

libsimpa

1.1.4

Généré par Doxygen 1.6.1

Fri Jan 29 11 :31 :36 2010

Table des matières

1	Index des modules	1
1.1	Modules	1
2	Index des espaces de nommage	3
2.1	Liste des espaces de nommage	3
3	Index des classes	5
3.1	Liste des classes	5
4	Documentation des modules	7
4.1	Ecriture d'un fichier de particule	7
4.1.1	Description détaillée	7
4.1.2	Documentation des fonctions	7
4.1.2.1	NewParticle	7
4.1.2.2	NewPositionParticle	7
4.1.2.3	SetHeaderData	7
4.2	Lecture d'un fichier de particule	9
4.2.1	Description détaillée	9
4.2.2	Documentation des fonctions	9
4.2.2.1	GetHeaderData	9
4.2.2.2	NextParticle	9
4.2.2.3	NextTimeStep	9
4.2.2.4	OpenForRead	10
5	Documentation des espaces de nommage	11
5.1	Référence de l'espace de nommage <code>CalculsGenerauxThermodynamique</code>	11
5.1.1	Description détaillée	11
5.2	Référence de l'espace de nommage <code>formatCoreBIN</code>	12
5.2.1	Description détaillée	12
5.3	Référence de l'espace de nommage <code>formatMBIN</code>	13

5.3.1	Description détaillée	13
5.4	Référence de l'espace de nommage formatRSBIN	14
5.4.1	Description détaillée	14
5.5	Référence de l'espace de nommage gabe_io	15
5.5.1	Description détaillée	15
5.6	Référence de l'espace de nommage particleio	16
5.6.1	Description détaillée	16
5.7	Référence de l'espace de nommage surf_merge	17
5.7.1	Description détaillée	17
6	Documentation des classes	19
6.1	Référence de la structure formatMBIN : :bintetraface	19
6.1.1	Description détaillée	19
6.1.2	Documentation des données membres	19
6.1.2.1	marker	19
6.1.2.2	neighbor	19
6.2	Référence de la structure formatMBIN : :bintetrahedre	20
6.2.1	Description détaillée	20
6.3	Référence de la classe CalculsGenerauxThermodynamique : :CCalculsGenerauxThermodynamique	21
6.4	Référence de la classe formatCoreBIN : :CformatBIN	22
6.4.1	Description détaillée	22
6.4.2	Documentation des constructeurs et destructeur	22
6.4.2.1	CformatBIN	22
6.4.2.2	~CformatBIN	22
6.4.3	Documentation des fonctions membres	22
6.4.3.1	ExportBIN	22
6.4.3.2	GetSurfAire	22
6.4.3.3	ImportBIN	23
6.5	Référence de la classe formatMBIN : :CMBIN	24
6.5.1	Description détaillée	24
6.5.2	Documentation des constructeurs et destructeur	24
6.5.2.1	CMBIN	24
6.5.2.2	~CMBIN	24
6.5.3	Documentation des fonctions membres	24
6.5.3.1	LoadMesh	24
6.5.3.2	SaveMesh	24

6.6	Référence de la classe <code>gabe_io : :Gabe_rw</code>	25
6.6.1	Description détaillée	25
6.6.2	Documentation des constructeurs et destructeur	25
6.6.2.1	<code>Gabe_rw</code>	25
6.6.3	Documentation des fonctions membres	25
6.6.3.1	<code>AppendFloatCol</code>	25
6.6.3.2	<code>AppendIntCol</code>	25
6.6.3.3	<code>AppendStrCol</code>	25
6.6.3.4	<code>GetColTitle</code>	26
6.6.3.5	<code>GetTabTypes</code>	26
6.6.3.6	<code>IsReadOnly</code>	26
6.6.3.7	<code>Load</code>	26
6.6.3.8	<code>operator[]</code>	26
6.6.3.9	<code>Save</code>	26
6.6.3.10	<code>SetReadOnly</code>	26
6.6.3.11	<code>size</code>	26
6.7	Référence de la structure <code>formatCoreBIN : :ioFace</code>	27
6.7.1	Description détaillée	27
6.7.2	Documentation des données membres	27
6.7.2.1	<code>a</code>	27
6.7.2.2	<code>b</code>	27
6.7.2.3	<code>c</code>	27
6.7.2.4	<code>idEn</code>	27
6.7.2.5	<code>idMat</code>	27
6.7.2.6	<code>idRs</code>	27
6.8	Référence de la structure <code>formatCoreBIN : :ioModel</code>	28
6.8.1	Description détaillée	28
6.8.2	Documentation des données membres	28
6.8.2.1	<code>faces</code>	28
6.8.2.2	<code>vertices</code>	28
6.9	Référence de la classe <code>particleio : :ParticuleIO</code>	29
6.9.1	Description détaillée	29
6.9.2	Documentation des fonctions membres	29
6.9.2.1	<code>Close</code>	29
6.10	Référence de la classe <code>progressionInfo</code>	30
6.10.1	Description détaillée	30

6.10.2	Documentation des constructeurs et destructeur	30
6.10.2.1	progressionInfo	30
6.10.3	Documentation des fonctions membres	30
6.10.3.1	GetMainOperation	30
6.10.3.2	OutputCurrentProgression	31
6.11	Référence de la classe progressOperation	32
6.11.1	Description détaillée	32
6.11.2	Documentation des constructeurs et destructeur	32
6.11.2.1	progressOperation	32
6.11.3	Documentation des fonctions membres	32
6.11.3.1	PushProgression	32
6.12	Référence de la classe formatRSBIN : :rsurf_data	33
6.12.1	Description détaillée	33
6.12.2	Documentation des fonctions membres	33
6.12.2.1	GetFaceEnergy	33
6.12.2.2	GetFaceInfo	34
6.12.2.3	GetFaceSumEnergy	34
6.12.2.4	GetFileInfos	34
6.12.2.5	GetNodePositionValue	35
6.12.2.6	GetRsCount	35
6.12.2.7	GetRsInfo	35
6.12.2.8	Make	35
6.12.2.9	MakeRs	36
6.12.2.10	SetFaceEnergy	36
6.12.2.11	SetFaceInfo	36
6.12.2.12	SetNodeValue	36
6.12.2.13	SetRsBinData	37
6.13	Référence de la classe formatRSBIN : :rsurf_io	38
6.14	Référence de la classe formatRSBIN : :rsurf_iso_contouring	39
6.14.1	Documentation des constructeurs et destructeur	39
6.14.1.1	rsurf_iso_contouring	39
6.14.2	Documentation des fonctions membres	39
6.14.2.1	GetIsoLevelCurves	39
6.14.2.2	GetMinMax	39
6.14.2.3	SmoothGrid	39
6.15	Référence de la classe surf_merge : :SurfaceMerging	40

6.15.1	Description détaillée	40
6.15.2	Documentation des fonctions membres	40
6.15.2.1	GetGroupInformation	40
6.15.2.2	GetGroups	40
6.15.2.3	LoadGroups	40
6.15.2.4	SavePolyToPolyFile	40
6.16	Référence de la structure formatMBIN : :t_binNode	41
6.17	Référence de la structure formatRSBIN : :t_curve	42
6.18	Référence de la classe formatCoreBIN : :t_pos	43
6.19	Référence de la structure formatMBIN : :trimeshmodel	44
6.20	Référence de la classe volumes_splitter : :VolumesSplitter	45
6.20.1	Description détaillée	45
6.20.2	Documentation des fonctions membres	45
6.20.2.1	GetInternalFaces	45
6.20.2.2	GetTetraByPosition	45
6.20.2.3	GetVolumeInformation	46
6.20.2.4	GetVolumes	46
6.20.2.5	GetVolumeXmlId	46
6.20.2.6	LoadDomain	46

Chapitre 1

Index des modules

1.1 Modules

Liste de tous les modules :

Ecriture d'un fichier de particule	7
Lecture d'un fichier de particule	9

Chapitre 2

Index des espaces de nommage

2.1 Liste des espaces de nommage

Liste de tous les espaces de nommage documentés avec une brève description :

CalculsGenerauxThermodynamique (Calculs de thermodynamique)	11
formatCoreBIN (Implémentation de l'interpréteur de fichier modèle Bin (*.bin))	12
formatMBIN (Import export du maillage tétraédrique)	13
formatRSBIN (Import export de l'animation des récepteurs surfaciques)	14
gabe_io (Fichier d'échange générique)	15
particleio (Lecture, écriture des fichiers de particules)	16
surf_merge	17

Chapitre 3

Index des classes

3.1 Liste des classes

Liste des classes, structures, unions et interfaces avec une brève description :

formatMBIN : :bintetraface (Structure d'une face d'un tetrahedre)	19
formatMBIN : :bintetrahedre (Structure d'un tétrahedre du maillage)	20
CalculsGenerauxThermodynamique : :CCalculsGenerauxThermodynamique	21
formatCoreBIN : :CformatBIN (Classe de sauvegarde et de chargement de fichier de modèle entre l'interface et le mode de calcul SPPS)	22
formatMBIN : :CMBIN (Classe de sauvegarde et de chargement du maillage tétrahedrique)	24
gabe_io : :Gabe_rw	25
formatCoreBIN : :ioFace (Structure d'une face, destiné à l'echange fichier<->CObjet3D)	27
formatCoreBIN : :ioModel (Structure du modèle, destiné à l'echange fichier<->CObjet3D)	28
particleio : :ParticuleIO	29
progressionInfo (Gestionnaire d'affichage de progression du calcul compatible avec le multi-threading)	30
progressOperation	32
formatRSBIN : :rsurf_data (Contient les données d'un fichier de résultat de récepteur de surfaces)	33
formatRSBIN : :rsurf_io	38
formatRSBIN : :rsurf_iso_contouring	39
surf_merge : :SurfaceMerging	40
formatMBIN : :t_binNode	41
formatRSBIN : :t_curve	42
formatCoreBIN : :t_pos	43
formatMBIN : :trimeshmodel	44
volumes_splitter : :VolumesSplitter	45

Chapitre 4

Documentation des modules

4.1 Ecriture d'un fichier de particule

Fonctions

- void `particleio : :ParticuleIO : :NewParticle` (const unsigned long &firsttimestep)
- void `particleio : :ParticuleIO : :NewPositionParticle` (const float &x, const float &y, const float &z, const float &energy)
- bool `particleio : :ParticuleIO : :OpenForWrite` (const std : :string &fileName)
- void `particleio : :ParticuleIO : :SetHeaderData` (const float &timeStep)

4.1.1 Description détaillée

Ces méthodes permettent de créer un nouveau fichier de particule

4.1.2 Documentation des fonctions

4.1.2.1 void `particleio : :ParticuleIO : :NewParticle` (const unsigned long & *firsttimestep*) [*inherited*]

Enregistre une nouvelle particule

4.1.2.2 void `particleio : :ParticuleIO : :NewPositionParticle` (const float & *x*, const float & *y*, const float & *z*, const float & *energy*) [*inherited*]

Enregistre une nouvelle position pour la particule courante

Paramètres:

- x* Position
- y* Position
- z* Position
- energy* Energie w

4.1.2.3 void `particleio : :ParticuleIO : :SetHeaderData` (const float & *timeStep*) [*inherited*]

Met à jour les données de l'entete du fichier

Paramètres:

timeStep Pas de temps en seconde

4.2 Lecture d'un fichier de particule

Fonctions

- void `particleio : :ParticuleIO : :GetHeaderData` (float &timeStep, unsigned long &nbParticles, unsigned long &nbStepMax)
- void `particleio : :ParticuleIO : :NextParticle` (unsigned long &firstTimeStep, unsigned long &nbTimeStep)
- void `particleio : :ParticuleIO : :NextTimeStep` (float &x, float &y, float &z, float &energy)
- bool `particleio : :ParticuleIO : :OpenForRead` (const std : :string &fileName)

4.2.1 Description détaillée

Ces méthodes permettent de lire un fichier de particule

4.2.2 Documentation des fonctions

4.2.2.1 void `particleio : :ParticuleIO : :GetHeaderData` (float & *timeStep*, unsigned long & *nbParticles*, unsigned long & *nbStepMax*) [*inherited*]

Permet d'accéder aux données d'entête du fichier de particule

Paramètres:

- *timeStep* Pas de temps en seconde
- *nbParticles* Nombre de particules
- *nbTimeStep* Nombre de particules

Python note :

Return a tuple containing (timeStep,nbParticles,nbStepMax)

4.2.2.2 void `particleio : :ParticuleIO : :NextParticle` (unsigned long & *firstTimeStep*, unsigned long & *nbTimeStep*) [*inherited*]

Passe à la prochaine particule

Paramètres:

- *firstTimeStep* Premier pas de temps
- *nbTimeStep* Nombre de pas de temps

Python note :

Return a tuple containing (firstTimeStep,nbTimeStep)

4.2.2.3 void `particleio : :ParticuleIO : :NextTimeStep` (float & *x*, float & *y*, float & *z*, float & *energy*) [*inherited*]

Lis une nouvelle position pour la particule courante

Paramètres:

- *x* Position

- **y** Position
- **z** Position
- **energy** Energie w

Python note :

Return a tuple containing (x,y,z,energy)

**4.2.2.4 bool particleio : :ParticuleIO : :OpenForRead (const std : :string & *fileName*)
[inherited]**

Lecture d'un fichier de particule

Paramètres:

fileName Chemin d'accès au fichier de particule

Chapitre 5

Documentation des espaces de nommage

5.1 Référence de l'espace de nommage `CalculsGenerauxThermodynamique`

Calculs de thermodynamique.

Classes

– class [CCalculsGenerauxThermodynamique](#)

5.1.1 Description détaillée

Calculs de thermodynamique.

5.2 Référence de l'espace de nommage formatCoreBIN

Implémentation de l'interpréteur de fichier modèle Bin (*.bin).

Classes

- class [CformatBIN](#)
Classe de sauvegarde et de chargement de fichier de modèle entre l'interface et le mode de calcul SPPS.
- struct [ioFace](#)
Structure d'une face, destiné à l'échange fichier<->CObjet3D.
- struct [ioModel](#)
Structure du modèle, destiné à l'échange fichier<->CObjet3D.
- class [t_pos](#)

5.2.1 Description détaillée

Implémentation de l'interpréteur de fichier modèle Bin (*.bin).

5.3 Référence de l'espace de nommage formatMBIN

Import export du maillage tétraédrique.

Classes

- struct `bintetraface`
Structure d'une face d'un tetraedre.
- struct `bintetrahedre`
Structure d'un tetraedre du maillage.
- class `CMBIN`
Classe de sauvegarde et de chargement du maillage tetraedrique.
- struct `t_binNode`
- struct `trimeshmodel`

Définition de type

- typedef unsigned short `bCourt`
- typedef bool `Boolb`
- typedef float `Floatb`
- typedef int `Intb`
- typedef unsigned long `Longb`

5.3.1 Description détaillée

Import export du maillage tétraédrique.

5.4 Référence de l'espace de nommage formatRSBIN

Import export de l'animation des recepteurs surfaciques.

Classes

- class [rsurf_data](#)
Contient les données d'un fichier de résultat de récepteur de surfaces.
- class [rsurf_io](#)
- class [rsurf_iso_contouring](#)
- struct [t_curve](#)

Définition de type

- typedef std::size_t [t_index](#) [3]
- typedef float [t_pos](#) [3]

5.4.1 Description détaillée

Import export de l'animation des recepteurs surfaciques.

5.5 Référence de l'espace de nommage gabe_io

Fichier d'échange générique.

Classes

- class [Gabe_rw](#)

Définition de type

- typedef formatGABE : :GABE_OBJECTTYPE COLTYPE
- typedef vector< float > **floatarray**
- typedef vector< int > **intarray**
- typedef vector< string > **stringarray**
- typedef vector< COLTYPE > **typearray**

Fonctions

- static void **impl_vectors** ()

5.5.1 Description détaillée

Fichier d'échange générique.

5.6 Référence de l'espace de nommage particuleio

Lecture, écriture des fichiers de particules.

Classes

– class [ParticuleIO](#)

5.6.1 Description détaillée

Lecture, écriture des fichiers de particules.

5.7 Référence de l'espace de nommage surf_merge

Classes

- class [SurfaceMerging](#)

5.7.1 Description détaillée

Permet de regrouper les faces selon la même norme et le même matériau

Chapitre 6

Documentation des classes

6.1 Référence de la structure `formatMBIN : :bintetraface`

Structure d'une face d'un tetrahedre.

Attributs publics

- Intb [marker](#)
- Intb [neighbor](#)
- Intb `sommets` [3]

6.1.1 Description détaillée

Structure d'une face d'un tetrahedre.

6.1.2 Documentation des données membres

6.1.2.1 Intb `formatMBIN : :bintetraface : :marker`

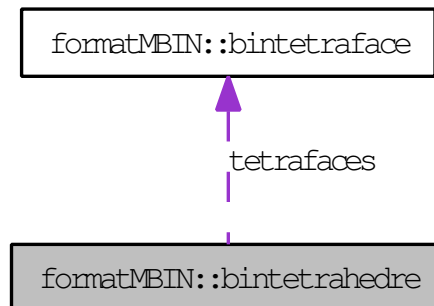
Face du modèle -1 aucune

6.1.2.2 Intb `formatMBIN : :bintetraface : :neighbor`

Indice de tetraHedre 0 à n. -2 pour aucun voisin sur cette face

6.2 Référence de la structure formatMBIN : :bintetrahedre

Structure d'un tétrahedre du maillage. Graphe de collaboration de formatMBIN : :bintetrahedre :



Fonctions membres publiques

- int **operator==** (const [bintetrahedre](#) &_f)

Attributs publics

- Intb **idVolume**
- Intb **sommets** [4]
- [bintetraface](#) **tetrafaces** [4]

6.2.1 Description détaillée

Structure d'un tétrahedre du maillage.

6.3 Référence de la classe CalculsGenerauxThermodynamique : :CCalculsGenerauxThermodynamique

Fonctions membres publiques statiques

- static double **c_son** (double x)
- static double **Coef_Att_Atmos** (double F, double H, double P, double K)
- static double **masse_vol** (double P, double K)

6.4 Référence de la classe formatCoreBIN : :CformatBIN

Classe de sauvegarde et de chargement de fichier de modèle entre l'interface et le mode de calcul SPPS.

Fonctions membres publiques

- CformatBIN ()
- bool ExportBIN (const char *strFileName, ioModel &modelExport)
- float GetSurfAire (const ioModel &model, const int &indiceface)
- bool ImportBIN (ioModel &modelImport, const char *strFileName)
- ~CformatBIN ()

6.4.1 Description détaillée

Classe de sauvegarde et de chargement de fichier de modèle entre l'interface et le mode de calcul SPPS.

Voir également:

- COBJET3D : :Save()
- COBJET3D : :Load()

6.4.2 Documentation des constructeurs et destructeur

6.4.2.1 formatCoreBIN : :CformatBIN : :CformatBIN ()

Constructeur

6.4.2.2 formatCoreBIN : :CformatBIN : :~CformatBIN ()

Destructeur

6.4.3 Documentation des fonctions membres

6.4.3.1 bool formatCoreBIN : :CformatBIN : :ExportBIN (const char * strFileName, ioModel & modelExport)

Méthode d'exportation d'un modèle 3D

Paramètres:

- strFileName* Chemin du fichier
- modelExport* Modèle à exporter, déjà converti par coordsOperation : :GlCoordsToCommonCoords()
- .
- matFacesAssociations* Lien entre les faces et les matériaux.

Renvoie:

- Vrai si l'opération a réussi

6.4.3.2 float formatCoreBIN : :CformatBIN : :GetSurfAire (const ioModel & model, const int & indiceface)

Retourne l'aire de la surface

6.4.3.3 `bool formatCoreBIN : :CformatBIN : :ImportBIN (ioModel & modellImport, const char * strFileName)`

Méthode d'importation d'un fichier binaire

Paramètres:

modellImport Structure de données où stocker le modèle à charger

strFileName Chemin du fichier

Renvoie:

Vrai si l'opération a réussi

6.5 Référence de la classe formatMBIN : :CMBIN

Classe de sauvegarde et de chargement du maillage tétraédrique.

Fonctions membres publiques

- [CMBIN \(\)](#)
- [trimeshmodel LoadMesh](#) (const char *strFileName)
- void [SaveMesh](#) (const char *strFileName, [trimeshmodel](#) &trimesh)
- [~CMBIN \(\)](#)

Fonctions membres publiques statiques

- static double [ComputeVolume](#) ([trimeshmodel](#) &trimesh)

6.5.1 Description détaillée

Classe de sauvegarde et de chargement du maillage tétraédrique.

6.5.2 Documentation des constructeurs et destructeur

6.5.2.1 formatMBIN : :CMBIN : :CMBIN () [inline]

Constructeur

6.5.2.2 formatMBIN : :CMBIN : :~CMBIN () [inline]

Destructeur

6.5.3 Documentation des fonctions membres

6.5.3.1 trimeshmodel formatMBIN : :CMBIN : :LoadMesh (const char * *strFileName*)

Méthode d'importation d'un fichier binaire

Paramètres:

strFileName Chemin du fichier

Renvoie:

Le maillage

6.5.3.2 void formatMBIN : :CMBIN : :SaveMesh (const char * *strFileName*, [trimeshmodel](#) & *trimesh*)

Méthode d'exportation d'un fichier binaire

Paramètres:

strFileName Chemin du fichier

trimesh Le maillage

6.6 Référence de la classe `gabe_io : :Gabe_rw`

Fonctions membres publiques

- void `AppendFloatCol` (const floatarray &coldata, const string &coltitle, const int &precision=12)
- void `AppendIntCol` (const intarray &coldata, const string &coltitle)
- void `AppendStrCol` (const stringarray &coldata, const string &coltitle)
- `Gabe_rw` (const size_t &sizeHint)
- string `GetColTitle` (const size_t &idcol)
- typearray `GetTabTypes` ()
- bool `IsReadOnly` ()
- bool `Load` (const string &fileName)
- stringarray `operator[]` (size_t indicecol)
- void `ReadColFloat` (const size_t &idcol, floatarray &coldata)
- void `ReadColInt` (const size_t &idcol, intarray &coldata)
- void `ReadColStr` (const size_t &idcol, stringarray &coldata)
- bool `Save` (const string &fileName)
- void `SetReadOnly` (const bool &ReadOnly=true)
- size_t `size` () const

6.6.1 Description détaillée

Gestion d'écriture et de lecture du format GABE lu par l'interface La première colonne correspond aux libellées des lignes. Chaque colonne peut être de 3 types différents, chaîne de caractère, nombre à décimale ou nombre entier.

6.6.2 Documentation des constructeurs et destructeur

6.6.2.1 `gabe_io : :Gabe_rw : :Gabe_rw (const size_t &sizeHint)`

Paramètres:

sizeHint A la génération d'un tableau ce paramètre permet une plus grande rapidité d'ajout des colonnes

6.6.3 Documentation des fonctions membres

6.6.3.1 void `gabe_io : :Gabe_rw : :AppendFloatCol (const floatarray & coldata, const string & coltitle, const int & precision = 12)`

Ajoute une colonne de nombres flottants à la fin du tableau

6.6.3.2 void `gabe_io : :Gabe_rw : :AppendIntCol (const intarray & coldata, const string & coltitle)`

Ajoute une colonne d'entier à la fin du tableau

6.6.3.3 void `gabe_io : :Gabe_rw : :AppendStrCol (const stringarray & coldata, const string & coltitle)`

Ajoute une colonne de chaîne de caractères à la fin du tableau

6.6.3.4 `string gabe_io : :Gabe_rw : :GetColTitle (const size_t & idcol)`

Lecture du titre d'une colonne

6.6.3.5 `typeid gabe_io : :Gabe_rw : :GetTabTypes ()`

Retourne les types de données sur les colonnes

6.6.3.6 `bool gabe_io : :Gabe_rw : :IsReadOnly ()`

Les données sont elles en lecture seule pour l'utilisateur.

6.6.3.7 `bool gabe_io : :Gabe_rw : :Load (const string & fileName)`

Chargement du tableau

Paramètres:

strFileName Chemin du fichier

6.6.3.8 `stringarray gabe_io : :Gabe_rw : :operator[] (size_t indicecol)`

Lire une colonne en tant que chaîne de caractères (compatible avec n'importe quel type de colonne)

6.6.3.9 `bool gabe_io : :Gabe_rw : :Save (const string & fileName)`

Sauvegarde du tableau

Paramètres:

strFileName Chemin du fichier

6.6.3.10 `void gabe_io : :Gabe_rw : :SetReadOnly (const bool & ReadOnly = true)`

Change le mode d'édition par l'utilisateur dans l'interface

6.6.3.11 `size_t gabe_io : :Gabe_rw : :size () const`**Renvoie:**

Nombre de colonnes

Python note :

Use the len function to give the number of cols.

6.7 Référence de la structure formatCoreBIN : :ioFace

Structure d'une face, destiné à l'échange fichier<->CObjet3D.

Fonctions membres publiques

- int operator==(const ioFace &_f)

Attributs publics

- indiceV a
- indiceV b
- indiceV c
- indiceEN idEn
- indiceMat idMat
- indiceRS idRs

6.7.1 Description détaillée

Structure d'une face, destiné à l'échange fichier<->CObjet3D.

6.7.2 Documentation des données membres

6.7.2.1 indiceV formatCoreBIN : :ioFace : :a

Indice du sommet

6.7.2.2 indiceV formatCoreBIN : :ioFace : :b

Indice du sommet

6.7.2.3 indiceV formatCoreBIN : :ioFace : :c

Indice du sommet

6.7.2.4 indiceEN formatCoreBIN : :ioFace : :idEn

Identifiant de l'encombrement (aucun -1)

6.7.2.5 indiceMat formatCoreBIN : :ioFace : :idMat

Identifiant du matériau associé à la face (aucun 0)

6.7.2.6 indiceRS formatCoreBIN : :ioFace : :idRs

Identifiant du récepteur surfacique (aucun -1)

6.8 Référence de la structure formatCoreBIN : :ioModel

Structure du modèle, destiné à l'échange fichier<->CObjet3D.

Attributs publics

- std : :vector< ioFace > faces
- std : :vector< t_pos > vertices

6.8.1 Description détaillée

Structure du modèle, destiné à l'échange fichier<->CObjet3D.

6.8.2 Documentation des données membres

6.8.2.1 std : :vector<ioFace> formatCoreBIN : :ioModel : :faces

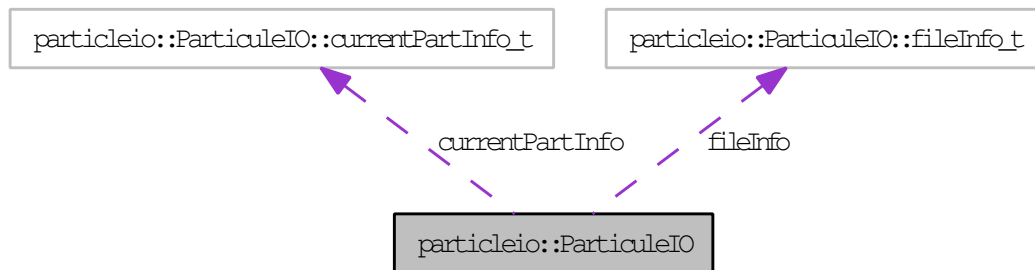
Faces de la scène

6.8.2.2 std : :vector<t_pos> formatCoreBIN : :ioModel : :vertices

Sommets des triangles

6.9 Référence de la classe `particleio : :ParticuleIO`

Grphe de collaboration de `particleio : :ParticuleIO` :



Classes

- struct `currentPartInfo_t`
- struct `fileInfo_t`

Fonctions membres publiques

- void `Close ()`
- void `GetHeaderData` (float &timeStep, unsigned long &nbParticles, unsigned long &nbStepMax)
- void `NewParticle` (const unsigned long &firsttimestep)
- void `NewPositionParticle` (const float &x, const float &y, const float &z, const float &energy)
- void `NextParticle` (unsigned long &firstTimeStep, unsigned long &nbTimeStep)
- void `NextTimeStep` (float &x, float &y, float &z, float &energy)
- bool `OpenForRead` (const std : :string &fileName)
- bool `OpenForWrite` (const std : :string &fileName)
- void `SetHeaderData` (const float &timeStep)

6.9.1 Description détaillée

Cette classe permet la gestion du fichier d'enregistrement de la propagation des particules. Les données sont conservées uniquement sur le disque durant toute la procédure d'écriture et de lecture car ce type de fichier pouvant être très volumineux.

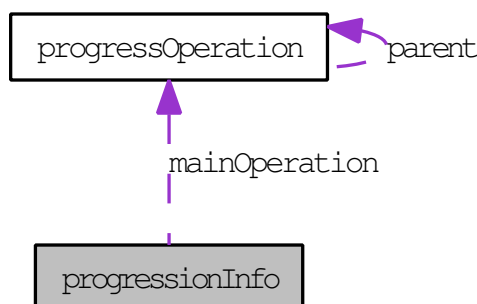
6.9.2 Documentation des fonctions membres

6.9.2.1 void `particleio : :ParticuleIO : :Close ()`

Fermeture du fichier. Cette méthode est automatiquement appelée lors de la destruction de l'instance de `ParticuleIO`.

6.10 Référence de la classe progressionInfo

Gestionnaire d'affichage de progression du calcul compatible avec le multithreading. Graphe de collaboration de progressionInfo :



Fonctions membres publiques

- float **GetCurrentProgression** ()
- [progressOperation](#) * **GetMainOperation** ()
- void **OutputCurrentProgression** ()
- [progressionInfo](#) (int firstLevelEstimation)

Fonctions membres protégées

- virtual void **DoShowProgression** (float newProg)

6.10.1 Description détaillée

Gestionnaire d'affichage de progression du calcul compatible avec le multithreading.

6.10.2 Documentation des constructeurs et destructeur

6.10.2.1 [progressionInfo](#) : :[progressionInfo](#) (int *firstLevelEstimation*) [[inline](#)]

Constructeur

Paramètres:

firstLevelEstimation Nombre de tache au premier niveau hiérarchique de taches

6.10.3 Documentation des fonctions membres

6.10.3.1 [progressOperation](#)* [progressionInfo](#) : :**GetMainOperation** () [[inline](#)]

Renvoie:

L'operation principale.

6.10.3.2 `void progressionInfo : :OutputCurrentProgression () [inline]`

Affiche la progression si celle-ci a changé

6.11 Référence de la classe progressOperation

Grphe de collaboration de progressOperation :



Fonctions membres publiques

- void **GetState** (float *thisLvlProg)
- [progressOperation](#) ([progressOperation](#) *parentOperation, unsigned int sonEstimation=1)

Fonctions membres protégées

- void [PushProgression](#) (float incProg)

6.11.1 Description détaillée

Représente une opération de calcul

6.11.2 Documentation des constructeurs et destructeur

- 6.11.2.1 **progressOperation : :progressOperation** ([progressOperation](#) * *parentOperation*, unsigned int *sonEstimation* = 1) [**inline**]

Constructeur

6.11.3 Documentation des fonctions membres

- 6.11.3.1 **void progressOperation : :PushProgression** (float *incProg*) [**inline**, **protected**]

Un des fils a changé d'état incProg constitue la différence entre la dernière mise à jour de sa progression et sa nouvelle progression

Paramètres:

incProg [0;1]

6.12 Référence de la classe formatRSBIN : :rsurf_data

Contient les données d'un fichier de résultat de récepteur de surfaces.

Fonctions membres publiques

- void **GetFaceEnergy** (const std : :size_t &rsIndex, const std : :size_t &faceIndex, const std : :size_t &recordIndex, std : :size_t &idstep, float &energy) const
- void **GetFaceInfo** (const std : :size_t &rsIndex, const std : :size_t &faceIndex, std : :size_t &vertexA, std : :size_t &vertexB, std : :size_t &vertexC, std : :size_t &recordCount) const
- float **GetFaceSumEnergy** (const std : :size_t &rsIndex, const std : :size_t &faceIndex)
- void **GetFileInfos** (std : :size_t &rs_size, std : :size_t &nodeds_size, std : :size_t &nbtimestep, float ×tep, std : :string &recordType) const
- void **GetNodePositionValue** (const std : :size_t &nodeIndex, float &x, float &y, float &z) const
- std : :size_t **GetRsCount** ()
- void **GetRsInfo** (const std : :size_t &rsIndex, std : :size_t &nbfaces, std : :string &rs_name, int &xmld) const
- void **Make** (const std : :size_t &tabNodesSize, const std : :size_t &tabRsSize, const int &nbtimestep, const float ×tep, const std : :string &record_type="SPL_STANDART")
- void **MakeRs** (const std : :size_t &rsIndex, const std : :size_t &nbfacesconst, const std : :string &rs_name, const int &xmld)
- void **SetFaceEnergy** (const std : :size_t &rsIndex, const std : :size_t &faceIndex, const std : :size_t &recordIndex, const std : :size_t &idstep, const float &energy)
- void **SetFaceInfo** (const std : :size_t &rsIndex, const std : :size_t &faceIndex, const std : :size_t &vertexA, const std : :size_t &vertexB, const std : :size_t &vertexC, const std : :size_t &recordCount)
- void **SetNodeValue** (const std : :size_t &nodeIndex, const float &x, const float &y, const float &z)
- void **SetRsBinData** (t_ExchangeData &rsData)
- void **SetRsBinData** (t_ExchangeData *rsData)

Fonctions membres protégées

- t_ExchangeData * **GetStruct** ()
- void **SetStruct** (t_ExchangeData *extData)

6.12.1 Description détaillée

Contient les données d'un fichier de résultat de récepteur de surfaces. Permet de créer et de lire les données

6.12.2 Documentation des fonctions membres

- 6.12.2.1 void formatRSBIN : :rsurf_data : :**GetFaceEnergy** (const std : :size_t & *rsIndex*, const std : :size_t & *faceIndex*, const std : :size_t & *recordIndex*, std : :size_t & *idstep*, float & *energy*) const

Accesseur Information sur un enregistrement d'une face

Paramètres:

- ← *rsIndex* Indice du récepteur de surface
- ← *faceIndex* Indice de la surface
- ← *recordIndex* Indice de l'enregistrement
- *idstep* Compteur du pas de temps
- *energy* Energie(w) ou temps(s)

Python note :

Return a tuple containing (idstep,energy)

6.12.2.2 void formatRSBIN : :rsurf_data : :GetFaceInfo (const std : :size_t & rsIndex, const std : :size_t & faceIndex, std : :size_t & vertexA, std : :size_t & vertexB, std : :size_t & vertexC, std : :size_t & recordCount) const

Accesneur Information sur une face

Paramètres:

- ← *rsIndex* Indice du récepteur de surface
- ← *faceIndex* Indice de la surface
- *vertexA* Indice du sommet
- *vertexB* Indice du sommet
- *vertexC* Indice du sommet
- *recordCount* Nombre d'enregistrement

Python note :

Return a tuple containing (vertexA,vertexB,vertexC,recordCount)

6.12.2.3 float formatRSBIN : :rsurf_data : :GetFaceSumEnergy (const std : :size_t & rsIndex, const std : :size_t & faceIndex)

Accesneur Information sur un enregistrement d'une face

Paramètres:

- ← *rsIndex* Indice du récepteur de surface
- ← *faceIndex* Indice de la surface

Renvoi:

Energie cumulé (w) ou temps (s)

6.12.2.4 void formatRSBIN : :rsurf_data : :GetFileInfos (std : :size_t & rs_size, std : :size_t & nodes_size, std : :size_t & nbtimestep, float & timestep, std : :string & recordType) const

Accesneur propriétés du fichier

Paramètres:

- *rs_size* Nombre de sommets de triangles
- *nodes_size* Nombre de récepteurs de surfaces
- *nbtimestep* Nombre de pas de temps maximum
- *timestep* Pas de temps en secondes
- *record_type* Type d'enregistrement. Valeurs possibles "SPL_STANDART" "SPL_GAIN" "TR" "EDT"

Python note :

Return a tuple containing (rs_size,nodes_size,nbtimestep,timestep,recordType)

6.12.2.5 void formatRSBIN : :rsurf_data : :GetNodePositionValue (const std : :size_t & nodeIndex, float & x, float & y, float & z) const

Accesseur valeur d'un noeud

Paramètres:

- ← *nodeIndex* Indice du sommet
- *x* X
- *y* Y
- *z* Z

Python note :

Return a tuple containing (x,y,z)

6.12.2.6 std : :size_t formatRSBIN : :rsurf_data : :GetRsCount ()**Renvoie:**

le nombre de récepteur surfaciques.

6.12.2.7 void formatRSBIN : :rsurf_data : :GetRsInfo (const std : :size_t & rsIndex, std : :size_t & nbfaces, std : :string & rs_name, int & xmlid) const

Accesseur Information sur un recepteur surfacique

Paramètres:

- ← *rsIndex* Indice du récepteur de surface
- *nbfaces* Nombre de surfaces
- *rs_name* Libellé du récepteur de surface
- *xmlid* Indice xml du récepteur

Python note :

Return a tuple containing (nbfaces,rs_name,xmlid)

6.12.2.8 void formatRSBIN : :rsurf_data : :Make (const std : :size_t & tabNodesSize, const std : :size_t & tabRsSize, const int & nbtimestep, const float & timestep, const std : :string & record_type = "SPL_STANDART")

1ere étape Création, Mise en place des récepteurs et des propriétés

Paramètres:

- ← *tabNodesSize* Nombre de sommets de triangles
- ← *tabRsSize* Nombre de récepteurs de surfaces
- ← *nbtimestep* Nombre de pas de temps maximum
- ← *timestep* Pas de temps en secondes
- ← *record_type* Type d'enregistrement. Valeurs possibles "SPL_STANDART" "SPL_GAIN" "TR" "EDT"

6.12.2.9 void formatRSBIN : :rsurf_data : :MakeRs (const std : :size_t & rsIndex, const std : :size_t & nbfacesconst, const std : :string & rs_name, const int & xmlid)

2eme étape Création, Affectation des propriétés d'un récepteur

Paramètres:

- ← *rsIndex* Indice du récepteur surfacique
- ← *nbfacesconst* Nombre de faces dans ce récepteur de surfaces.
- ← *rs_name* Nom du récepteur de surface
- ← *xmlid* Indice XML du récepteur de surface

6.12.2.10 void formatRSBIN : :rsurf_data : :SetFaceEnergy (const std : :size_t & rsIndex, const std : :size_t & faceIndex, const std : :size_t & recordIndex, const std : :size_t & idstep, const float & energy)

4eme étape Création, Affectation de la valeur de pression d'un pas de temps sur une face

Paramètres:

- ← *rsIndex* Indice du récepteur de surface
- ← *faceIndex* Indice de la face
- ← *recordIndex* Indice de l'enregistrement
- ← *idstep* xeme pas de temps (x=y sec/timestep)
- ← *energy* Niveau de pression (SPL) ou temps (TR EDT)

6.12.2.11 void formatRSBIN : :rsurf_data : :SetFaceInfo (const std : :size_t & rsIndex, const std : :size_t & faceIndex, const std : :size_t & vertexA, const std : :size_t & vertexB, const std : :size_t & vertexC, const std : :size_t & recordCount)

3eme étape Création, Affectation des propriétés d'une face

Paramètres:

- ← *rsIndex* Indice du récepteur de surface
- ← *faceIndex* Indice de la face
- ← *vertexA* Indice du sommet
- ← *vertexB* Indice du sommet
- ← *vertexC* Indice du sommet
- ← *recordCount* Nombre d'enregistrements de niveaux (energy !=0)

6.12.2.12 void formatRSBIN : :rsurf_data : :SetNodeValue (const std : :size_t & nodeIndex, const float & x, const float & y, const float & z)

2eme étape Création, Affectation de la valeur d'un noeud

Paramètres:

- ← *nodeIndex* Indice du sommet
- ← *x* Position X (m)
- ← *y* Position Y (m)
- ← *z* Position Z (m)

6.12.2.13 void formatRSBIN : :rsurf_data : :SetRsBinData (t_ExchangeData * rsData)

Affecte directement les données à cette classe à partir de la structure correspondance.

Avertissement:

Cette classe supprimera ces données, à moins que NULL soit passé en paramètre avant la suppression de [rsurf_data](#)

Python note :

Non disponible en python

6.13 Référence de la classe formatRSBIN : :rsurf_io

Fonctions membres publiques statiques

- static bool **Load** (const std : :string &fileName, [rsurf_data](#) &data)
- static bool **Save** (const std : :string &fileName, [rsurf_data](#) &data)

6.14 Référence de la classe formatRSBIN : :rsurf_iso_contouring

Fonctions membres publiques

- void [GetIsoLevelCurves](#) (const float &isolvl, std : :list< [t_curve](#) > &curvlst)
- void [GetMinMax](#) (float &min, float &max)
- [rsurf_iso_contouring](#) (const [rsurf_data](#) &data, const std : :size_t &rsIndex)
- void [SmoothGrid](#) (const int &smoothcount)

6.14.1 Documentation des constructeurs et destructeur

- 6.14.1.1** `formatRSBIN : :rsurf_iso_contouring : :rsurf_iso_contouring (const rsurf_data & data, const std : :size_t & rsIndex)`

/~english Iso Contouring initialisation step.

6.14.2 Documentation des fonctions membres

- 6.14.2.1** `void formatRSBIN : :rsurf_iso_contouring : :GetIsoLevelCurves (const float & isolvl, std : :list< t_curve > & curvlst)`

Calcul la position de segments

- 6.14.2.2** `void formatRSBIN : :rsurf_iso_contouring : :GetMinMax (float & min, float & max)`

Retourne les valeurs min,max

- 6.14.2.3** `void formatRSBIN : :rsurf_iso_contouring : :SmoothGrid (const int & smoothcount)`

Afin de lisser les isocourbes on peut multiplier le nombre de triangle et interpoler la valeur des nouveaux sommets.

Paramètres:

smoothcount Nombre de subdivision de la grille.

6.15 Référence de la classe surf_merge : :SurfaceMerging

Fonctions membres publiques

- void [GetGroupInformation](#) (const int &idel, std : :vector< std : :size_t > &faceIds)
- int [GetGroups](#) ()
- void [LoadGroups](#) (const [formatCoreBIN](#) : :ioModel &model)
- void [SavePolyToPolyFile](#) (const char *strFileName)

6.15.1 Description détaillée

Regrouper les faces selon la même norme et le même matériau

Permet de regrouper les faces selon la même norme et le même matériau

6.15.2 Documentation des fonctions membres

6.15.2.1 void surf_merge : :SurfaceMerging : :GetGroupInformation (const int & *idel*, std : :vector< std : :size_t > & *faceIds*)

Donne les données pour un groupe

Paramètres:

- ← *idel* Indice du groupe [0-GetGroups()]
- *faceIds* Faces du groupe

Python note :

Return faceIds vector, use list(SurfaceMerging : :GetGroupInformation(idel)) python built-in function to translate into an integer list.

6.15.2.2 int surf_merge : :SurfaceMerging : :GetGroups ()

Retourne le nombre de groupes de surfaces détectés

6.15.2.3 void surf_merge : :SurfaceMerging : :LoadGroups (const [formatCoreBIN](#) : :ioModel & *model*)

Lectures des structures de données à la recherche des faces coplanaires

Paramètres:

- model* Modèle de la scène

6.15.2.4 void surf_merge : :SurfaceMerging : :SavePolyToPolyFile (const char * *strFileName*)

Sauvegarde les triangles fusionnés dans le format de fichier .poly

Paramètres:

- strFileName* Nom et chemin du fichier de sortie

6.16 Référence de la structure formatMBIN : :t_binNode

Fonctions membres publiques

- int **operator==** (const t_binNode &f)
- t_binNode (const Floatb &x, const Floatb &y, const Floatb &z)

Attributs publics

- Floatb **node** [3]

6.17 Référence de la structure formatRSBIN : :t_curve

Attributs publics

- t_pos **A**
- t_pos **B**

6.18 Référence de la classe formatCoreBIN : :t_pos

Fonctions membres publiques

- **operator float * ()**
- **int operator==** (const **t_pos** & _v)
- **const float & operator[]** (int _i) const
- **t_pos** (float _x, float _y, float _z)

Attributs publics

- union {
 - struct {
 - float x
 - float y
 - float z
- float v [3]

6.19 Référence de la structure formatMBIN : :trimeshmodel

Attributs publics

- std : :vector< [t_binNode](#) > nodes
- std : :vector< [bintetrahedre](#) > tetrahedres

6.20 Référence de la classe `volumes_splitter : :VolumesSplitter`

Classes

- struct `facemodel_t`
- struct `vol_t`

Fonctions membres publiques

- void `GetInternalFaces` (std : :vector< std : :size_t > &internalFaceIds, const int &volumeXmlIdFilter=-1)
- int `GetTetraByPosition` (const `formatMBIN : :trimeshmodel` &tetramodel, const float &x, const float &y, const float &z, const bool &reloadTetraModel=false)
- void `GetVolumeInformation` (const int &idel, int &volumeXmlId, double &volume, std : :vector< std : :size_t > &faceIds, std : :vector< std : :size_t > &volNeigh)
- int `GetVolumes` ()
- int `GetVolumeXmlId` (const int &idel)
- void `LoadDomain` (const `formatCoreBIN : :ioModel` &model, const `formatMBIN : :trimeshmodel` &tetramodel)

6.20.1 Description détaillée

Séparation d'un modèle en volumes

Cette classe permet de regrouper les faces par rapport aux volumes auxquels ils appartiennent

6.20.2 Documentation des fonctions membres

6.20.2.1 void `volumes_splitter : :VolumesSplitter : :GetInternalFaces` (std : :vector< std : :size_t > & *internalFaceIds*, const int & *volumeXmlIdFilter* = -1)

Retourne la liste des limites internes (faces ayant un volume de chaque coté)

Paramètres:

→ *internalFaceIds* Faces propre au volume

int] *volumeXmlIdFilter* Limiter les faces internes appartenant à ce volume (-1 aucun filtre)

Python note :

Return *internalFaceIds*

6.20.2.2 int `volumes_splitter : :VolumesSplitter : :GetTetraByPosition` (const `formatMBIN : :trimeshmodel` & *tetramodel*, const float & *x*, const float & *y*, const float & *z*, const bool & *reloadTetraModel* = false)

Cherche la tétraèdre contenant le point indiqué en paramètre.

Avertissement:

Il n'est pas nécessaire d'appeler la méthode `LoadDomain` avant d'utiliser cette fonction

Paramètres:

tetramodel Maillage de la scène

x Position

y Position

z Position

reloadTetraModel Recharge le premier paramètre dans la table de test de position. Mettez le à vrai si vous avez modifié la structure tetramodel entre deux appel de GetTetraByPosition.

Renvoie:

L'indice du tétraèdre ou -1 si non trouvé.

6.20.2.3 `void volumes_splitter : :VolumesSplitter : :GetVolumeInformation (const int & idel, int & volumeXmlId, double & volume, std : :vector< std : :size_t > & faceIds, std : :vector< std : :size_t > & volNeigh)`

Donne les données pour un volume

Paramètres:

- ← *idel* Indice du volume [0-GetVolumes()]
- *volumeXmlId* Indice Xml du volume
- *volume* Volume (m3)
- *faceIds* Faces propre au volume
- *volNeigh* Volumes voisins du volume idel

Python note :

Read through VolumesSplitterInstance[*valid*] and return a tuple containing (volumeXmlId,volume,faceIds,volNeigh)

6.20.2.4 `int volumes_splitter : :VolumesSplitter : :GetVolumes () [inline]`

Retourne le nombre de volumes détectés

Python note :

Read through len(VolumesSplitterInstance)

6.20.2.5 `int volumes_splitter : :VolumesSplitter : :GetVolumeXmlId (const int & idel)`

Donne l'Indice Xml du volume

Paramètres:

- ← *idel* Indice du volume [0-GetVolumes()]

Renvoie:

Indice Xml du volume

6.20.2.6 `void volumes_splitter : :VolumesSplitter : :LoadDomain (const formatCoreBIN : :ioModel & model, const formatMBIN : :trimeshmodel & tetramodel)`

Lectures des structures de données à la recherche des volumes

Paramètres:

- model* Modèle de la scène
- tetramodel* Maillage de la scène

Index

- ~CMBIN
 - formatMBIN : :CMBIN, 24
- ~CformatBIN
 - formatCoreBIN : :CformatBIN, 22
- a
 - formatCoreBIN : :ioFace, 27
- AppendFloatCol
 - gabe_io : :Gabe_rw, 25
- AppendIntCol
 - gabe_io : :Gabe_rw, 25
- AppendStrCol
 - gabe_io : :Gabe_rw, 25
- b
 - formatCoreBIN : :ioFace, 27
- c
 - formatCoreBIN : :ioFace, 27
 - CalculsGenerauxThermodynamique, 11
 - CalculsGenerauxThermodynamique : :CCalculsGenerauxThermodynamique, 21
 - CformatBIN
 - formatCoreBIN : :CformatBIN, 22
 - Close
 - particleio : :ParticuleIO, 29
 - CMBIN
 - formatMBIN : :CMBIN, 24
 - Ecriture d'un fichier de particule, 7
 - ExportBIN
 - formatCoreBIN : :CformatBIN, 22
 - faces
 - formatCoreBIN : :ioModel, 28
 - formatCoreBIN, 12
 - formatCoreBIN : :CformatBIN, 22
 - ~CformatBIN, 22
 - CformatBIN, 22
 - ExportBIN, 22
 - GetSurfAire, 22
 - ImportBIN, 22
 - formatCoreBIN : :ioFace, 27
 - a, 27
 - b, 27
 - c, 27
 - idEn, 27
 - idMat, 27
 - idRs, 27
 - formatCoreBIN : :ioModel, 28
 - faces, 28
 - vertices, 28
 - formatCoreBIN : :t_pos, 43
 - formatMBIN, 13
 - formatMBIN : :bintetraface, 19
 - marker, 19
 - neighbor, 19
 - formatMBIN : :bintetrahedre, 20
 - formatMBIN : :CMBIN, 24
 - ~CMBIN, 24
 - CMBIN, 24
 - LoadMesh, 24
 - SaveMesh, 24
 - formatMBIN : :t_binNode, 41
 - formatMBIN : :trimeshmodel, 44
 - formatRSBIN, 14
 - formatRSBIN : :rsurf_data, 33
 - GetFaceEnergy, 33
 - GetFaceInfo, 34
 - GetFaceSumEnergy, 34
 - GetFileInfos, 34
 - GetNodePositionValue, 34
 - GetRsCount, 35
 - GetRsInfo, 35
 - Make, 35
 - MakeRs, 35
 - SetFaceEnergy, 36
 - SetFaceInfo, 36
 - SetNodeValue, 36
 - SetRsBinData, 36
 - formatRSBIN : :rsurf_io, 38
 - formatRSBIN : :rsurf_iso_contouring, 39
 - GetIsoLevelCurves, 39
 - GetMinMax, 39
 - rsurf_iso_contouring, 39
 - SmoothGrid, 39
 - formatRSBIN : :t_curve, 42
 - gabe_io, 15
 - gabe_io : :Gabe_rw, 25
 - AppendFloatCol, 25

- AppendIntCol, 25
- AppendStrCol, 25
- Gabe_rw, 25
- GetColTitle, 25
- GetTabTypes, 26
- IsReadOnly, 26
- Load, 26
- Save, 26
- SetReadOnly, 26
- size, 26
- Gabe_rw
 - gabe_io : :Gabe_rw, 25
- GetColTitle
 - gabe_io : :Gabe_rw, 25
- GetFaceEnergy
 - formatRSBIN : :rsurf_data, 33
- GetFaceInfo
 - formatRSBIN : :rsurf_data, 34
- GetFaceSumEnergy
 - formatRSBIN : :rsurf_data, 34
- GetFileInfos
 - formatRSBIN : :rsurf_data, 34
- GetGroupInformation
 - surf_merge : :SurfaceMerging, 40
- GetGroups
 - surf_merge : :SurfaceMerging, 40
- GetHeaderData
 - methodreadpart, 9
- GetInternalFaces
 - volumes_splitter : :VolumesSplitter, 45
- GetIsoLevelCurves
 - formatRSBIN : :rsurf_iso_contouring, 39
- GetMainOperation
 - progressionInfo, 30
- GetMinMax
 - formatRSBIN : :rsurf_iso_contouring, 39
- GetNodePositionValue
 - formatRSBIN : :rsurf_data, 34
- GetRsCount
 - formatRSBIN : :rsurf_data, 35
- GetRsInfo
 - formatRSBIN : :rsurf_data, 35
- GetSurfAire
 - formatCoreBIN : :CformatBIN, 22
- GetTabTypes
 - gabe_io : :Gabe_rw, 26
- GetTetraByPosition
 - volumes_splitter : :VolumesSplitter, 45
- GetVolumeInformation
 - volumes_splitter : :VolumesSplitter, 46
- GetVolumes
 - volumes_splitter : :VolumesSplitter, 46
- GetVolumeXmlId
 - volumes_splitter : :VolumesSplitter, 46
- idEn
 - formatCoreBIN : :ioFace, 27
- idMat
 - formatCoreBIN : :ioFace, 27
- idRs
 - formatCoreBIN : :ioFace, 27
- ImportBIN
 - formatCoreBIN : :CformatBIN, 22
- IsReadOnly
 - gabe_io : :Gabe_rw, 26
- Lecture d'un fichier de particule, 9
- Load
 - gabe_io : :Gabe_rw, 26
- LoadDomain
 - volumes_splitter : :VolumesSplitter, 46
- LoadGroups
 - surf_merge : :SurfaceMerging, 40
- LoadMesh
 - formatMBIN : :CMBIN, 24
- Make
 - formatRSBIN : :rsurf_data, 35
- MakeRs
 - formatRSBIN : :rsurf_data, 35
- marker
 - formatMBIN : :bintetraface, 19
- methodreadpart
 - GetHeaderData, 9
 - NextParticle, 9
 - NextTimeStep, 9
 - OpenForRead, 10
- methodwritepart
 - NewParticle, 7
 - NewPositionParticle, 7
 - SetHeaderData, 7
- neighbor
 - formatMBIN : :bintetraface, 19
- NewParticle
 - methodwritepart, 7
- NewPositionParticle
 - methodwritepart, 7
- NextParticle
 - methodreadpart, 9
- NextTimeStep
 - methodreadpart, 9
- OpenForRead
 - methodreadpart, 10
- OutputCurrentProgression
 - progressionInfo, 30
- particleio, 16
- particleio : :ParticuleIO, 29

- Close, 29
- progressionInfo, 30
 - GetMainOperation, 30
 - OutputCurrentProgression, 30
 - progressionInfo, 30
- progressOperation, 32
 - progressOperation, 32
 - PushProgression, 32
- PushProgression
 - progressOperation, 32
- rsurf_iso_contouring
 - formatRSBIN : :rsurf_iso_contouring, 39
- Save
 - gabe_io : :Gabe_rw, 26
- SaveMesh
 - formatMBIN : :CMBIN, 24
- SavePolyToPolyFile
 - surf_merge : :SurfaceMerging, 40
- SetFaceEnergy
 - formatRSBIN : :rsurf_data, 36
- SetFaceInfo
 - formatRSBIN : :rsurf_data, 36
- SetHeaderData
 - methodwritepart, 7
- SetNodeValue
 - formatRSBIN : :rsurf_data, 36
- SetReadOnly
 - gabe_io : :Gabe_rw, 26
- SetRsBinData
 - formatRSBIN : :rsurf_data, 36
- size
 - gabe_io : :Gabe_rw, 26
- SmoothGrid
 - formatRSBIN : :rsurf_iso_contouring, 39
- surf_merge, 17
- surf_merge : :SurfaceMerging, 40
 - GetGroupInformation, 40
 - GetGroups, 40
 - LoadGroups, 40
 - SavePolyToPolyFile, 40
- vertices
 - formatCoreBIN : :ioModel, 28
- volumes_splitter : :VolumesSplitter, 45
 - GetInternalFaces, 45
 - GetTetraByPosition, 45
 - GetVolumeInformation, 46
 - GetVolumes, 46
 - GetVolumeXmlId, 46
 - LoadDomain, 46