

## Solving Wave Equation with MPI paralleling

Generated by Doxygen 1.9.1



<b>1 ACSE-6 MPI Programming Assignment – Solving the Wave Equation</b>	<b>1</b>
1.1 Compile and run the task	1
1.2 User Handbook	1
1.2.1 About parameter settings	1
1.2.2 Postprocessing outputs	2
1.2.3 Something important to note	3
<b>2 Namespace Index</b>	<b>5</b>
2.1 Namespace List	5
<b>3 Class Index</b>	<b>7</b>
3.1 Class List	7
<b>4 File Index</b>	<b>9</b>
4.1 File List	9
<b>5 Namespace Documentation</b>	<b>11</b>
5.1 animation Namespace Reference	11
5.1.1 Function Documentation	12
5.1.1.1 update_grid()	12
5.1.2 Variable Documentation	12
5.1.2.1 ani_name	12
5.1.2.2 anim	12
5.1.2.3 arr	12
5.1.2.4 ax	12
5.1.2.5 boundary_type	12
5.1.2.6 cax	13
5.1.2.7 cb	13
5.1.2.8 dpi	13
5.1.2.9 dt	13
5.1.2.10 ele_i	13
5.1.2.11 ele_j	13
5.1.2.12 ele_x	13
5.1.2.13 ele_y	14
5.1.2.14 fig	14
5.1.2.15 fontsize	14
5.1.2.16 fps	14
5.1.2.17 i_loc	14
5.1.2.18 imax	14
5.1.2.19 j_loc	14
5.1.2.20 jmax	14
5.1.2.21 length	15
5.1.2.22 norm	15
5.1.2.23 num_pics	15

5.1.2.24 parameters	15
5.1.2.25 speedratio	15
5.1.2.26 t_out	15
5.1.2.27 text	15
5.1.2.28 width	15
5.1.2.29 x_label	16
5.1.2.30 y_label	16
5.2 loaddata Namespace Reference	16
5.2.1 Function Documentation	16
5.2.1.1 load_array()	16
5.2.1.2 load_binary_file()	16
5.2.1.3 read_parameters()	16
5.3 pic Namespace Reference	17
5.3.1 Variable Documentation	17
5.3.1.1 arr	17
5.3.1.2 ax	17
5.3.1.3 boundary_type	18
5.3.1.4 cax	18
5.3.1.5 cb	18
5.3.1.6 data	18
5.3.1.7 dpi	18
5.3.1.8 dt	18
5.3.1.9 ele_i	18
5.3.1.10 ele_j	18
5.3.1.11 ele_x	19
5.3.1.12 ele_y	19
5.3.1.13 fig	19
5.3.1.14 fontsize	19
5.3.1.15 i_loc	19
5.3.1.16 imax	19
5.3.1.17 j_loc	19
5.3.1.18 jmax	19
5.3.1.19 length	20
5.3.1.20 norm	20
5.3.1.21 parameters	20
5.3.1.22 pic_name	20
5.3.1.23 picnum	20
5.3.1.24 t	20
5.3.1.25 t_out	20
5.3.1.26 text	21
5.3.1.27 width	21
5.3.1.28 x_label	21

5.3.1.29 y_label . . . . .	21
<b>6 Class Documentation</b>	<b>23</b>
6.1 read_parameters Class Reference . . . . .	23
6.1.1 Constructor & Destructor Documentation . . . . .	23
6.1.1.1 read_parameters() . . . . .	24
6.2 wave_data Class Reference . . . . .	24
6.2.1 Constructor & Destructor Documentation . . . . .	25
6.2.1.1 wave_data() . . . . .	25
6.2.1.2 ~wave_data() . . . . .	25
6.2.2 Member Function Documentation . . . . .	26
6.2.2.1 allocate_grid() . . . . .	26
6.2.2.2 build_communication_type() . . . . .	26
6.2.2.3 build_OutputType() . . . . .	26
6.2.2.4 do_communication() . . . . .	26
6.2.2.5 do_iteration() . . . . .	26
6.2.2.6 find_neighbour_for_other_processor() . . . . .	26
6.2.2.7 MPI_grid_to_file() . . . . .	26
6.2.2.8 run_simulation() . . . . .	27
6.2.2.9 set_initial_condition() . . . . .	27
6.2.2.10 set_Neumman_boundary_new_u() . . . . .	27
6.2.2.11 set_Neumman_boundary_old_u() . . . . .	27
6.2.2.12 set_Neumman_boundary_u() . . . . .	27
6.2.2.13 setting_paramters() . . . . .	27
6.2.3 Member Data Documentation . . . . .	27
6.2.3.1 block_lengths . . . . .	27
6.2.3.2 c . . . . .	28
6.2.3.3 display . . . . .	28
6.2.3.4 down_boundaries . . . . .	28
6.2.3.5 down_edge_length . . . . .	28
6.2.3.6 dt . . . . .	28
6.2.3.7 dt_out . . . . .	28
6.2.3.8 dx . . . . .	28
6.2.3.9 dy . . . . .	28
6.2.3.10 end_index . . . . .	29
6.2.3.11 imax . . . . .	29
6.2.3.12 index_connect_down_edge . . . . .	29
6.2.3.13 index_connect_left_edge . . . . .	29
6.2.3.14 index_connect_right_edge . . . . .	29
6.2.3.15 index_connect_up_edge . . . . .	29
6.2.3.16 index_isolated . . . . .	29
6.2.3.17 index_not_isolated . . . . .	29

6.2.3.18 intdiv	30
6.2.3.19 isolated_num	30
6.2.3.20 jmax	30
6.2.3.21 left_boundaries	30
6.2.3.22 left_down_corner	30
6.2.3.23 left_edge_length	30
6.2.3.24 left_up_corner	30
6.2.3.25 line_end	30
6.2.3.26 line_feeds	31
6.2.3.27 line_start	31
6.2.3.28 mod	31
6.2.3.29 n	31
6.2.3.30 n_line	31
6.2.3.31 new_u	31
6.2.3.32 non_isolated_num	31
6.2.3.33 old_u	31
6.2.3.34 output	32
6.2.3.35 parameters	32
6.2.3.36 receive_from_neighbour	32
6.2.3.37 right_boundaries	32
6.2.3.38 right_down_corner	32
6.2.3.39 right_edge_length	32
6.2.3.40 right_up_corner	32
6.2.3.41 send_length	32
6.2.3.42 start_index	33
6.2.3.43 t	33
6.2.3.44 t_max	33
6.2.3.45 t_out	33
6.2.3.46 u	33
6.2.3.47 up_boundaries	33
6.2.3.48 up_edge_length	33
6.2.3.49 x_index	33
6.2.3.50 x_max	34
6.2.3.51 y_index	34
6.2.3.52 y_max	34
<b>7 File Documentation</b>	<b>35</b>
7.1 animation.py File Reference	35
7.2 loaddata.py File Reference	36
7.3 main.cpp File Reference	36
7.3.1 Function Documentation	36
7.3.1.1 main()	36

7.4 pic.py File Reference	37
7.5 read_parameters.cpp File Reference	37
7.5.1 Macro Definition Documentation	38
7.5.1.1 COMMENT_CHAR	38
7.5.2 Function Documentation	38
7.5.2.1 deleteSpace()	38
7.5.2.2 extractArray()	38
7.5.2.3 IsCommentChar()	38
7.5.2.4 TrimMarks()	38
7.5.2.5 TrimSpaces()	39
7.6 read_parameters.h File Reference	39
7.7 README.md File Reference	39
7.8 wave.cpp File Reference	39
7.8.1 Function Documentation	40
7.8.1.1 find_processor()	41
7.8.2 Variable Documentation	41
7.8.2.1 add_start	41
7.8.2.2 cnt	41
7.8.2.3 down_1	41
7.8.2.4 down_2	41
7.8.2.5 down_3	41
7.8.2.6 DownEdge	41
7.8.2.7 elapsed	42
7.8.2.8 finish	42
7.8.2.9 id	42
7.8.2.10 idx	42
7.8.2.11 index_down_for_processor	42
7.8.2.12 index_left_for_processor	42
7.8.2.13 index_right_for_processor	42
7.8.2.14 index_up_for_processor	42
7.8.2.15 it	43
7.8.2.16 left_1	43
7.8.2.17 left_2	43
7.8.2.18 left_3	43
7.8.2.19 num	43
7.8.2.20 num_allocates	43
7.8.2.21 out_cnt	43
7.8.2.22 OutputType	43
7.8.2.23 OutputType2	44
7.8.2.24 OutputType3	44
7.8.2.25 p	44
7.8.2.26 parameters	44

7.8.2.27 receive_index . . . . .	44
7.8.2.28 requests . . . . .	44
7.8.2.29 right_1 . . . . .	44
7.8.2.30 right_2 . . . . .	44
7.8.2.31 right_3 . . . . .	45
7.8.2.32 Send_to_neighbour . . . . .	45
7.8.2.33 Send_to_neighbour2 . . . . .	45
7.8.2.34 Send_to_neighbour3 . . . . .	45
7.8.2.35 Send_to_neighbour_address . . . . .	45
7.8.2.36 Send_to_neighbour_address2 . . . . .	45
7.8.2.37 Send_to_neighbour_address3 . . . . .	45
7.8.2.38 Send_to_neighbour_displacements . . . . .	45
7.8.2.39 Send_to_neighbour_displacements2 . . . . .	46
7.8.2.40 Send_to_neighbour_displacements3 . . . . .	46
7.8.2.41 start . . . . .	46
7.8.2.42 start_indexs . . . . .	46
7.8.2.43 tag_num . . . . .	46
7.8.2.44 up_1 . . . . .	46
7.8.2.45 up_2 . . . . .	46
7.8.2.46 up_3 . . . . .	46
7.8.2.47 UpEdge . . . . .	47
7.8.2.48 zero . . . . .	47
7.9 wave.h File Reference . . . . .	47
<b>Index</b>	<b>49</b>



## Chapter 1

# ACSE-6 MPI Programming Assignment – Solving the Wave Equation

This piece of coursework was created by Jin Yu (acse-jy220), github handle: <https://github.com/acse-jy220>

### 1.1 Compile and run the task

You will need a compiler which supports MPI on your system to compile and run this task.

```
$ mpic++ main.cpp -o main -std=c++11
$ mpiexec -n (number of nodes) ./main
```

### 1.2 User Handbook

#### 1.2.1 About parameter settings

The initial settings for the problem is located at `parameters.ini`(do not change its name! ), it offers a number of changable parameters:

-----Physical parameters-----

`length`: the range of the x-coordinate of the region

`width`: the range of the y-coordinate of the region

`imax`: number of grids in the x-coordinate

`jmax`: number of grids in the y-coordinate

`c`: the speed of the wave

`dt`: the numerical time step we use in the forward updating scheme.

-----Initial condition parameters-----

for the Initial condition setting, we just use sinusoidal disturbances over a period, where  $u_0 = 5.0 * (\cos(\text{distance} / \text{radius} * \pi) + 1.0)$  for some disturbance.

`rs`: array of radius for the disturbances

`xs`: array of the x-coordinate of centers for the disturbances

`ys`: array of the y-coordinate of centers for the disturbances

Note as show in the picture, a column of `rs`, `xs` and `ys` represents the radius, x-coordinate and y-coordinate for **one** specific disturbance, if you input multiple disturbances, please make sure they are on the correct indexes.

-----Output parameters-----

`output`: if turned on, binary files will be created in the `./output` folder, if not, just purely run the simulation without any outputs. `t_out`: the output time steps, or say, output images every '`t_out`' seconds.

`t_end`: the end time of the simulation. (of course the start time is 0)

-----Boundary types-----

`boundary_type`: There are three boundary types for choose, they are 'periodic', 'Neumann' and 'Dirichlet'.

**periodic boundary**: eable wave to passing through boundaries

(i.e the left nerighbour of a grid on the left physical boundary is the grid on the right physical boundary)

**Neumann boundary**: the divergence of 'u' is zero at the boundary, on square grid it is equal to etting the value on the boundary equal to the neighbouring point.

**Dirichlet boundary**: 'u' fixed with a value of zero on boundaries

-----Display options-----

`display_option`: if turned on, you will see the output information throughout the simulation, like the picture given followed:

if you truned it off, a `std::out` sentence only indicates the running time of the code will be given.

## 1.2.2 Postprocessing outputs

After you run a successful simulation, the output files would be saved at the `./output` folder in a binary format. (check they are there! )

You could then use python scripts to generate plots and animations from those binary files.

-----Generate a single shot for a specific time point in the simulation-----

If you are interested at the image of a single output file and want to view it, just type

```
$ python3 pic.py (number of the output)
```

It would give you a picture shot of the output state you are interested in, like following:

```

```

If you want to save this figure, add "-save" flag at the end of the command, like:

```
$ python3 pic.py (number of the output) -save
```

The picture would be save at the `./pic` folder.

-----Generate an animation for the simulation-----

If you want to view an animation for the simulation, just type

```
$ python3 animation.py
```

You can also enter any integer after a '-s' flag, indicating a speedup ratio, such as

```
$ python3 animation.py -s 8
```

To make the animation play 8 times faster.

Note that if you do not add the '-s' flag, the speed will just set default as 1.

If you want to save the animation, just add '-save' flag at the end, like

```
$ python3 animation.py -save
```

The animation will be save at `./anime` folder, with name like "anime [length] \* [width] grid (Mesh size: [imax] \* [jmax]) dt = xxx, boundary\_type = xxx.mp4".

I have attached the videos for three\_example runs in the `./anime` folder, with intial condition being two splashes of radius 1.12381 and 3.2313, locating at (3.12312, 3.31312) and (9.2313, 9.12312) respectively.

### 1.2.3 Something important to note

Before you run another simulation, if you are not changing 'mesh size' or 'end\_time', then it would be possibly be safe to rerun directly, but it's safer to clear the outputs of the last run before running a new simulation. Here I provide a script to clear the outputs, just use

```
bash ./clear_output.sh
```

to clear the outputs and then change the parameters in `parameters.ini` and run a new simulation.

Many thanks for using this piece of software!!!!



## Chapter 2

# Namespace Index

### 2.1 Namespace List

Here is a list of all namespaces with brief descriptions:

<a href="#">animation</a>	11
<a href="#">loaddata</a>	16
<a href="#">pic</a>	17



## Chapter 3

# Class Index

### 3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

<a href="#">read_parameters</a>	.....	<a href="#">23</a>
<a href="#">wave_data</a>	.....	<a href="#">24</a>





## Chapter 4

# File Index

### 4.1 File List

Here is a list of all files with brief descriptions:

<a href="#">animation.py</a>	35
<a href="#">loaddata.py</a>	36
<a href="#">main.cpp</a>	36
<a href="#">pic.py</a>	37
<a href="#">read_parameters.cpp</a>	37
<a href="#">read_parameters.h</a>	39
<a href="#">wave.cpp</a>	39
<a href="#">wave.h</a>	47



## Chapter 5

# Namespace Documentation

### 5.1 animation Namespace Reference

#### Functions

- `def update_grid (int picnum)`

#### Variables

- `num_pics = len(glob.glob('./output/*'))`
- `parameters = loaddata.read_parameters()`
- `t_out = float(parameters['t_out'])`
- `width = int(parameters['width'])`
- `length = int(parameters['length'])`
- `imax = int(parameters['imax'])`
- `jmax = int(parameters['jmax'])`
- `dt = float(parameters['dt'])`
- `boundary_type = str(parameters['boundary_type'])`
- `ele_x = int(length / 5)`
- `ele_y = int(width / 5)`
- `ele_i = int(imax / 5)`
- `ele_j = int(jmax / 5)`
- `list x_label = [ele_x * i for i in range(6)]`
- `list y_label = [ele_y * i for i in range(6)]`
- `list i_loc = [ele_i * i for i in range(6)]`
- `list j_loc = [ele_j * i for i in range(6)]`
- `speedratio = int(sys.argv[2])`
- `fig = plt.figure()`
- `ax = fig.add_subplot(111)`
- `fontsize`
- `text = plt.text(x = 0.95, y = 0.95, s='t = 0 s', fontdict=dict(fontsize=12, color='r', family='monospace',))`
- `arr = loaddata.load_binary_file(0).reshape((imax, jmax))`
- `norm = mpl.colors.Normalize(vmin = np.min(arr), vmax = np.max(arr))`
- `cax = ax.imshow(arr, interpolation='nearest', cmap = 'winter', norm = norm, origin = 'lower')`
- `cb = fig.colorbar(cax)`
- `anim = animation.FuncAnimation(fig, update_grid, frames = np.arange(0, num_pics, speedratio), blit=True)`
- `string ani_name = './anime/%d * %d grid (Mesh size: %d * %d) dt = %f, boundary_type = %s.mp4' % (length, width, imax, jmax, dt, boundary_type )`
- `fps`
- `dpi`

## 5.1.1 Function Documentation

### 5.1.1.1 update\_grid()

```
def animation.update_grid (
    int picnum )
```

Here is the call graph for this function:

## 5.1.2 Variable Documentation

### 5.1.2.1 ani\_name

```
animation.ani_name = './anime/%d * %d grid (Mesh size:  %d * %d) dt = %f, boundary_type = %s.%s.%s' % (length, width, imax, jmax, dt, boundary_type )
```

### 5.1.2.2 anim

```
animation.anim = animation.FuncAnimation(fig, update_grid, frames = np.arange(0, num_pics, speedratio), blit=True)
```

### 5.1.2.3 arr

```
animation.arr = loaddata.load_binary_file(0).reshape((imax, jmax))
```

### 5.1.2.4 ax

```
animation.ax = fig.add_subplot(111)
```

### 5.1.2.5 boundary\_type

```
animation.boundary_type = str(parameters['boundary_type'])
```

#### 5.1.2.6 cax

```
animation.cax = ax.imshow(arr, interpolation='nearest', cmap = 'winter', norm = norm, origin =  
'lower')
```

#### 5.1.2.7 cb

```
animation.cb = fig.colorbar(cax)
```

#### 5.1.2.8 dpi

```
animation.dpi
```

#### 5.1.2.9 dt

```
animation.dt = float(parameters['dt'])
```

#### 5.1.2.10 ele\_i

```
animation.ele_i = int(imax / 5)
```

#### 5.1.2.11 ele\_j

```
animation.ele_j = int(jmax / 5)
```

#### 5.1.2.12 ele\_x

```
animation.ele_x = int(length / 5)
```

#### 5.1.2.13 ele\_y

```
animation.ele_y = int(width / 5)
```

#### 5.1.2.14 fig

```
animation.fig = plt.figure()
```

#### 5.1.2.15 fontsize

```
animation.fontsize
```

#### 5.1.2.16 fps

```
animation.fps
```

#### 5.1.2.17 i\_loc

```
list animation.i_loc = [ele_i * i for i in range(6)]
```

#### 5.1.2.18 imax

```
animation.imax = int(parameters['imax'])
```

#### 5.1.2.19 j\_loc

```
list animation.j_loc = [ele_j * i for i in range(6)]
```

#### 5.1.2.20 jmax

```
animation.jmax = int(parameters['jmax'])
```

#### 5.1.2.21 length

```
animation.length = int(parameters['length'])
```

#### 5.1.2.22 norm

```
animation.norm = mpl.colors.Normalize(vmin = np.min(arr), vmax = np.max(arr))
```

#### 5.1.2.23 num\_pics

```
animation.num_pics = len(glob.glob('./output/*'))
```

#### 5.1.2.24 parameters

```
animation.parameters = loaddata.read_parameters()
```

#### 5.1.2.25 speedratio

```
int animation.speedratio = int(sys.argv[2])
```

#### 5.1.2.26 t\_out

```
animation.t_out = float(parameters['t_out'])
```

#### 5.1.2.27 text

```
animation.text = plt.text(x = 0.95, y = 0.95, s='t = 0 s', fontdict=dict(fontsize=12, color='r', family='monospa
```

#### 5.1.2.28 width

```
animation.width = int(parameters['width'])
```

#### 5.1.2.29 x\_label

```
list animation.x_label = [ele_x * i for i in range(6)]
```

#### 5.1.2.30 y\_label

```
list animation.y_label = [ele_y * i for i in range(6)]
```

## 5.2 loaddata Namespace Reference

### Functions

- def [load\\_binary\\_file](#) (int i)
- def [load\\_array](#) (int i)
- def [read\\_parameters](#) ()

### 5.2.1 Function Documentation

#### 5.2.1.1 load\_array()

```
def loaddata.load_array (  
    int i )
```

#### 5.2.1.2 load\_binary\_file()

```
def loaddata.load_binary_file (  
    int i )
```

#### 5.2.1.3 read\_parameters()

```
def loaddata.read_parameters ( )
```



## 5.3 pic Namespace Reference

### Variables

- `parameters` = `loaddata.read_parameters()`
- `t_out` = `float(parameters['t_out'])`
- `width` = `int(parameters['width'])`
- `length` = `int(parameters['length'])`
- `imax` = `int(parameters['imax'])`
- `jmax` = `int(parameters['jmax'])`
- `dt` = `float(parameters['dt'])`
- `boundary_type` = `str(parameters['boundary_type'])`
- `picnum` = `int(sys.argv[1])`
- `data` = `loaddata.load_binary_file(picnum).reshape((imax, jmax))`
- `ele_x` = `int(length / 5)`
- `ele_y` = `int(width / 5)`
- `ele_i` = `int(imax / 5)`
- `ele_j` = `int(jmax / 5)`
- `list_x_label` = `[ele_x * i for i in range(6)]`
- `list_y_label` = `[ele_y * i for i in range(6)]`
- `list_i_loc` = `[ele_i * i for i in range(6)]`
- `list_j_loc` = `[ele_j * i for i in range(6)]`
- `t` = `t_out * picnum`
- `fig` = `plt.figure()`
- `ax` = `fig.add_subplot(111)`
- `fontsize`
- `text` = `plt.text(x = 0.95, y = 0.95, s='t = %.3f s' % (t), fontdict=dict(fontsize=12, color='r', family='monospace',))`
- `arr` = `data`
- `norm` = `mpl.colors.Normalize(vmin = np.min(arr), vmax = np.max(arr))`
- `cax` = `ax.imshow(arr, interpolation='nearest', cmap = 'winter', norm = norm, origin = 'lower')`
- `cb` = `fig.colorbar(cax)`
- `string pic_name` = `'./pic/%d * %d grid (Mesh size: %d * %d) t = %f, boundary_type = %s.png' % (length, width, imax, jmax, t, boundary_type)`
- `dpi`

### 5.3.1 Variable Documentation

#### 5.3.1.1 `arr`

```
pic.arr = data
```

#### 5.3.1.2 `ax`

```
pic.ax = fig.add_subplot(111)
```

### 5.3.1.3 boundary\_type

```
pic.boundary_type = str(parameters['boundary_type'])
```

### 5.3.1.4 cax

```
pic.cax = ax.imshow(arr, interpolation='nearest', cmap = 'winter', norm = norm, origin = 'lower')
```

### 5.3.1.5 cb

```
pic.cb = fig.colorbar(cax)
```

### 5.3.1.6 data

```
pic.data = loaddata.load_binary_file(picnum).reshape((imax, jmax))
```

### 5.3.1.7 dpi

```
pic.dpi
```

### 5.3.1.8 dt

```
pic.dt = float(parameters['dt'])
```

### 5.3.1.9 ele\_i

```
pic.ele_i = int(imax / 5)
```

### 5.3.1.10 ele\_j

```
pic.ele_j = int(jmax / 5)
```

#### 5.3.1.11 ele\_x

```
pic.ele_x = int(length / 5)
```

#### 5.3.1.12 ele\_y

```
pic.ele_y = int(width / 5)
```

#### 5.3.1.13 fig

```
pic.fig = plt.figure()
```

#### 5.3.1.14 fontsize

```
pic.fontsize
```

#### 5.3.1.15 i\_loc

```
list pic.i_loc = [ele_i * i for i in range(6)]
```

#### 5.3.1.16 imax

```
pic.imax = int(parameters['imax'])
```

#### 5.3.1.17 j\_loc

```
list pic.j_loc = [ele_j * i for i in range(6)]
```

#### 5.3.1.18 jmax

```
pic.jmax = int(parameters['jmax'])
```

#### 5.3.1.19 length

```
pic.length = int(parameters['length'])
```

#### 5.3.1.20 norm

```
pic.norm = mpl.colors.Normalize(vmin = np.min(arr), vmax = np.max(arr))
```

#### 5.3.1.21 parameters

```
pic.parameters = loaddata.read_parameters()
```

#### 5.3.1.22 pic\_name

```
pic.pic_name = './pic/%d * %d grid (Mesh size: %d * %d) t = %f, boundary_type = %s.png' %  
(length, width, imax, jmax, t, boundary_type)
```

#### 5.3.1.23 picnum

```
int pic.picnum = int(sys.argv[1])
```

#### 5.3.1.24 t

```
pic.t = t_out * picnum
```

#### 5.3.1.25 t\_out

```
pic.t_out = float(parameters['t_out'])
```

#### 5.3.1.26 text

```
pic.text = plt.text(x = 0.95, y = 0.95, s='t = %.3f s' % (t), fontdict=dict(fontsize=12, color='r', family='monospace'))
```

#### 5.3.1.27 width

```
pic.width = int(parameters['width'])
```

#### 5.3.1.28 x\_label

```
list pic.x_label = [ele_x * i for i in range(6)]
```

#### 5.3.1.29 y\_label

```
list pic.y_label = [ele_y * i for i in range(6)]
```



## Chapter 6

# Class Documentation

### 6.1 read\_parameters Class Reference

```
#include <read_parameters.h>
```

#### Public Member Functions

- [read\\_parameters](#) ()
- [~read\\_parameters](#) ()
- string [getValue](#) (const string &name)
- void [set\\_splashes](#) ()
- void [print\\_parameters](#) ()

#### Public Attributes

- ifstream \* [infile](#)
- double [length](#)
- double [width](#)
- int [imax](#)
- int [jmax](#)
- double [c](#)
- double [dt](#)
- string [boundary\\_type](#)
- bool [display\\_option](#)
- bool [output](#)
- double [t\\_end](#)
- double [t\\_out](#)
- double \* [rs](#)
- double \* [xs](#)
- double \* [ys](#)
- int [num\\_splash](#)

#### 6.1.1 Constructor & Destructor Documentation

### 6.1.1.1 read\_parameters()

```
read_parameters::read_parameters ( )
```

Here is the call graph for this function:

## 6.2 wave\_data Class Reference

```
#include <wave.h>
```

Collaboration diagram for wave\_data:

### Public Member Functions

- [wave\\_data](#) ()
- [~wave\\_data](#) ()
- void [build\\_OutputType](#) ()
- void [setting\\_paramaters](#) ()
- void [allocate\\_grid](#) ()
- void [set\\_initial\\_condition](#) ()
- void [MPI\\_grid\\_to\\_file](#) ()
- void [set\\_Neumman\\_boundary\\_u](#) ()
- void [set\\_Neumman\\_boundary\\_new\\_u](#) ()
- void [set\\_Neumman\\_boundary\\_old\\_u](#) ()
- void [find\\_neighbour\\_for\\_other\\_processor](#) ()
- void [build\\_communication\\_type](#) ()
- void [do\\_communication](#) ()
- void [do\\_iteration](#) ()
- void [run\\_simulation](#) ()

### Public Attributes

- double \* [u](#)
- double \* [old\\_u](#)
- double \* [new\\_u](#)
- [read\\_parameters](#) parameters
- int [imax](#)
- int [jmax](#)
- double [t\\_max](#)
- double [t](#) = 0.0
- double [t\\_out](#) = 0.0
- double [dt\\_out](#)
- double [dt](#)
- double [y\\_max](#)
- double [x\\_max](#)
- double [dx](#)
- double [dy](#)
- double [c](#)
- bool [display](#)
- bool [output](#)



- int `n`
- int `intdiv`
- int `mod`
- int `start_index`
- int `end_index`
- int \* `x_index`
- int \* `y_index`
- int `line_feeds`
- int `line_start`
- int `line_end`
- int `n_line`
- int `isolated_num`
- int `non_isolated_num`
- int \* `block_lengths`
- vector< int > `index_connect_up_edge`
- vector< int > `index_connect_down_edge`
- vector< int > `index_connect_left_edge`
- vector< int > `index_connect_right_edge`
- int `up_edge_length`
- int `down_edge_length`
- int `left_edge_length`
- int `right_edge_length`
- double \* `up_boundaries`
- double \* `down_boundaries`
- double \* `left_boundaries`
- double \* `right_boundaries`
- int \* `send_length`
- double \*\* `receive_from_neighbour`
- vector< int > `index_isolated`
- vector< int > `index_not_isolated`
- bool `left_up_corner`
- bool `right_up_corner`
- bool `left_down_corner`
- bool `right_down_corner`

## 6.2.1 Constructor & Destructor Documentation

### 6.2.1.1 wave\_data()

```
wave_data::wave_data ( )
```

### 6.2.1.2 ~wave\_data()

```
wave_data::~~wave_data ( )
```

## 6.2.2 Member Function Documentation

### 6.2.2.1 allocate\_grid()

```
void wave_data::allocate_grid ( )
```

### 6.2.2.2 build\_communication\_type()

```
void wave_data::build_communication_type ( )
```

### 6.2.2.3 build\_OutputType()

```
void wave_data::build_OutputType ( )
```

### 6.2.2.4 do\_communication()

```
void wave_data::do_communication ( )
```

### 6.2.2.5 do\_iteration()

```
void wave_data::do_iteration ( )
```

### 6.2.2.6 find\_neighbour\_for\_other\_processor()

```
void wave_data::find_neighbour_for_other_processor ( )
```

Here is the call graph for this function:

### 6.2.2.7 MPI\_grid\_to\_file()

```
void wave_data::MPI_grid_to_file ( )
```

#### 6.2.2.8 run\_simulation()

```
void wave_data::run_simulation ( )
```

#### 6.2.2.9 set\_initial\_condition()

```
void wave_data::set_initial_condition ( )
```

#### 6.2.2.10 set\_Neumman\_boundary\_new\_u()

```
void wave_data::set_Neumman_boundary_new_u ( )
```

#### 6.2.2.11 set\_Neumman\_boundary\_old\_u()

```
void wave_data::set_Neumman_boundary_old_u ( )
```

#### 6.2.2.12 set\_Neumman\_boundary\_u()

```
void wave_data::set_Neumman_boundary_u ( )
```

#### 6.2.2.13 setting\_paramters()

```
void wave_data::setting_paramters ( )
```

### 6.2.3 Member Data Documentation

#### 6.2.3.1 block\_lengths

```
int* wave_data::block_lengths
```

#### 6.2.3.2 c

```
double wave_data::c
```

#### 6.2.3.3 display

```
bool wave_data::display
```

#### 6.2.3.4 down\_boundaries

```
double* wave_data::down_boundaries
```

#### 6.2.3.5 down\_edge\_length

```
int wave_data::down_edge_length
```

#### 6.2.3.6 dt

```
double wave_data::dt
```

#### 6.2.3.7 dt\_out

```
double wave_data::dt_out
```

#### 6.2.3.8 dx

```
double wave_data::dx
```

#### 6.2.3.9 dy

```
double wave_data::dy
```

### 6.2.3.10 end\_index

```
int wave_data::end_index
```

### 6.2.3.11 imax

```
int wave_data::imax
```

### 6.2.3.12 index\_connect\_down\_edge

```
vector<int> wave_data::index_connect_down_edge
```

### 6.2.3.13 index\_connect\_left\_edge

```
vector<int> wave_data::index_connect_left_edge
```

### 6.2.3.14 index\_connect\_right\_edge

```
vector<int> wave_data::index_connect_right_edge
```

### 6.2.3.15 index\_connect\_up\_edge

```
vector<int> wave_data::index_connect_up_edge
```

### 6.2.3.16 index\_isolated

```
vector<int> wave_data::index_isolated
```

### 6.2.3.17 index\_not\_isolated

```
vector<int> wave_data::index_not_isolated
```

**6.2.3.18 intdiv**

```
int wave_data::intdiv
```

**6.2.3.19 isolated\_num**

```
int wave_data::isolated_num
```

**6.2.3.20 jmax**

```
int wave_data::jmax
```

**6.2.3.21 left\_boundaries**

```
double* wave_data::left_boundaries
```

**6.2.3.22 left\_down\_corner**

```
bool wave_data::left_down_corner
```

**6.2.3.23 left\_edge\_length**

```
int wave_data::left_edge_length
```

**6.2.3.24 left\_up\_corner**

```
bool wave_data::left_up_corner
```

**6.2.3.25 line\_end**

```
int wave_data::line_end
```

#### 6.2.3.26 line\_feeds

```
int wave_data::line_feeds
```

#### 6.2.3.27 line\_start

```
int wave_data::line_start
```

#### 6.2.3.28 mod

```
int wave_data::mod
```

#### 6.2.3.29 n

```
int wave_data::n
```

#### 6.2.3.30 n\_line

```
int wave_data::n_line
```

#### 6.2.3.31 new\_u

```
double* wave_data::new_u
```

#### 6.2.3.32 non\_isolated\_num

```
int wave_data::non_isolated_num
```

#### 6.2.3.33 old\_u

```
double* wave_data::old_u
```

#### 6.2.3.34 output

```
bool wave_data::output
```

#### 6.2.3.35 parameters

```
read_parameters wave_data::parameters
```

#### 6.2.3.36 receive\_from\_neighbour

```
double** wave_data::receive_from_neighbour
```

#### 6.2.3.37 right\_boundaries

```
double* wave_data::right_boundaries
```

#### 6.2.3.38 right\_down\_corner

```
bool wave_data::right_down_corner
```

#### 6.2.3.39 right\_edge\_length

```
int wave_data::right_edge_length
```

#### 6.2.3.40 right\_up\_corner

```
bool wave_data::right_up_corner
```

#### 6.2.3.41 send\_length

```
int* wave_data::send_length
```



#### 6.2.3.42 start\_index

```
int wave_data::start_index
```

#### 6.2.3.43 t

```
double wave_data::t = 0.0
```

#### 6.2.3.44 t\_max

```
double wave_data::t_max
```

#### 6.2.3.45 t\_out

```
double wave_data::t_out = 0.0
```

#### 6.2.3.46 u

```
double* wave_data::u
```

#### 6.2.3.47 up\_boundaries

```
double* wave_data::up_boundaries
```

#### 6.2.3.48 up\_edge\_length

```
int wave_data::up_edge_length
```

#### 6.2.3.49 x\_index

```
int* wave_data::x_index
```

**6.2.3.50 x\_max**

```
double wave_data::x_max
```

**6.2.3.51 y\_index**

```
int* wave_data::y_index
```

**6.2.3.52 y\_max**

```
double wave_data::y_max
```

The documentation for this class was generated from the following files:

- [wave.h](#)
- [wave.cpp](#)

## Chapter 7

# File Documentation

### 7.1 animation.py File Reference

#### Namespaces

- [animation](#)

#### Functions

- def [animation.update\\_grid](#) (int picnum)

#### Variables

- [animation.num\\_pics](#) = len(glob.glob('./output/\*'))
- [animation.parameters](#) = [loaddata.read\\_parameters](#)()
- [animation.t\\_out](#) = float([parameters](#)['t\_out'])
- [animation.width](#) = int([parameters](#)['width'])
- [animation.length](#) = int([parameters](#)['length'])
- [animation.imax](#) = int([parameters](#)['imax'])
- [animation.jmax](#) = int([parameters](#)['jmax'])
- [animation.dt](#) = float([parameters](#)['dt'])
- [animation.boundary\\_type](#) = str([parameters](#)['boundary\_type'])
- [animation.ele\\_x](#) = int(length / 5)
- [animation.ele\\_y](#) = int(width / 5)
- [animation.ele\\_i](#) = int(imax / 5)
- [animation.ele\\_j](#) = int(jmax / 5)
- list [animation.x\\_label](#) = [ele\_x \* i for i in range(6)]
- list [animation.y\\_label](#) = [ele\_y \* i for i in range(6)]
- list [animation.i\\_loc](#) = [ele\_i \* i for i in range(6)]
- list [animation.j\\_loc](#) = [ele\_j \* i for i in range(6)]
- [animation.speedratio](#) = int(sys.argv[2])
- [animation.fig](#) = plt.figure()
- [animation.ax](#) = fig.add\_subplot(111)
- [animation.fontsize](#)
- [animation.text](#) = plt.text(x = 0.95, y = 0.95, s='t = 0 s', fontdict=dict(fontsize=12, color='r', family='monospace',))
- [animation.arr](#) = [loaddata.load\\_binary\\_file](#)(0).reshape((imax, jmax))

- [animation.norm](#) = `mpl.colors.Normalize(vmin = np.min(arr), vmax = np.max(arr))`
- [animation.cax](#) = `ax.imshow(arr, interpolation='nearest', cmap = 'winter', norm = norm, origin = 'lower')`
- [animation.cb](#) = `fig.colorbar(cax)`
- [animation.anim](#) = `animation.FuncAnimation(fig, update_grid, frames = np.arange(0, num_pics, speedratio), blit=True)`
- string [animation.ani\\_name](#) = `'./anime/%d * %d grid (Mesh size: %d * %d) dt = %f, boundary_type = %s.mp4'`  
% (length, width, imax, jmax, dt, boundary\_type )
- [animation.fps](#)
- [animation.dpi](#)

## 7.2 loaddata.py File Reference

### Namespaces

- [loaddata](#)

### Functions

- def [loaddata.load\\_binary\\_file](#) (int i)
- def [loaddata.load\\_array](#) (int i)
- def [loaddata.read\\_parameters](#) ()

## 7.3 main.cpp File Reference

```
#include "wave.cpp"
Include dependency graph for main.cpp:
```

### Functions

- int [main](#) (int argc, char \*argv[])

### 7.3.1 Function Documentation

#### 7.3.1.1 main()

```
int main (
    int argc,
    char * argv[] )
```

Here is the call graph for this function:

## 7.4 pic.py File Reference

### Namespaces

- [pic](#)

### Variables

- [pic.parameters](#) = [loaddata.read\\_parameters\(\)](#)
- [pic.t\\_out](#) = [float\(parameters\['t\\_out'\]\)](#)
- [pic.width](#) = [int\(parameters\['width'\]\)](#)
- [pic.length](#) = [int\(parameters\['length'\]\)](#)
- [pic.imax](#) = [int\(parameters\['imax'\]\)](#)
- [pic.jmax](#) = [int\(parameters\['jmax'\]\)](#)
- [pic.dt](#) = [float\(parameters\['dt'\]\)](#)
- [pic.boundary\\_type](#) = [str\(parameters\['boundary\\_type'\]\)](#)
- [pic.picnum](#) = [int\(sys.argv\[1\]\)](#)
- [pic.data](#) = [loaddata.load\\_binary\\_file\(picnum\).reshape\(\(imax, jmax\)\)](#)
- [pic.ele\\_x](#) = [int\(length / 5\)](#)
- [pic.ele\\_y](#) = [int\(width / 5\)](#)
- [pic.ele\\_i](#) = [int\(imax / 5\)](#)
- [pic.ele\\_j](#) = [int\(jmax / 5\)](#)
- list [pic.x\\_label](#) = [\[ele\\_x \\* i for i in range\(6\)\]](#)
- list [pic.y\\_label](#) = [\[ele\\_y \\* i for i in range\(6\)\]](#)
- list [pic.i\\_loc](#) = [\[ele\\_i \\* i for i in range\(6\)\]](#)
- list [pic.j\\_loc](#) = [\[ele\\_j \\* i for i in range\(6\)\]](#)
- [pic.t](#) = [t\\_out \\* picnum](#)
- [pic.fig](#) = [plt.figure\(\)](#)
- [pic.ax](#) = [fig.add\\_subplot\(111\)](#)
- [pic.fontsize](#)
- [pic.text](#) = [plt.text\(x = 0.95, y = 0.95, s='t = %.3f s' % \(t\), fontdict=dict\(fontsize=12, color='r', family='monospace',\)\)](#)
- [pic.arr](#) = [data](#)
- [pic.norm](#) = [mpl.colors.Normalize\(vmin = np.min\(arr\), vmax = np.max\(arr\)\)](#)
- [pic.cax](#) = [ax.imshow\(arr, interpolation='nearest', cmap = 'winter', norm = norm, origin = 'lower'\)](#)
- [pic.cb](#) = [fig.colorbar\(cax\)](#)
- string [pic.pic\\_name](#) = ['./pic/%d \\* %d grid \(Mesh size: %d \\* %d\) t = %f, boundary\\_type = %s.png' % \(length, width, imax, jmax, t, boundary\\_type\)](#)
- [pic.dpi](#)

## 7.5 read\_parameters.cpp File Reference

```
#include "read_parameters.h"
```

Include dependency graph for read\_parameters.cpp: This graph shows which files directly or indirectly include this file:

### Macros

- [#define COMMENT\\_CHAR](#) `'.'`

## Functions

- bool [IsCommentChar](#) (char c)
- void [deleteSpace](#) (string &str)
- void [TrimSpaces](#) (string &str)
- void [TrimMarks](#) (string &str)
- vector< string > [extractArray](#) (string &str)

## 7.5.1 Macro Definition Documentation

### 7.5.1.1 COMMENT\_CHAR

```
#define COMMENT_CHAR '-'
```

## 7.5.2 Function Documentation

### 7.5.2.1 deleteSpace()

```
void deleteSpace (  
    string & str )
```

### 7.5.2.2 extractArray()

```
vector<string> extractArray (  
    string & str )
```

Here is the call graph for this function:

### 7.5.2.3 IsCommentChar()

```
bool IsCommentChar (  
    char c )
```

### 7.5.2.4 TrimMarks()

```
void TrimMarks (  
    string & str )
```

### 7.5.2.5 TrimSpaces()

```
void TrimSpaces (
    string & str )
```

Here is the call graph for this function:

## 7.6 read\_parameters.h File Reference

```
#include <ctype.h>
#include <fstream>
#include <iostream>
#include <vector>
```

Include dependency graph for read\_parameters.h: This graph shows which files directly or indirectly include this file:

### Classes

- class [read\\_parameters](#)

## 7.7 README.md File Reference

## 7.8 wave.cpp File Reference

```
#include <mpi.h>
#include <iostream>
#include <sstream>
#include <fstream>
#include <chrono>
#include <cstdlib>
#include <time.h>
#include <vector>
#include <cmath>
#include "wave.h"
#include "read_parameters.cpp"
```

Include dependency graph for wave.cpp: This graph shows which files directly or indirectly include this file:

### Functions

- int [find\\_processor](#) (int a)

## Variables

- [read\\_parameters](#) parameters
- int [out\\_cnt](#) = 0
- int [it](#) = 0
- int [num](#)
- int [idx](#)
- chrono::high\_resolution\_clock::time\_point [start](#)
- chrono::high\_resolution\_clock::time\_point [finish](#)
- std::chrono::duration< double > [elapsed](#)
- int [id](#)
- int [p](#)
- int [tag\\_num](#) = 1
- const double [zero](#) = 0
- int \* [start\\_indexs](#)
- int \* [num\\_allocates](#)
- int \*\* [receive\\_index](#)
- MPI\_Datatype \* [Send\\_to\\_neighbour](#)
- MPI\_Datatype \* [Send\\_to\\_neighbour2](#)
- MPI\_Datatype \* [Send\\_to\\_neighbour3](#)
- MPI\_Aint \*\* [Send\\_to\\_neighbour\\_displacements](#)
- MPI\_Aint \*\* [Send\\_to\\_neighbour\\_displacements2](#)
- MPI\_Aint \*\* [Send\\_to\\_neighbour\\_displacements3](#)
- MPI\_Aint \*\* [Send\\_to\\_neighbour\\_address](#)
- MPI\_Aint \*\* [Send\\_to\\_neighbour\\_address2](#)
- MPI\_Aint \*\* [Send\\_to\\_neighbour\\_address3](#)
- MPI\_Request \* [requests](#)
- int [cnt](#)
- vector< int > \* [index\\_right\\_for\\_processor](#)
- vector< int > \* [index\\_left\\_for\\_processor](#)
- vector< int > \* [index\\_up\\_for\\_processor](#)
- vector< int > \* [index\\_down\\_for\\_processor](#)
- double \*\* [left\\_1](#)
- double \*\* [left\\_2](#)
- double \*\* [left\\_3](#)
- double \*\* [right\\_1](#)
- double \*\* [right\\_2](#)
- double \*\* [right\\_3](#)
- double \*\* [up\\_1](#)
- double \*\* [up\\_2](#)
- double \*\* [up\\_3](#)
- double \*\* [down\\_1](#)
- double \*\* [down\\_2](#)
- double \*\* [down\\_3](#)
- MPI\_Datatype [UpEdge](#)
- MPI\_Datatype [OutputType](#)
- MPI\_Datatype [DownEdge](#)
- MPI\_Datatype [OutputType2](#)
- MPI\_Datatype [OutputType3](#)
- MPI\_Aint [add\\_start](#)

### 7.8.1 Function Documentation



### 7.8.1.1 find\_processor()

```
int find_processor (
    int a )
```

## 7.8.2 Variable Documentation

### 7.8.2.1 add\_start

```
MPI_Aint add_start
```

### 7.8.2.2 cnt

```
int cnt
```

### 7.8.2.3 down\_1

```
double** down_1
```

### 7.8.2.4 down\_2

```
double** down_2
```

### 7.8.2.5 down\_3

```
double** down_3
```

### 7.8.2.6 DownEdge

```
MPI_Datatype DownEdge
```

#### 7.8.2.7 elapsed

```
std::chrono::duration<double> elapsed
```

#### 7.8.2.8 finish

```
chrono::high_resolution_clock::time_point finish
```

#### 7.8.2.9 id

```
int id
```

#### 7.8.2.10 idx

```
int idx
```

#### 7.8.2.11 index\_down\_for\_processor

```
vector<int>* index_down_for_processor
```

#### 7.8.2.12 index\_left\_for\_processor

```
vector<int>* index_left_for_processor
```

#### 7.8.2.13 index\_right\_for\_processor

```
vector<int>* index_right_for_processor
```

#### 7.8.2.14 index\_up\_for\_processor

```
vector<int>* index_up_for_processor
```

#### 7.8.2.15 it

```
int it = 0
```

#### 7.8.2.16 left\_1

```
double** left_1
```

#### 7.8.2.17 left\_2

```
double** left_2
```

#### 7.8.2.18 left\_3

```
double** left_3
```

#### 7.8.2.19 num

```
int num
```

#### 7.8.2.20 num\_allocates

```
int* num_allocates
```

#### 7.8.2.21 out\_cnt

```
int out_cnt = 0
```

#### 7.8.2.22 OutputType

```
MPI_Datatype OutputType
```

#### 7.8.2.23 OutputType2

`MPI_Datatype OutputType2`

#### 7.8.2.24 OutputType3

`MPI_Datatype OutputType3`

#### 7.8.2.25 p

`int p`

#### 7.8.2.26 parameters

`read_parameters parameters`

#### 7.8.2.27 receive\_index

`int** receive_index`

#### 7.8.2.28 requests

`MPI_Request* requests`

#### 7.8.2.29 right\_1

`double** right_1`

#### 7.8.2.30 right\_2

`double** right_2`

### 7.8.2.31 right\_3

```
double** right_3
```

### 7.8.2.32 Send\_to\_neighbour

```
MPI_Datatype* Send_to_neighbour
```

### 7.8.2.33 Send\_to\_neighbour2

```
MPI_Datatype* Send_to_neighbour2
```

### 7.8.2.34 Send\_to\_neighbour3

```
MPI_Datatype* Send_to_neighbour3
```

### 7.8.2.35 Send\_to\_neighbour\_address

```
MPI_Aint** Send_to_neighbour_address
```

### 7.8.2.36 Send\_to\_neighbour\_address2

```
MPI_Aint** Send_to_neighbour_address2
```

### 7.8.2.37 Send\_to\_neighbour\_address3

```
MPI_Aint** Send_to_neighbour_address3
```

### 7.8.2.38 Send\_to\_neighbour\_displacements

```
MPI_Aint** Send_to_neighbour_displacements
```

#### 7.8.2.39 Send\_to\_neighbour\_displacements2

```
MPI_Aint** Send_to_neighbour_displacements2
```

#### 7.8.2.40 Send\_to\_neighbour\_displacements3

```
MPI_Aint** Send_to_neighbour_displacements3
```

#### 7.8.2.41 start

```
chrono::high_resolution_clock::time_point start
```

#### 7.8.2.42 start\_indexs

```
int* start_indexs
```

#### 7.8.2.43 tag\_num

```
int tag_num = 1
```

#### 7.8.2.44 up\_1

```
double** up_1
```

#### 7.8.2.45 up\_2

```
double** up_2
```

#### 7.8.2.46 up\_3

```
double** up_3
```

#### 7.8.2.47 UpEdge

```
MPI_Datatype UpEdge
```

#### 7.8.2.48 zero

```
const double zero = 0
```

## 7.9 wave.h File Reference

```
#include "read_parameters.h"
```

Include dependency graph for wave.h: This graph shows which files directly or indirectly include this file:

### Classes

- class [wave\\_data](#)





# Index

- ~wave\_data
  - wave\_data, [25](#)
- add\_start
  - wave.cpp, [41](#)
- allocate\_grid
  - wave\_data, [26](#)
- ani\_name
  - animation, [12](#)
- anim
  - animation, [12](#)
- animation, [11](#)
  - ani\_name, [12](#)
  - anim, [12](#)
  - arr, [12](#)
  - ax, [12](#)
  - boundary\_type, [12](#)
  - cax, [12](#)
  - cb, [13](#)
  - dpi, [13](#)
  - dt, [13](#)
  - ele\_i, [13](#)
  - ele\_j, [13](#)
  - ele\_x, [13](#)
  - ele\_y, [13](#)
  - fig, [14](#)
  - fontsize, [14](#)
  - fps, [14](#)
  - i\_loc, [14](#)
  - imax, [14](#)
  - j\_loc, [14](#)
  - jmax, [14](#)
  - length, [14](#)
  - norm, [15](#)
  - num\_pics, [15](#)
  - parameters, [15](#)
  - speedratio, [15](#)
  - t\_out, [15](#)
  - text, [15](#)
  - update\_grid, [12](#)
  - width, [15](#)
  - x\_label, [15](#)
  - y\_label, [16](#)
- animation.py, [35](#)
- arr
  - animation, [12](#)
  - pic, [17](#)
- ax
  - animation, [12](#)
  - pic, [17](#)
- block\_lengths
  - wave\_data, [27](#)
- boundary\_type
  - animation, [12](#)
  - pic, [17](#)
- build\_communication\_type
  - wave\_data, [26](#)
- build\_OutputType
  - wave\_data, [26](#)
- c
  - wave\_data, [27](#)
- cax
  - animation, [12](#)
  - pic, [18](#)
- cb
  - animation, [13](#)
  - pic, [18](#)
- cnt
  - wave.cpp, [41](#)
- COMMENT\_CHAR
  - read\_parameters.cpp, [38](#)
- data
  - pic, [18](#)
- deleteSpace
  - read\_parameters.cpp, [38](#)
- display
  - wave\_data, [28](#)
- do\_communication
  - wave\_data, [26](#)
- do\_iteration
  - wave\_data, [26](#)
- down\_1
  - wave.cpp, [41](#)
- down\_2
  - wave.cpp, [41](#)
- down\_3
  - wave.cpp, [41](#)
- down\_boundaries
  - wave\_data, [28](#)
- down\_edge\_length
  - wave\_data, [28](#)
- DownEdge
  - wave.cpp, [41](#)
- dpi
  - animation, [13](#)
  - pic, [18](#)
- dt
  - animation, [13](#)

- pic, [18](#)
  - wave\_data, [28](#)
- dt\_out
  - wave\_data, [28](#)
- dx
  - wave\_data, [28](#)
- dy
  - wave\_data, [28](#)
- elapsed
  - wave.cpp, [41](#)
- ele\_i
  - animation, [13](#)
  - pic, [18](#)
- ele\_j
  - animation, [13](#)
  - pic, [18](#)
- ele\_x
  - animation, [13](#)
  - pic, [18](#)
- ele\_y
  - animation, [13](#)
  - pic, [19](#)
- end\_index
  - wave\_data, [28](#)
- extractArray
  - read\_parameters.cpp, [38](#)
- fig
  - animation, [14](#)
  - pic, [19](#)
- find\_neighbour\_for\_other\_processor
  - wave\_data, [26](#)
- find\_processor
  - wave.cpp, [40](#)
- finish
  - wave.cpp, [42](#)
- fontsize
  - animation, [14](#)
  - pic, [19](#)
- fps
  - animation, [14](#)
- i\_loc
  - animation, [14](#)
  - pic, [19](#)
- id
  - wave.cpp, [42](#)
- idx
  - wave.cpp, [42](#)
- imax
  - animation, [14](#)
  - pic, [19](#)
  - wave\_data, [29](#)
- index\_connect\_down\_edge
  - wave\_data, [29](#)
- index\_connect\_left\_edge
  - wave\_data, [29](#)
- index\_connect\_right\_edge
  - wave\_data, [29](#)
- index\_connect\_up\_edge
  - wave\_data, [29](#)
- index\_down\_for\_processor
  - wave.cpp, [42](#)
- index\_isolated
  - wave\_data, [29](#)
- index\_left\_for\_processor
  - wave.cpp, [42](#)
- index\_not\_isolated
  - wave\_data, [29](#)
- index\_right\_for\_processor
  - wave.cpp, [42](#)
- index\_up\_for\_processor
  - wave.cpp, [42](#)
- intdiv
  - wave\_data, [29](#)
- IsCommentChar
  - read\_parameters.cpp, [38](#)
- isolated\_num
  - wave\_data, [30](#)
- it
  - wave.cpp, [42](#)
- j\_loc
  - animation, [14](#)
  - pic, [19](#)
- jmax
  - animation, [14](#)
  - pic, [19](#)
  - wave\_data, [30](#)
- left\_1
  - wave.cpp, [43](#)
- left\_2
  - wave.cpp, [43](#)
- left\_3
  - wave.cpp, [43](#)
- left\_boundaries
  - wave\_data, [30](#)
- left\_down\_corner
  - wave\_data, [30](#)
- left\_edge\_length
  - wave\_data, [30](#)
- left\_up\_corner
  - wave\_data, [30](#)
- length
  - animation, [14](#)
  - pic, [19](#)
- line\_end
  - wave\_data, [30](#)
- line\_feeds
  - wave\_data, [30](#)
- line\_start
  - wave\_data, [31](#)
- load\_array
  - loaddata, [16](#)
- load\_binary\_file
  - loaddata, [16](#)

- loaddata, 16
  - load\_array, 16
  - load\_binary\_file, 16
  - read\_parameters, 16
- loaddata.py, 36
- main
  - main.cpp, 36
- main.cpp, 36
  - main, 36
- mod
  - wave\_data, 31
- MPI\_grid\_to\_file
  - wave\_data, 26
- n
  - wave\_data, 31
- n\_line
  - wave\_data, 31
- new\_u
  - wave\_data, 31
- non\_isolated\_num
  - wave\_data, 31
- norm
  - animation, 15
  - pic, 20
- num
  - wave.cpp, 43
- num\_allocates
  - wave.cpp, 43
- num\_pics
  - animation, 15
- old\_u
  - wave\_data, 31
- out\_cnt
  - wave.cpp, 43
- output
  - wave\_data, 31
- OutputType
  - wave.cpp, 43
- OutputType2
  - wave.cpp, 43
- OutputType3
  - wave.cpp, 44
- p
  - wave.cpp, 44
- parameters
  - animation, 15
  - pic, 20
  - wave.cpp, 44
  - wave\_data, 32
- pic, 17
  - arr, 17
  - ax, 17
  - boundary\_type, 17
  - cax, 18
  - cb, 18
  - data, 18
  - dpi, 18
  - dt, 18
  - ele\_i, 18
  - ele\_j, 18
  - ele\_x, 18
  - ele\_y, 19
  - fig, 19
  - fontsize, 19
  - i\_loc, 19
  - imax, 19
  - j\_loc, 19
  - jmax, 19
  - length, 19
  - norm, 20
  - parameters, 20
  - pic\_name, 20
  - picnum, 20
  - t, 20
  - t\_out, 20
  - text, 20
  - width, 21
  - x\_label, 21
  - y\_label, 21
- pic.py, 37
- pic\_name
  - pic, 20
- picnum
  - pic, 20
- read\_parameters, 23
  - loaddata, 16
  - read\_parameters, 23
- read\_parameters.cpp, 37
  - COMMENT\_CHAR, 38
  - deleteSpace, 38
  - extractArray, 38
  - IsCommentChar, 38
  - TrimMarks, 38
  - TrimSpaces, 38
- read\_parameters.h, 39
- README.md, 39
- receive\_from\_neighbour
  - wave\_data, 32
- receive\_index
  - wave.cpp, 44
- requests
  - wave.cpp, 44
- right\_1
  - wave.cpp, 44
- right\_2
  - wave.cpp, 44
- right\_3
  - wave.cpp, 44
- right\_boundaries
  - wave\_data, 32
- right\_down\_corner
  - wave\_data, 32
- right\_edge\_length

- wave\_data, 32
- right\_up\_corner
  - wave\_data, 32
- run\_simulation
  - wave\_data, 26
- send\_length
  - wave\_data, 32
- Send\_to\_neighbour
  - wave.cpp, 45
- Send\_to\_neighbour2
  - wave.cpp, 45
- Send\_to\_neighbour3
  - wave.cpp, 45
- Send\_to\_neighbour\_address
  - wave.cpp, 45
- Send\_to\_neighbour\_address2
  - wave.cpp, 45
- Send\_to\_neighbour\_address3
  - wave.cpp, 45
- Send\_to\_neighbour\_displacements
  - wave.cpp, 45
- Send\_to\_neighbour\_displacements2
  - wave.cpp, 45
- Send\_to\_neighbour\_displacements3
  - wave.cpp, 46
- set\_initial\_condition
  - wave\_data, 27
- set\_Neumman\_boundary\_new\_u
  - wave\_data, 27
- set\_Neumman\_boundary\_old\_u
  - wave\_data, 27
- set\_Neumman\_boundary\_u
  - wave\_data, 27
- setting\_paramters
  - wave\_data, 27
- speedratio
  - animation, 15
- start
  - wave.cpp, 46
- start\_index
  - wave\_data, 32
- start\_indexes
  - wave.cpp, 46
- t
  - pic, 20
  - wave\_data, 33
- t\_max
  - wave\_data, 33
- t\_out
  - animation, 15
  - pic, 20
  - wave\_data, 33
- tag\_num
  - wave.cpp, 46
- text
  - animation, 15
  - pic, 20
- TrimMarks
  - read\_parameters.cpp, 38
- TrimSpaces
  - read\_parameters.cpp, 38
- u
  - wave\_data, 33
- up\_1
  - wave.cpp, 46
- up\_2
  - wave.cpp, 46
- up\_3
  - wave.cpp, 46
- up\_boundaries
  - wave\_data, 33
- up\_edge\_length
  - wave\_data, 33
- update\_grid
  - animation, 12
- UpEdge
  - wave.cpp, 46
- wave.cpp, 39
  - add\_start, 41
  - cnt, 41
  - down\_1, 41
  - down\_2, 41
  - down\_3, 41
  - DownEdge, 41
  - elapsed, 41
  - find\_processor, 40
  - finish, 42
  - id, 42
  - idx, 42
  - index\_down\_for\_processor, 42
  - index\_left\_for\_processor, 42
  - index\_right\_for\_processor, 42
  - index\_up\_for\_processor, 42
  - it, 42
  - left\_1, 43
  - left\_2, 43
  - left\_3, 43
  - num, 43
  - num\_allocates, 43
  - out\_cnt, 43
  - OutputType, 43
  - OutputType2, 43
  - OutputType3, 44
  - p, 44
  - parameters, 44
  - receive\_index, 44
  - requests, 44
  - right\_1, 44
  - right\_2, 44
  - right\_3, 44
  - Send\_to\_neighbour, 45
  - Send\_to\_neighbour2, 45
  - Send\_to\_neighbour3, 45
  - Send\_to\_neighbour\_address, 45

- Send\_to\_neighbour\_address2, [45](#)
- Send\_to\_neighbour\_address3, [45](#)
- Send\_to\_neighbour\_displacements, [45](#)
- Send\_to\_neighbour\_displacements2, [45](#)
- Send\_to\_neighbour\_displacements3, [46](#)
- start, [46](#)
- start\_indexes, [46](#)
- tag\_num, [46](#)
- up\_1, [46](#)
- up\_2, [46](#)
- up\_3, [46](#)
- UpEdge, [46](#)
- zero, [47](#)
- wave.h, [47](#)
- wave\_data, [24](#)
  - ~wave\_data, [25](#)
  - allocate\_grid, [26](#)
  - block\_lengths, [27](#)
  - build\_communication\_type, [26](#)
  - build\_OutputType, [26](#)
  - c, [27](#)
  - display, [28](#)
  - do\_communication, [26](#)
  - do\_iteration, [26](#)
  - down\_boundaries, [28](#)
  - down\_edge\_length, [28](#)
  - dt, [28](#)
  - dt\_out, [28](#)
  - dx, [28](#)
  - dy, [28](#)
  - end\_index, [28](#)
  - find\_neighbour\_for\_other\_processor, [26](#)
  - imax, [29](#)
  - index\_connect\_down\_edge, [29](#)
  - index\_connect\_left\_edge, [29](#)
  - index\_connect\_right\_edge, [29](#)
  - index\_connect\_up\_edge, [29](#)
  - index\_isolated, [29](#)
  - index\_not\_isolated, [29](#)
  - intdiv, [29](#)
  - isolated\_num, [30](#)
  - jmax, [30](#)
  - left\_boundaries, [30](#)
  - left\_down\_corner, [30](#)
  - left\_edge\_length, [30](#)
  - left\_up\_corner, [30](#)
  - line\_end, [30](#)
  - line\_feeds, [30](#)
  - line\_start, [31](#)
  - mod, [31](#)
  - MPI\_grid\_to\_file, [26](#)
  - n, [31](#)
  - n\_line, [31](#)
  - new\_u, [31](#)
  - non\_isolated\_num, [31](#)
  - old\_u, [31](#)
  - output, [31](#)
  - parameters, [32](#)
  - receive\_from\_neighbour, [32](#)
  - right\_boundaries, [32](#)
  - right\_down\_corner, [32](#)
  - right\_edge\_length, [32](#)
  - right\_up\_corner, [32](#)
  - run\_simulation, [26](#)
  - send\_length, [32](#)
  - set\_initial\_condition, [27](#)
  - set\_Neumman\_boundary\_new\_u, [27](#)
  - set\_Neumman\_boundary\_old\_u, [27](#)
  - set\_Neumman\_boundary\_u, [27](#)
  - setting\_paramters, [27](#)
  - start\_index, [32](#)
  - t, [33](#)
  - t\_max, [33](#)
  - t\_out, [33](#)
  - u, [33](#)
  - up\_boundaries, [33](#)
  - up\_edge\_length, [33](#)
  - wave\_data, [25](#)
  - x\_index, [33](#)
  - x\_max, [33](#)
  - y\_index, [34](#)
  - y\_max, [34](#)
- width
  - animation, [15](#)
  - pic, [21](#)
- x\_index
  - wave\_data, [33](#)
- x\_label
  - animation, [15](#)
  - pic, [21](#)
- x\_max
  - wave\_data, [33](#)
- y\_index
  - wave\_data, [34](#)
- y\_label
  - animation, [16](#)
  - pic, [21](#)
- y\_max
  - wave\_data, [34](#)
- zero
  - wave.cpp, [47](#)