AGRIF

Adaptive **G**rid **R**efinement **I**n **F**ortran

Reference Manual

Version 1.3

27 novembre 2006

AGRIF is a package for the integration of adaptive mesh refinement (AMR) features within a multidimensional model written in Fortran and discretized on a structured grid.

This document deals with all functions and subroutines which can be used by AGRIF user's.

Contents

Agrif_Bc_variable

Type of procedure

Subroutine

Description

This routine calculates the boundary conditions for the array mentioned in argument.

More, if a fine grid has a common border with the root coarse grid, then boundary conditions are never calculated on this border when the Agrif_Bc_variable subroutine is called on this fine grid. However, if the user really wants to do this calculation, he must use the `Agrif_InterpNearBorderDim` and `Agrif_InterpDistantBorderDim` procedures where dim is the name of the dimension (x, y or z).

Arguments

```
subroutine Agrif_Bc_variable(Variable_Result,Variable,calledweight,procnam
```

`Variable` : grid variable
`Variable_Result` : result of the interpolation
`calledweight` : weight for the time interpolation during the calculation of the boundary conditions. It is a optional argument.
If it is present, then coefficients used for this linear interpolation are calledweight and 1-calledweight; otherwise they are equals to $n * 1/rhot$ and $1 - n * 1/rhot$ where rhot is the time refinement factor of the fine grid.

`procname` : It is the name of an user program which compute the interpolation. It is an optional argument.
It is it present, this program is use to compute the interpolation else it is the AGRIF interpolation which is used.

Example

```
call AGRIF_BC_variable(tab,u,0.5)
```

Conv effects

The name of the variable is replaced by its correspondant indice in the tabvars table.

Agrif_Cfixed

Type of procedure

Function

Description

This function returns the number of the current grid as a string.

Arguments

None

Result

Character*3

Example

```
character*3 :: name

   name = Agrif_Cfixed()
```

Conv effects

None.

Agrif_Childgrid_to_Parentgrid

Type of procedure

Subroutine

Description

This subroutine allows to be located on the parent grid of the current grid. A call to the AGRIF_Parentgrid_to_Childgrid allows to come back on this current grid.

Arguments

None

Example

Call Agrif_Childgrid_to_Parentgrid()

Conv effects

None.

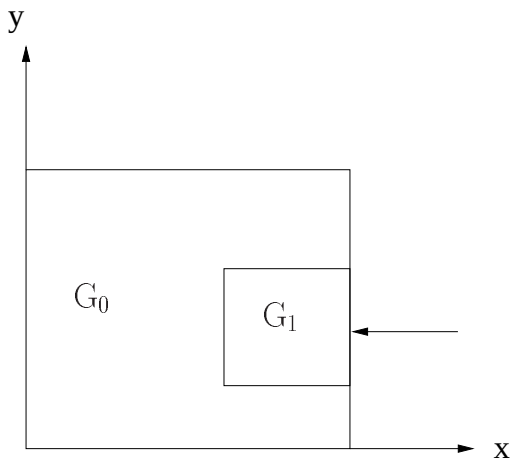
Agrif_DistantCommonBorderX

Type of procedure

Function

Description

This function indicates if the current grid as its distant border common with one of the root grid (in the x direction).



Arguments

None

Result

Logical

Example

If Agrif_DistantCommonBorderX() ...

Conv effects

Replace by its value.

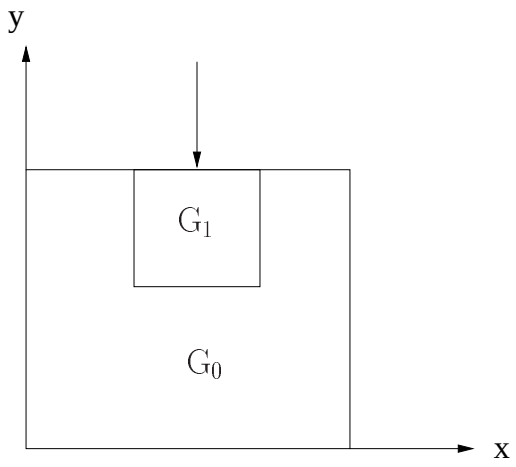
Agrif_DistantCommonBorderY

Type of procedure

Function

Description

This function indicates if the current grid as its distant border common with one of the root grid (in the y direction).



Arguments

None

Result

Logical

Example

If Agrif_DistantCommonBorderY() ...

Conv effects

Replace by its value.

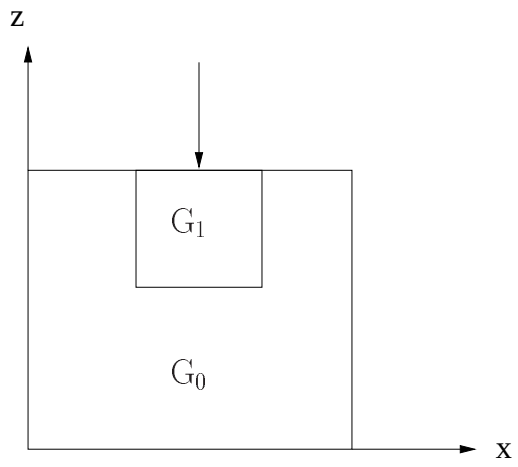
Agrif_DistantCommonBorderZ

Type of procedure

Function

Description

This function indicates if the current grid as its distant border common with one of the root grid (in the z direction).



Arguments

None

Result

Logical

Example

If Agrif_DistantCommonBorderZ() ...

Conv effects

Replace by its value.

Agrif_Fixed

Type of procedure

Function

Description

This function returns the number of the current grid. More precisely, it returns 0 for the coarse grid, a positive number for a fixed grid and -1 for a non fixed grid.

Arguments

None

Result

Integer

Example

```
integer :: nb

   nb = Agrif_Fixed()
```

Conv effects

None.

Agrif_Get_Unit

Type of procedure

Function

Description

This function returns a unit not connected to any file.
This unit could not be one of units listed in the file Agrif_forbiddenUnit.txt.

Arguments

None

Result

Integer

Example

```
integer :: nunit

   nunit = Agrif_Get_Unit()

   open(nunit,file='test.dat')
```

Conv effects

None.

Agrif_Init_Grids

Type of procedure

Subroutine

Description

This routine initialises the grid hierarchy. It must only be called once.
Let's note the number of cells of the root grid have to be initialized before calling this routine.
This calling should be the first line written in the original code.

Arguments

None

Example

Call Agrif_Init_Grids()

Conv effects

None.

Agrif_Init_variable

Type of procedure

Subroutine

Description

The subroutine is used for the initialization of the fine grids.

Arguments

```
subroutine Agrif_Init_variable(Variable_Result,Variable)
```

`Variable_Result` : result of the interpolation `Variable` : name of the grid variable

Example

Call Agrif_Init_variable(tab,u)

Conv effects

The name of the variable is replaced by its correspondant indice in the tabvars table.

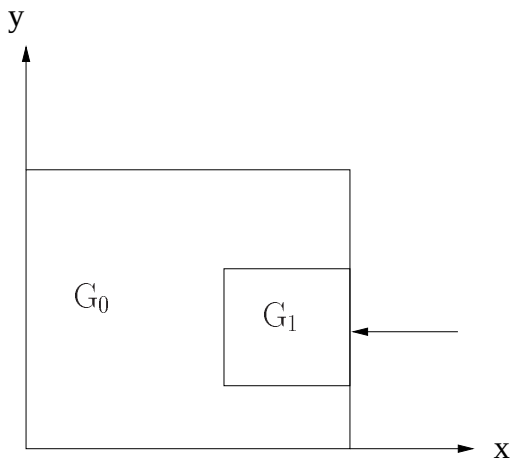
Agrif_InterpDistantBorderX

Type of procedure

Subroutine

Description

When this routine is called before the Agrif_Bc_variable subroutine, it allows the calculation of the boundary conditions (in the x direction) on a distant border of the current grid common with one of the borders of the root grid (by default, this calculation is not done).



Arguments

None

Example

Call Agrif_InterpDistantBorderX()

Conv effects

None.

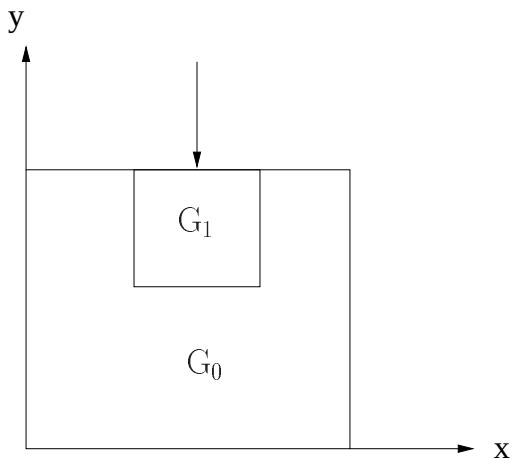
Agrif_InterpDistantBorderY

Type of procedure

Subroutine

Description

When this routine is called before the Agrif_Bc_variable subroutine, it allows the calculation of the boundary conditions (in the y direction) on a distant border of the current grid common with one of the borders of the root grid (by default, this calculation is not done).



Arguments

None

Example

Call Agrif_InterpDistantBorderY()

Conv effects

None.

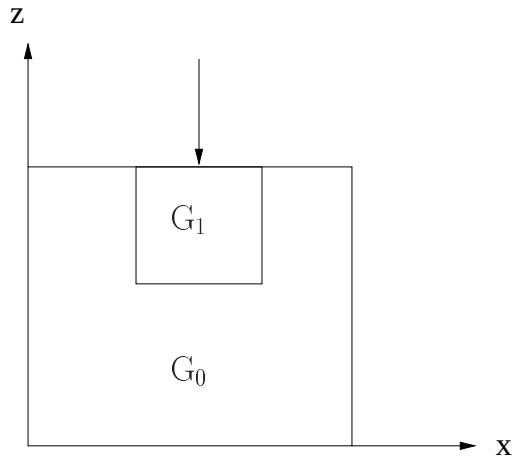
Agrif_InterpDistantBorderZ

Type of procedure

Subroutine

Description

When this routine is called before the Agrif_Bc_variable subroutine, it allows the calculation of the boundary conditions (in the z direction) on a distant border of the current grid common with one of the borders of the root grid (by default, this calculation is not done).



Arguments

None

Example

Call Agrif_InterpDistantBorderZ()

Conv effects

None.

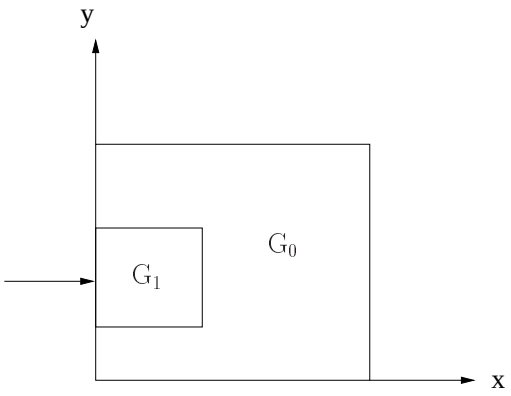
Agrif_InterpNearBorderX

Type of procedure

Subroutine

Description

When this routine is called before the Agrif_Bc_variable subroutine, it allows the calculation of the boundary conditions (in the x direction) on a near border of the current grid common with one of the borders of the root grid (by default, this calculation is not done).



Arguments

None

Example

Call Agrif_InterpNearBorderX()

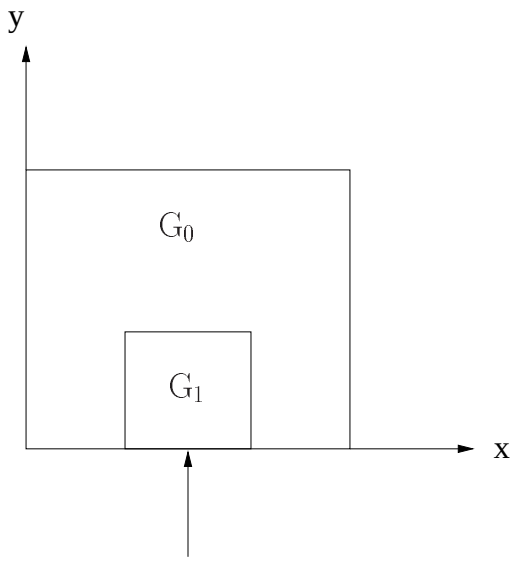
Agrif_InterpNearBorderY

Type of procedure

Subroutine

Description

When this routine is called before the Agrif_Bc_variable subroutine, it allows the calculation of the boundary conditions (in the y direction) on a near border of the current grid common with one of the borders of the root grid (by default, this calculation is not done).



Arguments

None

Example

Call Agrif_InterpNearBorderY()

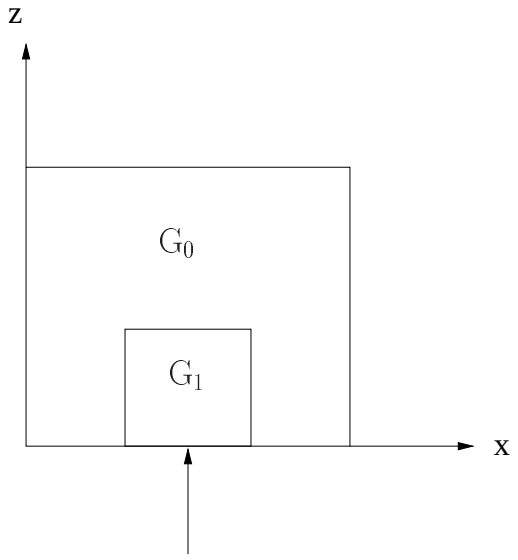
Agrif_InterpNearBorderZ

Type of procedure

Subroutine

Description

When this routine is called before the Agrif_Bc_variable subroutine, it allows the calculation of the boundary conditions (in the z direction) on a near border of the current grid common with one of the borders of the root grid (by default, this calculation is not done).



Arguments

None

Example

Call Agrif_InterpNearBorderZ()

Agrif_Interp_variable

Type of procedure

Subroutine

Description

This subroutine allows an interpolation of a fine grid variable when it is called.
8 methods of interpolation are available : linear,lagrange, spline, etc.

However, when the user calls an Agrif_Interp_VarName procedure, points where interpolation is done are different for staggered and for no staggered variable.

- For a no staggered variable, calculations are done on the inner points to the domain and on points located on the border of the grid (figure 1.1).
- For a staggered variable, calculations are done only on points located inside the grid (figure 2.1).

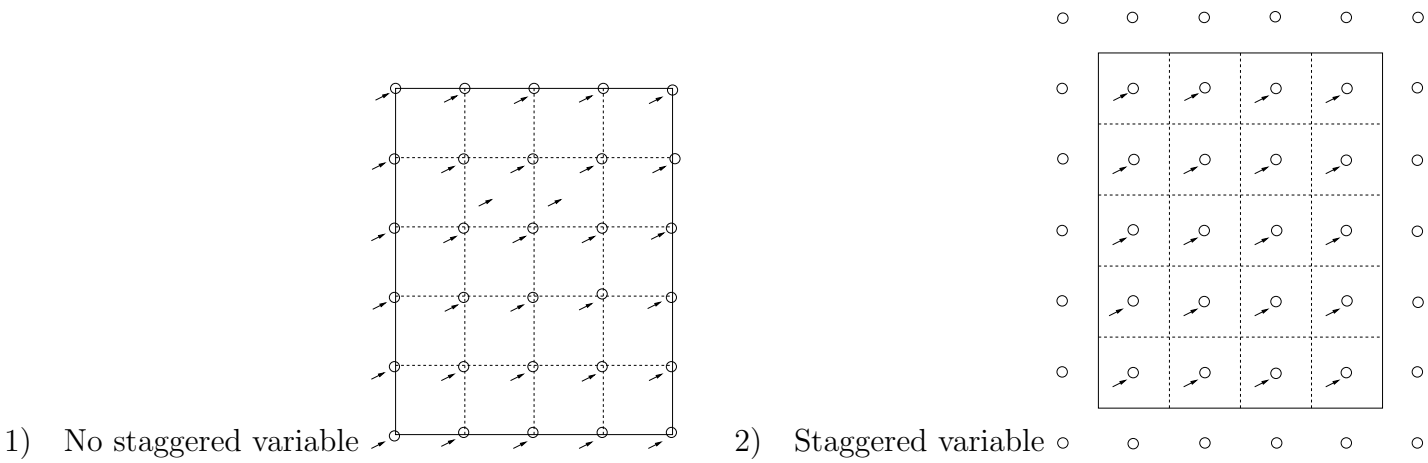


Figure 1: Interpolation of the grid variables (arrows indicate points which are interpolated)

Arguments

```
subroutine Agrif_Interp_variable(Variable_Result,Variable)
```

`Variable_Result` : result of the interpolation `Variable` : name of the grid variable

Example

```
call AGRIF_Interp_variable(u,u)
```
where u is the name of the interpolated grid variable; it can also be the name of an independent array.

Conv effects

The name of the variable is replaced by its correspondant indice in the tabvars table.

Agrif_IRhox

Type of procedure

Function

Description

This function indicates the space refinement factor of the current grid in the x direction.

Arguments

None

Result

Integer

Example

integer :: nb
nb = Agrif_IRhox()

Conv effects

Replace by its value.

Agrif_IRhoy

Type of procedure

Function

Description

This function indicates the space refinement factor of the current grid in the y direction.

Arguments

None

Result

Integer

Example

integer :: nb
nb = Agrif_IRhoy()

Conv effects

Replace by its value.

Agrif_IRhoz

Type of procedure

Function

Description

This function indicates the space refinement factor of the current grid in the z direction.

Arguments

None

Result

Integer

Example

integer :: nb
nb = Agrif_IRhoz()

Conv effects

Replace by its value.

Agrif_Is_Fixed

Type of procedure

Function

Description

This function returns true if the grid is fixed (including the root grid).

Arguments

None

Result

Logical

Example

If (Agrif_Is_Fixed()) Then
...
End If

Conv effects

None.

Agrif_Ix

Type of procedure

Function

Description

This function returns the minimum position of the current grid (in case of a fixed grid) on its parent grid for the x direction.

Arguments

None

Result

Integer

Example

integer :: nb
nb = Agrif_Ix()

Conv effects

Replace by its value.

Agrif_Iy

Type of procedure

Function

Description

This function returns the minimum position of the current grid (in case of a fixed grid) on its parent grid for the y direction.

Arguments

None

Result

Integer

Example

integer :: nb
nb = Agrif_Iy()

Conv effects

Replace by its value.

Agrif_Iz

Type of procedure

Function

Description

This function returns the minimum position of the current grid (in case of a fixed grid) on its parent grid for the z direction.

Arguments

None

Result

Integer

Example

integer :: nb
nb = Agrif_Iz()

Conv effects

Replace by its value.

Agrif_Nb_Fixed_Grids

Type of procedure

Function

Description

This function returns the number of fixed grids of the grid hierarchy.

Arguments

None

Result

Integer

Example

```
integer :: nb
nb = Agrif_Nb_Fixed_Grids()
```

Conv effects

Replace by its value.

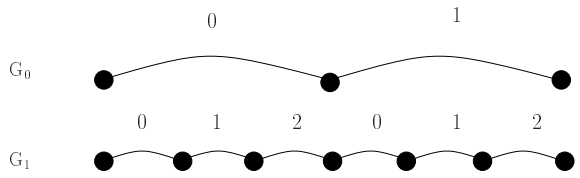
Agrif_Nbstepint

Type of procedure

Function

Description

This function indicates the local number of time steps of the current grid inside one time step on its parent grid.



Arguments

None

Result

Integer

Example

integer :: nb
nb = Agrif_NbStepInt()

Conv effects

None.

Agrif_Nb_Step

Type of procedure

Function

Description

This function indicates the number of iterations done on the current grid since the start of the time integration of the grid.

Arguments

None

Result

Integer

Example

integer :: nb
nb = Agrif_Nb_Step()

Conv effects

Replace by its value.

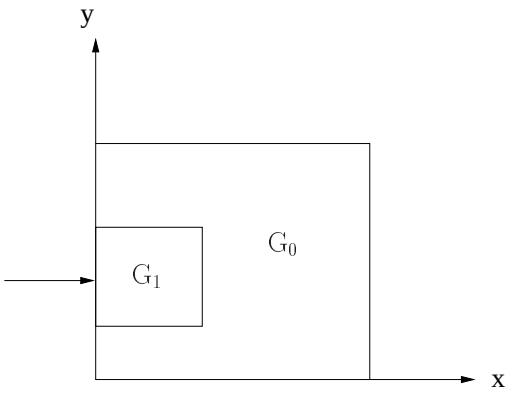
Agrif_NearCommonBorderX

Type of procedure

Function

Description

This function indicates if the current grid as its near border common with one of the root grid (in the x direction).



Arguments

None

Result

Logical

Example

If Agrif_NearCommonBorderX() ...

Conv effects

Replace by its value.

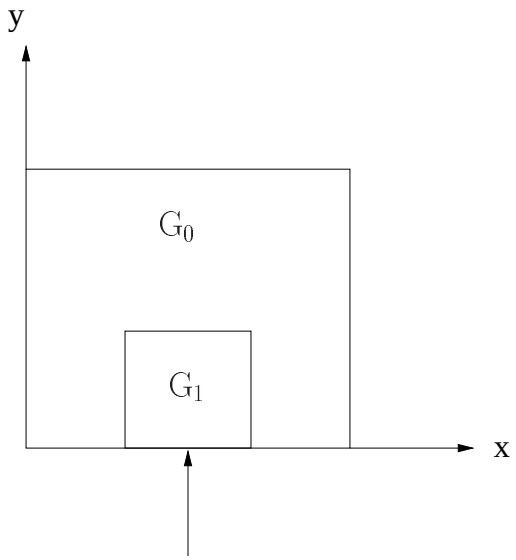
Agrif_NearCommonBorderY

Type of procedure

Function

Description

This function indicates if the current grid as its near border common with one of the root grid (in the y direction).



Arguments

None

Result

Logical

Example

If Agrif_NearCommonBorderY() ...

Conv effects

Replace by its value.

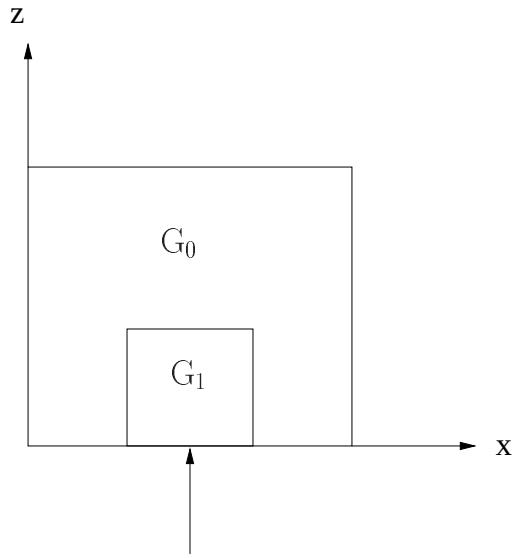
Agrif_NearCommonBorderZ

Type of procedure

Function

Description

This function indicates if the current grid as its near border common with one of the root grid (in the z direction).



Arguments

None

Result

Logical

Example

If Agrif_NearCommonBorderZ ...

Conv effects

Replace by its value.

Agrif_Parent_Cfixed

Type of procedure

Function

Description

This function returns the number of the parent grid of the current grid as a string.

Arguments

None

Result

Character*3

Example

```
character*3 :: name
name = Agrif_Parent_Cfixed()
```

Conv effects

None.

Agrif_Parent_Fixed

Type of procedure

Function

Description

This function returns the number of the parent grid of the current grid. More precisely, it returns 0 for the coarse grid, a positive number for a fixed grid and -1 for a no fixed grid.

Arguments

None

Result

Integer

Example

integer :: nb
nb = Agrif_Parent_Fixed()

Conv effects

None.

Agrif_Parentgrid_to_Childgrid

Type of procedure

Subroutine

Description

This subroutine allows to come back on the grid which was the current grid before the call to the Agrif_Childgrid_to_Parentgrid subroutine.

Arguments

None

Example

Call Agrif_Parentgrid_to_Childgrid()

Conv effects

None.

Agrif_Parent_IRhox

Type of procedure

Function

Description

This function indicates the space refinement factor of the parent grid of the current grid in the x direction.

Arguments

None

Result

Integer

Example

integer :: nb
nb = Agrif_Parent_IRhox()

Conv effects

Replace by its value.

Agrif_Parent_IRhoy

Type of procedure

Function

Description

This function indicates the space refinement factor of the parent grid of the current grid in the y direction.

Arguments

None

Result

Integer

Example

```
integer :: nb
nb = Agrif_Parent_IRhoy()
```

Conv effects

Replace by its value.

Agrif_Parent_IRhoz

Type of procedure

Function

Description

This function indicates the space refinement factor of the parent grid of the current grid in the z direction.

Arguments

None

Result

Integer

Example

integer :: nb
nb = Agrif_Parent_IRhoz()

Conv effects

Replace by its value.

Agrif_Parent_Is_Fixed

Type of procedure

Function

Description

This function returns true if the parent grid of the current grid is fixed (including the root grid).

Arguments

None

Result

Logical

Example

If (Agrif_Parent_Is_Fixed()) Then
...
End If

Conv effects

None.

Agrif_Parent_Ix

Type of procedure

Function

Description

This function returns the minimum position of the parent grid (in case of a fixed grid) on its parent grid for the x direction.

Arguments

None

Result

Integer

Example

integer :: nb
nb = Agrif_Parent_Ix()

Conv effects

Replace by its value.

Agrif_Parent_Iy

Type of procedure

Function

Description

This function returns the minimum position of the parent grid (in case of a fixed grid) on its parent grid for the y direction.

Arguments

None

Result

Integer

Example

integer :: nb
nb = Agrif_Parent_Iy()

Conv effects

Replace by its value.

Agrif_Parent_Iz

Type of procedure

Function

Description

This function returns the minimum position of the parent grid (in case of a fixed grid) on its parent grid for the z direction.

Arguments

None

Result

Integer

Example

integer :: nb
nb = Agrif_Parent_Iz()

Conv effects

Replace by its value.

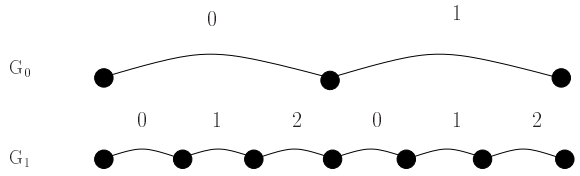
Agrif_Parent_Nbstepint

Type of procedure

Function

Description

This function indicates the local number of time steps of the parent grid of the current grid inside one time step on its parent grid.



Arguments

None

Result

Integer

Example

integer :: nb
nb = Agrif_Parent_NbStepInt()

Conv effects

None.

Agrif_Parent_Nb_Step

Type of procedure

Function

Description

This function indicates the number of iterations done on the parent grid of the current grid since the start of the time integration of the grid.

Arguments

None

Result

Integer

Example

integer :: nb
nb = Agrif_Parent_Nb_Step()

Conv effects

None.

Agrif_Parent_Rhot

Type of procedure

Function

Description

This function indicates the time refinement factor of the parent grid of the current grid.

Arguments

None

Result

Real

Example

real :: nb
nb = Agrif_Parent_Rhot()

Conv effects

Replace by its value.

Agrif_Parent_Rhox

Type of procedure

Function

Description

This function indicates the space refinement factor of the parent grid of the current grid in the x direction.

Arguments

None

Result

Real

Example

real :: nb
nb = Agrif_Parent_Rhox()

Conv effects

Replace by its value.

Agrif_Parent_Rhoy

Type of procedure

Function

Description

This function indicates the space refinement factor of the parent grid of the current grid in the y direction.

Arguments

None

Result

Real

Example

real :: nb
nb = Agrif_Parent_Rhoy()

Conv effects

Replace by its value.

Agrif_Parent_Rhoz

Type of procedure

Function

Description

This function indicates the space refinement factor of the parent grid of the current grid in the z direction.

Arguments

None

Result

Real

Example

real :: nb
nb = Agrif_Parent_Rhoz()

Conv effects

Replace by its value.

Agrif_Parent_Root

Type of procedure

Function

Description

This function indicates if the parent grid of the current grid is the root grid (true) or not (false).

Arguments

None

Result

Logical

Example

If Agrif_Parent_Root() ...

Conv effects

None.

Agrif_Parent

Type of procedure

Function

Description

This function gives the values of the grid variable on the parent grid.

Arguments

`function Agrif_Parent(Variable)`

`Variable` : name of the grid variable.

Result

Scalar or array pointer

Example

For a scalar variable :

Real :: dtparent

dtparent=Agrif_Parent(dt)
ou
Agrif_Parent(dt) = dtparent

For a multidimensionnal variable :

Real, Dimension(:,:), pointer :: parent_u
parent_u ⇒ Agrif_Parent(u)

Conv effects

Replace by its value.

Agrif_Rhot

Type of procedure

Function

Description

This function indicates the time refinement factor of the current grid.

Arguments

None

Result

Real

Example

real :: nb
nb = Agrif_Rhot()

Conv effects

None.

Agrif_Rhox

Type of procedure

Function

Description

This function indicates the space refinement factor of the current grid in the x direction.

Arguments

None

Result

Real

Example

real :: nb
nb = Agrif_Rhox()

Conv effects

None.

Agrif_Rhoy

Type of procedure

Function

Description

This function indicates the space refinement factor of the current grid in the y direction.

Arguments

None

Result

Real

Example

real :: nb
nb = Agrif_Rhoy()

Conv effects

None.

Agrif_Rhoz

Type of procedure

Function

Description

This function indicates the space refinement factor of the current grid in the z direction.

Arguments

None

Result

Real

Example

real :: nb
nb = Agrif_Rhoz()

Conv effects

None.

Agrif_Root

Type of procedure

Function

Description

This function indicates if the current grid is the root grid (true) or not (false).

Arguments

None

Result

Logical

Example

If Agrif_Root() ...

Conv effects

None.

Agrif_Set_parent

Type of procedure

Subroutine

Description

This subroutine is used to give a value to the parent grid variable.

Arguments

```
subroutine Agrif_Set_parent(Variable,Value)
```

`Variable` : grid variable
`Value` : Future value of the parent grid variable.

Example

```
call AGRIF_Set_parent(u,2)
```

Conv effects

The name of the variable is replaced by its correspondant indice in the tabvars table.

Agrif_Step

Type of procedure

Subroutine

Description

This routine replaces the user's time integration procedure. It integrates the model forward on all grids and performs a regridding at regular interval.

Arguments

None or the name of the user's calculation procedure.

Example

Call Agrif_Step()
or Call Agrif_Step(step)

Conv effects

None.

Agrif_update_variable

Type of procedure

Subroutine

Description

This subroutine allows to make update on a fine grid variable when it is called.
3 methods of update are available : copy, average and Full Weighting.

Arguments

```
subroutine Agrif_Update_variable(Variable_Result,Variable,locupdate,procna
```

`Variable_Result` : result of the update `Variable` : name of the grid variable
`locupdate` :
`procname` : Name of a user program which make the update computation.

Example

```
call AGRIF_Update_variable(u,u)
```

Conv effects

The name of the variable is replaced by its correspondant indice in the tabvars table.

Index