

# High Performance Computing Coursework

Generated by Doxygen 1.8.14



# Contents

<b>1</b>	<b>File Index</b>	<b>1</b>
1.1	File List . . . . .	1
<b>2</b>	<b>File Documentation</b>	<b>3</b>
2.1	init.cpp File Reference . . . . .	3
2.1.1	Function Documentation . . . . .	3
2.1.1.1	zeros() [1/2] . . . . .	3
2.1.1.2	zeros() [2/2] . . . . .	3
2.2	io.cpp File Reference . . . . .	4
2.2.1	Function Documentation . . . . .	4
2.2.1.1	def() . . . . .	4
2.2.1.2	errorMessage() . . . . .	4
2.2.1.3	green() . . . . .	5
2.2.1.4	printMatrix() [1/2] . . . . .	5
2.2.1.5	printMatrix() [2/2] . . . . .	5
2.2.1.6	red() . . . . .	5
2.2.1.7	WriteVtkFile() . . . . .	5
2.3	main.cpp File Reference . . . . .	6
2.3.1	Macro Definition Documentation . . . . .	6
2.3.1.1	F77NAME . . . . .	7
2.3.2	Function Documentation . . . . .	7
2.3.2.1	detMatrix2() . . . . .	7
2.3.2.2	dgesv() . . . . .	7
2.3.2.3	getInd_i() . . . . .	7
2.3.2.4	getInd_j() . . . . .	7
2.3.2.5	invMatrix2() . . . . .	8
2.3.2.6	main() . . . . .	8



# Chapter 1

## File Index

### 1.1 File List

Here is a list of all files with brief descriptions:

<a href="#">init.cpp</a>	.....	3
<a href="#">io.cpp</a>	.....	4
<a href="#">main.cpp</a>	.....	6



## Chapter 2

# File Documentation

### 2.1 init.cpp File Reference

```
#include "init.h"
```

#### Functions

- void **zeros** (double \*a, int length)  
*Sets each entry in an array of doubles to zero.*
- void **zeros** (int \*a, int length)  
*Sets each entry in an array of integers to zero.*

#### 2.1.1 Function Documentation

##### 2.1.1.1 zeros() [1/2]

```
void zeros (  
    double * a,  
    int length )
```

Sets each entry in an array of doubles to zero.

##### 2.1.1.2 zeros() [2/2]

```
void zeros (  
    int * a,  
    int length )
```

Sets each entry in an array of integers to zero.

## 2.2 io.cpp File Reference

```
#include "io.h"  
#include "colormod.h"  
#include <fstream>  
#include <iomanip>  
#include <iostream>
```

### Functions

- Color::Modifier [def](#) (Color::FG\_DEFAULT)  
*Call in a cout stream to change color to default for the console.*
- Color::Modifier [red](#) (Color::FG\_RED)  
*Call in a cout stream to change console output color to red.*
- Color::Modifier [green](#) (Color::FG\_GREEN)  
*Call in a cout stream to change console output color to green.*
- void [printMatrix](#) (double \*a, int M, int N)  
*Prints an [M x N] array of doubles to the console.*
- void [printMatrix](#) (int \*a, int M, int N)  
*Prints an [M x N] array of integers to the console.*
- void [errorMessage](#) (int e)  
*Throws an error message to the console and provides user with possible solutions.*
- void [WriteVtkFile](#) (int nnode\_elem, int nnode, int nelelem, double \*Coord, int \*ElemNode, double \*T, int casenum)  
*Creates a VTK file of the solutions.*

### 2.2.1 Function Documentation

#### 2.2.1.1 [def\(\)](#)

```
Color::Modifier def (  
    Color::FG_DEFAULT )
```

Call in a cout stream to change color to default for the console.

#### 2.2.1.2 [errorMessage\(\)](#)

```
void errorMessage (  
    int e )
```

Throws an error message to the console and provides user with possible solutions.



## Parameters

<b>e</b>	Error number generated by an active if-statement check in the code.
----------	---

Default error is general.

### 2.2.1.3 green()

```
Color::Modifier green (
    Color::FG_GREEN )
```

Call in a cout stream to change console output color to green.

### 2.2.1.4 printMatrix() [1/2]

```
void printMatrix (
    double * a,
    int M,
    int N )
```

Prints an [M x N] array of doubles to the console.

### 2.2.1.5 printMatrix() [2/2]

```
void printMatrix (
    int * a,
    int M,
    int N )
```

Prints an [M x N] array of integers to the console.

### 2.2.1.6 red()

```
Color::Modifier red (
    Color::FG_RED )
```

Call in a cout stream to change console output color to red.

### 2.2.1.7 WriteVtkFile()

```
void WriteVtkFile (
    int nnode_elem,
    int nnode,
    int nelem,
    double * Coord,
    int * ElemNode,
    double * T,
    int casenum )
```

Creates a VTK file of the solutions.

## Parameters

<i>nnode_elem</i>	Number of elements per node
<i>nnode</i>	Total number of nodes
<i>nelem</i>	Total number of elements
<i>Coord</i>	[nnode x 2] List of [x,y]-values for each node (stored in row-major format)
<i>ElemNode</i>	[nelem x (nnode_elem + 1)] List of elements and their associated nodes
<i>T</i>	[1 x nnode] Array of nodal temperatures
<i>casenum</i>	Case associated with the solution (if not case 1, 2 or 3, will set to user defined). Used to generate .vtk filename

## 2.3 main.cpp File Reference

```
#include <iostream>
#include <cmath>
#include <iomanip>
#include <fstream>
#include <limits>
#include <cstdlib>
#include <stdio>
#include <vector>
#include <numeric>
#include <functional>
#include <algorithm>
#include <ctime>
#include <valarray>
#include <mpi.h>
#include <cstring>
#include "cblas.h"
#include "io.h"
#include "init.h"
```

### Macros

- `#define F77NAME(x) x##_`

### Functions

- void `F77NAME()` `dgesv` (const int &n, const int &nrhs, const double \*A, const int &lda, int \*ipiv, double \*B, const int &ldb, int &info)
- double `detMatrix2` (double \*)
- void `invMatrix2` (double \*, double \*)
- int `getInd_i` (int, int)
- int `getInd_j` (int, int)
- int `main` (int argc, char \*argv[])

#### 2.3.1 Macro Definition Documentation

### 2.3.1.1 F77NAME

```
#define F77NAME(  
    x ) x##_
```

## 2.3.2 Function Documentation

### 2.3.2.1 detMatrix2()

```
double detMatrix2 (  
    double * A )
```

### 2.3.2.2 dgesv()

```
void F77NAME() dgesv (  
    const int & n,  
    const int & nrhs,  
    const double * A,  
    const int & lda,  
    int * ipiv,  
    double * B,  
    const int & ldb,  
    int & info )
```

### 2.3.2.3 getInd\_i()

```
int getInd_i (  
    int node,  
    int nnode_y )
```

### 2.3.2.4 getInd\_j()

```
int getInd_j (  
    int node,  
    int nnode_y )
```

### 2.3.2.5 invMatrix2()

```
void invMatrix2 (
    double * J,
    double * invJ )
```

### 2.3.2.6 main()

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