Kelvin-Helmholtz instability

Generated by Doxygen 1.11.0

Chapter 1

File Index

1.1 File List

Here is a list of all files with brief descriptions:

Chapter 2

File Documentation

2.1 finitevol.c File Reference

```
#include <math.h>
#include <mpi.h>
#include <omp.h>
#include <stdbool.h>
#include <unistd.h>
#include <stdio.h>
```

Functions

- void getConserved (double rho, double vx, double vy, double P, double gamma, double vol, double *Mass_o, double *Momx_o, double *Momy_o, double *Energy_o)
- void getPrimitive (double Mass, double Momx, double Momy, double Energy, double gamma, double vol, double *rho_o, double *vx_o, double *vy_o, double *P_o)
- void getGradient (double f, double f_down, double f_up, double f_right, double f_left, double dx, double *f_dx_o, double *f_dy_o)
- void extrapolateInSpaceToFace (double f, double f_dx, double f_dy, double dx, double *f_XL_o, double *f_XR_o, double *f_YL_o, double *f_YR_o)
- double applyFluxes (double F, double flux_F_x, double flux_F_x_up, double flux_F_y_left, double flux_F_y, double dx, double dt)
- void getFlux (double rho_L, double rho_R, double vx_L, double vx_R, double vy_L, double vy_R, double P_L, double P_R, double gamma, double *flux_Mass_o, double *flux_Momx_o, double *flux_Momy_o, double *flux_Energy_o)
- void linspace (double start, double stop, size_t num, double *output)
- void meshgrid (double *xv, double *yv, size_t N, double *x_res, double *y_res)
- void printM (double *a, int w, int h)
- void writeToFile (FILE *stream, double *m, int w, int h)

2.1.1 Function Documentation

2.1.1.1 applyFluxes()

```
double applyFluxes (
            double F,
            double flux_F_x,
            double flux_F_x_up,
            double flux_F_y_left,
            double flux_F_y,
            double dx,
            double dt)
```

Applies the fluxes to the conserved variables

Parameters

F	- Conserved variable
flux_F_x	- Flux of the conserved variable in the x direction
flux_F_x_up	- Flux of the conserved variable above F in the x direction
flux_F_y_left	- Flux of the conserved variable to the left of F in the y direction
flux_F_y	- Flux of the conserved variable in the y direction
dx	- Cell size
dt	- Time step

2.1.1.2 extrapolateInSpaceToFace()

```
void extrapolateInSpaceToFace (
            double f,
            double f_dx,
            double f_dy,
            double dx,
            double * f_XL_o,
            double * f_XR_o,
            double * f_YL_o,
            double * f_YR_o)
```

Performed spatial extrapolation on a cell in a field to each of the 4 faces of a cell (Its four neighbors up, down, left and right)

Parameters

f	- Element in a field
f_dx	- Derivative of f along the x-axis
f_dy	- Derivate of f along the y-axis
dx	- Cell size
f_XL_o	- Pointer to the buffer to store the extrapolated variable to the left of f
f_XR_o	- Pointer to the buffer to store the extrapolated variable to the right of f
f_YL_o	- Pointer to the buffer to store the extrapolated variable below f
f_YR_o	- Pointer to the buffer to store the extrapolated variable above f

2.1.1.3 getConserved()

```
void getConserved (
             double rho,
             double vx,
             double vy,
             double P,
             double gamma,
             double vol,
             double * Mass_o,
             double * Momx_o,
             double * Momy_o,
             double * Energy_o)
```

Calculates the conserved variables from the primitive variables

Parameters

rho	- Density
vx	- Velocity in the x direction
vy	- Velocity in the y direction
P	- Pressure
gamma	- Ideal gas gamma
vol	- Volume of the cell
Mass_o	- Pointer to the buffer to store the mass
Momx_o	- Pointer to the buffer to store the x momentum
Momy_o	- Pointer to the buffer to store the y momentum
Energy_o	- Pointer to the buffer to store the ernergy

2.1.1.4 getFlux()

```
void getFlux (
             double rho_L,
             double rho_R,
             double vx_L,
             double vx_R,
             double vy_L,
             double vy_R,
             double P_L,
             double P_R,
             double gamma,
             double * flux_Mass_o,
             double * flux_Momx_o,
             double * flux_Momy_o,
             double * flux_Energy_o)
```

Calculates the fluxes of a between two states

Parameters

rho_L	- Density of the left state
rho_R	- Density of the right state

Parameters

vx_L	- Velocity in the x direction of the left state
vx_R	- Velocity in the x direction of the right state
vy_L	- Velocity in the y direction of the left state
vy_R	- Velocity in the y direction of the right state
P_L	- Pressure of the left state
P_R	- Pressure of the right state
gamma	- Ideal gas gamma
flux_Mass_o	- Pointer to the buffer to store the mass flux
flux_Momx_o	- Pointer to the buffer to store the x momentum flux
flux_Momy_o	- Pointer to the buffer to store the y momentum flux
flux_Energy_o	- Pointer to the buffer to store the energy flux

2.1.1.5 getGradient()

```
void getGradient (
            double f,
            double f_down,
            double f_up,
            double f_right,
            double f_left,
            double dx,
            double * f_dx_o,
            double * f_dy_o)
```

Calculate the gradients of an element in a field

Parameters

f	- A variable in the field
f_down	- The variable below f in the field
f_up	- The variable above f in the field
f_right	- The variable to the right of f in the field
f_left	- The variable to the left of f in the field
dx	- Size of a cell in the field
f_dx_o	- Where to store the derivative of f in the x-direction
f_dy_o	- Where to store the derivative of f in the y-direction

2.1.1.6 getPrimitive()

```
void getPrimitive (
            double Mass,
            double Momx,
            double Momy,
            double Energy,
```

```
            double gamma,
            double vol,
            double * rho_o,
            double * vx_o,
            double * vy_o,
            double * P_o)
```

Calculates the primitive variables from the conserved variables

Parameters

Mass	- Mass
Momx	- Momentum in the x direction
Momy	- Momentum in the y direction
Energy	- Energy
gamma	- Ideal gas gamma
vol	- Volume of the cell
rho_o	- Pointer to the buffer to store the density
vx_o	- Pointer to the buffer to store the x velocity
vy_o	- Pointer to the buffer to store the y velocity
P_o	- Pointer to the buffer to store the pressure

2.1.1.7 linspace()

```
void linspace (
            double start,
            double stop,
            size_t num,
            double * output)
```

Implementation of the numpy function linspace generate a vector of num evenly distributed elements from start to stop

Parameters

start	- Start of the vector
stop	- End of the vector
num	- Number of elements in the vector
output	- Buffer to store the vector

2.1.1.8 meshgrid()

```
void meshgrid (
            double * xv,
            double * yv,
            size_t N,
            double * x_res,
            double * y_res)
```

Implementation of the numpy function meshgrid generate a two square matricies by repeating the row vector xv and the column vector yv

Parameters

xv	- Row vector to generate x_res from
yv	- Column vector to generate y_res from
N	- Length of xv and yv vectors
x_res	- Preallocated buffer to store matrix generated from xv in
y_res	- Preallocated buffer to store matrix generated from yv in

2.1.1.9 printM()

```
void printM (
            double * a,
            int w,
            int h)
```

Prints a matrix.

Parameters

a	- buffer were the matrix is stored
w	- Width of the matrix stored in a
h	- Height of the matrix stored in a

2.1.1.10 writeToFile()

```
void writeToFile (
            FILE * stream,
            double * m,
            int w,
            int h)
```

Writes a matrix to a file in binary format. The first 4 bytes in the file contain the matrixes width The next 4 bytes contain its height The following w $*$ h sections of 8 bytes contain the values of the matrix

Parameters

stream	- File output stream opened in "wb" mode to write to
m	- buffer were matrix is stored
w	- Width of the matrix stored in m
h	- Height of the matrix stored in m

Index