# NUMERICAL CONVERGENCE STUDY FOR SPH SOLUTIONS

A thesis submitted in partial fulfillment for the
Bachelor of Technology

by

Mrinalgouda Patil

Under the guidance of

Prabhu Ramachandran

Department of Aerospace Engineering
Indian Institute of Technology, Bombay

April 2017

*“So many of our dreams at first seem impossible, then they seem improbable, and then, when we summon the will, they soon become inevitable”*

Christopher Reeve

# *Acknowledgements*

I am really thankful to Prof. Prabhu Ramachandran for his expert guidance and continuous encouragement throughout my work. I owe a lot for all the courses we took under him. I would also like to mention that his advice outside the Academics helped me to decide about my future. I would extend our gratitude to Prof. Kowsik Bodi, who suggested to pursue our computational interests under our guide. I also thank my friends Koushik, Vyas, Vinod, Vikas, Suraj, Shinde, Rushi and many others for their help in clearing doubts. Finally, I thank to the Almighty for his grace throughout this project. . . .

# Contents

# List of Figures

# List of Tables

# Abbreviations

| | |
|---|---|
| **ADKE** | **A**daptive **D**ensity **K**ernel **E**stimation |
| **SPH** | **S**moothed **P**article **H**ydrodynamics |

# Symbols

| | | |
|---|---|---|
| $W$ | Kernel | |
| $p$ | Pressure | (Nm$^{-2}$) |
| $\rho$ | Density | (kgm$^{-3}$) |
| $u$ | Velocity | (ms$^{-1}$) |
| $E$ | Energy | (Nm$kg^{-1}$) |
| $hdx$ | Kernel factor | |
| $\pi$ | Viscous-factor | N$m^4kg^{-2}$ |

# Chapter 1

# Introduction

Smoothed particle hydrodynamics (SPH) is a method for obtaining approximate numerical solutions of the equations of fluid dynamics by replacing the fluid with a set of particles. It was developed by Gingold and Monaghan (1977) and Lucy (1977) initially for astrophysical problems.The method has a number of attractive features. It is a tool to simulate the flow for various problems. SPH uses Lagragian frame based equations to solve the problem. It is a mesh less framework.

A lot of advantages are obtained by using SPH. The first of these is that pure advection is treated exactly, which we will see in this report through an example. For example, if the particles are given a colour, and the velocity is specified, the transport of colour by the particle system is exact. Modern finite difference methods give reasonable results for advection but the algorithms are not Galilean invariant so that, when a large constant velocity is superposed, the results can be badly corrupted. The second advantage is that with more than one material, each described by its own set of particles, interface problems are often trivial for SPH but difficult for finite difference schemes. The third advantage is that particle methods bridge the gap between the continuum and fragmentation in a natural way.

In today's world, SPH is used in many fields, particularly theoretical astrophysics and solid mechanics. We can use SPH to simulate various fluid problems like dam break problem, also it can be used to simulate the various gas-dynamics problems. In this report, we will go through the error analysis for schemes used in SPH to solve Partial Differential Equations. We start with the study of kernels and their convergence rates and then we look into the famous Inviscid

Burgers' equation. In the end, we will look into more on Sod Shock Tube which uses Euler equations to solve it.

Usually in SPH, convergence studies are rarely made as they happen to show bad results as compared to other computational techniques. This report gives an overview for the convergence studies in a basic level, which can be extended to a further level.

# Chapter 2

# Kernel Convergence

One of the central issues for the meshfree methods is how to effectively perform function approximation based on a set of nodes scattered in an arbitrary manner without using a predefined mesh or grid that provides the connectivity of the nodes. The SPH method employs the integral representation using a smoothing function. The smoothing function (also called smoothing kernel function, smoothing kernel or simply kernel in many literatures) is of utmost importance since it not only determines the pattern for the function approximation, defines the dimension of the support domain of particles, but also determines the consistency and hence the accuracy of both the kernel and particle approximations.

Different kernels have been used for computing better SPH approximations. Some of the requirements for the smoothing functions are listed as follows:

- The kernel must be normalized over its support domain.

$$\int_{\Omega} W(x - x', h) dx' = 1 \tag{2.1}$$

- The kernel should be compactly supported(Compact Support).

$$W(x - x') = 0, for |x - x'| > \kappa h \tag{2.2}$$

- The kernel should satisfy the Dirac delta function condition as the smoothing length approaches to zero (Delta function property).

- The smoothing function value for a particle should be monotonically decreasing with the increase of the distance away from the particle (Decay)
- The kernel should be an even function (Symmetric)

## 2.1 Kernels

In our study, we have considered seven different kernels namely,

- Cubic Spline
- Gaussian
- Quintic Spline
- Super Gaussian
- Wendland Quintic C2
- Wendland Quintic C4
- Wendland Quintic C6

### 2.1.1 Cubic Spline

The Cubic Spline kernel in 1D is given by:

$$\frac{2}{3h}(1.0 - 3/2q^2(1 - q/2)) \text{ if } q < 1.0 \tag{2.3}$$

$$\frac{2}{12h}(2-q)^3 \quad \text{if } 1.0 < q < 2.0 \tag{2.4}$$

$$0 \quad \text{if } q > 2 \tag{2.5}$$

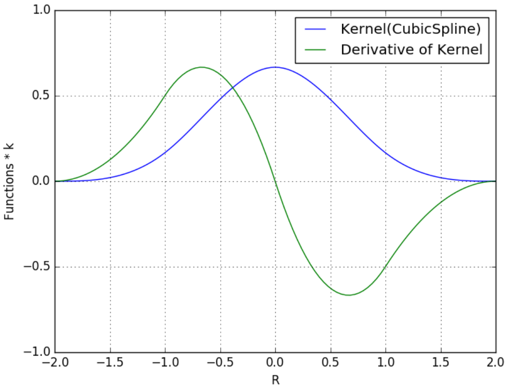


FIGURE 2.1: The Cubic Spline kernel and its derivative.

### 2.1.2 Gaussian

The Gaussian Kernel is given by:

$$\frac{1}{\sqrt{\pi}h}e^{-q^2} \tag{2.6}$$

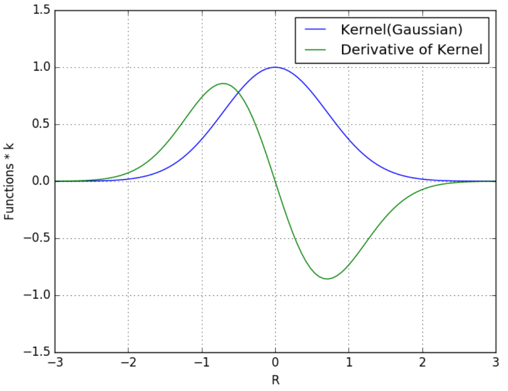


FIGURE 2.2: The Gaussian kernel and its derivative.

### 2.1.3 Quintic Spline

The Quintic Spline kernel in 1D is given by:

$$\frac{120}{h}((3-q)^5 - 6(2-q)^5 + 15(1-q)^5) \text{ if } q < 1.0 \tag{2.7}$$

$$\frac{120}{h}((3-q)^5 - 6(2-q)^5) \quad \text{if } 1.0 < q < 2.0 \tag{2.8}$$

$$\frac{120}{h}(3-q)^5 \quad \text{if } 2.0 < q < 3.0 \tag{2.9}$$

$$0 \quad \text{if } q > 3 \tag{2.10}$$

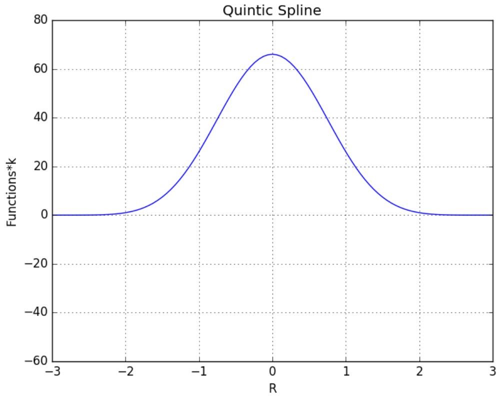


FIGURE 2.3: The Quintic Spline kernel.

### 2.1.4 Super Gauss

The Super Gaussian kernel in 1D is given by:

$$\frac{1}{\sqrt{\pi}h}(1.5 - q^2)e^{-q^2} \quad \text{if } q < 2.0 \tag{2.11}$$

$$0 \quad \text{if } q > 2 \tag{2.12}$$

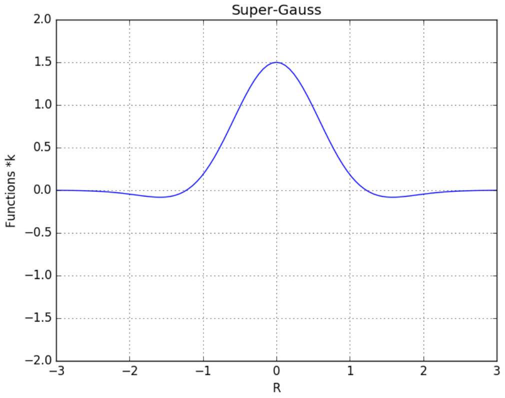


FIGURE 2.4: The Super Gaussian kernel.

### 2.1.5 Wendland Quintic C2 and C4

The Wendland C2 kernel in 1D is given by:

$$\frac{5}{8h}((1.0-0.5q)^3)(1.5q+1) \text{ if } q < 2.0 \tag{2.13}$$

$$0 \quad \text{if } q > 2 \tag{2.14}$$

The Wendland C4 kernel in 1D is given by:

$$\frac{3}{4h}((1-0.5q)^5)(2.5q+1+2.0q^2) \text{ if } q < 2.0 \tag{2.15}$$

$$0 \quad \text{if } q > 2 \tag{2.16}$$

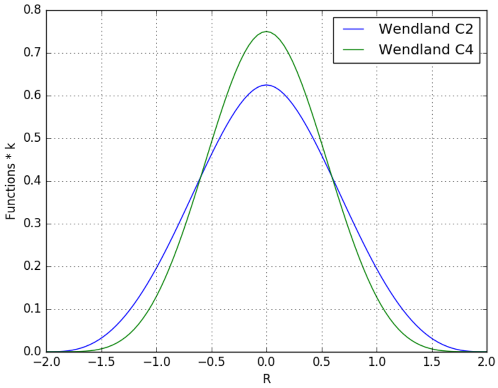


FIGURE 2.5: The Wendland C2 and C4 kernel.

### 2.1.6 Wendland Quintic C6

The Wendland C6 kernel in 1D is given by:

$$\frac{55}{64h}((1-0.5q)^7)(3.5q+1+\frac{19}{4}(q^2)+\frac{21}{8}(q^3)) \text{ if } q < 1.0 \tag{2.17}$$

$$0 \quad \text{if } q > 1 \tag{2.18}$$

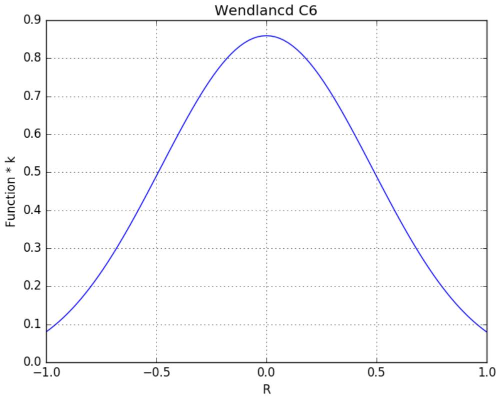


FIGURE 2.6: The Wendland C6 kernel.

## 2.2 Convergence Study

We will look into the convergence study for the 1D function approximation using the above mentioned kernels. We will consider two cases here, namely with or without noise.

### 2.2.1 With Noise

#### 2.2.1.1 Cubic Spline

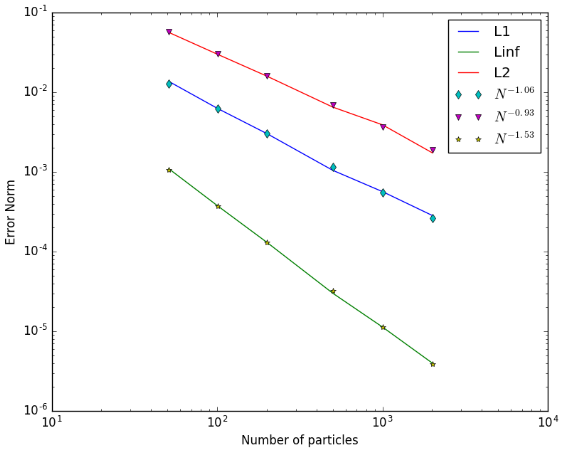


FIGURE 2.7: Error norm vs no. of particles for Cubic Spline Kernel

From the above figure, we see that infinity norm error converges with order 1 as we increase the number of particles, whereas the L2 norm converges at the order of 1.5.

#### 2.2.1.2 Gauss

The below figure shows the convergence rates for function approximation using Gaussian kernel. The results obtained are similar with that of Cubic Spline kernel.

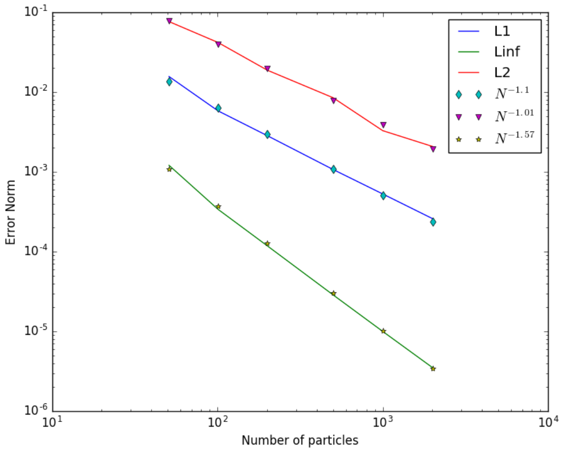


FIGURE 2.8: Error norm vs no. of particles for Gaussian Kernel

#### 2.2.1.3 Quintic Spline

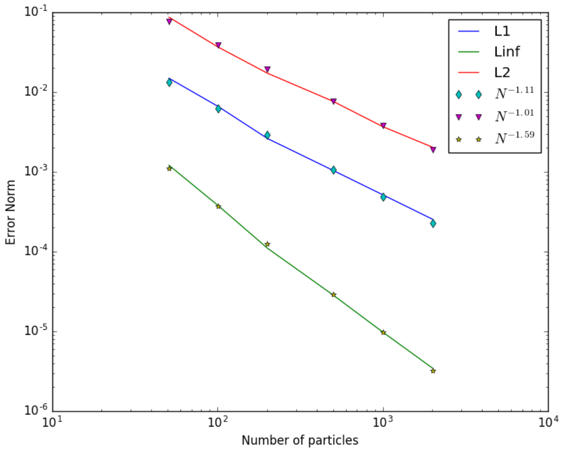


FIGURE 2.9: Error norm vs no. of particles for Quintic Spline Kernel.

From the above figure, we see that infinity norm error converges with order 1 as we increase the number of particles, whereas the L2 norm converges at the order of 1.5.

#### 2.2.1.4 Super Gauss

The below figure shows the convergence rates for function approximation using Gaussian kernel. The results obtained are similar with that of Quintic Spline kernel.

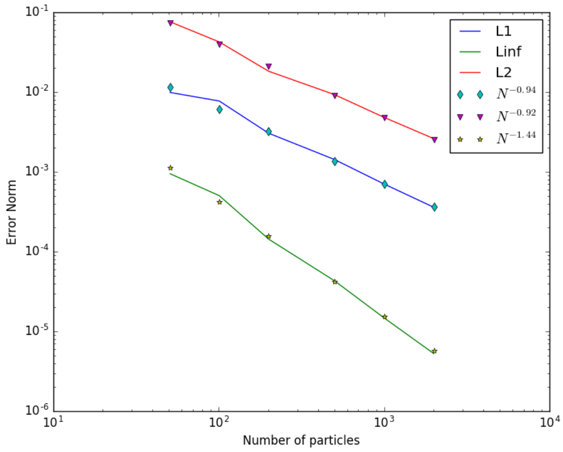


FIGURE 2.10: Error norm vs no. of particles for Super Gaussian Kernel.

#### 2.2.1.5 Wendland C2

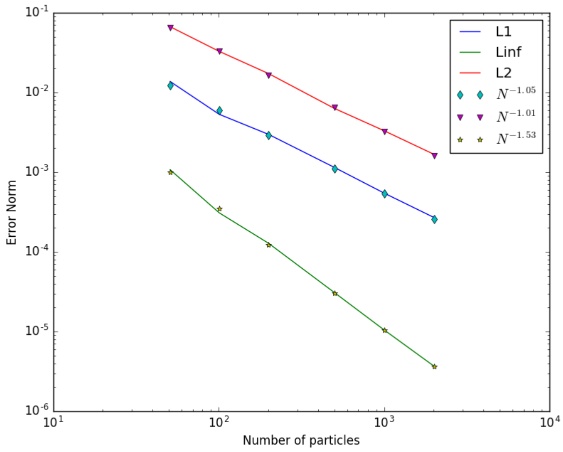


FIGURE 2.11: Error norm vs no. of particles for WC2 Kernel.

#### 2.2.1.6 Wendland C4

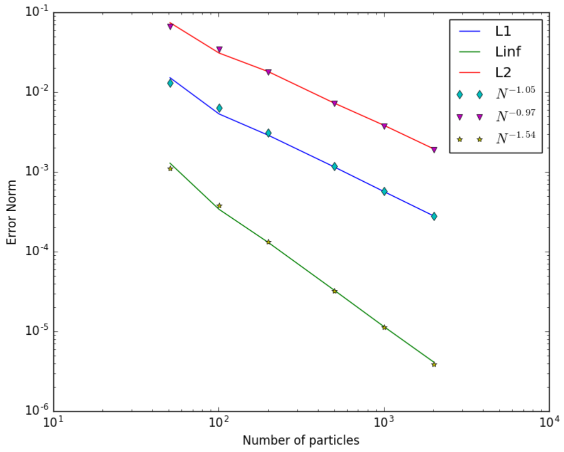


FIGURE 2.12: Error norm vs no. of particles for WC4 Kernel.

#### 2.2.1.7 Wendland C6

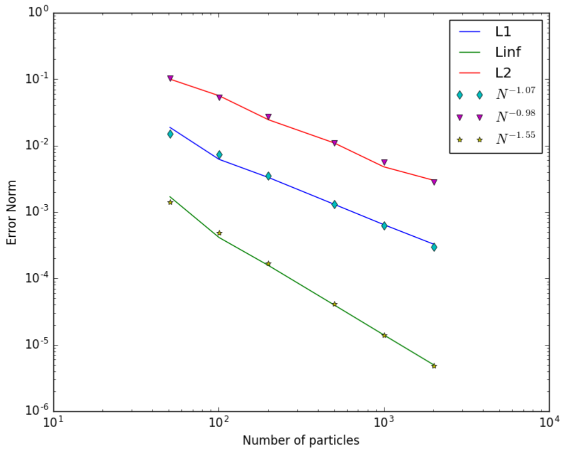


FIGURE 2.13: Error norm vs no. of particles for WC6 Kernel.

From all the above figures corresponding to Wendland Kernels, all the kernels show similar results as compared to that of other kernels. This is not the case when we remove noise, which we will see in next section.

### 2.2.2 Without Noise

#### 2.2.2.1 Cubic Spline

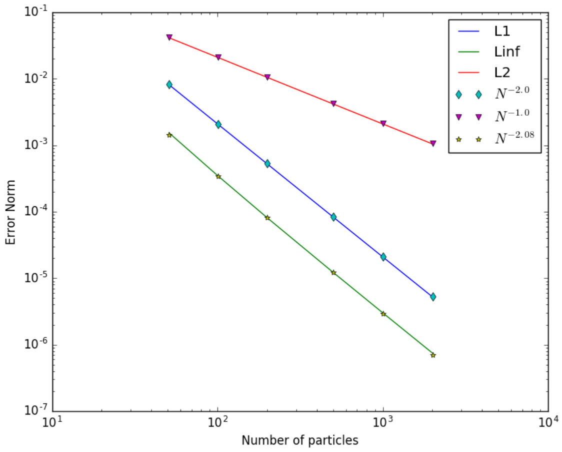


FIGURE 2.14: Error norm vs no. of particles for Cubic Spline Kernel

In the case of no noise added, the cubic spline shows a higher convergence rate as compared to the one with noise.

#### 2.2.2.2 Gauss

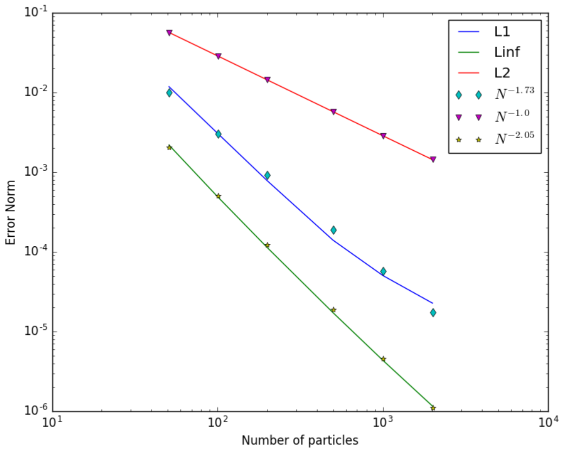


FIGURE 2.15: Error norm vs no. of particles for Gaussian Kernel

#### 2.2.2.3 Quintic Spline

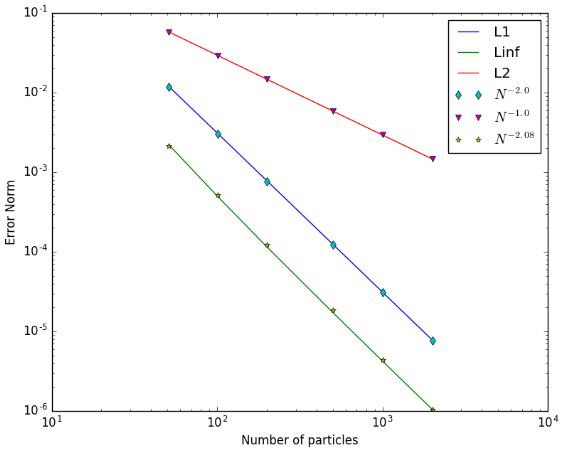


FIGURE 2.16: Error norm vs no. of particles for Quintic Spline Kernel.

#### 2.2.2.4 Super Gauss

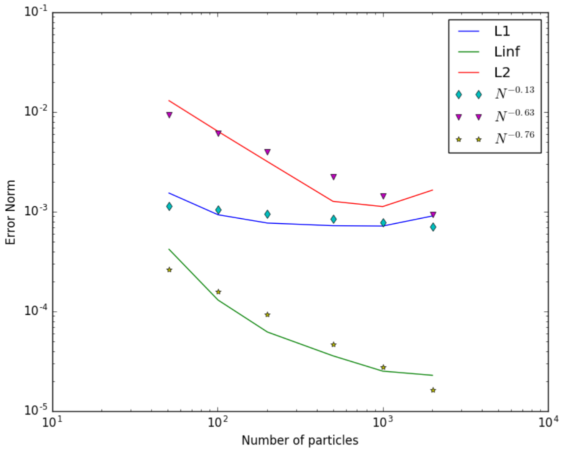


FIGURE 2.17: Error norm vs no. of particles for Super Gaussian Kernel.

### 2.2.2.5 Wendland C2

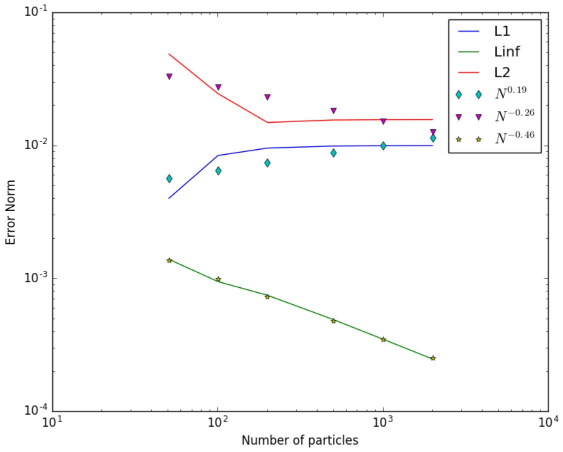


FIGURE 2.18: Error norm vs no. of particles for WC2 Kernel.

### 2.2.2.6 Wendland C4

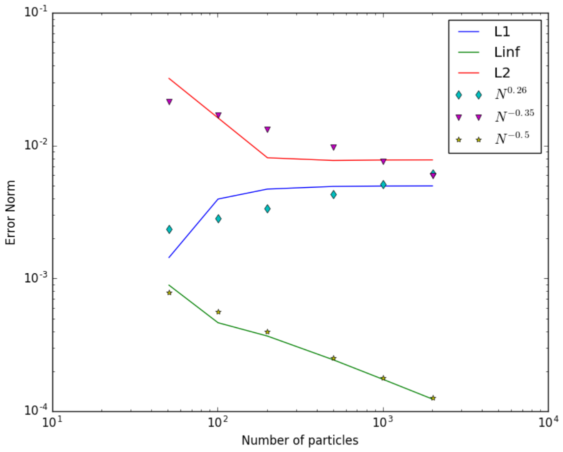


FIGURE 2.19: Error norm vs no. of particles for WC4 Kernel.

#### 2.2.2.7 Wendland C6

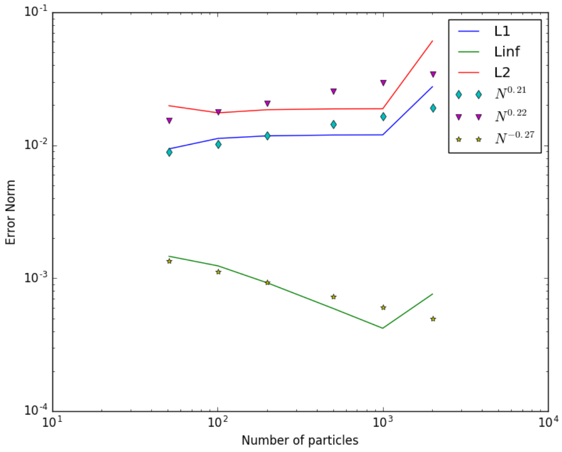


FIGURE 2.20: Error norm vs no. of particles for WC6 Kernel.

From all the above figures corresponding to Wendland Kernels, we can see that the results obtained are not good. There is no improvement in the solution as we increase the N, if we see it in terms of L1 error norm.

## 2.3 Summary

TABLE 2.1: Kernel Convergence rates -Without noise

| Kernel | Linf | L1 | L2 |
|---|---|---|---|
| Cubic Spline | 1 | 2 | 2.08 |
| Quintic Spline | 1 | 2 | 2.08 |
| Gauss | 1 | 1.72 | 2.05 |
| S. Gauss | 0.13 | 0.63 | 0.76 |
| WC2 | 0.46 | 0.19 | 0.26 |
| WC4 | 0.5 | 0.26 | 0.35 |
| WC6 | 0.27 | 0.21 | 0.22 |

TABLE 2.2: Kernel Convergence rates - With noise

| Kernel | Linf | L1 | L2 |
|---|---|---|---|
| Cubic Spline | 1.5 | 1.1 | 1.01 |
| Quintic Spline | 1.59 | 1.11 | 1.01 |
| Gauss | 1.57 | 1.1 | 1.01 |
| S. Gauss | 1.44 | 0.94 | 0.92 |
| WC2 | 1.53 | 1.01 | 1.03 |
| WC4 | 1.54 | 0.97 | 1.02 |
| WC6 | 1.53 | 0.98 | 1.01 |

- From the above plots and the tabular data, we can see that the convergence rates for kernels without noise is around 2 for spline and Gauss kernels.
- However the order of convergence is very less for Wendland Kernels, which have not much change in L1 error norm.
- Considering with noise addition, all kernels show similar results.

# Chapter 3

# Burgers' Equation

In this section, we will see how SPH is used to solve some of the basic differential equations namely inviscid Burgers' equation.

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = 0 \tag{3.1}$$

To solve the inviscid berger equation, we have to use the correction term known as XSPH correction for the velocity. The XSPH correction is shown in the below equation:

$$\frac{dx_j}{dt} = u_j + \varepsilon \sum_j \frac{m_j}{\rho_j}(u_j - u_i)W_{ij} \tag{3.2}$$

The $\varepsilon$ in the above expression is taken to be around 0.5 for solving problems.

Consider the initial condition given by:

$$u(x,0) = sin(2\pi x) \tag{3.3}$$

The initial condition of the problem along with the solution at various time steps is shown in the below figure.

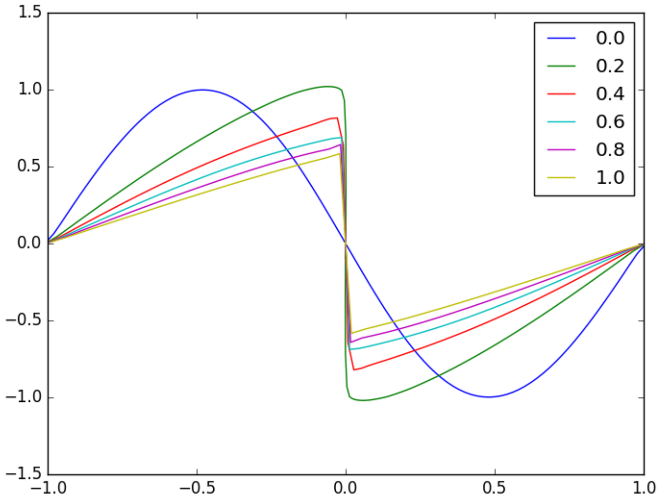


FIGURE 3.1: Solution at various times.

We have solved the above inviscid burger equation with the help of SPH, using different approaches. For all these approaches, a detailed convergence study has been performed. Since the exact solution for the above equation is available before the appearance of shock, all the results shown below are valid before the shock appears. Before explaining the approaches involved in computing the SPH solution, let's look at how we got to the exact solution.

## 3.1 Exact solution

Consider the inviscid Burgers equation below:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = 0 \tag{3.4}$$

$$\frac{\partial u}{\partial t} + \frac{1}{2}\frac{\partial u^2}{\partial x} = 0 \tag{3.5}$$

Given initial condition: $u(x,0) = sin(2\pi x)$

Exact solution is given by : $u(x,t) = sin(2\pi[x-ut])$

To compute the exact solution, we have used the Newton Raphson method for each time ’t’, since it behaves like a non-linear equation of one variable at constant ’t’. The following plot shows the exact solution of the inviscid Burgers’ equation:

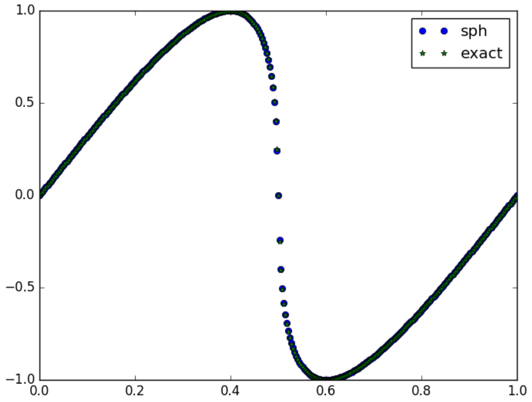


FIGURE 3.2: Exact and SPH solution for Inviscid Burgers’ equation

## 3.2 SPH solution

The following are the approaches used in computing the solution for the inviscid Burgers’ equation.

- Constant ’h’
- ADKE
- Remesh using Scipy

### 3.2.1 Constant ’h’

In this method, we obtain the SPH solution using constant ’h’. The parameter ’h’ is taken to be the initial spacing of the particles. This is not changed throughout the simulation. Below figures show the variation of error norm with number of particles for different kernels.

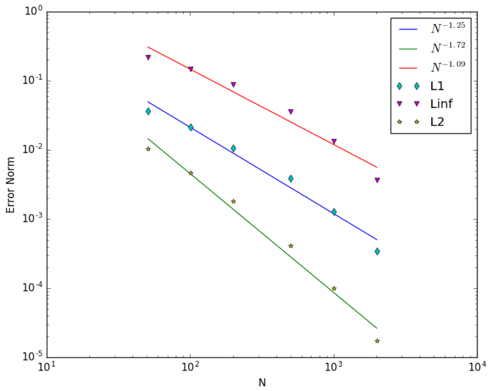


FIGURE 3.3: Error vs N - Spline Kernel

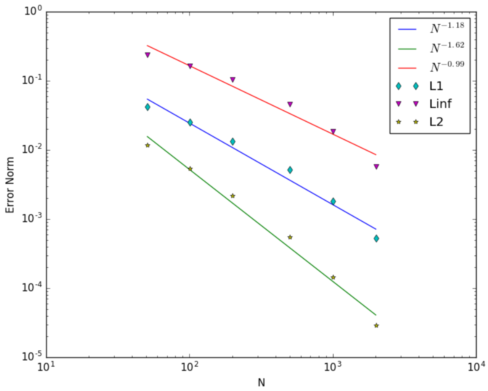


FIGURE 3.4: Error vs N - Gauss Kernel

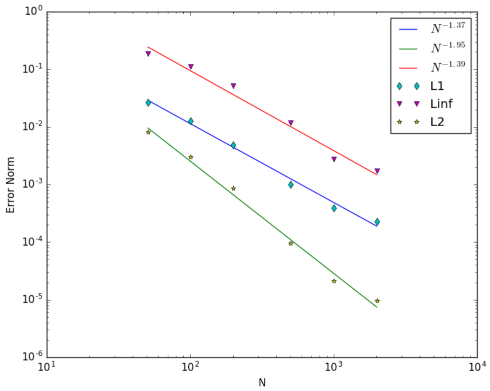


FIGURE 3.5: Error vs N - S. Gauss Kernel

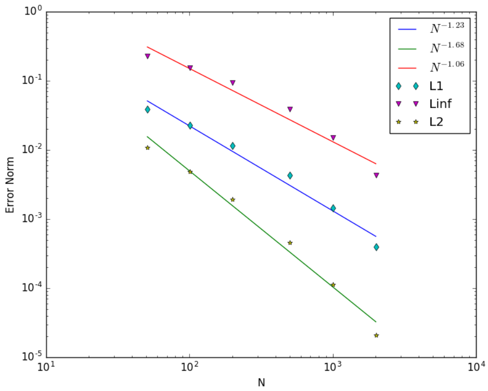


FIGURE 3.6: Error vs N - WC2 Kernel

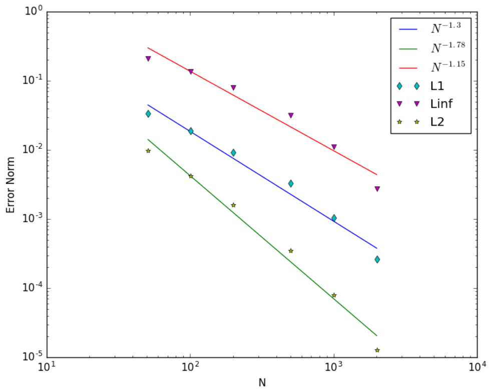


FIGURE 3.7: Error vs N - WC4 Kernel

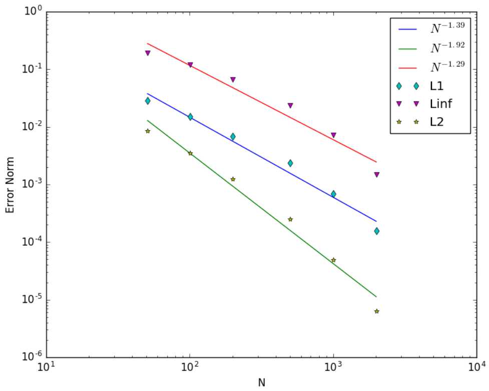


FIGURE 3.8: Error vs N - WC6 Kernel

### 3.2.2 ADKE Scheme

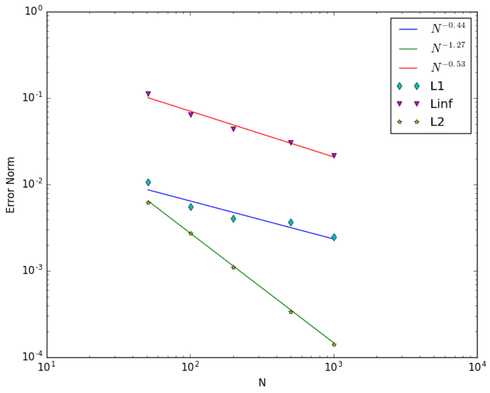


FIGURE 3.9: Error vs N - C. Spline Kernel

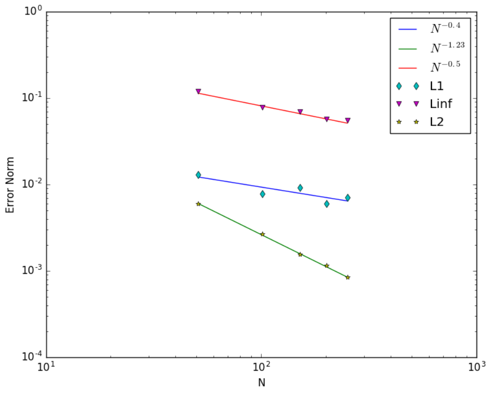


FIGURE 3.10: Error vs N - Gauss Kernel

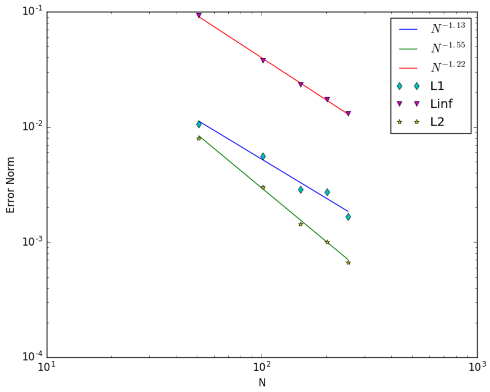


FIGURE 3.11: Error vs N - S. Gauss Kernel

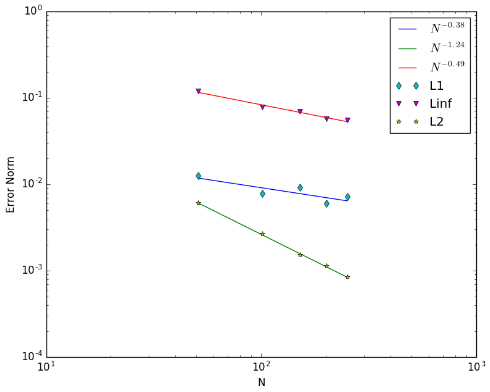


FIGURE 3.12: Error vs N - Q. Spline Kernel

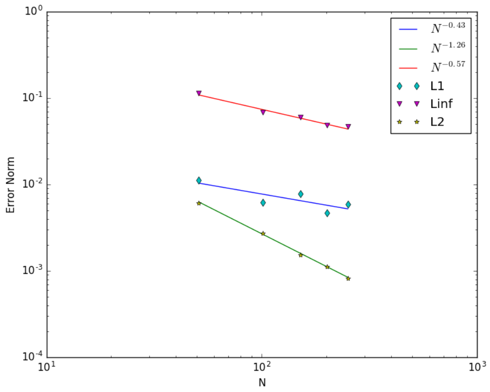


FIGURE 3.13: Error vs N - WC2 Kernel

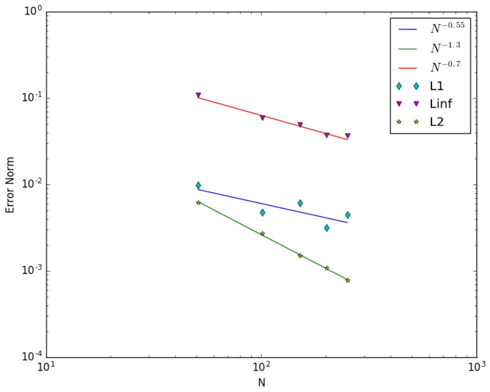


FIGURE 3.14: Error vs N - WC4 Kernel

In this method, we use variable ’h’, which is obtained from ADKE scheme. The above plots depict the variation of error norm with the number of particles for different kernels. We can see that the error magnitude has decreased and it gives better results as compared to constant ’h’ method. However the time required to compute the solution is higher, as iterations get increased in each time-step.

### 3.2.3 Remesh - Scipy Interpolation

In this method, we use the strategy of Remeshing the solution on to a fine grid after certain time-steps. This method actualy gives a better solution and happens to be the best among all methods. For remeshing, we use interpolate module from scipy, which interpolates the set of points to a fine grid. We can see from the below plots that, the error magnitude has decreased and the convergence rate has also increased for almost all kernels.

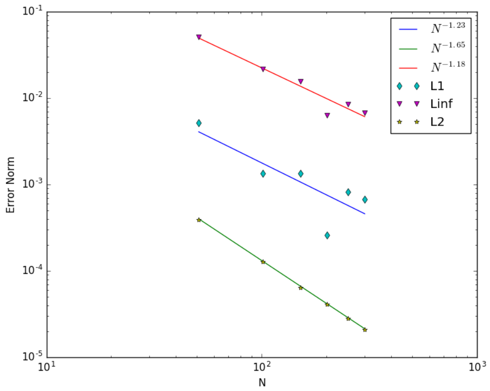


FIGURE 3.15: Error vs N - Spline Kernel

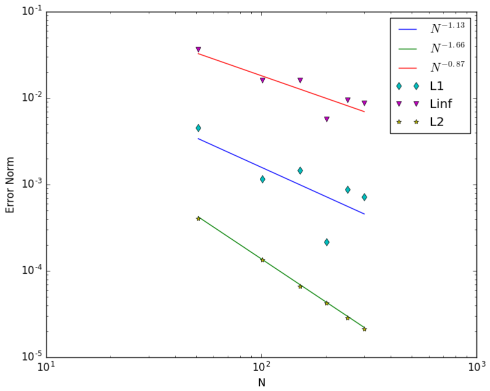


FIGURE 3.16: Error vs N - Gauss Kernel

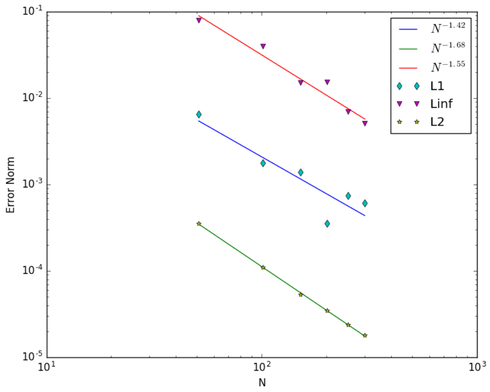


FIGURE 3.17: Error vs N - S. Gauss Kernel

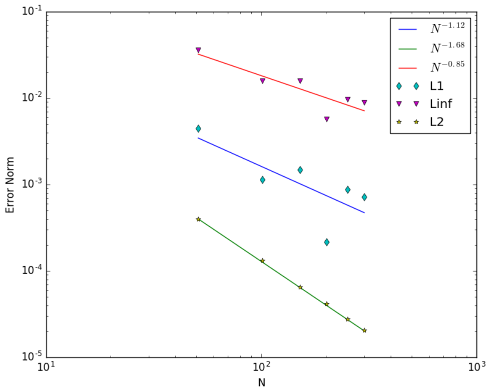


FIGURE 3.18: Error vs N - Q. Spline Kernel

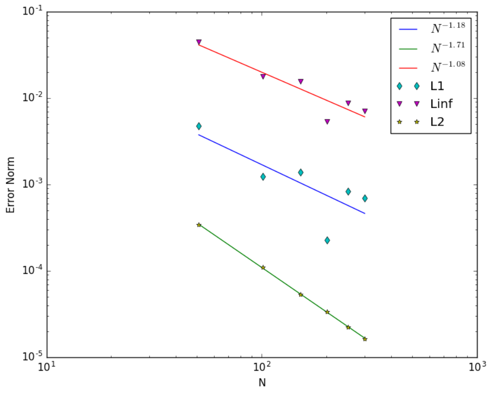


FIGURE 3.19: Error vs N - WC2 Kernel

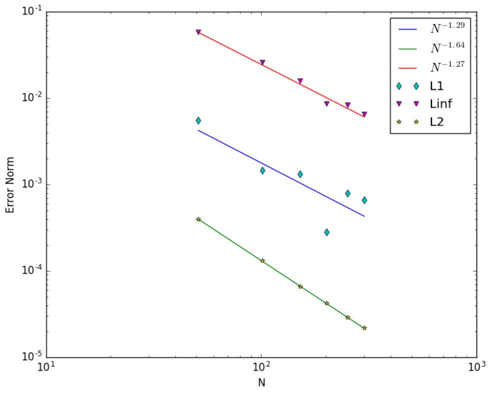


FIGURE 3.20: Error vs N - WC4 Kernel

### 3.2.4 Remesh - Centred Interpolation

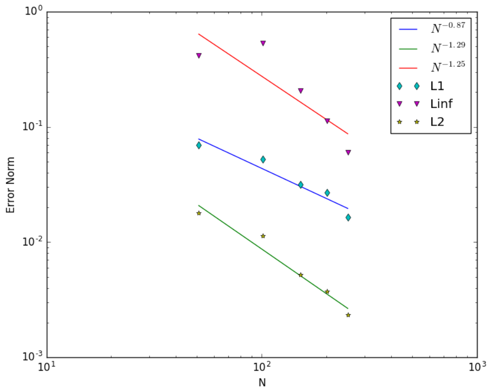


FIGURE 3.21: Error vs N - Spline Kernel

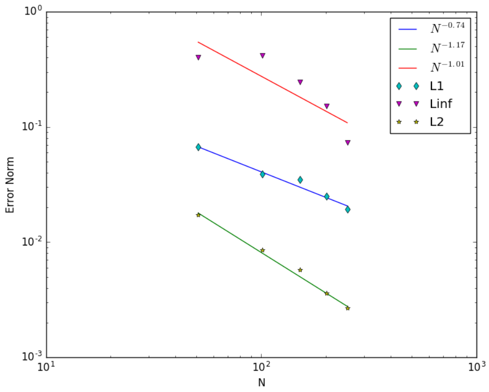


FIGURE 3.22: Error vs N - Gauss Kernel

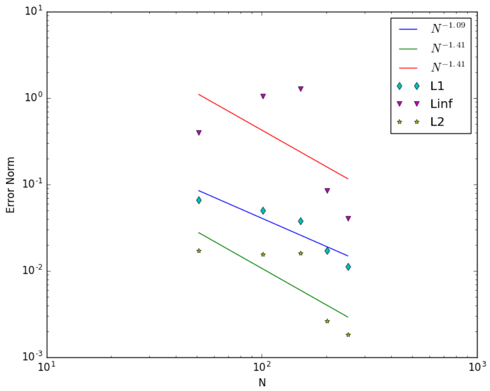


FIGURE 3.23: Error vs N - S.Gauss Kernel

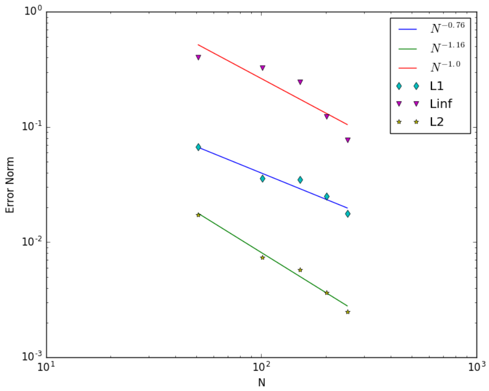


FIGURE 3.24: Error vs N - Q.Spline Kernel

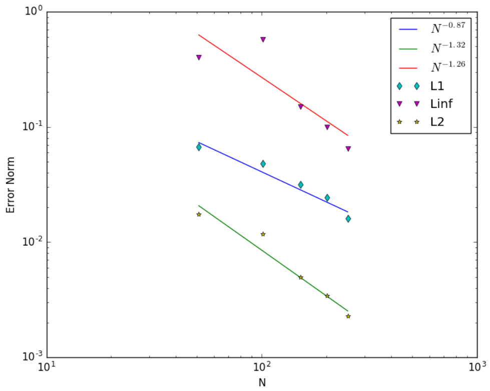


FIGURE 3.25: Error vs N - WC2 Kernel

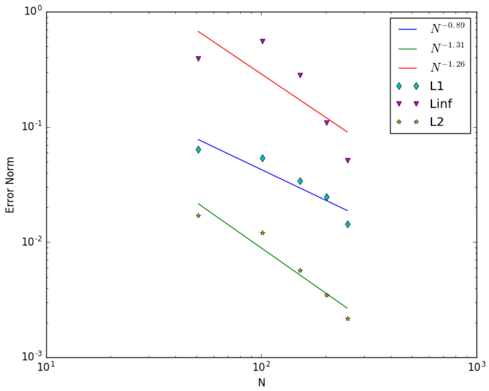
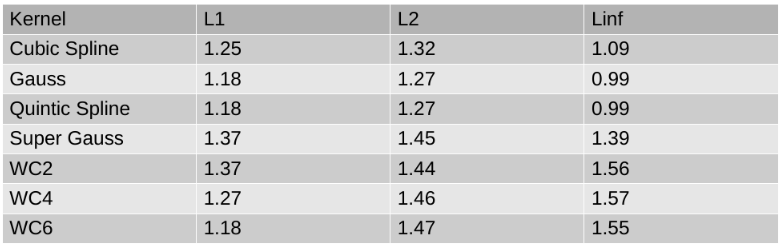


FIGURE 3.26: Error vs N - WC4 Kernel

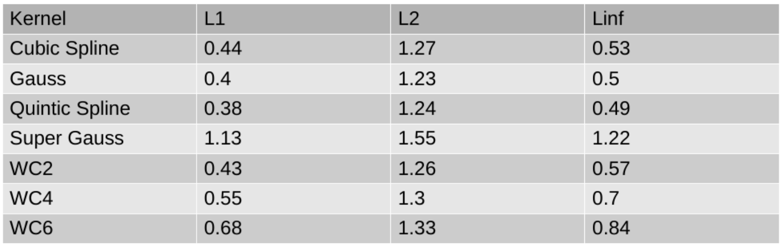
This method is another version of the above method, where we interpolate the set of points based on centred interpolation of order 2 and 3. The above plots the variation of error norm as we increase the number of particles for different kernels. One advantage of this method over the scipy one is that this requires less time for computation.

## 3.3 Summary



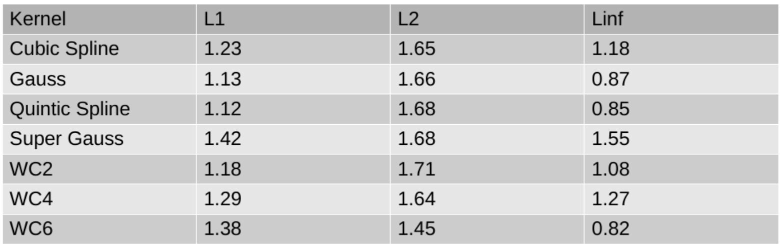
| Kernel | L1 | L2 | Linf |
|---|---|---|---|
| Cubic Spline | 1.25 | 1.32 | 1.09 |
| Gauss | 1.18 | 1.27 | 0.99 |
| Quintic Spline | 1.18 | 1.27 | 0.99 |
| Super Gauss | 1.37 | 1.45 | 1.39 |
| WC2 | 1.37 | 1.44 | 1.56 |
| WC4 | 1.27 | 1.46 | 1.57 |
| WC6 | 1.18 | 1.47 | 1.55 |

FIGURE 3.27: Convergence rates for different kernels- constant 'h' method



| Kernel | L1 | L2 | Linf |
|---|---|---|---|
| Cubic Spline | 0.44 | 1.27 | 0.53 |
| Gauss | 0.4 | 1.23 | 0.5 |
| Quintic Spline | 0.38 | 1.24 | 0.49 |
| Super Gauss | 1.13 | 1.55 | 1.22 |
| WC2 | 0.43 | 1.26 | 0.57 |
| WC4 | 0.55 | 1.3 | 0.7 |
| WC6 | 0.68 | 1.33 | 0.84 |

FIGURE 3.28: Convergence rates for different kernels - ADKE approach



| Kernel | L1 | L2 | Linf |
|---|---|---|---|
| Cubic Spline | 1.23 | 1.65 | 1.18 |
| Gauss | 1.13 | 1.66 | 0.87 |
| Quintic Spline | 1.12 | 1.68 | 0.85 |
| Super Gauss | 1.42 | 1.68 | 1.55 |
| WC2 | 1.18 | 1.71 | 1.08 |
| WC4 | 1.29 | 1.64 | 1.27 |
| WC6 | 1.38 | 1.45 | 0.82 |

FIGURE 3.29: Convergence rates for different kernels - Remeshing strategy

- From all the above methods implemented, we can see that the one using remeshing technique with scipy module gave better results than the rest.

- However, this method requires large amount of computational time to compute the solution.
- Scipy ¿ ADKE ¿ Constant 'h' method, this is the decreasing order of accuracy of solutions obtained.

# Chapter 4

# Sod Shock Tube

The Sod shock tube problem, named after Gary A. Sod, is a common test for the accuracy of computational fluid codes. The time evolution of this problem can be described by solving the Euler equations, which leads to three characteristics, describing the propagation speed of the various regions of the system. Namely the rarefaction wave, the contact discontinuity and the shock discontinuity. The initial conditions for the problem describe two states (left and right) of a quiescent gas separated by an imaginary diaphragm. The states are given as $(\rho_l, p_l, u_l)$ = (1.0, 1.0, 0.0) and $(\rho_r, p_r, u_r)$ = (0.125, 0.1, 0) on the left and right hand sides of the diaphragm which is placed at x=0. For the simualtion, we have used 360 particles in the domain [-0.5, 0.5].

The density for a target particle is determined from the particle distribution as

$$\rho_i = \sum_N m_j W_{ij} \tag{4.1}$$

From the momentum equation, we have:

$$\frac{dv_i}{dt} = -\sum_j m_j \left(\frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2} + \pi_{ij}\right) Dw_{ij} \tag{4.2}$$

The artificial viscosity $\pi_{ij}$ is introduced as a viscous pressure in this equation. This viscosity is constructed to approximate a shear and bulk viscosity in the continuum limit. It is given as:

$$\pi_{ij} = \frac{-\alpha c_{ij}\mu_{ij} + \beta \mu_{ij}^2}{\rho_{ij}} \tag{4.3}$$

$$\mu_{ij} = \frac{h_{ij} v_{ij}.x_{ij}}{|x_{ij}^2| + \eta^2} \tag{4.4}$$

and is only activated for approaching particle pairs ($v_{ij}.r_{ij} \leq 0$). The viscosity depends on two numerical parameters $\alpha$ and $\beta$. The terms $\rho_{ij}$ and $c_{ij}$ denotes the arithmetic mean of the density and sound speed between two particles. The energy equation is discretized as

$$\frac{de_i}{dt} = \frac{1}{2} \sum_j m_j (\frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2} + \pi_{ij}) v_{ij} D w_{ij} \tag{4.5}$$

The artificial viscosity $\pi_{ij}$ is constructed to ensure that the viscous contribution to the thermal energy is always positive definite.

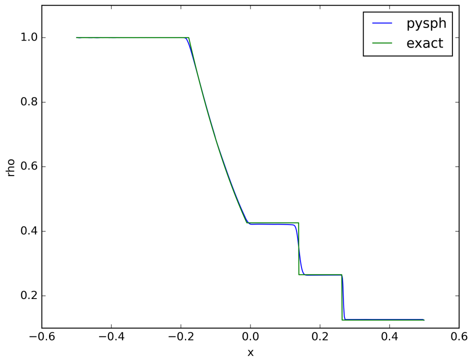


FIGURE 4.1: Sod-Shock Tube Density solution at t=0.2s

The above figure shows the plot of density profile at t=0.2s for SodShock case using ADKE scheme of SPH.

## 4.1 Convergence Study

Let's look at the error analysis for the SodShock case obtained from ADKE scheme SPH. Before we begin, we will be studying four parameters namely density, pressure, velocity and energy. Each parameter is studied separately and independent of each other. The whole domain is divided into four regions as shown below.

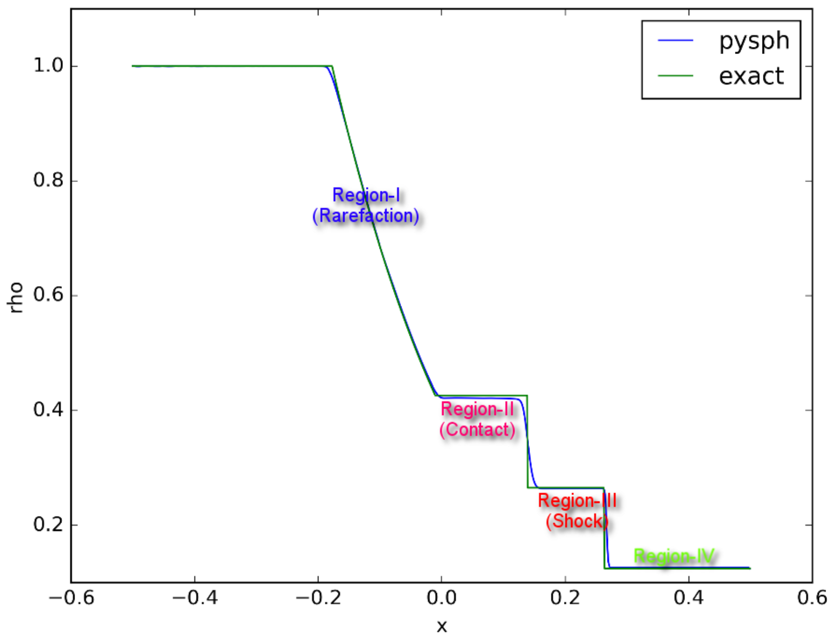


FIGURE 4.2: Nomenclature of each region

We will look into the error analysis in each region as well as the overall domain.

### 4.1.1 Density

First, we look at the infinty norm or the maximum for the overall region. The maximum norm increases as we increase the number of particles used in the simulation. Here we use constant kernel factor for the simulation.

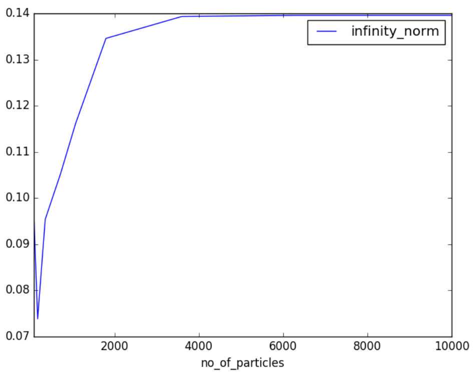


FIGURE 4.3: Infinity norm for overall region.

Coming to the L1 error norm, we see the error decreases as we increase the number of particles. Below figure, shows the loglog variation of L1 norm with number of particles. All the regions show a similar trend of convergence rate of the order 0.5.

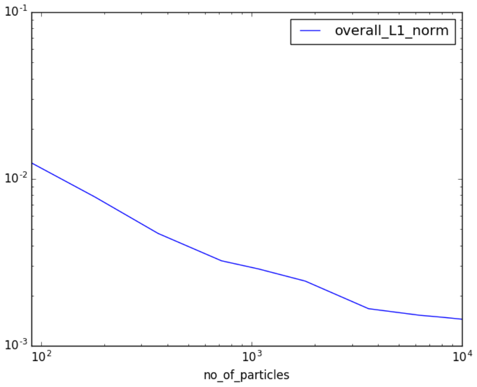


FIGURE 4.4: L1 norm for overall region of Density profile

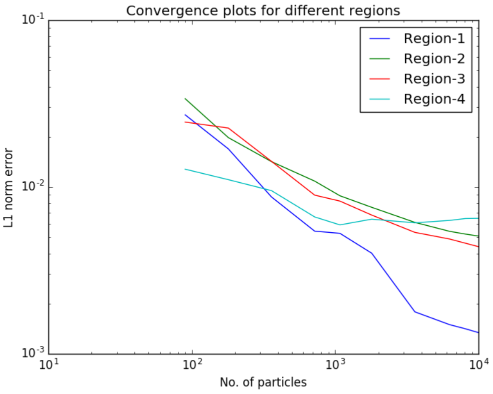


FIGURE 4.5: L1 Norm region wise for Density profile

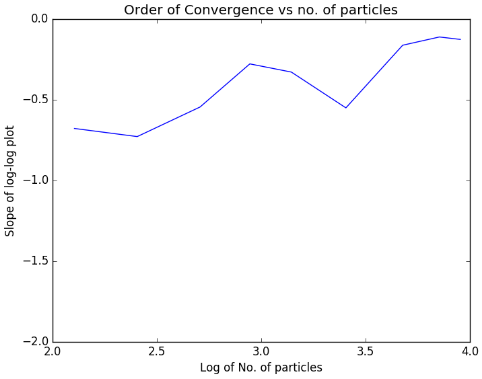


FIGURE 4.6: Comvergence rates with no. of particles

The above figure gives an overall picture of convergencce rates as we increase the no. of particles. We see that the order of convergence decreases as we increase N.The maximum convergence order is around 0.5.

### 4.1.2 Pressure

As seen above, the maximum norm in case of Pressure variable increases as we increase the number of particles.

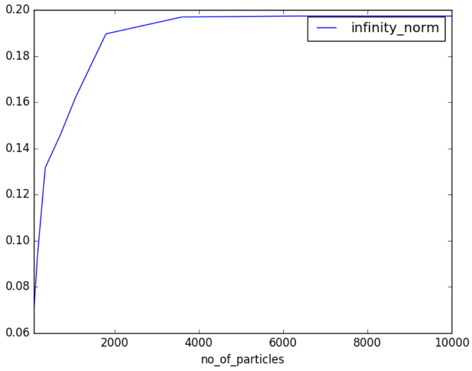


FIGURE 4.7: Infinity norm for overall region.

Coming to the L1 error norm, we see the error decreases as we increase the number of particles. Below figure, shows the loglog variation of L1 norm with number of particles. All the regions show a similar trend of convergence rate of the order 0.4.

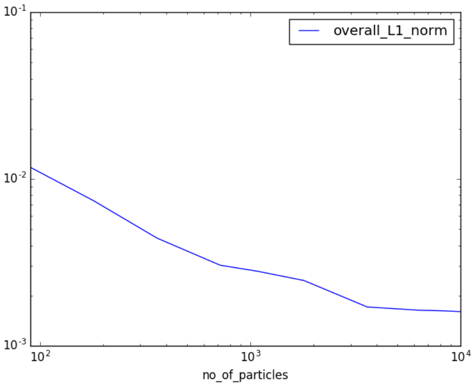


FIGURE 4.8: L1 norm for overall region of Pressure profile

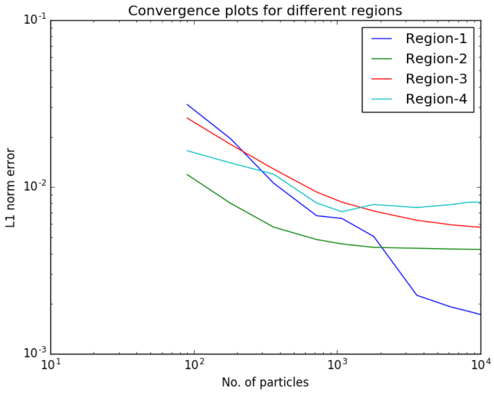


FIGURE 4.9: L1 Norm region wise for Pressure profile

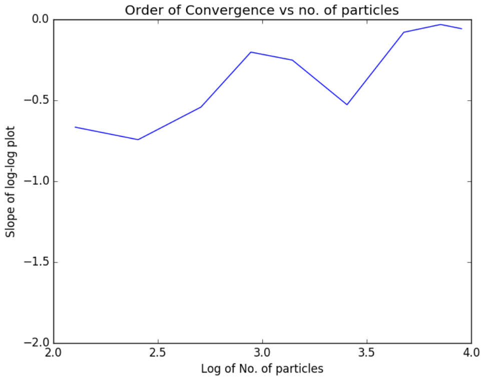


FIGURE 4.10: Comvergence rates with no. of particles

The above figure gives an overall picture of convergencce rates as we increase the no. of particles. We see that the order of convergence decreases as we increase N.The maximum convergence order is around 0.65.

### 4.1.3 Velocity

First, we look at the infinty norm or the maximum for the overall region. The maximum norm increases as we increase the number of particles used in the simulation. Here we use constant kernel factor for the simulation.

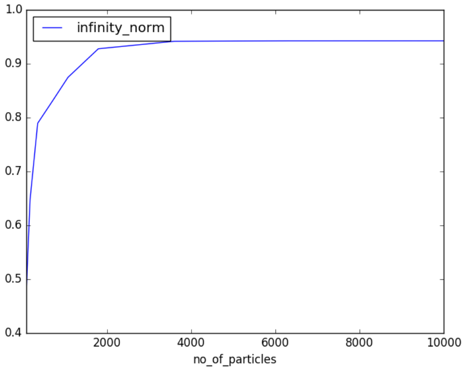


FIGURE 4.11: Infinity norm for overall region.

Coming to the L1 error norm, we see the error decreases as we increase the number of particles. Below figure, shows the loglog variation of L1 norm with number of particles. All the regions show a similar trend of convergence rate of the order 0.4.

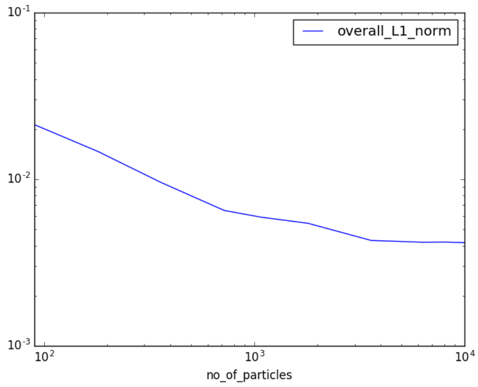


FIGURE 4.12: L1 norm for overall region of Velocity profile

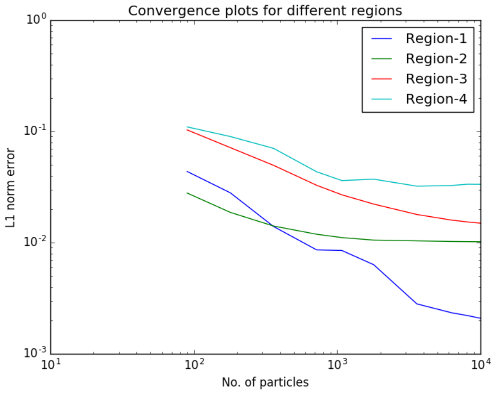


FIGURE 4.13: L1 Norm region wise for Velocity profile

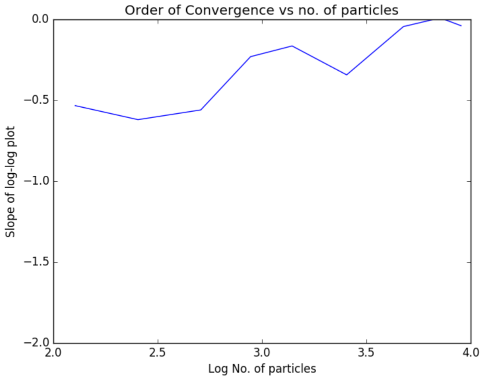


FIGURE 4.14: Comvergence rates with no. of particles

The above figure gives an overall picture of convergencce rates as we increase the no. of particles. We see that the order of convergence decreases as we increase N.The maximum convergence order is around 0.6.

### 4.1.4 Energy

As seen above, the maximum norm in case of Pressure variable increases as we increase the number of particles.

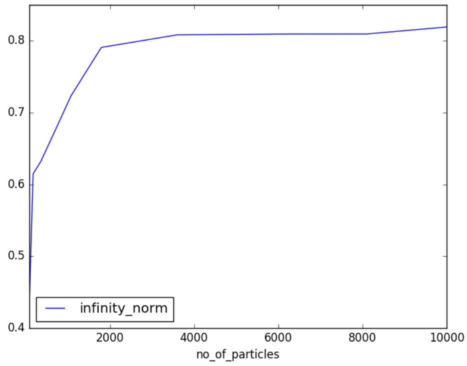


FIGURE 4.15: Infinity norm for overall region.

Coming to the L1 error norm, we see the error decreases as we increase the number of particles. Below figure, shows the loglog variation of L1 norm with number of particles. All the regions show a similar trend of convergence rate of the order 0.4.

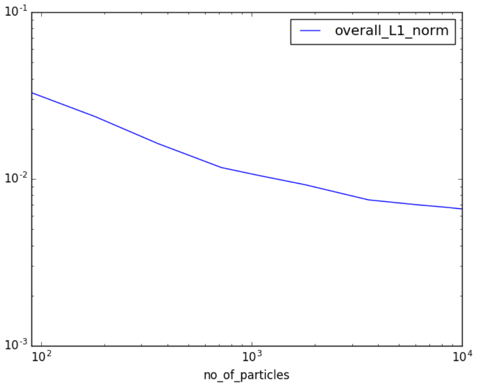


FIGURE 4.16: L1 norm for overall region of Energy profile

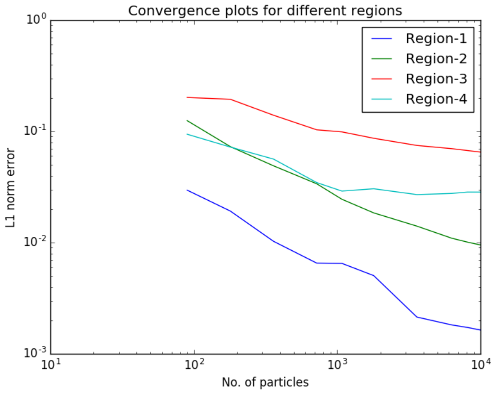


FIGURE 4.17: L1 Norm region wise for Energy profile

TABLE 4.1: Convergence rates for different regions

| | Density | Pressure | Velocity | Energy |
|---|---|---|---|---|
| **Regions** | **Order** | **Order** | **Order** | **Order** |
| Overall | 0.45 | 0.33 | 0.4 | 0.33 |
| Region-1 | 0.64 | 0.65 | 0.62 | 0.62 |
| Region-2 | 0.38 | 0.18 | 0.18 | 0.53 |
| Region-3 | 0.39 | 0.41 | 0.30 | 0.25 |
| Region-4 | 0.14 | 0.26 | 0.14 | 0.25 |

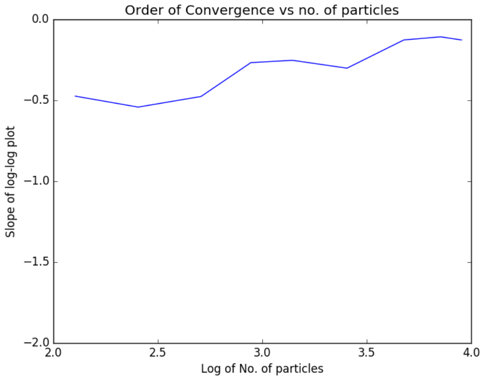


FIGURE 4.18: Comvergence rates with no. of particles

The above figure gives an overall picture of convergencce rates as we increase the no. of particles. We see that the order of convergence decreases as we increase N.The maximum convergence order is around 0.5.

## 4.2 Summary

From the above table, we can see that the order of convergence rates are very less and are around 0.5. We can conclude the following from the data obtained:

- When we consider overall region, the infinty norm increases as we increase the number of particles.
- When we consider overall region, the L1 error norm has an average order of convergence of 0.4.

- Considering region-wise convergence study, Region-1, which is the Rarefaction region, shows a higher order convergence as compared to others. This is expected as the profile happens to be smooth.
- The maximum norm usually occurs at the shock for each case.

# Chapter 5

# Future Work

The following tasks have been finished pertaining to PySPH as of now:

- Implemented Exact Riemann Solver in Python
- Tested the Exact Riemann Solver for all test cases (Toro)
- Added command line arguments
- Implemented Woodward Colella Problem
- Implemented Wendland Quintic Kernels
- Error Analysis for smooth solution problems.
- Error Analysis for SodShock case.

We can extend this work further in studying convergence study for different schemes and different problems in SPH. The following can thought as future work in this field:

- Extend to different schemes.
- Extend to different problems apart from Riemann.
- Compare SPH wwith other computational algorithms like Galerkin methods, Finite Differences. problems

# Chapter 6

# References

- E.F.Toro, Riemann solvers and numerical methods for fluid dynamics, Springer, 2009
- J.J. Monaghan, Smoothed Particle Hydrodynamics
- Kunal Puri, Prabhu Ramachandran, A comparison of solvers of SPH schemes for the compressible Euler equations, Journal of Computational Physics, 256:308-333, September 2013
- Prabhu Ramachandran, PySPH, https://pysph.readthedocs.io/en/latest/