

Projection of a field on a grid

Summary:

The order PROJ_CHAMP allows “to project” fields (known on a grid *ma1*) on another grid *ma2* .

In this document, one describes the various accessible methods of projection in this order. The paragraph [§6] gives some elements of validation of these methods.

Contents

1	General information on the order PROJ_CHAMP.....	3
2	Method 'COLLOCATION'.....	4
2.1	Algorithm implemented.....	4
2.1.1	Notice.....	4
2.2	Encountered difficulties and their treatment.....	4
2.2.1	Stage 1.....	4
2.2.2	Stages 2 and 3.....	5
2.2.3	Stage 4.....	5
2.3	Details concerning stages 1 and 2.....	5
2.3.1	Node N2 "interior".....	6
2.3.2	Node N2 "external".....	6
2.3.3	Case of the grids "hull".....	7
3	Method 'NUAGE_DEG_0' or 'NUAGE_DEG_1'.....	7
3.1	Principle of the method.....	7
3.2	Choice of weights of the points of the cloud.....	9
3.3	Current restriction of the programming.....	9
4	Method 'ECLA_PG'.....	10
5	Method 'COUPLING'.....	11
6	Elements of validation for the projection of the fields to the nodes.....	12
6.1	Method: 'COLLOCATION'.....	13
6.1.1	Quadratic initial grid.....	13
6.2	Method: 'NUAGE_DEG_0'.....	14
6.2.1	Influence of the form of exponential decreasing.....	14
6.3	Method: 'NUAGE_DEG_1'.....	16
6.3.1	Influence of the form of exponential decreasing.....	17
6.4	Choice of the best method of projection for the fields with the nodes.....	17
7	Bibliography.....	17

1 General information on the order PROJ_CHAMP

The order PROJ_CHAMP allows to project the fields with the nodes or the fields by elements on another grid.

For the fields with the nodes, one has 4 methods:

```
METHOD: 'COLLOCATION'  
METHOD: 'NUAGE_DEG_0'  
METHOD: 'NUAGE_DEG_1'  
METHOD: 'COUPLING'
```

Method 'COLLOCATION' uses the functions of form of the elements of the grid *ma1*. It is detailed in the paragraph [§2].

The 2 other methods use a smoothing of the values of the field in the vicinity of the point where one wants to project the field. These 2 methods are detailed in the paragraph [§3].

Lastly, method 'COUPLING' is an alternative of the method 'COLLOCATION'. It is developed especially for the coupling fluid-structure with *Code_Saturne* and does not have normally not to thus be used apart from this framework.

For the fields by elements 'ELNO', one can use the method 'COLLOCATION'. The problem is the same one as for a field with the nodes but the field project is always continuous whereas the initial field is discontinuous. Moreover, if a node of the grid *ma2* is located at the border of 2 elements of the grid *ma1*, the projected value will be random: it will come from one or the other on the sides of the border to the liking of the classification of the elements. Therefore it is disadvised projecting the fields 'ELNO'.

For the fields by elements 'ELEM', one can still use the method 'COLLOCATION'. The value of the field (constant by element) is duplicated on the nodes of the element and one is reduced to the case of the fields 'ELNO'.

For the fields by elements 'ELGA', a dedicated method ('ECLA_PG') is described in the paragraph [§4]

Whatever the method, the user has the possibility of projecting only one "piece" of field on a "piece" of the grid *ma2*. This opportunity is given by the keyword factor VIS_A_VIS. A piece of field is the restriction of the field on a set of nodes (or meshes) of the grid *ma1*. A piece of the grid *ma2* is a subset of the nodes of *ma2*.

In the continuation of the document, one will not speak any more a subset of a grid, one will make as if all the grid were projected *ma1* on all the grid *ma2*.

2 Method 'COLLOCATION'

Notice on the vocabulary:

The word "to project" is sometimes ambiguous in this document.

When one says "to project" the field of $ma1$ towards $ma2$, one seeks the field on $ma2$ knowing that on $ma1$: projection goes from 1 towards 2.

For the method 'COLLOCATION', it is necessary to find for each node $n2$ of $ma2$ which is the point of $ma1$ who occupies the same position as $n2$, for that one projects the node $n2$ on the grid $ma1$: projection goes from 2 towards 1.

2.1 Algorithm implemented

One buckles on all the nodes of the grid $ma2$:

for each node ($n2$), one proceeds in 4 stages:

- 1) One seeks which is the mesh $m1$ of $ma1$ who "contains" the node geometrically $n2$,
- 2) One determines an approximate position of $n2$ in $m1$ (i.e its coordinates in the element of reference associated with the mesh $m1$),
- 3) One "refines" the position calculated previously by solving a nonlinear problem to take account owing to the fact that the meshes are not linear (the jacobien of the geometrical transformation varies on the mesh).
- 4) The functions of form of the mesh are used $m1$ to determine the value of the field on $n2$ knowing the value of the field on the nodes of $m1$.

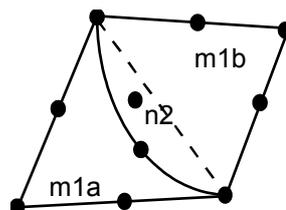
2.1.1 Notice

The third stage shows that this method supposes that all the nodes of the mesh $m1$ know the field to be projected. For example, one could not project a field which would carry different degrees of freedom on its nodes "tops" and on his nodes "mediums of edge". The projection of such a field is possible on the other hand with the 2 other methods of projection [§3].

2.2 Encountered difficulties and their treatment

2.2.1 Stage 1

- One does not treat the possible curve of the edges of the elements. For example, in figure Ci - below (problem plan), the node $n2$ will be stated to belong to the mesh $m1a$ whereas it belongs to the mesh $m1b$. Actually, this problem is rare because, generally, the "curved" edges exist only on the border of the fields with a grid (the internal facets are plane).



- If the node $n2$ is actually "external" with the grid $ma1$, the mesh will be affected to him $m1$ nearest to him. This behavior makes it possible to project, without stopping in fatal error, a field on a grid whose border differs slightly from that of the initial grid (what is always the case in practice).

2.2.2 Stages 2 and 3

To find the point of the element of reference which gives by the geometrical transformation the node $n2$, in general should be solved a non-linear problem, because there is only for the triangle with 3 nodes (in 2D) and for the tetrahedron with 4 nodes (in 3D) that the geometrical transformation is linear.

It is supposed initially that the mesh is linear by forgetting its nodes mediums and by arbitrarily cutting out it in triangles or tetrahedrons. In this "linearized" mesh one determines an approximate position of the point in with respect to the node $n2$.

Then one solves by an algorithm of Newton the non-linear problem of the precise research of the point on the basis of the approximation previously calculated.

Note: the non-linear problem is solved in two 2D cases (projection of a grid 2D on another grid 2D) or 3D (projection of a grid 3D on another grid 3D) but not for the cases 2.5D (hulls plunged in space 3D) or 1.5D (telegraphic grid in 2D),

2.2.3 Stage 4

One always does not use the true functions of form of the elements of the initial grid. Indeed, in *Code_Aster*, in fact the finite elements choose their functions of form: a triangle of thermics is not obliged to choose the same functions of form as a triangle of mechanics. An element can also not need functions of form, or it can choose different functions according to the variables to be interpolated. The formulation of the element either always does not reveal of element of reference and associated geometrical transformation.

For all these reasons and so that programming of `PROJ_CHAMP` that is to say independent of the finite elements present in the model, one assigns to all the meshes of `mal`, functions of form of the isoparametric elements 1D, 2D or 3D [R3.01.01].

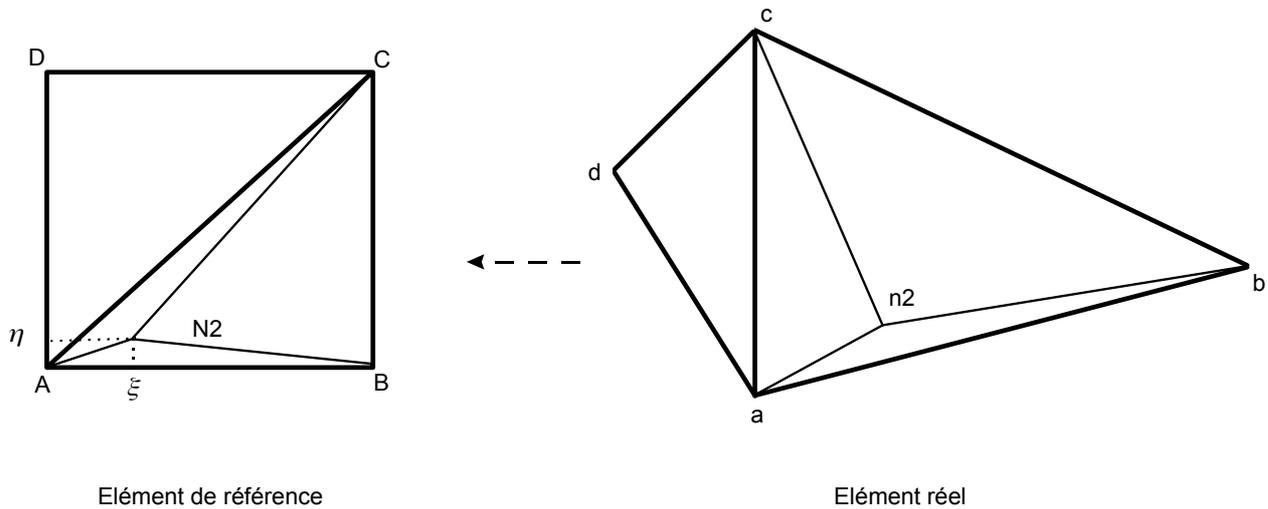
2.3 Details concerning stages 1 and 2

By means of computer, stages 1 and 2 are carried out simultaneously. We will discuss successively in the way of dealing with the 3 following problems:

- treatment of a node $n2$ finding itself inside the border of the grid `mal` (the most frequent case),
- treatment of a node $n2$ outside the border of `mal`,
- treatment of the case of the grids of type "hull" (surfaces plunged in \mathbb{R}^3).

2.3.1 Node N2 "interior"

To understand the treatment of an interior node, let us take the case of a mesh 2D QUAD8 ($abcd$). One starts by forgetting his nodes mediums (and thus their possible curve) then one cuts out it in 2 triangles (abc and acd). This cutting is arbitrary (it depends on the local classification of the nodes on QUAD8). Let us note that other possible cutting (another diagonal) would in general give another point in the element of reference.



$n2$ belongs to the triangle abc . One seeks his barycentric coordinates in this triangle. They are the 3 numbers xa, xb, xc such that one can write: $n2 = xa \times a + xb \times b + xc \times c$

The point N2 element of reference retained by the algorithm will be:

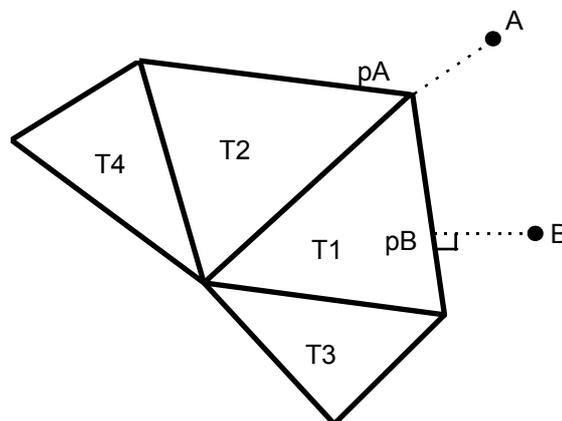
$$N2 = xa \times A + xb \times B + xc \times C$$

For the voluminal meshes (Hexahedrons, Pentahedrons, Pyramids and Tetrahedrons), one proceeds in the same way: one forgets the nodes mediums and one cuts out the nontetrahedral meshes in tetrahedrons.

2.3.2 Node N2 "external"

A node $n2$ is declared external with the grid mal if no mesh was found for which it is interior. When this is noted, the mesh is searched $m1$ nearest to $n2$. The calculated distance is that which separates the node $n2$ and the border of the mesh $m1$. Let us call $p2$ the point of $m1$ nearest to $n2$. This point can be on a face of a voluminal element or on an edge or a top.

Example (in 2D):



With the node B , the point is associated pB obtained here by orthogonal projection of B on an edge of $T1$.

With the node A , the point is associated pA triangle $T1$. One could just as easily have associated the point to him pA triangle $T2$, but that would not have changed anything since the field to be projected is known with the nodes of the grid. It is thus continuous between the adjacent elements.

Once found the point $p2$ mesh $m1$ who carries out the minimum of distance with $n2$, one is reduced to the preceding problem: the point $N2$ element of reference which will be associated with $n2$ will be the corresponding one of $p2$ by the procedure of the paragraph [§2.3.1].

Note:

For a triangle (or a tetrahedron) given T , there is only one point $p2$ carrying out the minimal distance enters $n2$ and T because T is convex. This property disappears if one took account of the curve of the edges of the meshes. It is seen there that two simplifications of the implementation (lapse of memory of the nodes mediums and cutting of the meshes in linear triangles) are dependent between them.

An external point will always have a value interpolated between the values of the nodes of the grid $ma1$ and ever extrapolated; what is not the case of the 2 methods `NUAGE_DEG_0/1`.

2.3.3 Case of the grids "hull"

When one seeks to project the nodes of a surface grid on another surface grid, one falls in general systematically on the case from the "external" points above. Indeed, the inaccuracy on the coordinates of the nodes makes that a node $n2$ is never rigorously in the plan of the triangles of the meshes of $ma1$.

It is thus the procedure of [the §2.3.2] which applies:

For each node $n2$:

- research of the triangle (or of the tetrahedron) which carries out the minimum of distance with $n2$. Identification of the point $p2$ who carries out this distance,
- calculation of the point $n2$ element of reference which corresponds to $p2$ by the procedure of [§3.3.1].

3 Method 'NUAGE_DEG_0' or 'NUAGE_DEG_1'

3.1 Principle of the method

These 2 methods are based on the same principle: one chooses a priori basic functions $F_i(x, y, z)$ (here of the polynomials of degree 0 or 1). One seeks in the vector space generated by these basic functions, the function $F = \sum (\alpha_i F_i)$ who carries out the distance minimum (within the meaning of least squares) with the "cloud" of the known points. Once this found function, one evaluates it at the sought point.

To reduce the notations, one places oneself in 2D (but calculations can be made in the same way in dimension 3 or more...). The field to be projected is a set of couples (X_j, V_j) where $X_j = (x_j, y_j)$ is a node of the grid $ma1$ and V_j is a reality (value of the field on this node). This field constitutes the cloud of the known points.

That is to say a node $n2(x, y)$ grid $ma2$ for which one wants to calculate the value of the field (V).

Choice of the basic functions:

Titre : Projection d'un champ sur un maillage
Responsable : PELLET Jacques

Date : 12/09/2013 Page : 8/17
Clé : R7.20.01 Révision :
4ba1d76a0f57

NUAGE_DEG_0 : only one function: $F1=1$
NUAGE_DEG_1 : 3 functions: $F1=1$; $F2=x$; $F3=y$

In the continuation of this paragraph, one will choose `NUAGE_DEG_1` so that the formulas do not degenerate too much.

The sought function is $F = \alpha_1 F_1 + \alpha_2 F_2 + \alpha_3 F_3$. One defines a kind of "distance" enters F and couples (X_j, V_j) :

$$D = \sum (w_j (F(X_j) - V_j)^2)$$

where them w_j are the weights assigned to each couple (X_j, V_j) .

One wants to minimize D compared to the 3 variables $\alpha_1, \alpha_2, \alpha_3$. D is a quadratic function of $\alpha_1, \alpha_2, \alpha_3$. To minimize D come down to cancel its derivative and thus to solve a linear system with 3 unknown factors: $\alpha_1, \alpha_2, \alpha_3$.

When this problem is solved, one calculates:

$$V = \alpha_1 + \alpha_2 x + \alpha_3 y$$

3.2 Choice of weights of the points of the cloud

All the trick of these methods is in the choice (difficult) weights w_j assigned to the points of the cloud.

- it is decided a priori that the weight of a point (X_j) depends only on the distance (d) separating this point from the node $n2$ (isotropic weighting),
- if w_j is a constant, the problem does not depend more of the unknown node $n2$. The function F is single (for all the nodes of $ma2$) : it is "the linear straight regression line" of the cloud,
- if $w_j(d)$ is a function too not very decreasing, the function F "is smoothed too much": the local "accidents" "are gummed" by a large number of remote points taken into account,
- if $w_j(d)$ is a too decreasing function, one takes the chance not to catch any point of the cloud. The digital consequence is that the linear system to solve becomes singular (and thus insoluble).

We chose to write $w(d)$ like exponential decreasing parameterized by 2 parameters: $dref$ and β :

$$w(d) = e^{-(d/dref)^{2\beta}}$$

$dref$ is a distance from reference (depend on the node $n2$). We will see below how it is calculated. β is a constant chosen to more or less quickly cancel the weight of the distant points of $n2$. In the code, β was selected to 0.75.

$dref$ is the distance from which one wishes to see the weight of the points decreasing quickly. In the programming, $dref$ is calculated like the product of a distance $d1$ by a coefficient $C1$. Today, we chose $C1=0.45$.

$d1$ is defined as follows:

- in 3D, $d1$ is the ray of the smallest ball of center $n2$ who contains 4 noncoplanar points of the cloud,
- in 2D, $d1$ is the ray of the smallest ball of center $n2$ who contains 3 points of the cloud not aligned.

The problem is known as 2D if all the nodes of the grid mal have the same coordinate z , it is known as 3D if not.

3.3 Current restriction of the programming

2 methods of projection NUAGE_DEG_0/1 are programmed only for the real fields (and not for the complex fields)

4 Method 'ECLA_PG'

This method is dedicated to the projection of the fields at the points of Gauss (ELGA).

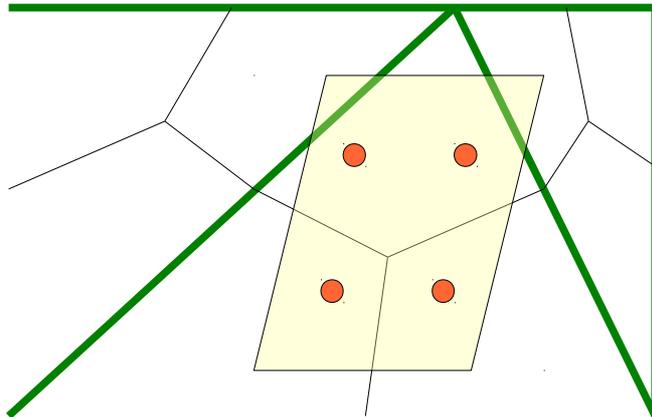
Let us illustrate this method on a 2D example.

It is supposed that the initial grid consists of 3 triangles (green) having 3 points of Gauss (see figure below).

Each triangle is cut out in as many subelements as of points of Gauss (here 3). The value carried by the point of Gauss is allotted to all the subelement.

A mesh of the final grid (parallelogram) is drawn in clear yellow. It has four points of Gauss (small rounds oranges).

The value of the field project on each one of these four points of Gauss will be that of the subelement in with respect to the point. In our example, the 2 higher points belong to the same subelement and they will carry the same value at the conclusion of projection.



5 Method 'COUPLING'

This method is dedicated to the coupling fluid-structure with *Code_Saturne* for the fluid field. It is based on the method 'COLLOCATION' by taking account of specificities of the problems fluid-structure considered (cf. [bib2]).

One thus seeks to project data of a fluid grid on a grid structure, and conversely.

The grid structure is based on a discretization EF, therefore with the knowledge of functions of forms associated, whereas the fluid grid is based on a discretization in finished volumes (VF), therefore without functions of forms *a priori*. The fields kinematics on fluid field VF are defined with the nodes tops whereas their duaux fields (efforts) are constant by faces and defined in the mediums of the faces.

Moreover, the grid structure is very generally definitely coarser than the fluid grid (which must be very refined with the wall in order to represent the boundary layer well).

The first stage of pairing of the nodes of the fluid grid on the solid is made by a method of the type 'COLLOCATION' classic. It should be noted that this method can be used only to generate the structure of data stamps correspondence of grid (CORRESP_2_MAILLA cf. U4.72.05 documentation). This structure of data will be then re-used by the operators of exchanges to the interfaces for the coupling fluid-structure:

- ENV_CINE_YACS (to send the fields displacement and speed to the fluid code, cf. documentation U7.06.11),
- MODI_CHAR_YACS (to recover the fluid efforts and to summon them with the existing loading structure, cf. U7.06.22 documentation).

Apart from this limitation, principal difference with the method 'COLLOCATION' will be in the construction of the structure of data stamps correspondence (CORRESP_2_MAILLA *confer* U4.72.05 documentation) at exit of PROJ_CHAMP.

Indeed, one will seek as well as possible to respect the conservation of energy to the interface, therefore the conservation of the work of the efforts discretized at the time of the fluid passage towards solid (cf. [bib2]). Obviously, this result is valid only under particular assumptions:

- fields of constant fluid effort by element,
- fluid grid finer than the solid grid,
- method of pairing of the type 'COLLOCATION' of *Code_Aster*.

In practice, for any problem coupled fluid-structure, one will have to use the method twice 'COUPLING' in order to calculate two structures of data of correspondence of grid. Indeed, one must separately manage the projection of the fields kinematics which is pressed on definite fluid fields with the nodes tops of the fluid grid, of that of the efforts which are defined in the mediums of the fluid faces. In each one of these two cases, the fluid grid correspondent will be either the grid of the nodes tops, or that of the nodes mediums of the faces. The method of projection is then the same one and the grid structure is, him, single. All the fluid fields (displacements, speeds and efforts with the interface) will be considered by *Code_Aster* like fields with the classical nodes.

6 Elements of validation for the projection of the fields to the nodes

To validate the 3 methods suggested, we will treat the example of the projection of a “unidimensional” field of temperature $T = T(x)$.

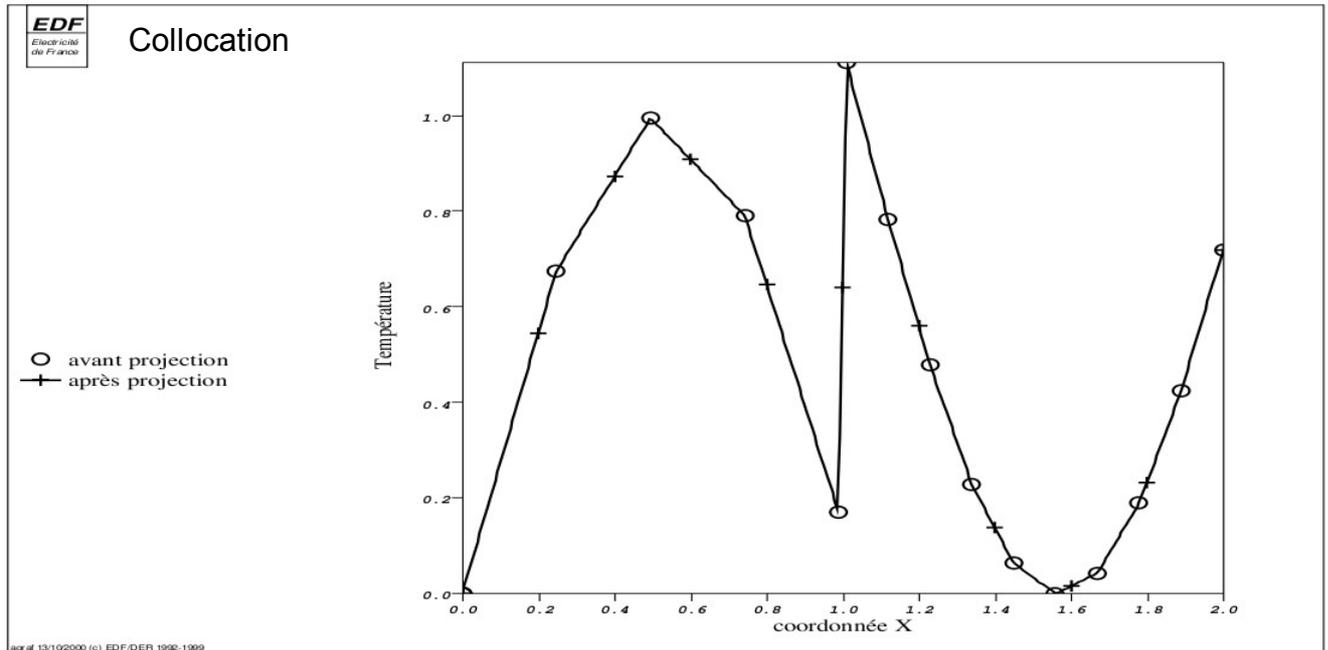
The field to be projected is worth:

$$T(x) = \sin(3x) + \text{Heavyside}(x-1)$$

This field is affected on the nodes of a grid *mal* linear rather coarse (14 meshes) segment $[0,2]$. This field has the property to be discontinuous (in theory) at the point $x=1$. Because of discretization on the grid *mal*, the field only seems to vary very brutally between the 2 points $x=0.99$ and $x=1.01$.

One projects this field on a grid *ma2* very fine of the same segment (300 elements length $2/300$).

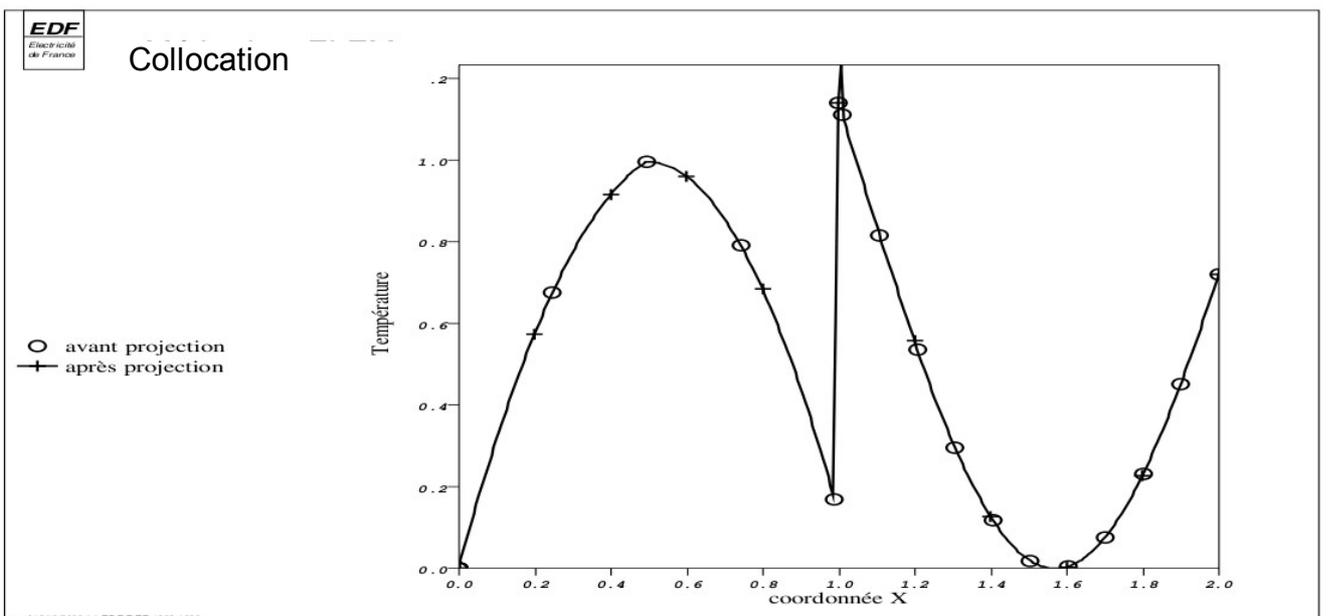
6.1 Method: 'COLLOCATION'



It is noted that with this method, the field project is without surprised: the value obtained by projection the is always interpolated linear one between the 2 nodes of *mal* where the field is known.

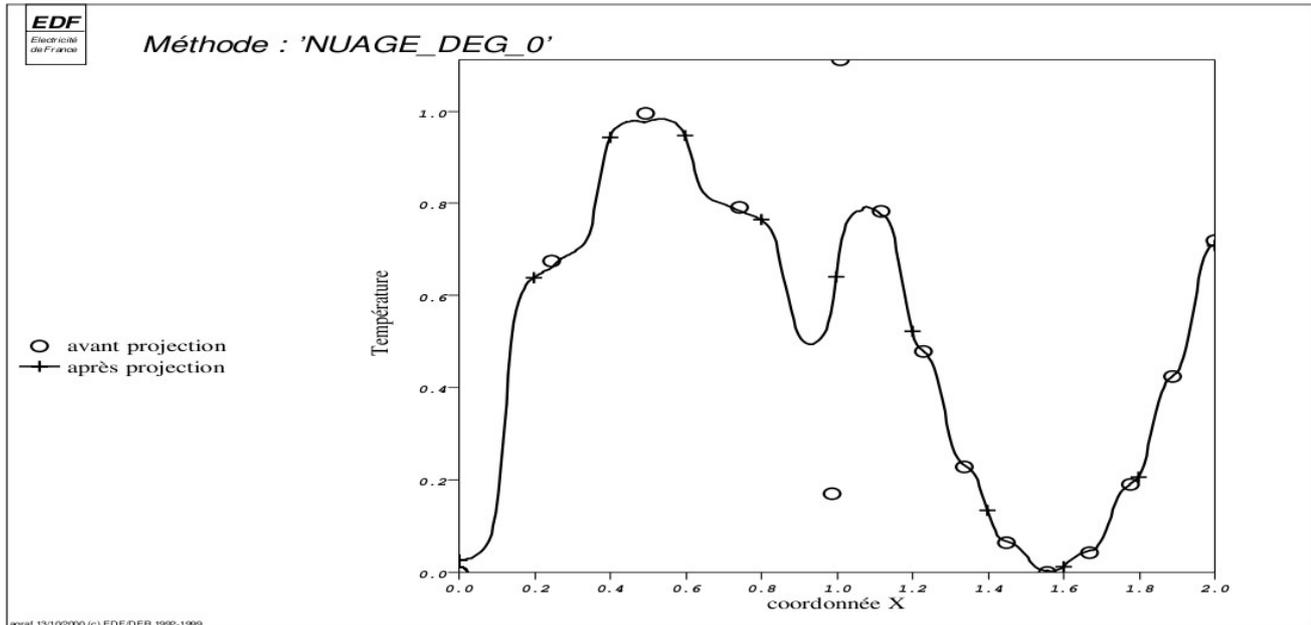
6.1.1 Quadratic initial grid

If one remakes same calculation by replacing the linear grid *mal* by a quadratic grid (containing approximately 2 times less meshes), one finds:



It is noted that the interpolation of the field is now parabolic between the nodes of *mal*. One thus approaches best the form of the initial function.

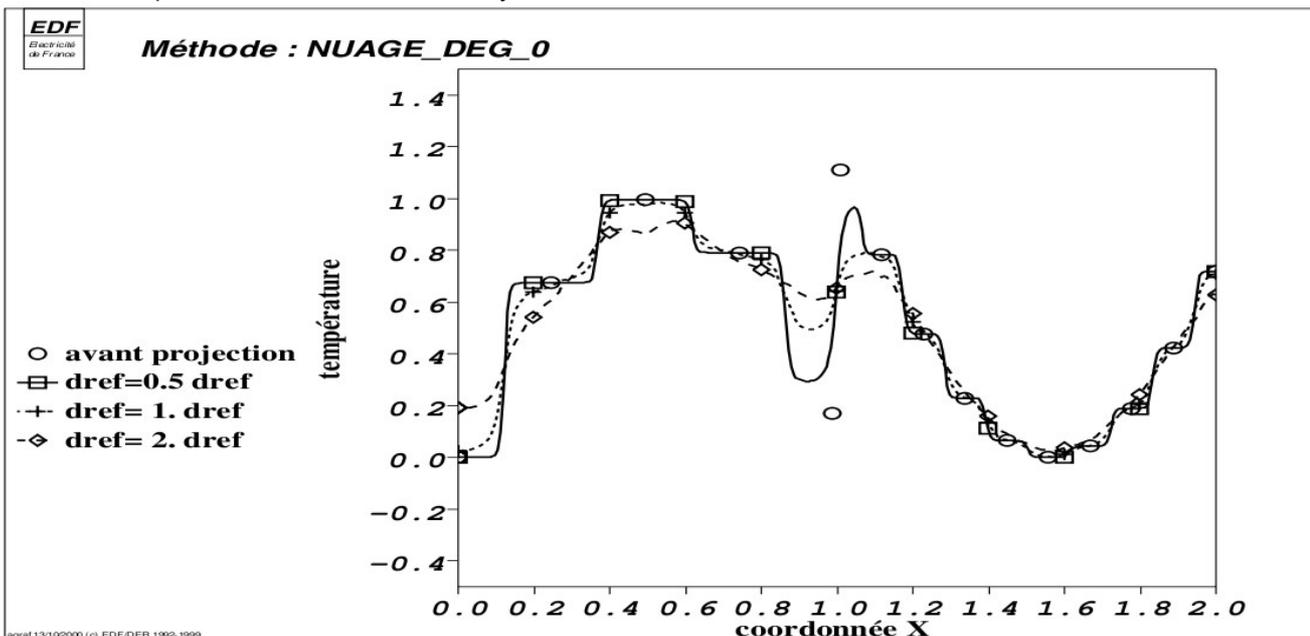
6.2 Method: 'NUAGE_DEG_0'



It is noted that with this method, the field project is presented in the form of a succession of small horizontal "stages" connected to each other. This aspect in staircase is related to the parameters (β and CI) chosen in "hard" in the code for the form of the exponential decay of the weight of the points. It is also seen that the discontinuity of the initial field is very strongly gummed.

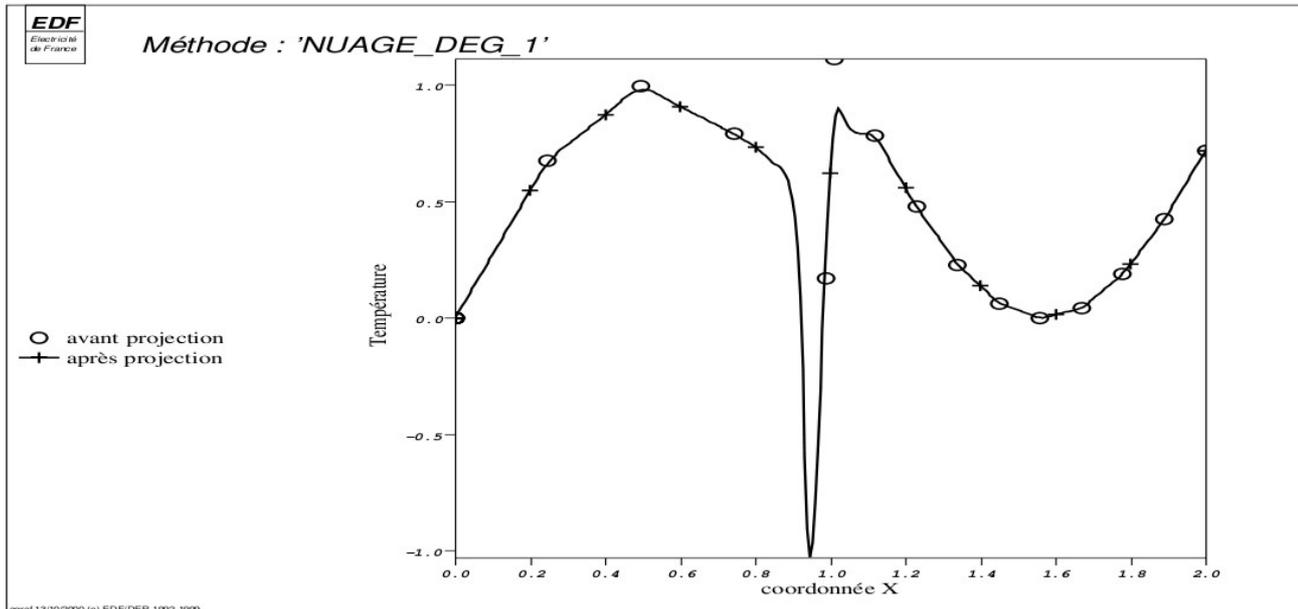
6.2.1 Influence of the form of exponential decreasing

On the following figure, we modified a parameter of exponential decreasing of the method 'NUAGE_DEG_0' : the parameter $dref$ (or CI what returns to same) was multiplied by 0.5 or 2. compared to the value retained by the code.



It is noted that the choice of *dref* influence the result much. If it is too large, one does not see any more discontinuity. If it is too small, the form in stair is accentuated.

6.3 Method: 'NUAGE_DEG_1'



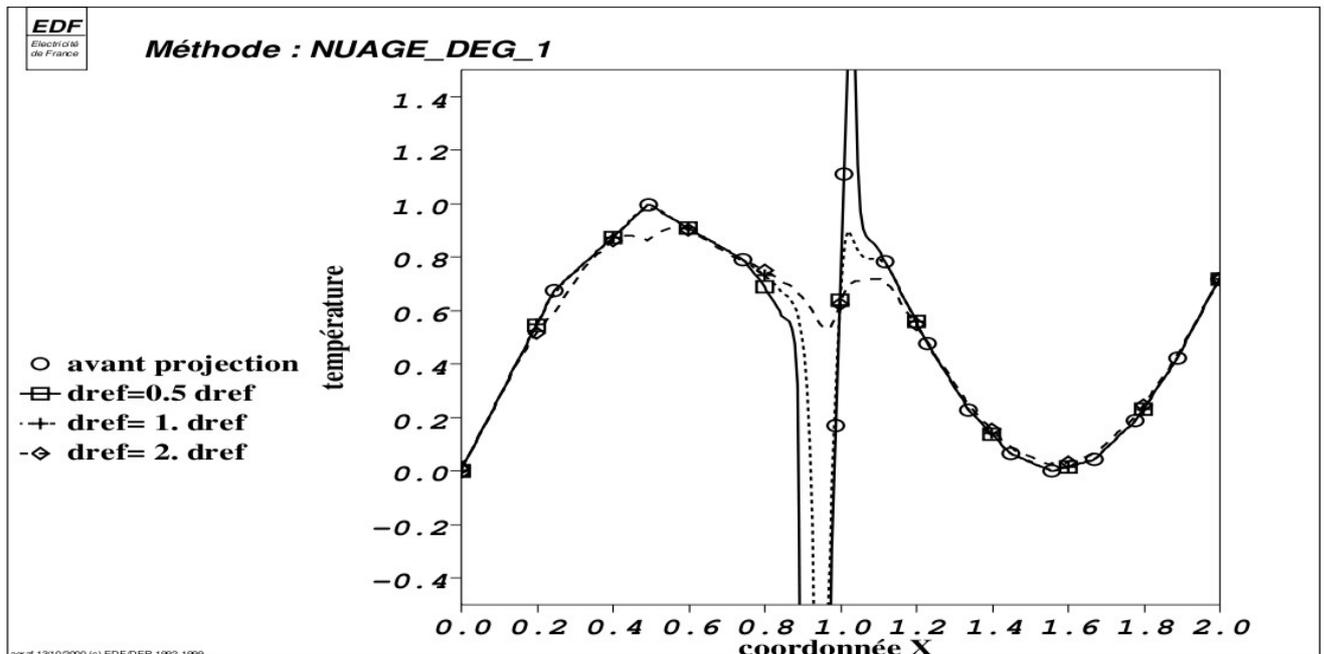
It is noted that with this method, the field project is correct in the 2 regular zones ($x < 0.99$ and $x > 1.01$).

On the other hand the discontinuity of the field between these 2 values is strongly exaggerated. This example illustrates well faculty that with this method to extrapolate the values of the initial points (because of estimate of the gradient of the field).

In the 2 regular parts, the field project is rather close to that obtained with the method 'COLLOCATION'. It is noticed simply that the method 'NUAGE_DEG_1' rounds a little the angles

6.3.1 Influence of the form of exponential decreasing

On the following figure, we modified a parameter of exponential decreasing of the method 'NUAGE_DEG_1' : the parameter *dref* (or *CI* what returns to same) was multiplied by 0.5 or 2. compared to the value retained by the code.



There still, it is noted that the choice of *dref* is crucial for the result: too much large, discontinuity is gummed, too small, discontinuity is exaggerated.

6.4 Choice of the best method of projection for the fields with the nodes

Within sight of some preceding curves, it appears clear that the method 'COLLOCATION' is in general preferable with the methods NUAGE_DEG_0/1. This method is "natural" within the framework of the finite elements and it does not depend on any digital coefficient of adjustment.

Methods NUAGE_DEG_0/1 must be reserved, in our opinion, for a little special uses:

- case of a grid *mal* "non-existent": one has only of the nodes but not the meshes (for example, the "nodes" of *mal* are by way of transmitters),
- case of a field (result of a calculation or obtained by measurements) that one wants to voluntarily smooth. But in this case, it would be necessary that 2 digital parameters (*CI* and β) are accessible to the user what is not the case today.

7 Bibliography

- 1) I. VAUTIER: "Isoparametric elements", Reference material of *Code_Aster* n° [R3.01.00]
- 2) NR. MOM and C. FARHAT: "Matching computations: parallel approach has." *Computers & Structures*, vol. 54 (4), p. 779-785, 1995.