \quad s \subset (0,T) \times (a,b) \tag{38}$$

3. If $E\Big(\int_0^T \int_a^b f^2(t,x) dx dt \Big) < \infty$, then

$$E\Bigg(\int_0^T \int_a^b f(t,x) dW(t,x) \Bigg)^2 = E\Bigg(\int_0^T \int_a^b f^2(t,x) dW(t,x) \Bigg) \tag{39}$$

Now we define the partition of $[0,T] \times [0,1]$ by rectangles $[t_i, t_{i+1}] \times [x_j, x_{j+1}]$ for $i = 1,2,...,N$ and $j = 1,2,...,M$ where $t_i = (i-1)\Delta t$, $x_j = (j-1)\Delta x, \Delta t = T/N$, and $\Delta x = (x_f - x_i)/M$.

Then a reasonable approximation on this partition is given by

$$\frac{\partial^2 \widehat{W}}{\partial t \partial x}(t,x) = \frac{1}{\Delta t \cdot \Delta x} \sum_{i=1}^{N} \sum_{j=1}^{M} \eta_{ij} \sqrt{\Delta t \cdot \Delta x}\ \chi_i(t) \cdot \chi_j(x) \tag{40}$$

Where χ_i is the characteristic function for the i^{th} sub-interval and

$$\eta_{ij} = \frac{1}{\sqrt{\Delta t \cdot \Delta x}} \int_{t_i}^{t^{i+1}} \int_{x_j}^{x^{j+1}} dW(t,x)$$

Hence,

$$\frac{\partial^2 \widehat{W}}{\partial t \partial x}(t_n, x_j) = \frac{1}{\sqrt{kh}} \cdot \eta_{ij} \cdot \eta_{ij} \sim N(0,1) \tag{41}$$

Finally, we get

$$a_j^n u_{j-1}^{n+1} + b_j^n u_j^{n+1} + c_j^n u_{j-1}^{n+1} = d_j^n \tag{42}$$

Where,

$$\begin{aligned}
a_j^n &= -\frac{\Delta t}{4\Delta x} u_{j-1}^n - \frac{s}{2} \\
b_j^n &= 1 + s \\
c_j^n &= -\frac{\Delta t}{4\Delta x} u_{j+1}^n - \frac{s}{2} \\
d_j^n &= 0.5 s u_{j-1}^n + (1-s) u_j^n + \frac{s}{2} u_{j+1}^n + \Delta t \times \frac{1}{\Delta x \Delta t} \times \eta_{nj} \\
s &= \frac{\nu \Delta t}{\Delta x^2} \\
\eta_{nj} &\sim N(0,1)
\end{aligned} \tag{43}$$

2.2.1 Regularization Parameter

We introduce a new parameter r called the Regularizing parameter. r Controls the noise added to the burger's equation.

$$d_j^n = 0.5 s u_{j-1}^n + (1-s) u_j^n + \frac{s}{2} u_{j+1}^n + \Delta t \times \frac{1}{\Delta x \Delta t} \times \eta_{nj} \times r \tag{44}$$

r varies from 0 to 1.

3 Implementation

3.1 Crank-Nicolson on Heat Equation

For $\alpha = 0.0625$,
$x_i = 0, x_f = 1$, $t_i = 0, t_f = 1$
Initial condition $= u(x, o) = u(x) = 100 \times sin(2 \times pi \times x)$
Boundary conditions $= u(x_i, t) = u(x_f, t) = 0$.

3.1.1 Code

Constant_sin.m

```
%Heat equation solution using FDM.
%First ,constants.m file has Constants.
% inputs by user -
x0 = 0;
x1 = 1;
t0 = 0;
t1 = 1;
dx = 1 / 20;
dt = 1 / 20;
alpha = 0.0625;
% Constants to be calculated -
nx = (x1-x0)/dx;
nt = (t1-t0)/dt;
a = -alpha/(2*(dx^2));
b = 1/dt + alpha/(dx^2);
c = -alpha/(2*(dx^2));
%matrices.
U = zeros(nt,nx);
D = zeros(nx);
Ut = zeros(nx);
```

Mat_AB.m

```
% file handleing creation of matrice A and A^-1.
% calling file where constants are declared.
constants_sin;
% making matrix A
A = zeros(nt,nx);
A(1,1) = 1;
A(nt,nx) = 1;
for i=2:1:nt-1
A(i,i-1) = a;
A(i,i) = b;
A(i,i+1) = c;
end
% making matrix B which is inverse of A
B = inv(A);
```

Conditions_sin.m

Mat_AB.m

```
%File for Initial Conditions and boundary condition.
%calling previous file which calls previous file and so on...
MatAB_sin;
%initial condition.
% at t=t0, U(x,t0) = Ut0(x)
% Ut0(x) = 100*sin(2*pi*x)
for i=1:1:nx
x =(x0 +(i)*dx);
U(1,i) = 100*sin(2*pi*x);
end
%boundary conditions.
% at x=xo, U(x0,t) = Ux0(t);
% at x=x1, U(x1,t) = Ux1(t);
% Ux0(t) = 0 ; Ux1 = 0;
for i=1:1:nt
%for x=x0
```

```
U(i,1) = 0;
%for x=x1
U(i,nx) = 0;
end
```

Main_FDM_sin.m

```
% File for solving the Heat Equation.
% calling previous file which calls previous file and so on...
Conditions_sin;
for i=1:1:nt-1
% making matrix D in AU=D
D(1) = 0;
D(nx) = 0;
for j=2:1:nx-1
D(j) = (-a)*U(i,j-1) + (1/dt +a*2)*U(i,j) -
c*U(i,j+1);
end
%making matrix Ut which is inv(A)* D
%as inv(A) * A U = inv(A) *D
Ut = B * D;
%setting values of Ut to U(x,t+1)
for j=2:1:nx-1
U(i+1,j) = Ut(j);
end
end
% making the surface
%for i=1:1:nt
% for j=1:1:nx
% T(t0+(i-1)*dt,x0+(i-1)*dx) = U(i,j);
% end
%end
x = linspace(0,1,20);
t = linspace(0,1,20);
surf(x,t,U);
title('Surface plot of heat equation solution');
xlabel('Distance x');
ylabel('Time t');
```

3.1.2 Surface Graph for heat equation solution

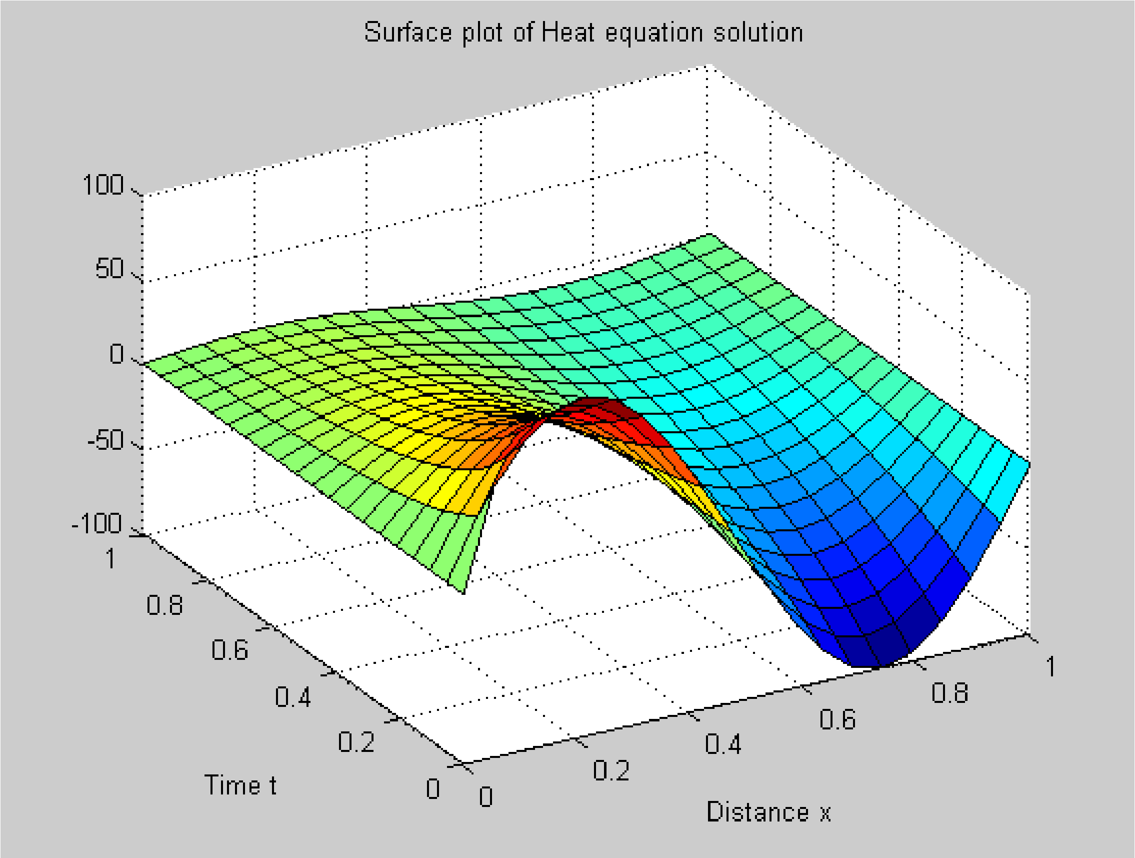

Figure 3: Surface Graph for heat equation solution.

3.2 Crank-Nicolson on Burger's Equation

Similar conditions as taken in heat equation problem.
Except, $\alpha = 0.01$
And $u(x, 0) = u(x) = sin(2 \times pi \times x)$

3.2.1 Code

Basic_CN.m

```
%Burger's equation solution
%using Crank-Nicolson Implicit finite-difference scheme
%constants
x0 = 0;
x1 = 1;
t0 = 0;
t1 = 1;
dx = 0.05;
dt = 0.05;
alpha = 0.01;
%constants
s = alpha * dt / dx / dx;
nx = (x1-x0)/dx;
nt = (t1-t0)/dt;
%matrices.
U = zeros(nt,nx);
D = zeros(nx);
Ut = zeros(nx);
A = zeros(nt,nx);
%CONDITIONS
%initial condition.
% at t=t0, U(x,t0) = Ut0(x)
% Ut0(x) = 100*sin(2*pi*x)
for i=1:1:nx
x =(x0 +(i)*dx);
U(1,i) = sin(2*pi*x);
end
%boundary conditions.
```

```
% at x=xo, U(x0,t) = Ux0(t);
% at x=x1, U(x1,t) = Ux1(t);
% Ux0(t) = 0 ; Ux1 = 0;
for i=1:1:nt
%for x=x0
U(i,1) = 0;
%for x=x1
U(i,nx) = 0;
end
```

FDM_CN_Burgers.m

```
%Scheme implimentation
Basic_CN
for i=1:1:nt-1
% making matrix D in AU=D
D(1) = 0;
D(nx) = 0;
for j=2:1:nx-1
D(j) = 0.5*s*U(i,j-1) + (1-s)*U(i,j) +
0.5*s*U(i,j+1);
end
%making matrix A in AU=D
A(1,1) = 1;
A(nt,nx) = 1;
for j=2:1:nx-1
A(j,j-1) = -1*dt/(4*dx)*U(i,j-1) - s/2;
A(j,j) = 1+s;
A(j,j+1) = dt/(4*dx)*U(i,j+1) - s/2;
end
%making Inverse matrix B
B = inv(A);
%Solution for time t = i+1
Ut = B * D;
%setting values of Ut to U(x,t+1)
for j=2:1:nx-1
U(i+1,j) = Ut(j);
```

```
end
end
%Making the surface
x = linspace(x0,x1,nx);
t = linspace(t0,t1,nt);
surf(x,t,U);
title('Surface plot of Burgers Equation solution');
xlabel('Distance x');
ylabel('Time t');
%Making the graphs
%making plot at t=0.1
%Ux = zeros(nx);
%ux = zeros(nx);
%for i=1:1:nx
% Ux(i)=U(20,i);
% ux(i)=U(40,i);
%end
%plot(x,Ux,x,ux);
```

3.2.2 Surface Graph for Burger's equation solution

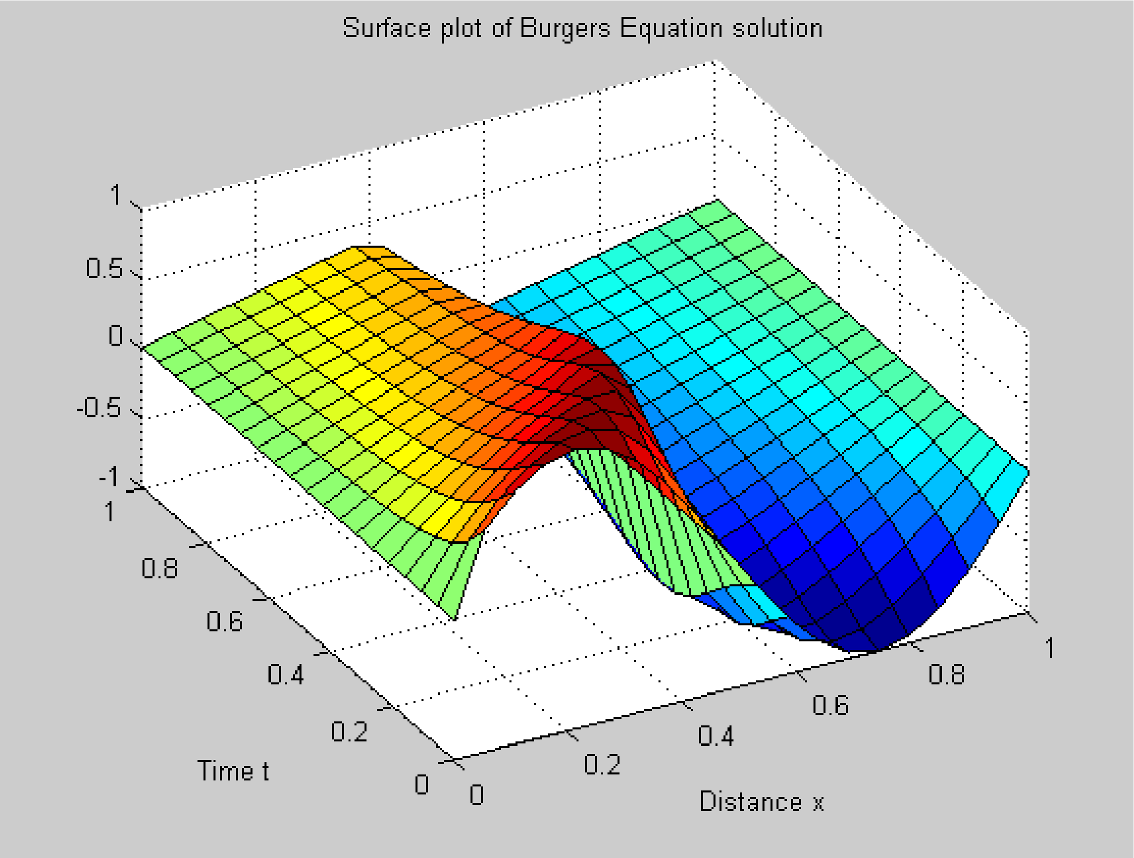

Figure 4: Surface Graph for Burger's equation solution.

3.3 Crank-Nicolson on Stochastic Burger's Equation

Taking the same conditions as burger's equation.

3.3.1 Code

Basic_CN_S.m

```
%Burger's equation solution
%using Crank-Nicolson Implicit finite-difference scheme
%constants
x0 = 0;
x1 = 1;
t0 = 0;
t1 = 1;
dx = 0.05;
dt = 0.05;
alpha = 0.01;
%r varies in each experiment
r = 1;
%constants
s = alpha * dt / dx / dx;
nx = (x1-x0)/dx;
nt = (t1-t0)/dt;
%matrices.
U = zeros(nt,nx);
D = zeros(nx);
Ut = zeros(nx);
A = zeros(nt,nx);
%CONDITIONS
%initial condition.
% at t=t0, U(x,t0) = Ut0(x)
% Ut0(x) = 100*sin(2*pi*x)
for i=1:1:nx
x =(x0 +(i)*dx);
U(1,i) = sin(2*pi*x);
end
%boundary conditions.
```

```
% at x=xo, U(x0,t) = Ux0(t);
% at x=x1, U(x1,t) = Ux1(t);
% Ux0(t) = 0 ; Ux1 = 0;
for i=1:1:nt
%for x=x0
U(i,1) = 0;
%for x=x1
U(i,nx) = 0;
end
```

FDM_CN_Burgers_S.m

```
%Scheme implimentation
Basic_CN_S
for i=1:1:nt-1
% making matrix D in AU=D
D(1) = 0;
D(nx) = 0;
for j=2:1:nx-1
D(j) = 0.5*s*U(i,j-1) + (1-s)*U(i,j) +
0.5*s*U(i,j+1);
%Adding Random Noise
randn('state',1 + i + j);
D(j) = D(j) + dt*((dx*dt)^(-1/2))*randn *r;
end
% W(j,m) can be made seperatly and added to D(j)
%Major PROBLEM.. W(nx) might not be zero.may be very far from
zero
dW = zeros(1,nx);
W = zeros(1,nx);
dW(1) = sqrt(dx)*randn;
W(1) = dW(1);
for j=2:nx
dW(j) = sqrt(dx)*randn;
W(j) = W(j-1) + dW(j);
%D(j) = D(j) + W(j)/10;
end
```

```
%making matrix A in AU=D
A(1,1) = 1;
A(nt,nx) = 1;
33
for j=2:1:nx-1
A(j,j-1) = -1*dt/(4*dx)*U(i,j-1) - s/2;
A(j,j) = 1+s;
A(j,j+1) = dt/(4*dx)*U(i,j+1) - s/2;
end
%making Inverse matrix B
B = inv(A);
%Solution for time t = i+1
Ut = B * D;
%setting values of Ut to U(x,t+1)
for j=2:1:nx-1
U(i+1,j) = Ut(j);
end
end
%Making the surface
x = linspace(x0,x1,nx);
t = linspace(t0,t1,nt);
surf(x,t,U);
title('Surface plot of Stochastic Burgers Equation solution');
xlabel('Distance x');
ylabel('Time t');
%Making the graphs
%making plot at t=0.1
%Ux = zeros(nx);
%ux = zeros(nx);
%for i=1:1:nx
% Ux(i)=U(20,i);
% ux(i)=U(40,i);
%end
%plot(x,Ux,x,ux);
```

3.3.2 Surface Graph for Stochastic Burger's equation solution

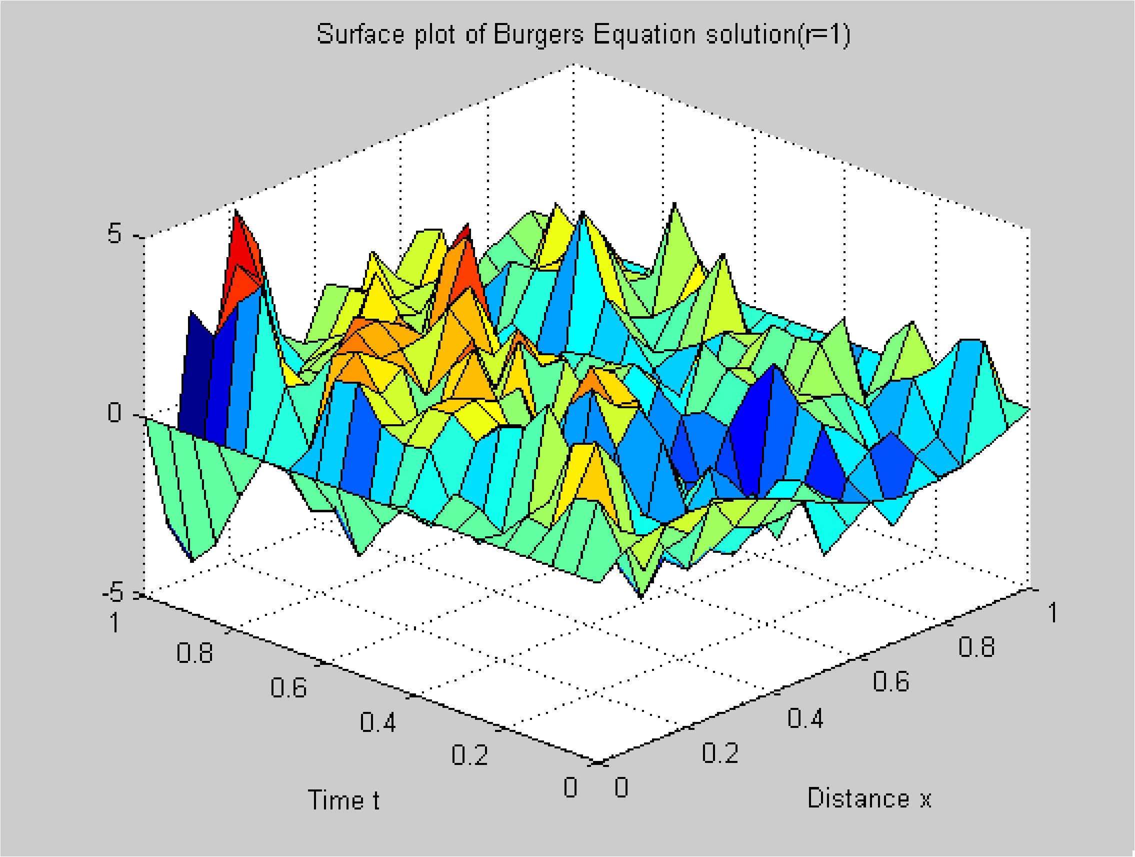

Figure 5: Surface Graph for Burger's equation solution with r = 1.

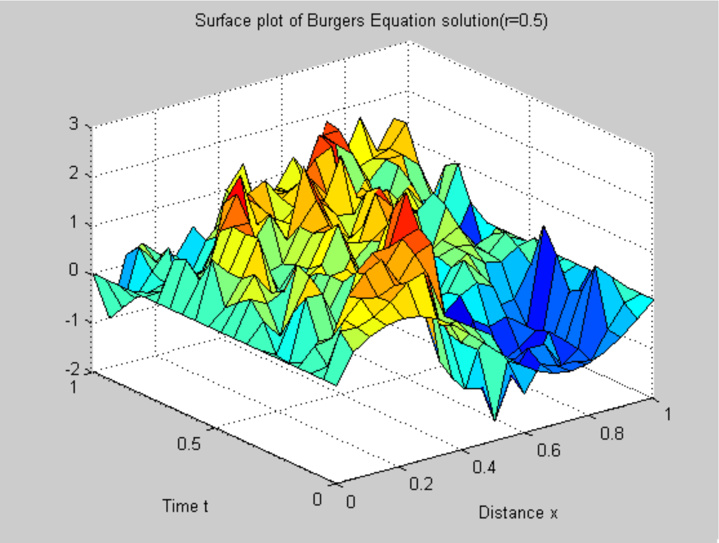

Figure 6: Surface Graph for Burger's equation solution with r = 0.5.

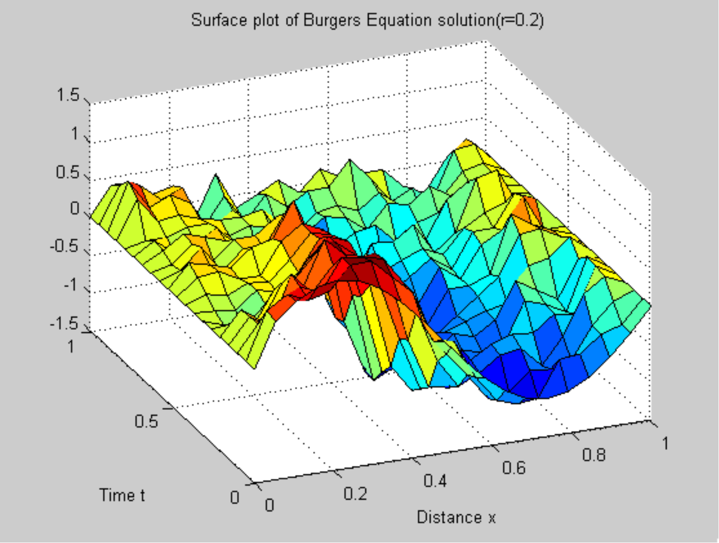

Figure 7: Surface Graph for Burger's equation solution with r = 0.2.

3.4 Comparison of Burger's equation and Stochastic Burger's equation Solutions

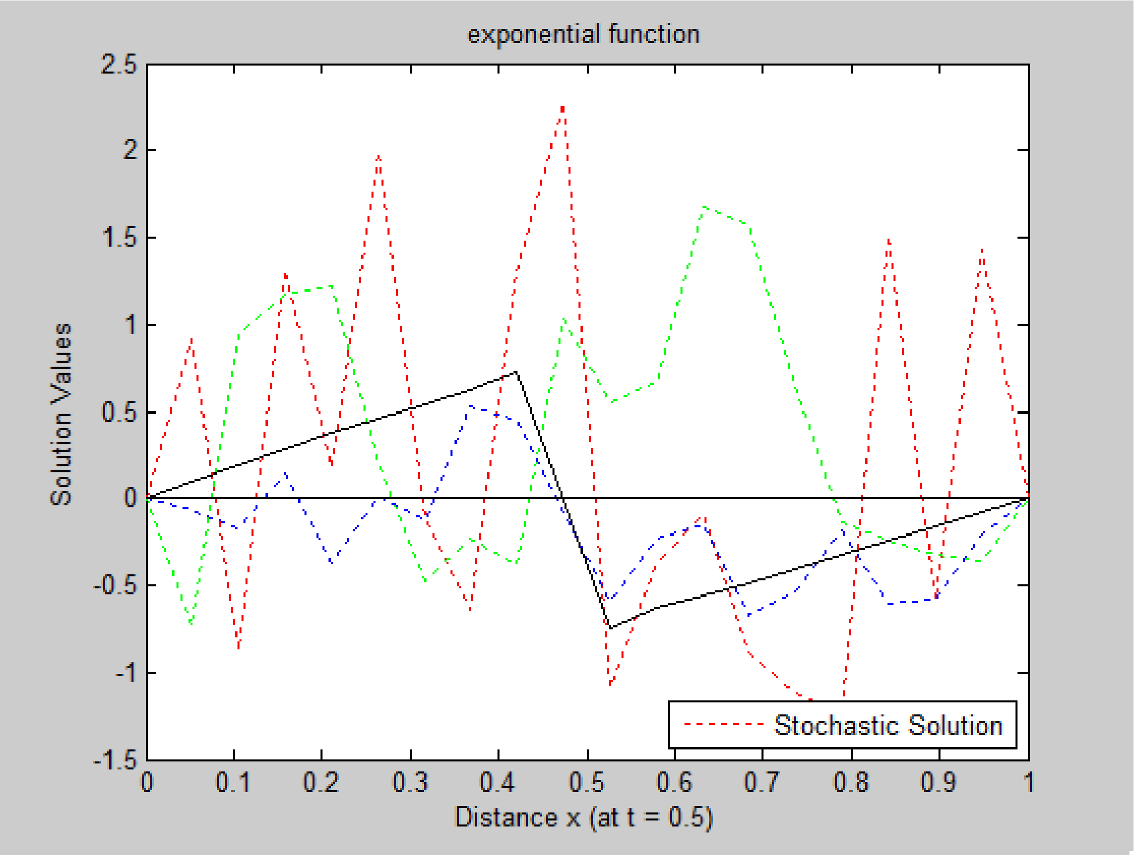

Figure 8: Surface Values at x = 0.5 for Various Systems

Red line ∼ r = 1
Green line ∼ r = 0.5.
Blue line ∼ r = 0.2.
Black line ∼ r = 0.

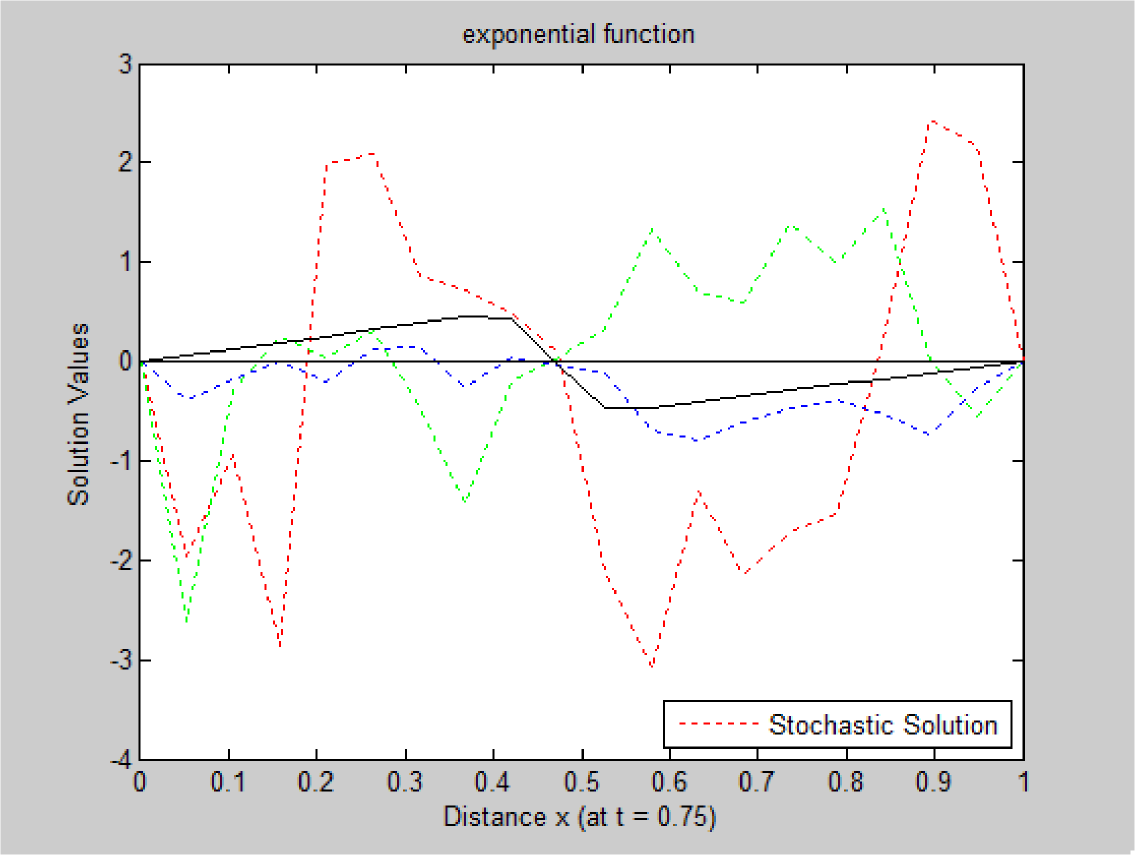

Figure 9: Surface Values at x = 0.75 for Various Systems

Red line $\sim$ r = 1
Green line $\sim$ r = 0.5.
Blue line $\sim$ r = 0.2.
Black line $\sim$ r = 0.

References

[1] I. G. Aureli Alabert, "On numerical approximation of stochastic burgers' equation," *From Stochastic Calculus to Mathematical Finance*, 2006.

[2] B. M. Patel, "Finite difference approximation for stochastic parabolic partial differential equations," Master's thesis, University of Chester, 2009.

[3] K. Hattori and T. Hattori, "Sales ranks, burgers-like equations, and least-recently-used caching," *Applications of renormalization group methods in mathematical sciences, RIMS Kôkyûroku Bessatsu B*, vol. 21, pp. 149–161, 2010.

[4] D. J. Higham, "An algorithmic introduction to numerical simulation of stochastic differential equations," *SIAM review*, vol. 43, no. 3, pp. 525–546, 2001.

[5] E. Alos, D. Nualart, and F. Viens, "Stochastic heat equation with white-noise drift," in *Annales de l'IHP Probabilités et statistiques*, vol. 36, pp. 181–218, 2000.

[6] G. W. Recktenwald, "Finite-difference approximations to the heat equation," -, 2004.

[7] L. C. Evans, *An introduction to stochastic differential equations.* American Mathematical Society, 2013.

[8] L. Ksenia, "Burgers' equation as a mo del for electricity spot price behavior," *From Stochastic Calculus to Mathematical Finance*, 2011.

[9] M. Hairer, "An introduction to stochastic pdes," -, 2009.

[10] M. Landajuela, "Burgers equation.," *BACM*, 2011.

[11] P. Howard, "Partial differential equations in matlab 7.0," -, 2005.

[12] T. Szabados, "An elementary introduction to the wiener process and stochastic integrals," *Studia Scientiarum Mathematicarum Hungarica*, vol. 31, pp. 249–297, 1996.

[13] Y. Chong and J. B. Walsh, "The roughness and smoothness of numerical solutions to the stochastic heat equation," *Potential Analysis*, vol. 37, no. 4, pp. 303–332, 2012.

[14] G. Da Prato, A. Debussche, and R. Temam, "Stochastic burgers' equation," *Nonlinear Differential Equations and Applications NoDEA*, vol. 1, no. 4, pp. 389–402, 1994.