Projection of a field on a Summarized

mesh:

Command PROJ_CHAMP makes it possible “to project” fields (known on a mesh $ma1$) on another mesh $ma2$.

In this document, one describes the various accessible methods of projection in this command. The paragraph [§612] gives some elements of validation of these methods.

Contents

1 General information on command PROJ_CHAMP

command PROJ_CHAMP make it possible to project the fields at nodes or the fields by elements on another mesh.

For the fields at nodes, one has 4 methods:

```
METHODE: "COLLOCATION"
METHODE: "NUAGE_DEG_0"
METHODE: "NUAGE_DEG_1"
METHODE: "COUPLAGE"
```

method "COLLOCATION" uses the shape functions of the elements of the mesh $ma1$. It is detailed in the paragraph [§24].

The 2 other methods use a lissage 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 [§37].

Lastly, method "COUPLAGE" is an alternative of method "COLLOCATION". It is developed especially for fluid-structure coupling with *Code_Saturne* and does not have normally not to thus be used outside this frame.

For the fields by elements "ELNO", one can use method "COLLOCATION". The problem is the same one as for a field at nodes but the field project is always continuous whereas the initial field is discontinuous. Moreover, if a node of the mesh $ma2$ is at the border of 2 elements of the mesh $ma1$, 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 fields "ELNO".

For the fields by elements "ELEM", one can still use 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 fields "ELNO".

For the fields by elements "ELGA", a dedicated method ("ECLA_PG") is described in the paragraph [§410]

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

In the continuation of the document, one will not speak any more a subset D" a mesh, one will make as if L" one projected all the mesh $ma1$ on all the mesh $ma2$.

2 Method "COLLOCATION"

Notices on the vocabulary:

> *The key "to project" is sometimes ambiguous in this document.*
>
> *When one says "to project" the field of $ma1$ worms $ma2$, one seeks the field on $ma2$ knowing that on $ma1$: projection goes from 1 towards 2.*
>
> *For method "COLLOCATION" , it should be found for each node $n2$ of $ma2$ which is the point of $ma1$ which occupies the same position that $n2$, for that one projects the node $n2$ on mesh: $ma1$ projection goes from 2 towards 1.*

2.1 Algorithm implemented

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

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

1) One seeks which is the mesh $m1$ of $ma1$ which "contains" the node N2 `geometrically`,
2) One determines an approximate position of $n2$ in $m1$ (i.e its coordinates in the element of reference associated with the mesh), $m1$ One
3) "refines" the position calculated previously by solving a nonlinear problem to take account owing to the fact that meshes are not linear (the jacobian of the geometrical transformation varies on the mesh). One
4) uses the shape functions of the mesh $m1$ to determine the value of the field on knowing $n2$ the value of the field on the nodes of. $m1$ Notice

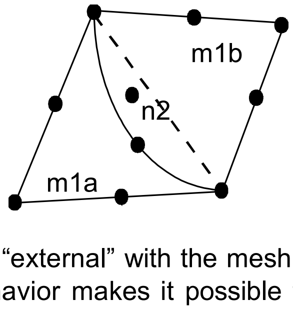
2.1.1

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].7

2.2 The encountered difficulties and their processing Stage

2.2.1 1 One

- does not treat the possible curvature of edges of the elements. For example, in figure Ci - below (plane problem), the node $n2$ will be stated to belong to the mesh whereas $m1a$ 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 is $n2$ actually "external" with the mesh, $ma1$ one will affect the mesh to him $m1$ nearest to him. This behavior makes it possible to project, without stopping in fatal error, a field on a mesh whose border differs slightly from that of the initial mesh (what is always the case in practice). Stages

2.2.2 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 nonlinear 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 nonlinear problem of the precise search of the point on the basis of the approximation previously calculated. Note:

the nonlinear problem is solved in two cases 2D (projection of a 2D mesh on another mesh 2D) or 3D (projection of a mesh 3D on another mesh 3D) but not for the cases 2.5D (shells plunged in space 3D) or 1.5D (telegraphic mesh in 2D), Stage

2.2.3 4 One

always does not use the true shape functions of the elements of the initial mesh. Indeed, in the Code_Aster *, in fact* the finite elements choose their shape functions: a triangle of thermal is not obliged to choose the same shape functions as a triangle of mechanics. An element can also not need shape functions, 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 the programming of PROJ_CHAMP `is` independent of the finite elements present in the model, one assigns to all meshes, $ma1$ the shape functions of the isoparametric elements 1D, 2D or 3D [R3.01.01]. Details

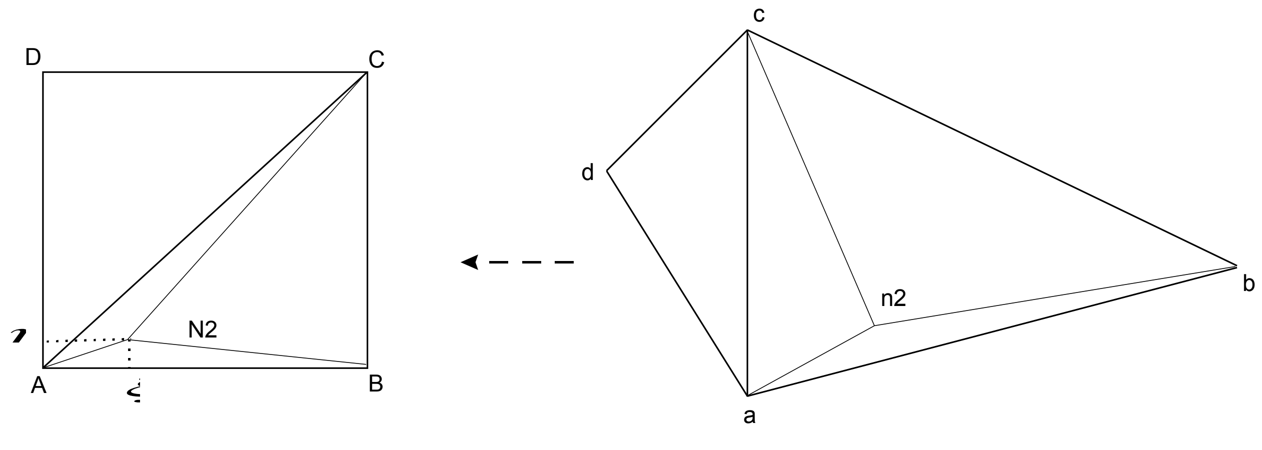
2.3 concerning stages 1 and 2 By means of computer

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

- of a node $n2$ being found inside the border of the mesh (the $ma1$ most frequent case), processing
- of one node outside $n2$ the border of, $ma1$ processing
- of the case of the meshes of type “shell” (surfaces plunged in). $\mathbb{R}^3$ Node

2.3.1 " `interior`" N2

to understand the processing of an interior node, let us take the case of a mesh 2D QUAD `8 ()` . $abcd$ One starts by forgetting his nodes mediums (and thus their possible curvature) then one cuts out it in 2 (et) triangles abc . acd This cutting is arbitrary (it depends on the local classification of the nodes of QUAD `8)`. Let us note that other possible cutting (another diagonal) would in general give another point in the element of reference. belongs



Elément de référence

Elément réel

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

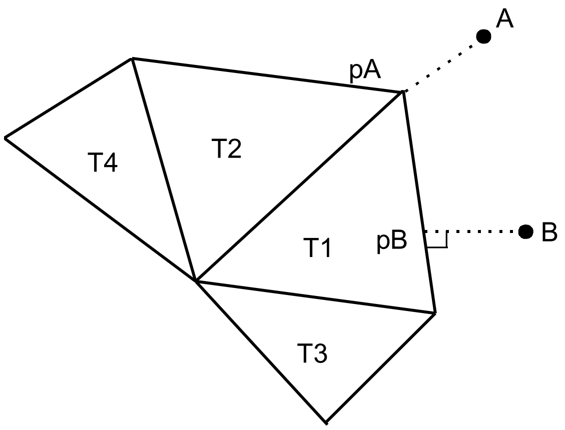
The point N2 `of` the element of reference retained by the algorithm will be: For $N2=xa\times A+xb\times B+xc\times C$

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

2.3.2 " `external`" N2

a node $n2$ is declared external with the mesh if $ma1$ one found no mesh for which it is interior. When this is noted, one searches the mesh $m1$ nearest to. $n2$ The calculated distance is that which separates the node and $n2$ 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 one associates the point obtained pB here by orthogonal projection of on B an edge of. $T1$ With

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

found the point of the mesh $p2$ which $m1$ carries out the minimum of distance with, $n2$ one is reduced to the preceding problem: the point of $N2$ the element of reference which will be associated with will be $n2$ the corresponding one from $p2$ the procedure of the paragraph [§2.3.16Note:

For

> *a triangle (or a tetrahedron) given, T there is only one point carrying out $p2$ the minimal distance enters and $n2$ because of being T T convex. This property disappears if one took account of the curvature of edges of meshes. It is seen there that two simplifications of the implementation (lapse of memory of the nodes mediums and cutting of meshes in linear triangles) are dependant between them.*
>
> *An external point will always have a value interpolated between the values of the nodes of the mesh and $ma1$ ever extrapolated; what is not the case of the 2 methods CLOUD `_DEG_0/1`. Cases*

2.3.3 of the meshes “shell” When

one seeks to project the nodes of a surface mesh on another surface mesh, one fall 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 plane of the triangles of meshes of. $ma1$ It

is thus the procedure of [§2.3.26 applies: For
each the node is outside the field of definition with a right profile of the EXCLU type node: $n2$ search

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

3 “NUAGE_DEG_0 ” or “NUAGE_DEG_1 ” Principle

3.1 of the method These

2 methods are based on the same principle: one chooses a priori elementary functions (here $Fi(x,y,z)$ polynomials of degree 0 or 1). One seeks in the vector space generated by these elementary functions, the function which $F=\sum(\alpha_i F_i)$ carries out the distance minimum (within the meaning of the 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 computations can be made in the same way in dimension 3 or more…). The field to be projected is a set of couples where (Xj,Vj) is $Xj=(xj,yj)$ a node of the mesh and $ma1$ is Vj a reality (value of the field on this node). This field constitutes the cloud of the known points. That is to say

a node of $n2$ (x,y) the mesh for $ma2$ which one wants to calculate the value of the field (). V Choice of the elementary functions: NUAGE_DEG_0

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

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

The sought function is. $F=\alpha 1 F1+\alpha 2 F2+\alpha 3 F3$ One and the defines a kind of “distance” between F couples: (Xj, Vj) where

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

are to them wj the weights assigned to each couple. (Xj, Vj) One

wants to minimize compared to D the 3 variables. $\alpha 1, \alpha 2, \alpha 3$ is D a quadratic function of. $\alpha 1, \alpha 2, \alpha 3$ To minimize come down D to cancel its derivatives and thus to solve a linear system with 3 unknowns: . $\alpha 1, \alpha 2, \alpha 3$ When

this problem is solved, one calculates: Choice of weights

$$V=\alpha 1+\alpha 2 x+\alpha 3 y$$

3.2 of the points of the cloud All

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

- is decided a priori that the weight of a point () Xj depends only on the distance () separating d this point from the node (isotropic $n2$ weighting), if
- is wj a constant, the problem does not depend more of the unknown node. $n2$ The function is F single (for all the nodes of) $ma2$: it is “the linear straight regression line” of the cloud, if
- is $wj(d)$ 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
- is $wj(d)$ a too decreasing function, one takes the chance not to catch any point of the cloud. The numerical consequence is that the linear system to solve becomes singular (and thus insoluble). We

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

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

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

$dref$ the distance from which one wishes to see the weight of the points decreasing quickly. In the programming, $dref$ is calculated like produces it D” a distance by $d1$ a coefficient. $C1$ Aujourd” today, we chose. $C1=0.45$ is

$d1$ defined as follows: in

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

The problem is known as 2D if all the nodes of the mesh have $ma1$ the same coordinate, z it is known as 3D if not. Current

3.3 restriction of the programming

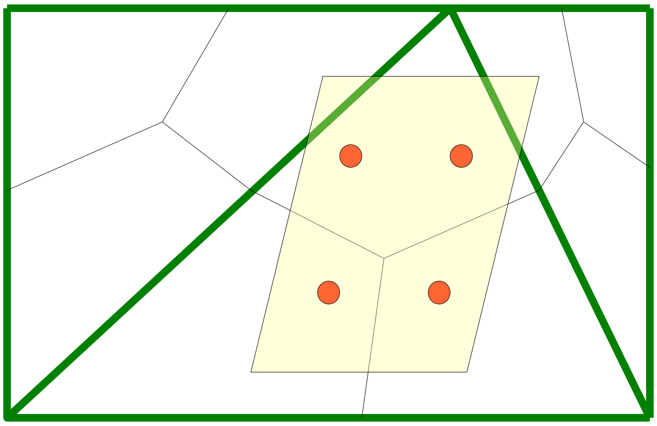
the 2 methods of projection CLOUD _DEG_0/1 are programmed only for the real fields (and not for the complex fields) Method

4 "ECLA_PG" This

method is dedicated to projection of the fields to Gauss points (ELGA). 2D

let us illustrate this method on an example. It
is supposed that the initial mesh consists of 3 triangles (green) having 3 Gauss points (see figure below). Each
triangle is cut out in as many subelements as of Gauss points (here 3). The value carried by the Gauss point is allotted to all the subelement.

A mesh of the final mesh (parallelogram) is drawn in clear yellow. It has four Gauss points (small rounds oranges).
The value of the field project on each of these four Gauss points 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. Method



5 "COUPLAGE" This

method is dedicated to fluid-structure coupling with Code _*Saturne for* the fluid field. It is based on method "COLLOCATION" by taking account of specificities of the problems fluid-structure considered (cf *[feeding-bottle* 2]). One
thus seeks to project data of a fluid mesh on a mesh structure, and conversely. The mesh
structure leans on a discretization EF, therefore with the knowledge of shape functions associated, whereas the fluid mesh is based on a discretization in finished volumes (VF), therefore without shape functions a priori . The kinematical fields on fluid field VF are defined with the nodes tops whereas their duaux fields (forces) are constant by sides and defined in the mediums of the sides. Moreover
, the mesh structure is very generally definitely coarser than the fluid mesh (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 mesh on solid is made by a method of the classical type " COLLOCATION" . It should be noted that this method can be used only to generate data structure stamps correspondence of mesh (CORRESP_2_MAILLA cf *U4.72.05* documentation). This data structure will be then re-used by the operators of exchanges to the interfaces for fluid-structure coupling: ENV

- _CINE_YACS (to send the fields displacement and velocity to the fluid code, cf *documentation* U7.06 .11), MODI
- _CHAR_YACS (to recover the fluid forces and to add with the existing loading structure, cf *U7.06.22* documentation). Apart from

this limitation, the principal difference with method "COLLOCATION " will be in the construction of data structure stamps correspondence (CORRESP_2_MAILLA confer *U4.72.05* documentation) in output 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 forces discretized during the fluid transition towards solid (cf *[feeding-bottle* 2]). Obviously, this result is valid only under particular assumptions: fields

- of constant fluid force by element, fluid
- mesh finer than the solid mesh, method
- of pairing of the type "COLLOCATION " of Code_Aster *.* In practice

, for any problem coupled fluid-structure, one will have to twice use method "COUPLAGE " in order to calculate two data structures of correspondence of mesh. Indeed, one must separately manage the projection of the kinematical fields which leans on definite fluid fields with the nodes tops of the fluid mesh, of that of the forces which are defined in the mediums of the fluid sides. In each one of these two cases, the fluid mesh corresponding will be either the mesh of the nodes tops, or that of the nodes mediums of the sides. The method of projection is then the same one and the mesh structure is, him, single. All the fluid fields (displacements, velocities and forces with the interface) will be regarded by Code_Aster *as* classical fields at nodes. Elements

6 of validation for the projection of the fields at nodes

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

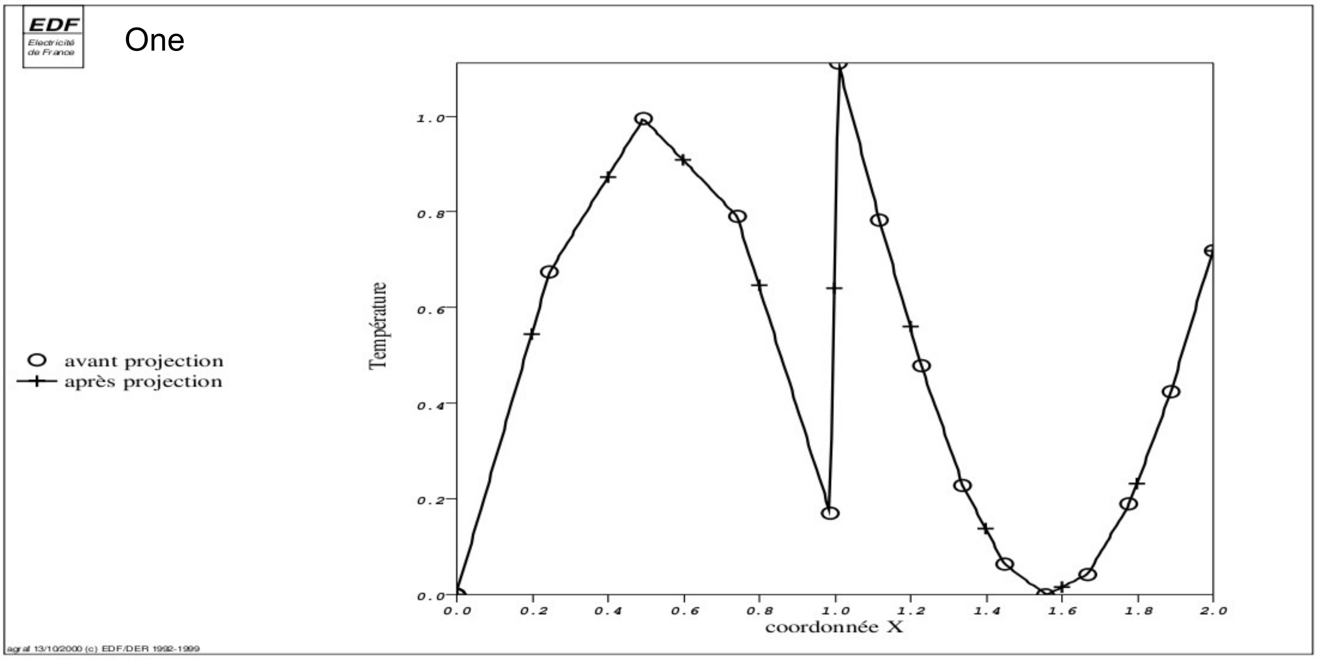
The field to be projected is worth: This

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

field is affected on the nodes of a rather coarse $ma1$ linear mesh (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 mesh, $ma1$ the field only seems to vary very brutally between the 2 points and $x=0.99$. $x=1.01$ One

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

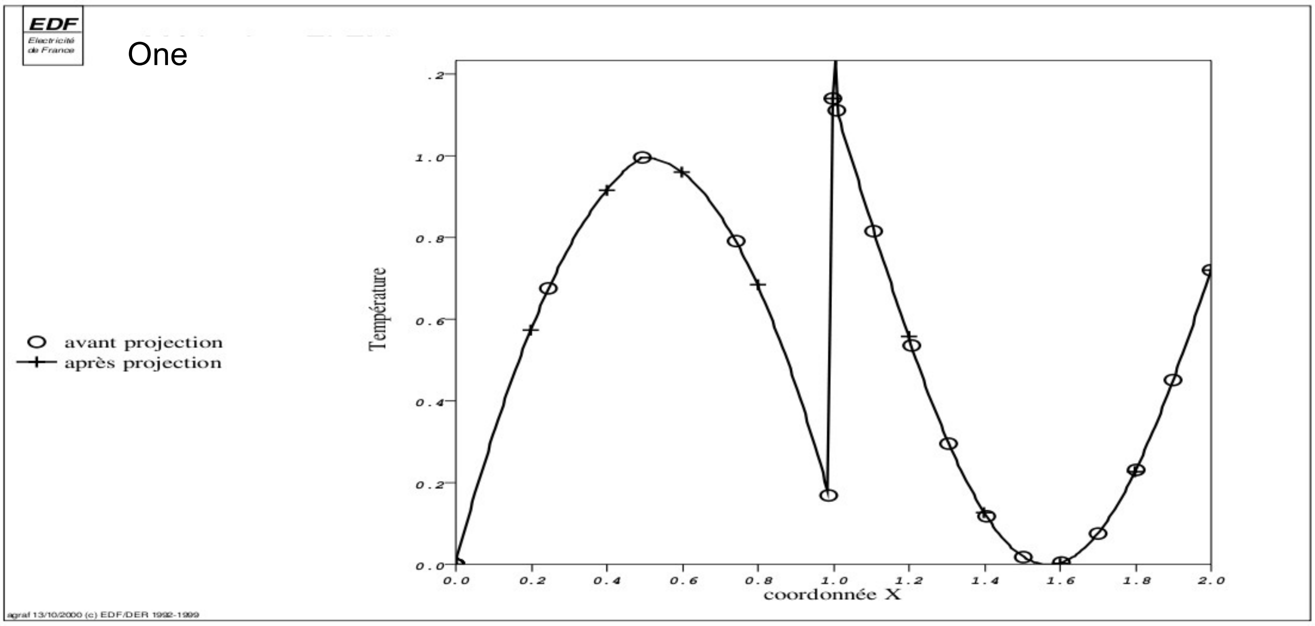
6.1 : “COLLOCATION ″ Collocation



notes 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 where $ma1$ the field is known. Quadratic

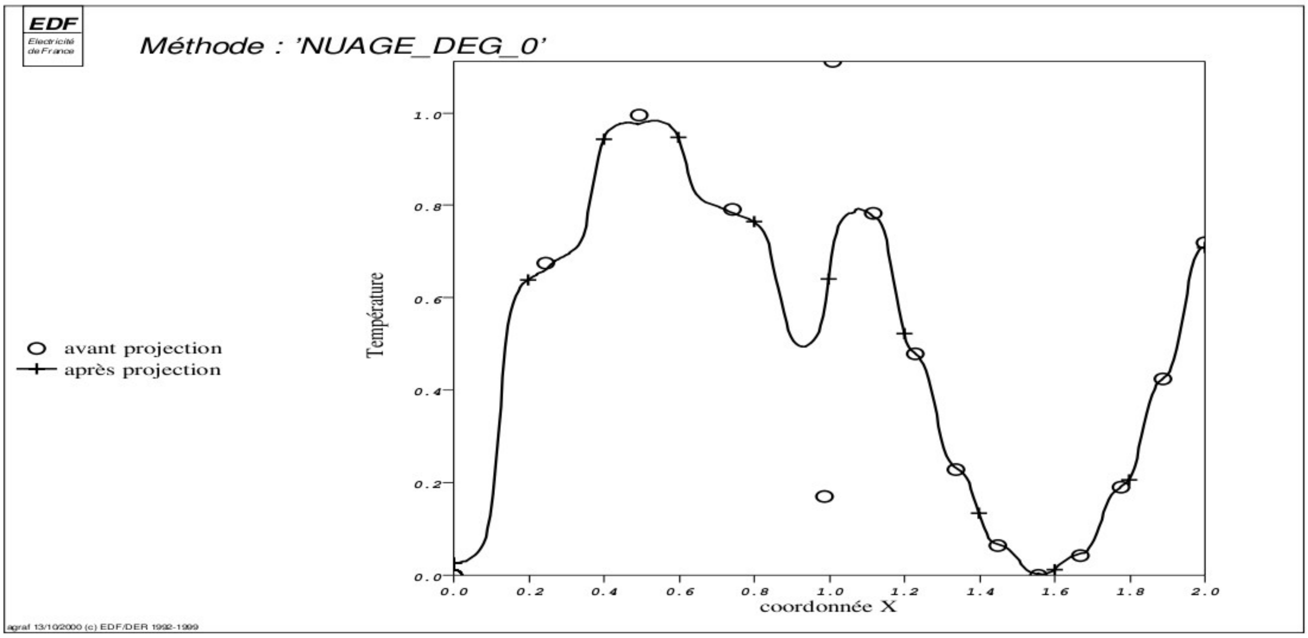
6.1.1 initial mesh If

one remakes same computation by replacing the linear mesh by $ma1$ a quadratic mesh (containing approximately 2 times less meshes), one finds: Collocation



notes that the interpolation of the field is now parabolic between the nodes of. $ma1$ One thus approaches best the form of the initial function. Method

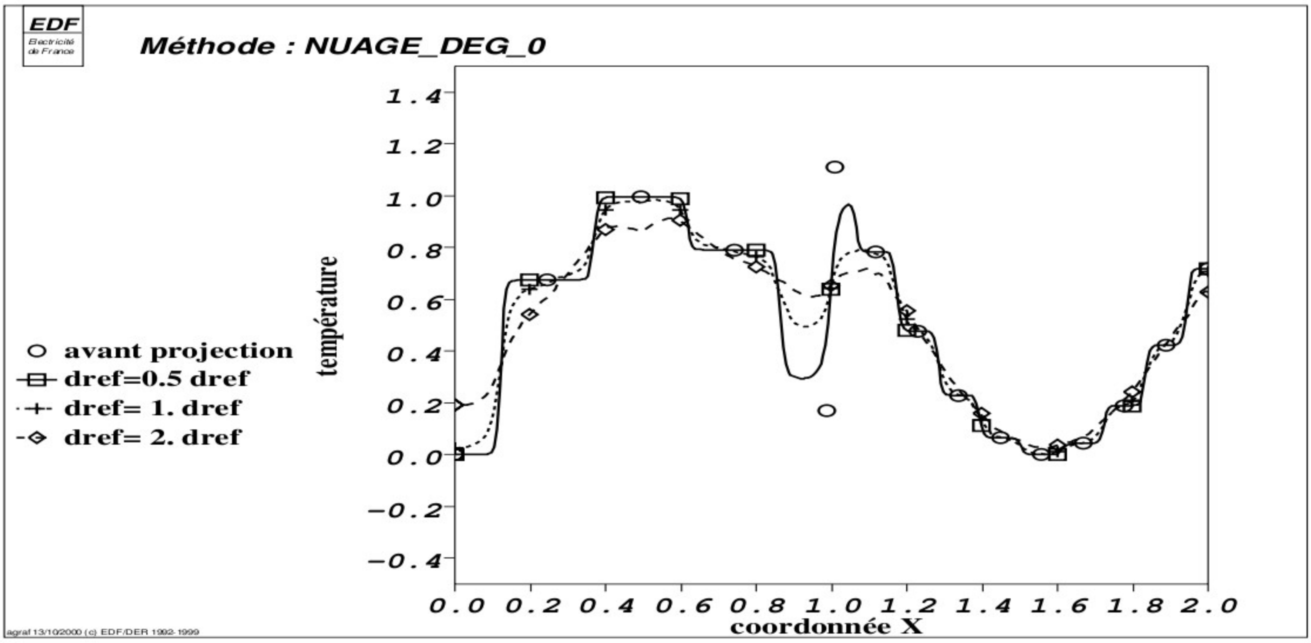
6.2 : “NUAGE_DEG_0 ″ One



notes that with this method, the field project is presented in the form of a succession of small horizontal “bearings” connected to each other. This aspect in staircase is related to the parameters (and β) $C1$ selected in “tough” 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. Influence

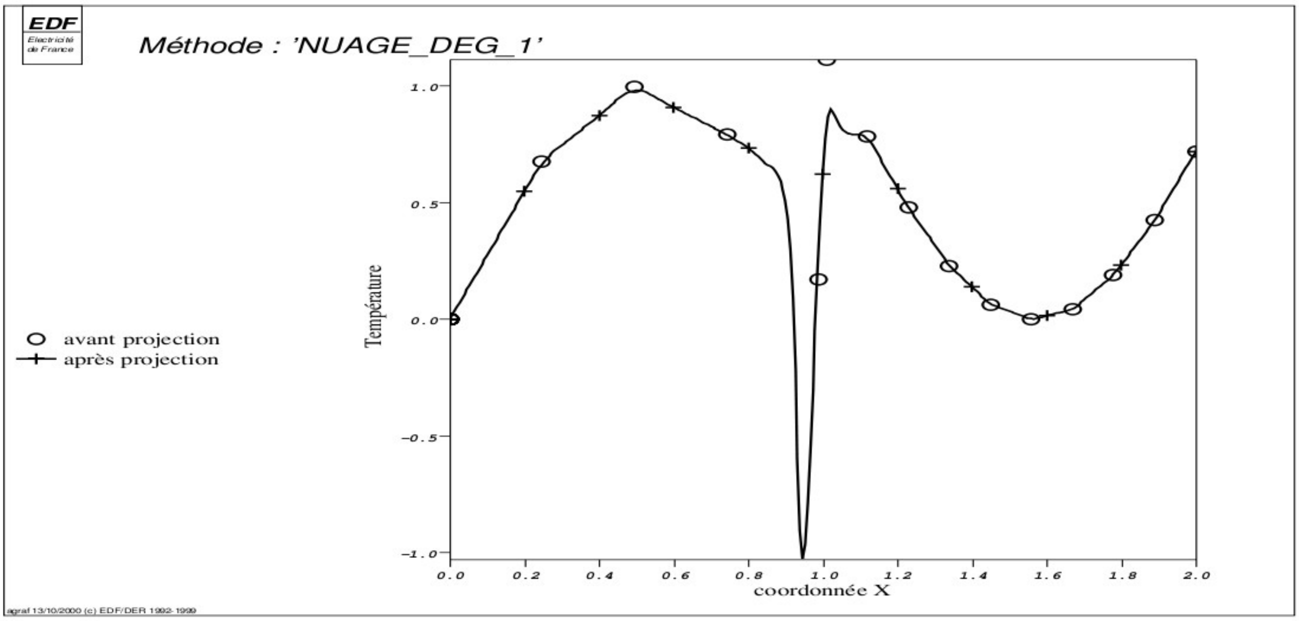
6.2.1 form of exponential decreasing On

the following figure, we modified a parameter of exponential decreasing of method “NUAGE_DEG_0 ″: the parameter (or $dref$ what $C1$ 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 influences $dref$ 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. Method

6.3 : “NUAGE_DEG_1 ″ One

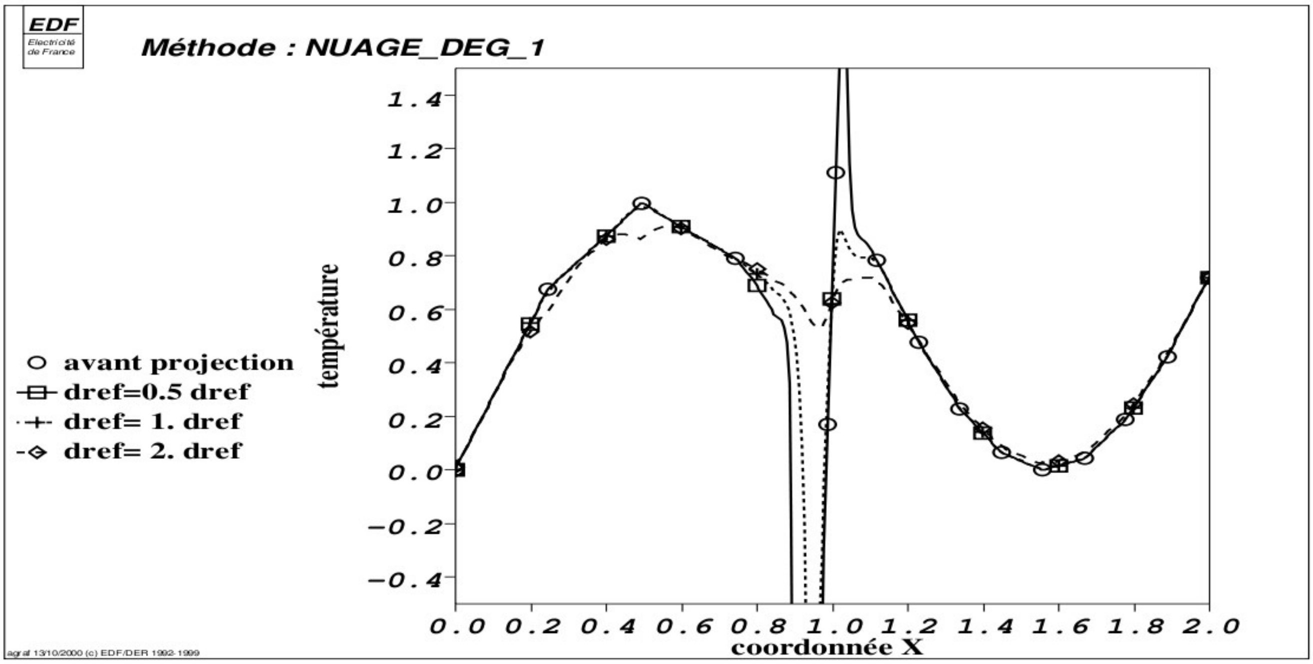


notes that with this method, the field project is correct in the 2 regular zones (and $x<0.99$). $x>1.01$ On the other hand
the discontinuity of the field between these 2 values is strongly exaggerated. This example illustrates well faculty that with this method to extrapolate the values of the initial 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 method “COLLOCATION ″. One notices simply that method “NUAGE_DEG_1 ″ rounds a little the angles Influences

6.3.1 form of exponential decreasing On

the following figure, we modified a parameter of exponential decreasing of method “NUAGE_DEG_1 ″: the parameter (or $dref$ what $C1$ 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 is $dref$ crucial for result: too much large, discontinuity is gummed, too small, discontinuity is exaggerated. Choice

6.4 of the best method of projection for the fields at nodes Within sight of

the few preceding curves, it appears clear that method “COLLOCATION ″ `is` in general preferable with the methods CLOUD `_DEG_0/1. This` method is “natural” in the frame of the finite elements and it does not depend on any numerical coefficient of adjustment.

The methods CLOUD `_DEG_0/1 must` be held, in our opinion, for a little special uses: case

- of a “non-existent” $ma1$ mesh: one has only of the nodes but not meshes (for example, the “nodes” of are $ma1$ by way of transmitters), case
- of a field (result of a computation or obtained by measurements) that one wants to voluntarily smooth. But in this case, it would be necessary that the 2 numerical parameters (and $C1$) β are accessible to the user what is not the case today. Bibliography

7 I.

1) VAUTIER: “Isoparametric Elements”, Documentation of reference of the Code_Aster *n°* [R3.01.00] N.

2) MOM and C. FARHAT: “Matching computations: parallel approach has.” Computers & Structures, vol. 54 (4), p. 779-785, 1995.