Numerical Modeling

department name(s) Semester Report

Antonio Peters

March 15, 2016

1 Introduction and Preliminaries

This report will detail the work done for the Numerical Modeling mathematics honors course. Whenever possible, decimal values will be calculated to double precision. Data will generally be stored in a *.dat*, using the $save\_to\_file$ function detailed in Appendix B.1 file which will then be read in and plotted with the function $plot\_file$ seen in Appendix B.2. The report will build up from analyzing the finite difference method, error approximation and convergence, and boundary conditions to understanding time evolutions and the Riemann problem and then will look at Gordinov's method in order to simulate high resolution shock capturing in order to solve Burger's equation.

2 Finite Difference

There are several different variants of the finite difference methods (FDMs), this is dependent on the stencil (number of points being used to determine a single value). The FDM uses Taylor expansion (Appendix A.1) to approximate the derivative of a function of finite points using the surrounding points.

The most basic of these is the first order FDMs

$$dv_i = \frac{v_{i+1} - v_i}{h} \tag{1}$$

where $dv_i \approx \frac{df(x_i)}{dx}$

A Mathematical Background

A.1 Taylor Expansion

B Utility Functions

B.1 Save to File

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Numerical Modelling
%Writes to a generic file
%Author: Antonio Peters
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function save_to_file( x, filename)

[p,n] = size(x);

fid = fopen(filename, 'w+');
for i=1:n
   for j=1:p
      fprintf(fid,'%f \t',x(j,i));
   end
   fprintf(fid,'\n');
end
fclose(fid);

end
```

B.2 Plot File

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Numerical Modelling
%Plots a generic file in MATLAB
%Author: Antonio Peters
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function plot_file(datfile)

data    = load(datfile);

[~, p]  = size(data);

hold on

for i=2:p
   plot( data(:,1), data(:,i));
end

hold off

end
```