



HAL
open science

Parallel Computational Acoustics Library - Mesh Generation Reference Manual

Frédéric Magoulès, François-Xavier Roux

► **To cite this version:**

Frédéric Magoulès, François-Xavier Roux. Parallel Computational Acoustics Library - Mesh Generation Reference Manual. [Intern report] A02-R-073 || magoules02b, 2002, 12 p. inria-00099430

HAL Id: inria-00099430

<https://inria.hal.science/inria-00099430v1>

Submitted on 26 Sep 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Parallel Computational Acoustics Library

Mesh Generation Reference Manual*

by F. Magoulès and F.-X. Roux

June 13, 2002

Contents

1	Modules	3
1.1	mesh_struct	3
2	Functions	4
2.1	split_mesh	4
2.2	split_element	4
2.3	split_facet	5
2.4	build_glob_dirichlet	5
2.5	build_glob_robin	6
2.6	build_glob_neumann	6
2.7	build_glob_mesh	7
2.8	imove	7
2.9	rmove	8
2.10	echo_dirichlet	8
2.11	echo_neumann	9
2.12	echo_coor	9
2.13	echo_geom	10
2.14	echo_mesh	10
2.15	echo_splitting	11
2.16	inivec	11

* Generated by adoC, awk documenting C, June 13, 2002

Index

Alphabetical

- build_glob_dirichlet, 5
- build_glob_mesh, 7
- build_glob_neumann, 6
- build_glob_robin, 6
- echo_coor, 9
- echo_dirichlet, 8
- echo_geom, 10
- echo_mesh, 10
- echo_neumann, 9
- echo_splitting, 11
- imove, 7
- inivec, 11
- mesh_struct, 3
- rmove , 8
- split_element, 4
- split_facet, 5
- split_mesh, 4

- echo_geom, 10
- echo_mesh, 10
- echo_neumann, 9
- echo_splitting, 11
- imove, 7
- inivec, 11
- rmove , 8
- split_element, 4
- split_facet, 5
- split_mesh, 4

Modules

- mesh_struct, 3

Files

- mesh_struct.f90
 - mesh_struct, 3
- splitmesh_main.f90
 - split_mesh, 4
- splitmesh_routines.f90
 - build_glob_dirichlet, 5
 - build_glob_mesh, 7
 - build_glob_neumann, 6
 - build_glob_robin, 6
 - echo_coor, 9
 - echo_dirichlet, 8
 - echo_geom, 10
 - echo_mesh, 10
 - echo_neumann, 9
 - echo_splitting, 11
 - imove, 7
 - inivec, 11
 - rmove , 8
 - split_element, 4
 - split_facet, 5

Sorted

Functions

- build_glob_dirichlet, 5
- build_glob_mesh, 7
- build_glob_neumann, 6
- build_glob_robin, 6
- echo_coor, 9
- echo_dirichlet, 8

1 Modules

1.1 **mesh_struct**

NAME

mesh_struct

SYNOPSIS

```
MODULE mesh_struct
```

DESCRIPTION

Mesh structure for split_mesh program.

ARGUMENTS

- nodes_per_element** – integer (= number of nodes per element)
- numb_elements** – integer (= number of elements)
- geometry** – integer array (= geometry)
- space_dim** – integer (= number of coordinates per node)
- numb_nodes** – integer (= number of nodes)
- coordinates** – real array (coordinates)
- numb_clamp_nodes** – integer (= number of clamped nodes)
- clamp_nodes** – integer array (= clamped nodes)
- nodes_per_facet** – integer (= number of nodes per facet)
- numb_facets** – integer (= number of facets)
- facets** – integer array (= facets)
- facet2elem** – integer array (= facet to element correspondance)
- numb_clamp_facets** – integer
- clamp_facets** – integer array (= clamped facets)

2 Functions

2.1 `split_mesh`

NAME

split_mesh

SYNOPSIS

```
PROGRAM split_mesh
```

DESCRIPTION

Build subdomains from a global domain divided in grid and mesh with regular hexahedron with all the same material properties: $\omega = 2\pi F/c$, with $c = 1$. The variables ω , F and c denotes respectively the wavenumber, the frequency, and the celerity.

ARGUMENTS

None

MODULES

```
USE mesh_struct
```

2.2 `split_element`

NAME

split_element

SYNOPSIS

```
SUBROUTINE split_element (ni,nj,nk,nsdi,nsdj,nsdk,           &
&                          subdom_num, num_elem, elem2subdom)
```

DESCRIPTION

Decompose regular hexaedric mesh of domain into subdomains and build elements to subdomain correspondance.

ARGUMENTS

ni – integer
nj – integer
nk – integer
nsdi – integer
nsdj – integer
nsdk – integer
subdom_num – integer
num_elem – integer
elem2subdom – integer array

2.3 split_facet

NAME

split_facet

SYNOPSIS

SUBROUTINE `split_facet` (`dom,elem2dom,facet2dom`)

DESCRIPTION

Detect facets on rear face of domain.

ARGUMENTS

dom – mesh structure
elem2dom – integer array
facet2dom – integer array

MODULES

USE `mesh_struct`**2.4 build_glob_dirichlet**

NAME

build_glob_dirichlet

SYNOPSIS

SUBROUTINE `build_glob_dirichlet` (`front,back,right,left,top,botton, &`
& `dom`)

DESCRIPTION

Detect clamped nodes on face of domain with the convention front ($x > 0$), back ($x < 0$), right ($y > 0$), left ($y < 0$), top ($z > 0$), botton ($z < 0$).

ARGUMENTS

front – integer
back – integer
right – integer
left – integer
top – integer
botton – integer
dom – mesh structure

MODULES

USE `mesh_struct`

2.5 build_glob_robin

NAME

build_glob_robin

SYNOPSIS

```

SUBROUTINE build_glob_robin (front,back,right,left,top,botton,    &
&                               dom)

```

DESCRIPTION

Detect robin facet on face of domain with the convention front ($x > 0$), back ($x < 0$), right ($y > 0$), left ($y < 0$), top ($z > 0$), botton ($z < 0$).

ARGUMENTS

front – integer
back – integer
right – integer
left – integer
top – integer
botton – integer
dom – mesh structure

MODULES

```

USE mesh_struct

```

2.6 build_glob_neumann

NAME

build_glob_neumann

SYNOPSIS

```

SUBROUTINE build_glob_neumann (front,back,right,left,top,botton,  &
&                               dom)

```

DESCRIPTION

Detect clamped facet on face of block with the convention front ($x > 0$), back ($x < 0$), right ($y > 0$), left ($y < 0$), top ($z > 0$), botton ($z < 0$).

ARGUMENTS

front – integer
back – integer
right – integer
left – integer
top – integer
botton – integer
dom – mesh structure

MODULES

USE mesh_struct

2.7 **build_glob_mesh**

NAME

build_glob_mesh

SYNOPSIS

SUBROUTINE build_glob_mesh (ni,nj,nk,dx,dy,dz,dom)

DESCRIPTION

Compute geometry and coordinates of regular hexaedric mesh.

ARGUMENTS

ni – integer
nj – integer
nk – integer
dx – real
dy – real
dz – real
dom – mesh structure

MODULES

USE mesh_struct

2.8 **imove**

NAME

imove

SYNOPSIS

SUBROUTINE imove (dim,x,y)

DESCRIPTION

Move integer array.

ARGUMENTS

dim – integer
x – integer array
y – integer array

2.9 **rmove**

NAME

rmove

SYNOPSIS

SUBROUTINE `rmove` (`dim,x,y`)

DESCRIPTION

Move real array.

ARGUMENTS

dim – integer
x – real array
y – real array

2.10 **echo_dirichlet**

NAME

echo_dirichlet

SYNOPSIS

SUBROUTINE `echo_dirichlet` (`file_num,dom,frequency,theta,phi`)

DESCRIPTION

Write list of clamped nodes with Dirichlet boundary conditions.

ARGUMENTS

file_num – integer
dom – mesh structure
frequency – real
theta – real
phi – real

MODULES

USE mesh_struct

2.11 **echo_neumann**

NAME

echo_neumann

SYNOPSIS

```

SUBROUTINE echo_neumann (file_num,dom,frequency,theta,phi,      &
&                        frontR,backR,rightR,leftR,topR,bottonR, &
&                        frontN,backN,rightN,leftN,topN,bottonN)

```

DESCRIPTION

Write rhs associated to Robin and/or Neumann boundary conditions.

ARGUMENTS

```

file_num – integer
dom – mesh structure
frequency – real
theta – real
phi – real
frontR – integer
backR – integer
rightR – integer
leftR – integer
topR – integer
bottonR – integer
frontN – integer
backN – integer
rightN – integer
leftN – integer
topN – integer
bottonN – integer

```

MODULES

USE mesh_struct

2.12 **echo_coor**

NAME

echo_coor

SYNOPSIS

```

SUBROUTINE echo_coor (file_num,space_dim,numb_nodes,coordinates)

```

DESCRIPTION

Write coordinates of nodes.

ARGUMENTS

file_num – integer
space_dim – integer
numb_nodes – integer
coordinates – real array

2.13 **echo_geom**

NAME

echo_geom

SYNOPSIS

```
SUBROUTINE echo_geom (file_num,nodes_per_element,numb_elements,    &
&                      geometry,region,type_elem)
```

DESCRIPTION

Write topology.

ARGUMENTS

file_num – integer
nodes_per_element – integer
numb_elements – integer
geometry – integer array
region – integer
type_elem – integer

2.14 **echo_mesh**

NAME

echo_mesh

SYNOPSIS

```
SUBROUTINE echo_mesh (file_num,dom)
```

DESCRIPTION

Write mesh.

ARGUMENTS

file_num – integer
dom – mesh structure

MODULES

USE mesh_struct

2.15 **echo_splitting**

NAME

echo_splitting

SYNOPSIS

```

SUBROUTINE echo_splitting (file_num,numb_elem,elem2subdom,      &
&                          numb_facet,facet2subdom)

```

DESCRIPTION

Write element to subdomain correspondance including facet element.

ARGUMENTS

file_num – integer
numb_elem – integer
elem2subdom – integer array
numb_facet – integer
facet2subdom – integer array

2.16 **inivec**

NAME

inivec

SYNOPSIS

```

SUBROUTINE inivec (lx,x)

```

DESCRIPTION

Pseudo random number generator.

ARGUMENTS

lx – integer
x – complex array

Acknowledgements

The authors acknowledge partial financial support by the European Community under the Enhancing Access to Research Infrastructure action of the Improving Human Potential programme, contract number HPRI-1999-CT-0026.