BEGINNING WITH CASTEM 2000

31 août 1995

Résumé

This document is a simple and easy way to start using CASTEM 2000 with lessons about GIBIANE. It also contains a complete example of input data for mechanical analysis and one advanced thermal analysis.

CASTEM 2000 is a multi-purpose finite element code, developped at the CEA (French Atomic Research Center). Domains of applications are Structural mechanics, Fluids mechanics, Thermic and Magnetic. CASTEM 2000 is portable on most of the actual hardware, and is free for Universities.

Table des matières

Table des figures

1 GIBIANE OBJECT ORIENTED LANGUAGE

CASTEM2000 works with the help of a an object oriented language. This language is based on the idea that a computation is a succession of independant process communicating by the mean of information strutures called objects. CASTEM2000 is the combination of a language named GIBIANE and a set of objects.

1.1 Generalities on GIBIANE

The data of the problem that we want to resolve are formulated with the help of GIBIANE instructions.

The goal of one instruction is generally to create a particular object made with the help of operators and objects that were previously created.

```
 RESULT     =         OPERATOR             OBJECT 1    OBJECT 2 ...;
result                  name              data needed to build
object              of the operator         the new object
```

- Each instruction is terminated with a ";", it can have 8 maximum lines with 72 characters/line.
- Each operator is coded with 4 characters: OPER has the same meaning that OPERATOR or OPERA or OPERO.
- The maximum number of characters for the name of an object is 8.

As the names of operators contains no numbers, it is recommended to name your objects with a number as the last character. For example:

```
RIGI = RIGIDITE MODEL1 MATER1 ;
```

Creates the rigidity matrix RIGI with the model object MODEL1 and the field of material properties MATER1. If one writes:

```
RIGI = RIGIDITE MODEL1 MATER1;
```

The object RIGI contains now the result of the operation and cannot be used anymore as an operator.

The operators are written in ESOPE which is an evolution of FORTRAN with some aspects of C language (See the chapter "Development on CASTEM 2000"). Some functionality's of CASTEM 2000 are directly written in GIBIANE. These functionality's are called PROCEDURES and can be used with the same manner than the operators. The standard procedures are coded in the file named GIBI.PROCEDURES. One can have it's own procedures coded in the data inputs or in a file named UTILPROC generated with the operator **UTILE.**

1.2 Essential

To begin with CASTEM type CASTEM at the shell prompt The standard input of CASTEM is the file ftn03, when the end of ftn03 is reached, CASTEM reads at the keyboard input. CASTEM duplicates all the commands in the file named ftn98.

As it is possible to share the input and outputs between different executions ofCASTEM, different CASTEM running in the same directories may produce some errors.

If the data's of your problem are already written in the file named toto.dgibi, type

```
"CASTEM toto"
```

or type

```
OPTI DONN 'toto.dgibi';
```

at the CASTEM prompt (these commands copy the file toto.dgibi in ftn03). The extension must be "dgibi".

Control characters in the data input such as TABS may produce errors.

To finish with CASTEM, type:

```
FIN;
```

at the CASTEM prompt. The universal separator for data is a blank character.

You can have some information's on the operator TOTO by executing the following command:

```
INFO TOTO;
```

You can ask for English message and manual by the command

```
OPTION LANG ANGLAIS;
```

All the information's are contained in the file named GIBI.MASTER, with is a direct access file created from a sequential file named gibi.master. One can translate these information's in an other language, add the information in the file at the good place and specifying the used language Italian for example. The user can specify the lang typing:

```
OPTION LANG ITALIAN;
```

You can see the containing of the object TITI typing:

```
LIST TITI;
```

Take care that if TITI is a big object, listing it's containing could take some time. Comments can be typed in the input data's on lines beginning with * at the first character. CASTEM opens only one graphic window, don't kill it, it will stop the program. If the window minds you, put it to the back or iconify it.

TAB. 1 - Main GIBIANE objects

Type of the object	CASTEM 2000 type	Examples of commands
Integer	Entier	I1 = 3; I2 = (3 + 2)/4; I3 = ENTIER 3.1415926;
Real	Flottant	X1 = 1.4142; X2 = (SIN 45)**0.5; X3 = FLOTTANT I3;
List of integers	Listenti	LIST1 = LECT 1 3 PAS 1 10;
List of reals	Listreel	LIST3 = PROG 0. PAS 0.2.; LIST4 = X2 * LIST3;
Discrete function	Evolution	EVOL1 = EVOL MANU 'x' LIST3 'F(x)' LIST4;
Table	Table	TAB1 = TABLE; TAB1.1 = X1; TAB1.'EVOLUTION' = EVOL1;

1.3 Some useful objects

CASTEM 2000 is not only a finite element code and can be used for solving different kind of problems. Some general objects are very useful and can be introduced as parameters of a problem.

1.4 Exercise 1

Title: Use of GIBIANE level 1.

Related operators: +, -, *, SIN, PROG, DIME, EVOL, DESSIN.

Subject: Plot the function $f(x) = x \times sin(3 \times x + 90)$ with $x \in [-360, 360]$ with steps of 5 .

Note: The operators +, -, *, SIN ... are available for complex objects. To plot the evolution EVOL1, type

```
DESSIN EVOL1;
```

1.5 More advanced functions of GIBIANE

1.5.1 Loops

GIBIANE can perform loops using the operator REPETER. We must specify a flag name as an argument. The number of times for the repeat operation may be specified in the command. All the instruction typed between the

```
REPETER FLAG1 ;
```

command and the

```
FIN FLAG1 ;
```

are repeated until the loops ends. The instruction

```
QUITTER FLAG1 ;
```

ends the loop.

1.5.2 Tests

The operator SI and FINSI has the same meaning in GIBIANE than IF and ENDIF in FORTRAN. The argument of SI must be a LOGICAL (LOGIQUE). The following instructions gives the value of factorial 9;

```
N = 9;
RESULT1 = 1;
REPETER FLAG1;
RESULT1 = RESULT1 * N;
SI (EGA 0 (N-1));
QUITTER FLAG1;
FINSI;
N = N-1;
FIN FLAG1;
MESSAGE 'FACT 9 EQUALS TO' RESULT1;
```

1.5.3 Procedures

One can program it's own PROCEDURES using the operator DEBPROC. The procedure may have input arguments that must be specified in the DEBPROC after the name of the procedure. Each argument must be followed with it's type.

1.5.4 Examples

```
DEBPROC HELLO USER1*MOT;
MESSAGE 'Hello user' USER1;
FINPROC;
DEBPROC FACT1 N*ENTIER;
RESULT1 = 1
REPETER FLAG1;
RESULT1 = RESULT1 * N;
SI (EGA 0 (N-1));
QUITTER FLAG1;
FINSI;
N = N-1;
FIN FLAG1;
FINPROC RESULT1;
```

We can now compute the factorial of a number with: RES1 = FACT1 12; for example. Note that a nice example of the factorial procedure using recursivity already exists and can be listed with:

```
LIST FACTORIE;
```

1.6 Exercise 2

Title: Use of GIBIANE level 2.

Related operators: DEBP, SI, REPETER.

Subject: Program a procedure that gives the zeros of a discrete function.

Note:
- The function is assumed to be linear between two discrete points.
- Use the function described in the exercise N 1 to test the procedure.

2 CREATING A GEOMETRY

Before creating a geometry one must specify the dimension of the working space:

```
OPTI DIMENSION N;
```

Where N can be 2 or 3. Two types of objects can represent a geometry in CASTEM.

2.1 The points

A point contains it's coordonates and a density. The density is used for the automatic mesh generation (the size of the elements generated around the specified point is around the density of the point). If the density is not specified, CASTEM use a zero value and the automatic meshing cannot be used. The density can be specified using the DENSITE operator.

Example:

```
OPTI DIME 2;
P1 = 0. 0.;
DENSITE 0.1;
P2 = 1. 0.;
DENSITE 0.2;
P3 = 1. 1.;
DENSITE 0.5;
P4 = 0. 1.;
```

2.2 The complex geometry's (type MAILLAGE)

A MAILLAGE typed object contains at least two points and can contain elements. Before generating elements, one must specify the kind of elements to use (see the information's on the OPTI operator in order to see the available elements). The elements generated for lines (or surfaces in tridimentional problems) are degenerated from the specified element.

2.2.1 Lines

CASTEM can generate straight lines, circles or curved lines. See the operators DROITE, CERCLE, COURBE. The characteristic points of the line must already exist and the densities are used for determining the sizes of the elements.

2.2.2 Surfaces

The operators SURFACE, DALLER, TRANS, ROTA generate irregular or regular surfaces based on their boundaries. These operators can use a field of density.

2.2.3 Volumes

The volumes are generated by the operator VOLUME.

2.3 Operators on geometry's

The operators PLUS and MOINS allows the user to reproduce an existing geometry with a positive or negative vector translation (a vector is the same object than a point). The operator TOURNER rotates an existing geometry. The operator SYMETRIE generates a geometry obtained by symmetry from an other geometry. The operator ET concatenates objects from the same type. One can plot a geometry with the operator TRACER. For tridimentional geometry's, the observation point must be specified.

2.4 Exercise 3

Title: Generating meshes.

Related operators: DROITE, CERCLE, SURFACE, DALLER,

Subject: Generate the following meshes.

- - Helicoid: Program a procedur that builds up an helicoid with given radius, twist, angle and number of elements
- - Square: Create a 1 by 1 square with density of 0.1 and 0.01 at opposite corner
- - Quarter of a disk: see figure 1

2.5 Representation of different fields

Two kinds of fields can be represented on a mesh.

2.5.1 The variables defined at the nodes

These variables are typed in CASTEM 2000 as CHPOINT (pronounce champoint). The classical examples of CHPOINT are loads, displacements, speeds ... etc. .... These fields are generally created when needed. A CHPOINT contains the type of the field, the name of the field, and for the different components, the pointer on the mesh, the values of the components.

2.5.2 The variables defined in the elements

These variables are typed in CASTEM 2000 as MCHAML (pronounce chamelem). Stresses, strains, damage, material parameters are some MCHAML. A MCHAML can be defined at different gauss points or at the nodes, it contains its title, the pointers on the meshes, the name of the components, the type of the components and their values.

As we need shape functions to represent continuous variables in the elements a MCHAML is often associated with an other object typed MMODEL which contains the information's on the shape functions.

The different fields can be mapped with the operator TRACER, with the MCHAML's you must specify the MMODEL. When mapping a MCHAML, the values are extrapolated at the nodes.

The evolution of a CHPOINT along a line can be built with the option CHPO of the operator EVOLUTION and plotted with DESSIN.

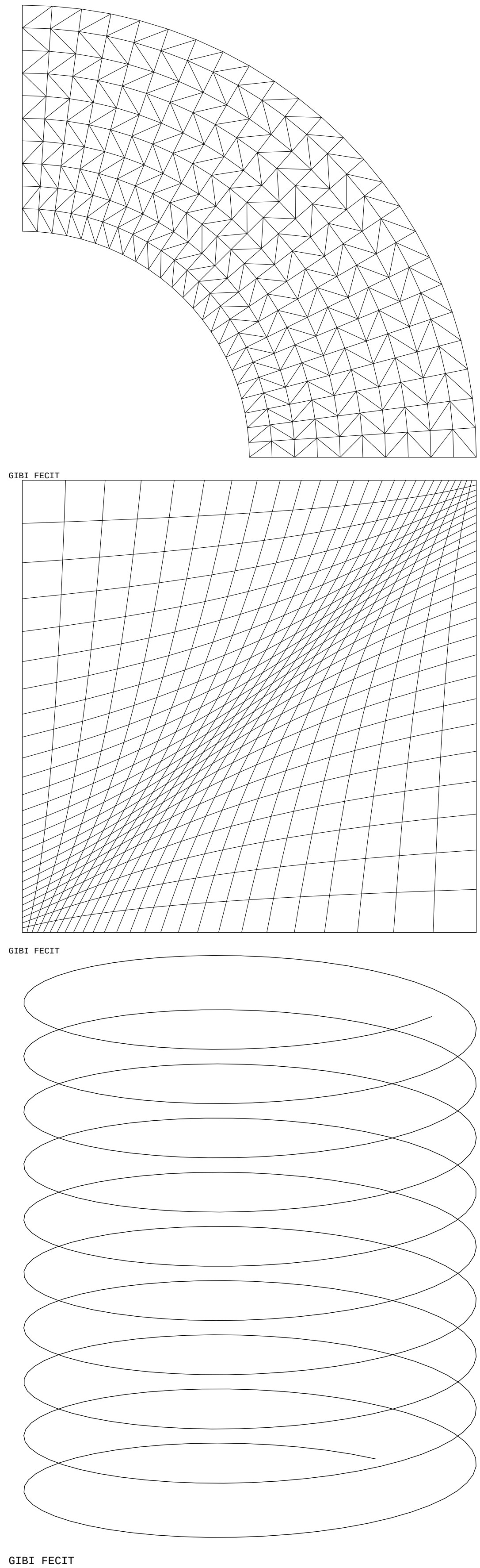


FIG. 1 - *Exercize 3 - Example of meshes*

3 LINEAR PROBLEMS IN MECHANIC

Assuming than the geometry has already been defined, the discretization of the problem is complete when adding the shape functions. The information's on the material model and properties completes the finite element model.

3.1 MMODEL object

The MMODEL object contains the information on the shape functions, the pointer on the mesh, and the mechanical model. A finite element is given by its geometry (specified with OPTI ELEM) and the associated shape functions. For a given geometry some trivial associations can be made. For example, when the geometry is a 3 nodes triangle, the element can be a constant stress triangle (trivial association) or DKT, DST, COQ3 (plate elements). The MMODEL's objects are built with the MODELISER operator, the finite element formulation is needed when not trivial.

3.2 The material properties

The material properties are given in a MCHAML which can be manually built or simply built with the MATER operator. If the structure contains structural elements, the geometrical parameters such as inertia's, thickness or sections must also been specified.

3.3 Exercise 4

Title: Influence of the mesh refinement on the results.

Related operators: MATE MODELISER RIGIDITE MASSE MANU BLOQUER SIGMA.

Subject: Bar subjected to its own weight.(see figure 2) A vertical bar has a length L = lm, a section S = 0.0lm2. The bar is embedded at its upper end and is subjected to its own weight. Mesh the bar with n elements $1 \leq n \leq 10$ and observe the influence of n on results on the stresses.

Note: In order to solve this problem, it is recommended to follow the following scheme:

- Build the geometry.
- Define the model and the material properties.
- Compute the weight by the following manner: Built the mass matrix and multiply it with a constant CHPOINT equal to -g on the vertical axis. For the compatibility with the mass matrix, the CHPOINT must have UX, UY as the names of the components.
- Compute the rigidity matrix.
- Give the boundary conditions.
- Solve the set of equation.
- Compute the stresses.
- Plot the stresses along the ordonate.

The material properties of the bar are:

- Young's modulus : E = 200000 MPa
- Poisson's ratio : $\nu = 0.3$
- Mass density : $\rho = 7800$ Kg/m3

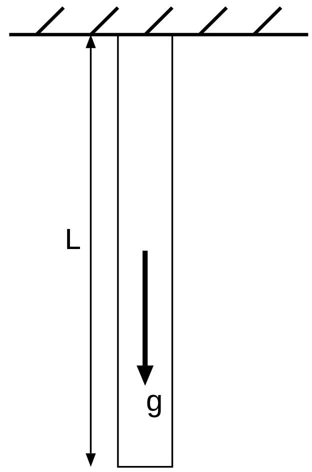


FIG. 2 - *Exercize 4 - Bar subjected to its own weight*

3.4 Exercise 5

Title: Influence of the type of element on the results.

Related operators: MATE MODELISER RIGIDITE MASSE MANU BLOQUER SIGMA.

Subject: Bending of a beam. (see figure 3)

- Resolve this problem with beam elements. (The beam elements are only available in 3D). Compare the deflexion with the beam theory.
- Use QUA4 elements for the same problem. Map the stresses on the deformed mesh. Same questions with a constant distribution of vertical load (see figure 4)

Note:

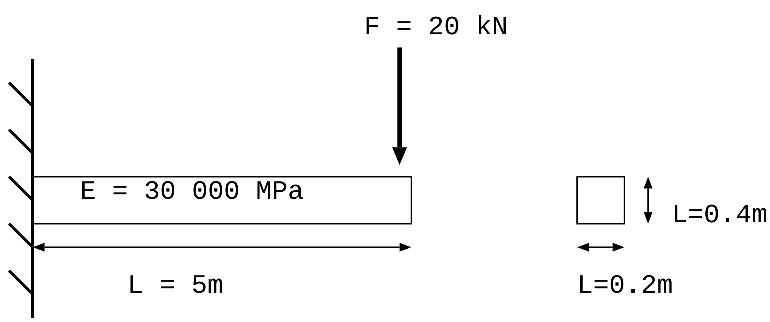


FIG. 3 - *Exercize 5 b - Constant bending of a beam*

3.5 Exercise 6

Title: Elastic problem, use of the symmetries.

Related operators: MATE MODELISER RIGIDITE BLOQUER SIGMA.

Subject: Plate with a hole (see figure 5). Mesh the geometry using the symmetries. Compute the stresses and remesh until the solution is good.

Note: – $a = 1m$

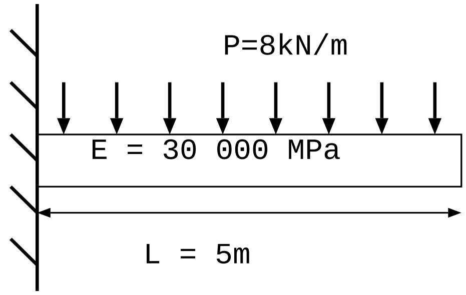


FIG. 4 - *Exercize 5 b - Linear bending of a beam*

- $d = a/10$ et $d = a/2$
- $p = 10kN.m^{-1}$
- $E = 200000MPa$
- $\nu = 0,3$

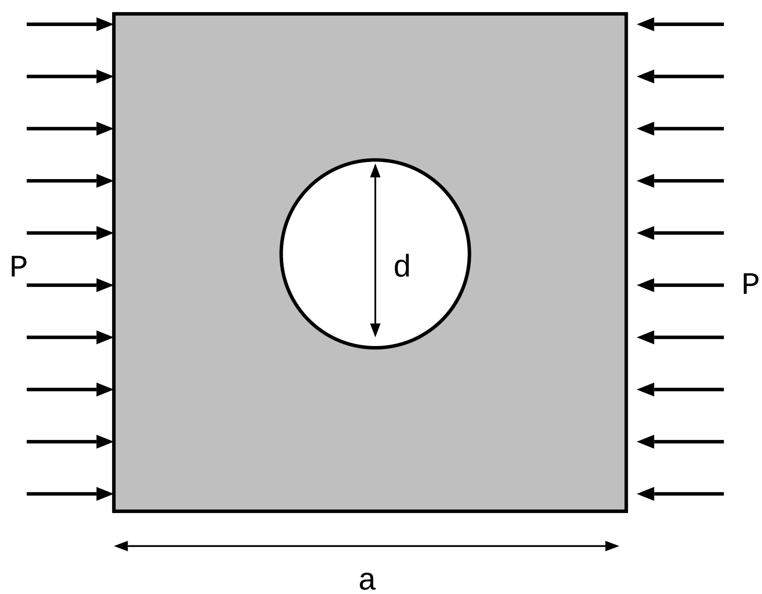


FIG. 5 - *Exercize 6 - Plate with a hole*

4 SOLUTIONS OF THE EXERCISES

4.1 Exercise 1

```
PROG1=PROG -360. PAS 5. 360.;
N1=DIME PROG1;
PROG2=PROG N1 * 90.;
PROG3=PROG1 * (SIN ((3 * PROG1) + PROG2));
EVOL1=EVOL MANU PROG1 PROG3;
DESSINER EVOL1;
LIST PROG3;
```

4.2 Exercise 2

```
DEBP ZERO1 EVOL1*EVOLUTION;
XABS=EXTR EVOL1 ABSC;
YORD=EXTR EVOL1 ORDO;
NBP=DIME XABS;
PROG1=PROG;
IBP=1;
Y1=EXTR YORD IBP;
REPETER BOU1 (NBP - 1);
 IBP=IBP + 1;
 Y2=EXTR YORD IBP;
   SI ((Y2 * Y1) < 0.);
     X1=EXTR XABS (IBP - 1);
     X2=EXTR XABS IBP;
     A=(Y2 - Y1) / (X2 - X1);
     B=Y1 - (A * X1);
     XZERO=-1. * B / A;
     OO=PROG XZERO;
     PROG1=PROG1 ET OO;
   FINSI;
   Y1 = Y2;
FIN BOU1;
N1 = DIME PROG1;
MESSAGE 'ON A TROUVE' N1'RACINES';
FINPROC PROG1;
PROG1=PROG -360. PAS 5. 360.;
N1=DIME PROG1;
PROG2=PROG N1 * 90.;
PROG3=PROG1 * (SIN ( (3 * PROG1) + PROG2) );
EVOL1=EVOL MANU PROG1 PROG3;
PZERO = ZERO1 EVOL1;
LIST PZERO;
DESS EVOL1;
```

4.3 Exercise 3

```
* QUART DE DISQUE TROUE
```

```
OPTI DIME 2 ELEM TRI3;
DENSITE 0.1;
P1=2. 0.;
P2=3. 0.;
D1=DROI P1 P2;
SURF1=D1 ROTA (1. 0.) 90.;
TRAC SURF1;
*CARRE DALLAGE IRREGULIER
OPTI ELEM QUA4;
DENSITE 0.01;
PP1=0. 0.;
DENSITE 0.1;
PP2=1. 0.;
DENSITE 0.01;
PP3=1. 1.;
DENSITE 0.1;
PP4=0. 1.;
DD1=DROI PP1 PP2;
DD2=DROI PP2 PP3;
N1=NBELE DD1;
N2=NBELE DD2;
DD3=DROI (-1 * N1) PP3 PP4;
DD4=DROI (-1 * N2) PP4 PP1;
SURF2=DALLER DD1 DD2 DD3 DD4;
TRAC SURF2;
TRAC (SURF1 ET SURF2);
*
* HELICOID
*
DEBP HELICE RAY1*FLOTTANT PAS1*FLOTTANT TETA1*ENTIER
N1*ENTIER;
DTETA=TETA1/N1;
DENSITE 1.;
XTETA=0.;
DPAS=DTETA * PAS1 / 360.;
XPAS=0.;
P1= (RAY1 * (COS XTETA)) (RAY1 * (SIN XTETA)) XPAS;
I=0;
REPETER BOU1 (N1 + 1);
I=I + 1;
XTETA = XTETA + DTETA;
XPAS = XPAS + DPAS;
P2 = (RAY1 * (COS XTETA)) (RAY1 * (SIN XTETA)) XPAS;
D1 = D 1 P1 P2;
SI (EGA I 1);
RESU1=D1;
SINON;
RESU1=RESU1 ET D1;
FINSI;
P1=P2;
```

```
FIN BOU1;
FINPROC RESU1;
OPTI DIME 3 ELEM SEG2;
HEL1=HELICE 2. 0.5 3600 500;
OEIL=-3000 -2000 1000;
TRAC OEIL HEL1;
```

4.4 Exercise 4

This procedure can be used for plotting the stresses.

```
DEBPROC TRACCHAM GEO1*MAILLAGE MOD1*MMODEL
                CHAM1*MCHAML;
PROG1=PROG;
PROG2=PROG;
NBN1=NBELE GEO1;
I=0;
X1=COORD 2 GEO1;
REPETER BOU1 NBN1;
I=I + 1;
ELE1=ELEM GEO1 I;
P1=ELE1 POINT 1;
P2=ELE1 POINT 2;
NB2=ELE1 NBNO;
X2=COORD 2 ELE1;
X1=COORD 1 ELE1;
X2=X1 + X2;
XX1=EXTR X2 SCAL P1;
XX2=EXTR X2 SCAL P2;
I1 = (2 * (I - 1) ) + 1;
I2=I1 + 1;
LIST I1 I2;
PROG1=INSE PROG1 I1 XX1;
PROG1=INSE PROG1 I2 XX2;
SIG1=EXTR CHAM1 EFFX 1 I 1;
PROG2=INSE PROG2 I1 SIG1;
PROG2=INSE PROG2 I2 SIG1;
FIN BOU1;
EVOL1=EVOL MANU PROG1 PROG2;
FINPROC EVOL1;
```

5 EXAMPLE OF THERMAL COMPUTATION

To illustrate the potential of CASTEM 2000, we present an example of thermic computation.

The gist of this computation is to determine the temperature distribution in a cooking pan as a permanent solution and as a time-varying solution.

The geometry of the cooking pan is presented on figure 6.

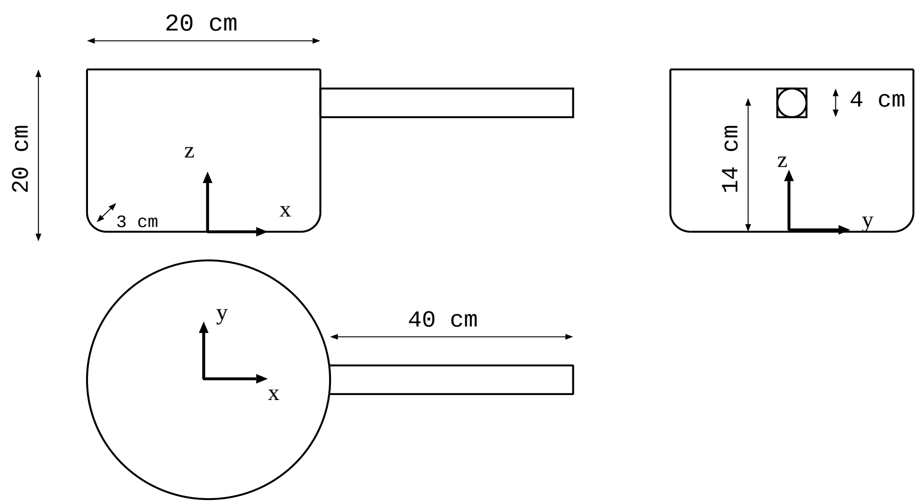


FIG. 6 - *Geometry of the structure*

5.1 Models

5.1.1 Steady problem

In the stationnary problem, the temperature is prescribed on the bottom of the pan. The temperature of the water inside the pan is uniform, its value is 100 Celsius degres. A forced convective model is used to modelize heat exchange between the water and the pan and between the pan and the surrounding air. Inside the pan heat transfer is governed by diffusion.

This model can be summerized by the following equations:

- The rate of heat flow ($W.m^2$)

$$j = k.\text{grad}T \quad \text{inside the cooking pan} \tag{1}$$
$$j = h.(T_{air} - T) \quad \text{where } T_{air} = 20 \text{ on the external surface} \tag{2}$$
$$j = h.(T_{water} - T) \quad \text{where } T_{water} = 100 \text{ on the internal surface} \tag{3}$$

  The coefficient k and h are assumed to be constant.

- Prescribed temperature at the bottom (Celsius)

$$T = T_{imposed} = 200 \text{ Celsius} \tag{4}$$

- The condition of equilibrium is

$$\text{div}j = 0 \quad \text{inside the pan, and} \quad j = 0 \text{ on the surface} \tag{5}$$

5.1.2 Transient problem

We are now interested by the evolution of the temperature with time. Instead of a prescribed temperature at the bottom we impose a given rate of heat flow on the surface. Diffusive and convective heat flow are modelized the same way. As the temperature of the pan and the temperature of the water change with time we need to introduce heat capacity. We assume that the temperature of the water is uniform. Heat capacity is defined as the derivative of enthalpy with respect to temperature.

$$c = \frac{\partial H}{\partial T} \tag{6}$$

c is assumed to be constant.

The following equation summerize our model:

- Rate of heat flow:

$$\begin{aligned} j &= k.\text{grad}T \quad \text{inside the cooking pan} && (7)\\ j &= h.(T_{air} - T) \quad \text{where } T_{air} = 20 \quad \text{on the external surface} && (8)\\ j &= h.(T_{water} - T) \quad \text{where } T_{water} \text{ is an unknown} && (9)\\ j &= \text{constant} \quad \text{on the bottom} && (10) \end{aligned}$$

- Heat capacity

$$\begin{aligned} \frac{\partial H_{pan}}{\partial T} &= c \quad \text{in } J.m^{-3}.C^{-1} \quad \text{in the pan} && (11)\\ \frac{\partial H_{water}}{\partial T_{water}} &= c_{water} \quad \text{in } J..C^{-1} && (12) \end{aligned}$$

- The energy equation gives

$$\begin{aligned} \text{For the pan} \quad \frac{\partial H}{\partial t} &= \text{div}j \quad \text{inside the pan, and } j = 0 \text{ on the surface} && (13)\\ \text{For the water} \quad \frac{\partial H}{\partial t} &= \int_{Surface} j.\partial S && (14) \end{aligned}$$

5.2 Finite Element formulation

CASTEM 2000 offers all the operations required to solve this problem.

5.2.1 Finite elment

We use isoparametric triangular and quadrangular linear shell elements for the cylinder and isoparametric massive cubic bilinear elements for the handle. For the massive element at each node corresponds one degree of freedom the value of T at this node. The value of T inside the element is as a linear combination of the interpolation function. For the shell element at each node there are three degrees of freedom: the temperature on the lower surface 'TINF' the temperature in the middle 'T' of the shell and the temperature on the upper surface 'TSUP'. Therefore temperature distribution is quadratic in the thickness and linear in the plane of the element.

5.2.2 Conduction Matrix

The conduction matrix is defined as

$$K_{i,j} = \int_{\Omega} \mathrm{grad}N_i \, . \, k \, . \, \mathrm{grad}N_j \, \partial\Omega \tag{15}$$

where N_i is the interpolation function for the i th degree of freedom and Ω is the domain defined by the mesh.

5.2.3 Convection matrix

The convection matrix is defined as

$$H_{i,j} = \int_S N_i \, . \, h \, . \, N_j \, \partial S \tag{16}$$

S is the surface of the domain where the forced convection model applies.

5.2.4 Capacity matrix

The capacity matrix is defined as

$$C_{i,j} = \int_S N_i \, . \, c \, . \, N_j \, \partial S \tag{17}$$

5.3 Meshing the pan

The mesh is generated with the help of GIBIANE. During this process one should remind that each part that bear a model or a boundary condition must be named. The GIBIANE file is provided in 6 and all the name we will refer to are in accordance to that file. To summarize the mesh is obtained in several steps.

* The main lateral part of the cylinder SU2 is obtained from the rotation of one line LI1 about a central axis Oz.
* The bottom part of the pan SU3 is obtained with the automatic meshing operator 'SURF'. The contour of this surface is obtained from the surface SU2 and with the help of the 'COTE' operator.
* For the handle, we start by determining the intersection BASE1 between the handle ('ELEM' operator) and the cylinder then we construct the handle HANDLE1 itself by translating this surface with 'TRAN' operator. At that point the handle is a parallepipedic volume. We shape it using a coordinate transform ('DEPL' operator). In the cylindrical coordinate system a point M of coordinate (r, θ) or (x, y) is transfom into $M\prime$ in accordance with

$$\overrightarrow{OM\prime} = \alpha(z)\left(\vec{OM} \times \frac{Max(|x|,|y|)}{r}\right) + (1 - \alpha(z)) \, . \, \overrightarrow{OM}$$

The external surface of the handle SU4 is obtained with 'ENVE' and 'DIFF' .

Some parameters have been defined which determine the refinement of the mesh. The resulting mesh is presented in figure 7.

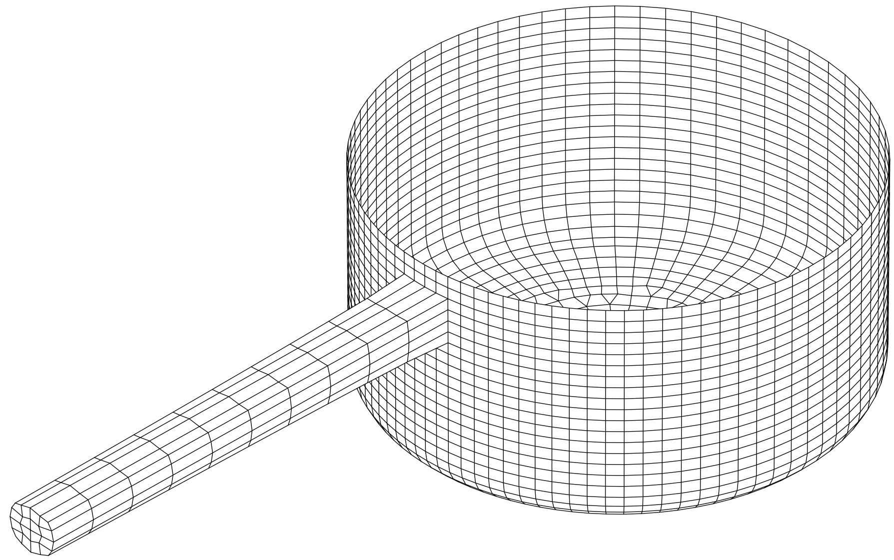


FIG. 7 - *Mesh of the cooking pan*

5.4 MMODEL and Material properties

In order to perform a computation we need to define models of computation an related material properties. The models are stored in 'MMODEL' objects: namely the geometrical mesh , the type of formulation ('THERMIQUE' or 'CONVECTION') and the type of finite element(shape function). The 'MODE' operator is used to define MMODEL object.

We define six different models:

- one for the diffusion in the pan: MOD1
- one for the diffusion in the handle: MOD2
- one for the convection on the surface of the handle: MOD3
- one for the convection on the internal surface of the cylinder: MOD4
- one for the convection on the external surface of the cylinder MOD5
- one for the convection on the upper surface of the bottom: MOD6

The material properties are defined using the 'MATE' operator. The properties are stored in field by element and each property is referenced by a key word , for instance 'H' for the coefficient of convection. In addition to material properties we need to define geometrical properties for the shell elements, namley the thickness 'EPAI'. In accordance with the MMODEL object we define 6 material properties field (MCHAML type object) MAT*.

5.5 Matrices generation

The capacity matrices are defined with the help of the 'CAPA' and 'CONDU' operators. Arguments of thoses operators are the model and the material properties. We store the matrices in the RIGIDITE (stiffness in French) type object named CAP* and COND*

5.6 Second member related to convection

Convection model is used to represent a rate of heat flux of the form

$$j = h.(T_{ext} - T)$$

where T_{ext} is a given temperature The finite element formulation is

$$Q = [H].(T_{ext} - T)$$

where Q is the equivalent nodal heat sources an T or T_{ext} are nodal value of temperature. In order to use this model in the determination of T we need to have the product

$$[H].T_{ext}$$

This product is stored in a field by point (CHPOINT type object) and is obtained with the 'CONV' operator. In the file of command we stored this second member in objects named FLX* .

5.6.1 Prescribed boundary condition

In order to prescribe temperature at given node of the mesh CASTEM 2000 uses langrangian multipliers. In other words, new unknowns are introduced: the equivalent nodal sources required to insure the prescribed temperature.

The problem can be presented as follow: we look for T the vector of nodal value of temperature such that

$$K.T = Q + q_i \qquad \text{avec } t_i = \overline{t_i} \tag{18}$$

K is (nxn) matrice and q_i is a new unknown, the nodal source at node i required to insure $t_i = \overline{t_i}$.

We call this new unknown λ_i and the preceding systee is rewrittten as

$$\begin{bmatrix} K & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} T \\ \lambda \end{bmatrix} = \begin{bmatrix} Q \\ \overline{t_i} \end{bmatrix} \tag{19}$$

Therefore prescribing temperature at node i requires to create the corresponding matrix and the corresponding second member.

This method makes the change of prescribed boundary condition easy to deal with, as it will appears in the file.

In our case we want to impose temperature on the bottom part of the pan. This part correspond to the surface SU3 and the unknown 'TINF'. We use the 'BLOQ' operator to create the matrix named BL1 and the 'DEPI' operator to create the second member named TIMP3 stored in 'CHPOINT' type object.

5.7 Solving the stationnary problem

The variational problem is

$$\forall w \qquad \int_\Omega w \,.\, \text{div} j(T) \, \partial\Omega + \int_S w \,.\, j(T).n \, \partial S \tag{20}$$

The discretized variational formulation of the problem is now under the form

$$[K].T = Q \tag{21}$$

The matrix $[K]$ is obtained by assembling the conduction, convection and langrangian multipliers associated matrix. The 'ET' (and in French) operator is used to assembly matrices and second members of the equation. The 'RESO' operator is used to solve the linear system. The result is a field by point of temperature on the surface.

The field of temperature can be visualized with the 'TRAC' operator. Since we use shell elements we must select (with the 'EXCO' operator) one of the three value of temperature defined at each node. We selected the value of 'T' which is the middle value. The result is presented on figure 8.

We are also interested by the heat exchanged with the water and the heat provided at the bottom. The heat provided is obtained with the 'REAC' operator, the langrangian multiplier associated matrix and the solution of the 'RESO' operator which contains the langrangian multipliers. With the help of 'REAC' we obtain a field of equivalent nodal sources. The 'RESU' operator sums up all this value in order to get the total value of heat exchanged.

The rate of heat exchanged with the water is obtained by computing

$$Q = [H].(T_{ext} - T)$$

using the corresponding matrices of convection. The '*' operator is a multiplication of a field by point by a matrix. The result is a field by point of equivalent nodal sources. 'RESU' is used the same way to get the total rate.

At last the temperature at the end of the handle is extracted from the solution of the problem with the 'EXTR' operator.

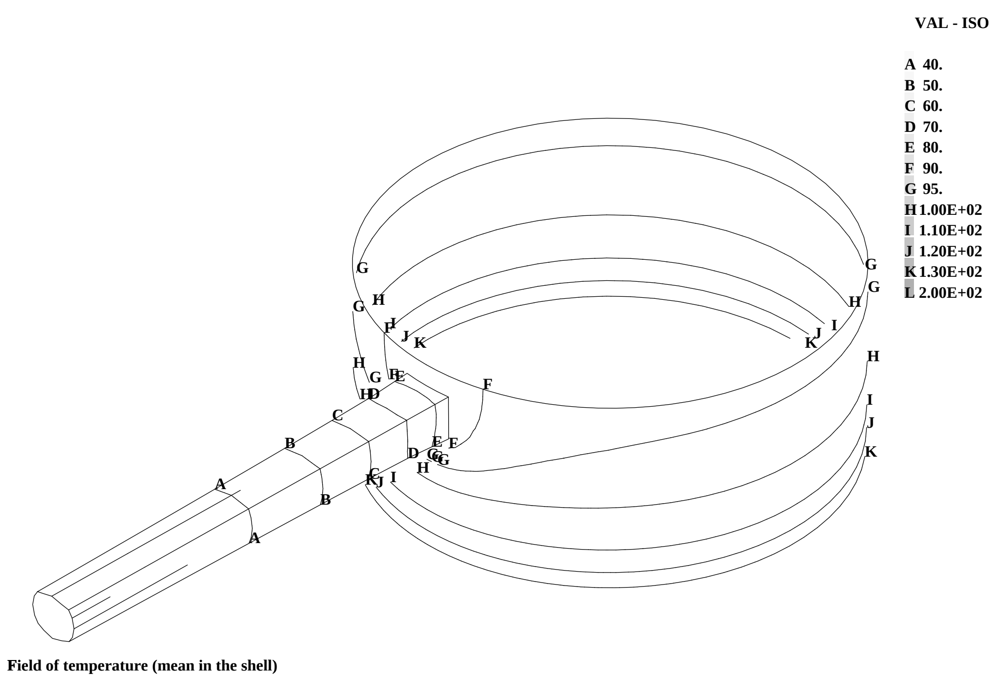


FIG. 8 - *Temperature distribution*

5.8 Solving the unsteady problem

Starting with a uniform temperature of the pan and the water of 20 C. We prescribe a given rate of heat flux on the lower surface ('FLUX' operator). In ordrer to find the temperature evolution we need to perform a time discretization.

5.8.1 Time discretization

The space discretized problem is under the form

$$\left\{ \begin{array}{l} [C].\dot{T} + [K].T = Q(t) \\ c_{water}.\dot{T}_{water} = q_{exchanged} \end{array} \right. \tag{22}$$

θ is the parameter for the theta-method of time discretization. If T^n is the temperature distribution at time t_n equation (22) becomes

$$\left\{ \begin{array}{c} (C + \Delta t.\theta\, .\, K)\, .\Delta T = \Delta t\, .\, (-K.T^n + Q^n) \\ t^{n+1}_{water} = t^n_{water} + \sum_i \frac{\Delta t}{c_{water}} \, .\, [H]\, .\, \left(\frac{T^{n+1}+T^n}{2} - T_{water} \right)_i \end{array} \right. \tag{23}$$

5.8.2 Loop over the time step in GIBIANE

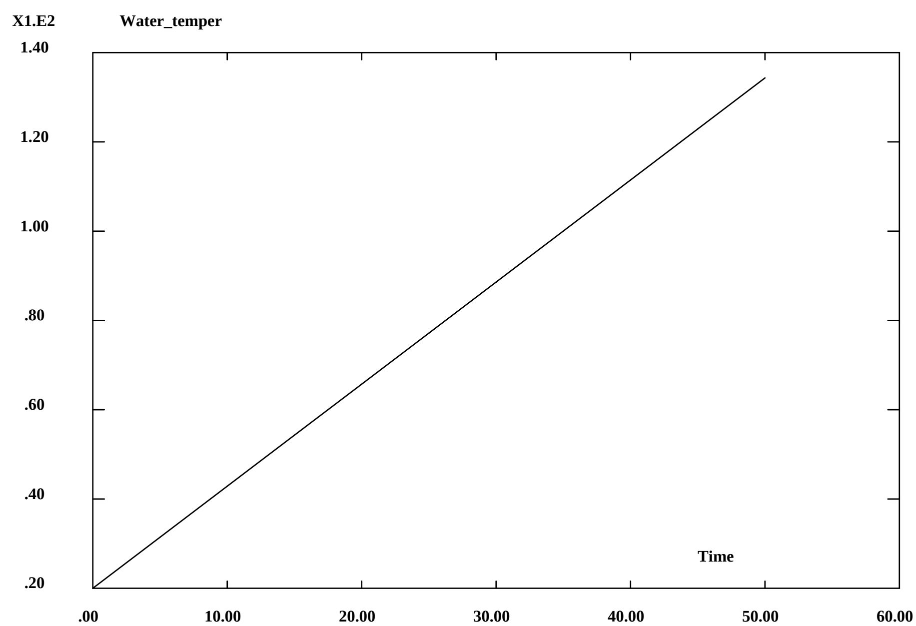
GIBIANE makes the programming of the algorithm very simple. First of all, one must assembly the matrix $C + \Delta t\, \theta.K$ and initialize temperature to 20 degres.

The time steps and the θ are defined as 'FLOTTANT' objects. From that point one can start the loop over the time steps. Performing loops with GIBIANE is possible by using the 'REPETER' operator associated with the 'FIN' operator.

For each time step

1. the second member of equation 23 is computed.
2. ΔT and T^{n+1} is obtained with 'RESO'
3. the heat exchanged with the water is computed by multipliing the convection matrices COND6 et COND4 with the mean of temperature during the step then substracting FLX6 and FLX4 .
4. the water temperature T_WATER is updated.
5. T^n is stored in a 'TABLE' object TABRES. Then T_2 becomes T_1.

The water temperature was stored each time in a 'LISTREEL' type object. It is possible to visualize the evolution of this quantity over the time steps by creating a 'EVOLUTION' type of object . The plot obtained with 'DESS' is represented in figure 9.



Heat exchanged with the oven

FIG. 9 - *Water temperature evolution with time*

6 GIBIANE FILE FOR THE THERMAL COMPUTATION

```
*                          MESH GENERATION

'OPTI' 'DIME' 3 'ELEM' 'SEG2' ;

*     DIMENSION OF THE PAN

*    radius of the cylinder
radius1 = 0.20 ;

*    thiCkness of the cylinder
thick1 = 0.01 ;

*    radius of the part joining the cylinder and the bottom
radius2 = 0.03 ;
*
*    height of the cylinder
height1 = 0.2 ;
*
*   height of the lower part of the handle
height2 = 0.14 ;

*   height of the upper part of the handle
height3 = 0.18 ;

*   width of the handle
width1 = 0.03 ;

*   length of the handle
lenght1 = 0.4 ;



*       POINTS TO GENERATE THE CYLINDER

p1     = radius1 0. height1 ;
p2     = radius1 0.  radius2   ;
p3     = (radius1 - radius2) 0. 0. ;
p4     =  radius1 0. (height2 + height3 / 2.d0) ;

*   PARAMETER CONTROLING THE REFINEMENT OF THE MESH

* number of elements on the vertical axis
n0 = 4 ;
```

```
n1 = 13 ;
n2 = 5 ;

* number of elements around the cylinder
n3 = -40 ;

* number of element along the handle
n4 = 10 ;

*       CYLINDER

c1      = (radius1 - radius2) 0. radius2 ;
li1     = 'DROI' p1 n0 p4 DROIT p2 n1 'CERC' c1 n2  p3 ;
*
'OPTI' 'ELEM' 'QUA4';
*
c2      = (0. 0. 0. ) ;
c3      = (0. 0. 10.) ;
*
su1     = li1 'ROTA' n3 179.99 'DINI' 0.01 'DFIN' 0.02 c2 c3 ;
su2     = su1 'ROTA' n3 179.99 'DINI' 0.02 'DFIN' 0.01 c2 c3 ;
su2     = 'ORIE' su2 'POINT' (0. 0. 0.1) ;

'ELIM' su2 0.001;

'TITRE' 'Main cylinder' ;
'TRAC' su2 (10. 0. 10.) 'CACH' ;

*      GENERATION OF THE BOTTOM OF THE PAN

ctr1     = 'COUL' ('COTE' 2 su2 ) 'TURQUOI' ;

su3      = 'COUL' ('SURF' ctr1 'PLAN') 'VERT' ;
su3      = 'ORIE' su3 'POINT' (0. 0. 0.1) ;

'TRAC' (su3 'ET' su2 ) ;

*      GENERATION OF THE HANDLE

*     identification of the contact between the cylinder and the
*     handle

su2z     = 'CHAN' 'CHAM' ( 'COOR' 3 su2 ) su2 ;
su2x     = 'CHAN' 'CHAM' ( 'COOR' 1 su2 ) su2 ;
theta    = width1 / 2. / radius1 ;
base1    = (su2z 'ELEM' 'COMPRIS' (height2+ 0.003) (height3 - 0.002))
                     'INTE'
   (su2x 'ELEM' 'COMPRIS' (radius1*( 1. - (theta**2/2.))) radius1) ;
base1    = 'COUL' base1 'BLEU' ;
```

```
'TRAC' (base1 'ET' su2 ) ;

'OPTI' 'ELEM' 'CUB8' ;

*   the handle is generated by translating base1  (the contact zone)

handle1 = base1 'VOLU' n4 'TRANS' (lenght1 0. 0.) ;

'TRAC' (su3 et su2 et handle1 );

*       EXTERNAL SURFACE OF THE HANDLE

su4     = ('ENVE'  handle1 ) 'DIFF' base1 ;

*        SHAPE OF THE HANDLE

*  A geometric transform is used to change the section from a
*   rectangle to a circle along the axis of the handle

*     Field of coordinates x y z :

handx1   = 'COOR' 1 handle1 ;
handy1   = 'COOR' 2 handle1 ;
handz1   = 'COOR' 3 handle1 - ('MANU' 'CHPO' handle1 1 'SCAL'
                                   (height3 + height2 / 2.d0));

*    sqrt( x**2 + y**2) :
handr0   = ( handy1**2 + (handx1 ** 2))** 0.5 ;

*    sqrt( y**2 + z**2) :
handr1   = ( handz1**2 + (handy1 ** 2))** 0.5 ;

*    max ( |y| , |z| )
handr2   = (('ABS' handy1) + (ABS handz1)) +
          ( 'ABS' (('ABS' handy1) - ('ABS' handz1)) ) / 2. ;

handy2   = handy1 / handr1 * handr2 ;
handz2   = handz1 / handr1 * handr2 ;

alpha    = ('ABS' (handr0 / lenght1
           - ('MANU' 'CHPO' handle1 1 'SCAL'
            (radius1 / lenght1 ))) ) ** 0.5;

chtraniy = (handy2 - handy1) *  alpha ;
chtraniz = (handz2 - handz1) * alpha ;

*  the points of handle1 change their coordinates

'DEPL' handle1 'PLUS' (('NOMC' chtraniy 'UY') +
                 ('NOMC' chtraniz 'UZ') +
```

```
                ('MANU' 'CHPO' handle1 1 'UX' 0.)) ;
'TRAC'  (handle1 'ET' su2 'ET' su3 ) 'CACH' (10. 10. 10.);

*********************************************************************

*                       THERMIC COMPUTATION

*
*        DIFFUSION IN THE PAN

* model
mod1 = 'MODE' (su2 'ET' su3) 'THERMIQUE' 'COQ4' 'COQ3';

* field of material properties

mat1 = 'MATE' mod1 'C' 4.d6 'RHO' 1. 'K' 25. 'EPAI' thick1  ;

*  matrices

capa1 = 'CAPA' mod1  mat1  ;
cond1 = 'CONDU' mod1  mat1  ;

*       DIFFUSION IN THE HANDLE

mod2  = 'MODE' handle1 'THERMIQUE' ;
mat2  = 'MATE' mod2 'C' 4.d6 'RHO' 1. 'K' 25. ;

capa2 = 'CAPA' mod2 mat2 ;
cond2 = 'CONDU' mod2 mat2 ;

*       CONVECTION ON THE SURFACE OF THE HANDLE

mod3  = 'MODE'  su4  'CONVECTION' ;
mat3  = 'MATE'  mod3 'H' 10. ;

cond3 = 'CONDU' mod3  mat3 ;
flx3  = 'CONV'  mod3  mat3 'T' 20. ;

*       CONVECTION ON THE SURFACE OF THE PAN

*  upper surface

mod4  = 'MODE'  su2    'CONVECTION' 'COQ4' ;
mat4  = 'MATE'  mod4 'H' 100. ;
cond4 = 'CONDU' mod4 mat4 'SUPE' ;
flx4  = 'CONV'  mod4 mat4 'T' 100. 'SUPE' ;

*  lower surface

mod5  = 'MODE'  (su2 'DIFF' base1 ) 'CONVECTION' 'COQ4' ;
```

```
mat5  = 'MATE'  mod5 'H' 10. ;

cond5 = 'CONDU' mod5 mat5 'INFE' ;
flx5  = 'CONV'  mod5 mat5 'T' 20. 'INFE';

*      CONVECTION ON THE UPPER SURFACE OF THE BOTTOM

mod6  = 'MODE'  su3  'CONVECTION' 'COQ4' 'COQ3' ;
mat6  = 'MATE'  mod6 'H' 100. ;

cond6 = 'CONDU' mod6 mat6 'SUPE' ;
flx6  = 'CONV'  mod6 mat6 'T' 100. 'SUPE';
*
*****************************************************************************

*                      STATIONNARY PROBLEM


*          ASSEMBLING THE MATRICES

captot  = capa1 'ET' capa2 ;
condtot = cond1 'ET' cond2 'ET' cond3 'ET' cond4 'ET' cond5
          'ET' cond6 ;

*          SECOND MEMBER

flxtot  = flx3 'ET' flx4 'ET' flx5 'ET' flx6  ;

*      The temperature is imposed on the bootom part of
*      the pan ( surface su3 unknowm TINF )


bl1     = 'BLOQ' su3 'TINF' ;
timp3   = 'DEPI' bl1 200. ;

*     SOLVING THE PROBLEM

oper1   = condtot 'ET' bl1 ;
Tres    = 'RESO' oper1 (flxtot 'ET' timp3 ) ;

*   VISUALIZE THE TEMPERATURE

'TITRE' 'Field of temperature (mean in the shell)' ;
'TRAC' ( 'EXCO' tres 'T' ) (su2 et su3 et su4 ) (prog 20. pas 10.
130. pas 150. 200. )  ;
'OPTI' 'ISOVAL' 'LIGNE' ;
'TRAC' ('EXCO' tres 'T') (su2 et su3 et su4 )
       ('ARET' (su2 et su3 et su4))
       (prog 40. pas 10. 90. 95. 100. PAS 10. 130. pas 150. 200. )
       'CACH' ;
```

```
opti donn 5;

*    HEAT EXCHANGED BETWEEN THE WATER AND THE PAN

qexch1  =  (cond4 'ET' cond6 ) * tres - flx4 - flx6  ;
qexch1b = 'RESU' qexch1 ;
qtot1   = 'EXTR' qexch1b ('POINT' 1 ('EXTR' qexch1b 'MAIL')) 'QSUP' ;

'MESS' 'Heat exchanged with the water = ' qtot1 ;
'TITRE' 'Heat exchanged with the water' ;

*    HEAT PROVIDED BY THE ELECTRIC OVEN

qexch2  = 'REAC' bl1 tres ;
qexch2b = 'RESU' qexch2 ;
qtot2   = 'EXTR' qexch2b ('POINT' 1 ('EXTR' qexch2b 'MAIL')) 'QINF' ;
'MESS' 'Heat exchanged with the oven' qtot2 ;

*     TEMPERATURE AT THE END OF THE HANDLE

* location of  the measure
pend1   = 'POINT' handle1 'PROC' ((lenght1 + radius1) 0.
                                  (height3 + height2 / 2.d0)) ;
tend1   =  'EXTR' tres 'T' pend1 ;
'MESS' 'Temperature at the end of the handle' ;
**********************************************************************

*                UNSTATIONNARY PROBLEM

*  capacity of the water
c_water = 4.19d3 * radius1 * radius1 * pi * height1  ;

*  parameters

theta = 0.5 ;
*   time step
dt    = 5. ;
i     = 0 ;


*  initializing the loop
qexwat  = 0. ;
t_water = 20. ;
lt_water = prog t_water ;

t_1 = ('MANU' 'CHPO' 1 handle1 'T' 20. 'NATURE' 'DIFFUS')  'ET'
     ('MANU' 'CHPO' 3 (su2 et su3) 'T' 20. 'TINF' 20. 'TSUP' 20.
      'NATURE' 'DIFFUS' ) ;

*   Table (array) to store the result
```

```
tabres = 'TABLE' ;


*  prescribed rate of heat

flx1  = 'FLUX' mod1 1000. su3 'INFE' ;


oper2 = captot 'ET' (condtot * (theta*dt)) ;

*  beginning of loop
'REPE' blok2 10 ;
*
  flx4      = 'CONV' mod4 mat4 'T' t_water 'SUPE';
  flx6      = 'CONV' mod6 mat6 'T' t_water 'SUPE';
  flcond    = condtot * t_1 ;
*
  flxtot    = (flx1 + flx3 + flx4 + flx5 + flx6 - flcond ) * dt ;
  dtemp     = 'RESO' oper2 flxtot ;
  t_2       = t_1 + dtemp ;
*
  qexch1    =  (cond4 'ET' cond6 ) * (t_2 + t_1 / 2.)
                                   - flx4 - flx6 * dt ;
  qexch1b   = 'RESU' qexch1 ;
  qexwat    = 'EXTR' qexch1b ('POINT' 1 ('EXTR' qexch1b 'MAIL'))
                                                         'QSUP' ;

  dt_water  = qexwat / c_water ;
  t_water   = t_water + dt_water ;
  lt_water  = lt_water 'ET' ('PROG' t_water ) ;

'MESS' 'Water temperature' t_water ;
*
  tabres . i = t_1 ;
  t_1        = t_2 ;
  i          = i + 1 ;


'FIN' blok2 ;

'TRAC' ( 'EXCO' 'T' t_2 ) (su2 et su3 et su4 ) ;

evtwet = 'EVOL' 'MANU' ('PROG' 0. 'PAS' dt (10.*dt)) 'Time'
         lt_water 'Water_temperature'  ;
'DESS' evtwet ;
```

7 Acknoledgment

The first part of this document is taken from "Beginning with Castem 2000" C Laborderie and E Jeanvoine Technical report DMT/94-356 CEA 1994.