

EES

Niveau intermédiaire

**Fonctions
Procédures
Boucles, Conditions, Redirections
Tables (LOOKUP Tables)
Tables paramétriques
Graphiques
Diagram window**

Par

Patrice Pinel (A-01)

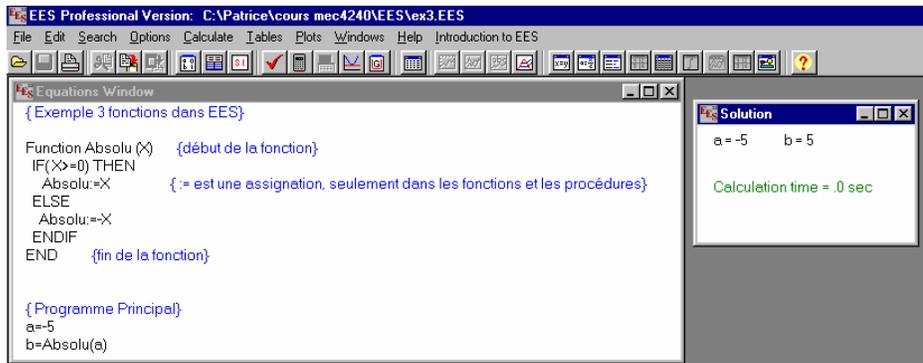
et

Michel Bernier (A-05)

1-Fonctions

On peut programmer des fonctions dans EES. La fonction débute par le mot clé FUNCTION et se termine par le mot END. L'utilisation d'une FUNCTION permet de retourner une variable (et une seule) au programme principal. Par exemple, dans l'exemple suivant la FUNCTION « Absolu (X) » retourne au programme principal la valeur absolue de la valeur envoyée.

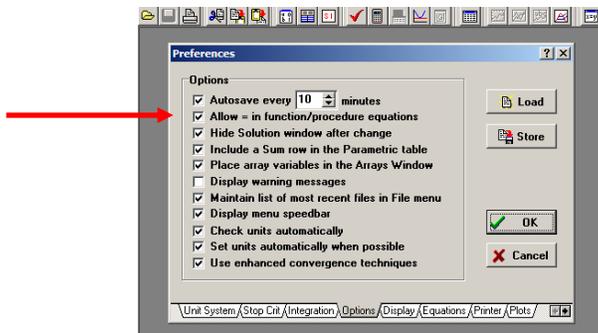
Les équations apparaissant dans une FUNCTION sont fondamentalement différentes de celles écrites dans le programme principal. En effet, ces équations sont plutôt des assignations. Ainsi le membre de gauche d'une déclaration d'assignation est fixé par le membre de droite. De plus, EES résout les équations d'une FUNCTION dans l'ordre dans lesquelles elles ont été écrites alors que dans le programme principal les équations sont résolues itérativement dans un ordre déterminé par EES.



Notes :

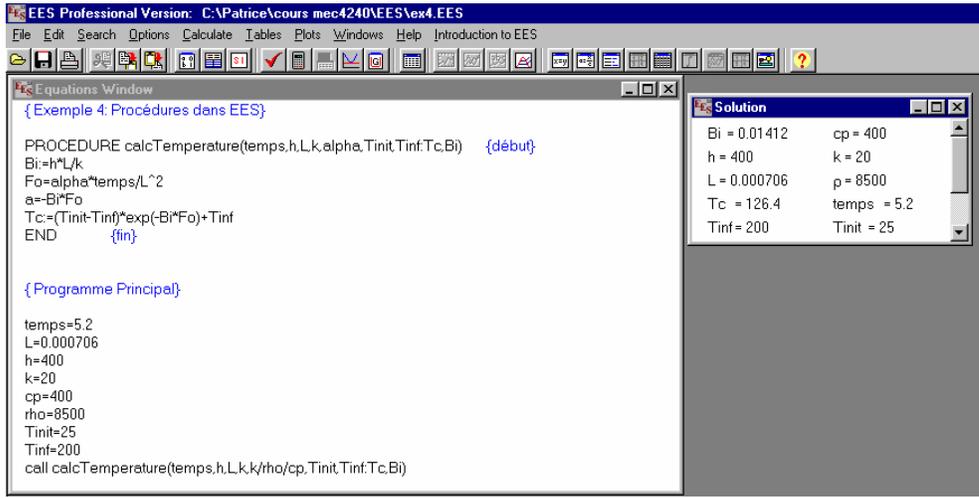
1- Il n'est pas nécessaire d'utiliser le « := » (comme dans l'énoncé « Absolu :=X » ci-dessus). Ainsi, si l'option « Allow = in function/procedure equations » est cochée dans la fenêtre « préférences/options » (voir figure suivante) on peut tout simplement utiliser le signe d'égalité.

2- Les FUNCTION doivent se trouver au début d'un programme soit avant le programme principal.



2- Procédures

On peut programmer des PROCEDURE dans EES. Les premières variables dans l'entête de la procédure (avant le ':') sont celles dont la procédure se sert sans les modifier alors que celles situées après peuvent être modifiées par la procédure. La procédure est appelée à l'aide de la commande call.

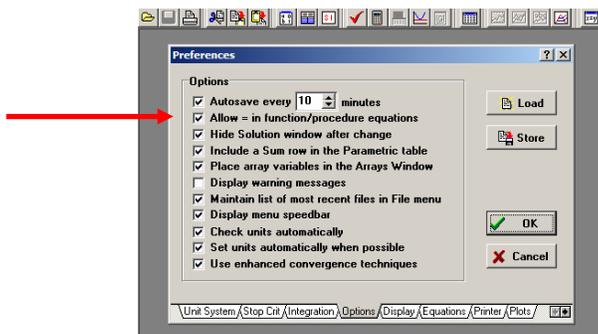


Les équations apparaissant dans une PROCEDURE sont fondamentalement différentes de celles écrites dans le programme principal. En effet, ces équations sont plutôt des assignations. Ainsi le membre de gauche d'une déclaration d'assignation est fixé par le membre de droite. De plus, EES résout les équations d'une fonction dans l'ordre dans lesquelles elles ont été écrites alors que dans le programme principal les équations sont résolues itérativement dans un ordre déterminé par EES.

Notes :

1- Il n'est pas nécessaire d'utiliser le « := » (comme dans l'énoncé « $Bi := h \cdot l / k$ » ci-dessus). Ainsi, si l'option « Allow = in function/procedure equations » est cochée dans la fenêtre « préférences/options » (voir figure suivante) on peut tout simplement utiliser le signe d'égalité.

2- Les PROCEDURE doivent se trouver au début d'un programme soit avant le programme principal.



3- Boucles, Conditions, Redirections

EES comprend plusieurs éléments de programmation tels les boucles (Repeat-Until), des directives conditionnelles (If-Then-Else) et des redirections (Goto). Les Boucles, Conditions, Redirections doivent obligatoirement se trouver dans une **FONCTION** ou une **PROCEDURE**.

Il est également possible d'utiliser les boucles avec compteurs (For, en C ou Pascal) en utilisant le mot clé Duplicate. Ces commandes sont décrites dans l'aide du logiciel.

The screenshot displays the EES Professional software interface. The main window, titled "Equations Window", contains the following code:

```
{ Exemple 5: directives de programmation dans EES}

{conditions IF THEN ELSE seulement dans les fonctions et procédures}
FUNCTION ifthenelse(a)
  IF(a=9) THEN
    ifthenelse=10
  ELSE ifthenelse:=a
END

{Boucle REPEAT UNTIL seulement dans les fonctions et procédures}
FUNCTION Som(x)
  Somx=0; i=0
  REPEAT
    i=i+1
    Somx= Somx+i
  UNTIL (i>=x)
  Som=Somx
END

{GOTO seulement dans les boucles IF THEN ELSE et REPEAT UNTIL}
FUNCTION Fact(N)
  F:=1; i:=1
  10: i=i+1 {Numérotation de la ligne}
  F=F*i
  IF Not (i>=N) Then GoTo 10 {renvoi à la ligne}
  Fact=F
End

{boucle duplicate et création d'un vecteur}
Duplicate i=1,4
vecteur[i]=10-i
End

{appels des fonctions}
x=ifthenelse(vecteur[2])
s=Som(x)
f=Fact(x)
```

The "Solution" window shows the results of the calculations:

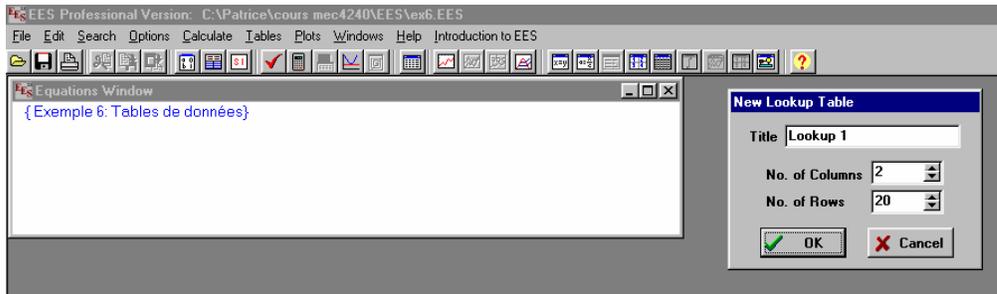
f = 40320 s = 36 x = 8
Calculation time = .0 sec

The "Arrays Table" window displays a table with the following data:

	vecteur _i
[1]	9
[2]	8
[3]	7
[4]	6

4-Tables et Recherches

On peut créer une table en utilisant la commande Tables/New Lookup Table dans le menu. Les « Lookup tables » servent généralement à stocker des données.



Après avoir donné un titre à la table et précisé le nombre de lignes et de colonnes, on obtient une table qu'on peut remplir.

	1	2	3	4
	Metal	K [(W/m²)]	ρ [Kg/m³]	Cp [J/Kg.K]
Row 1	Aluminium	237	2702	903
Row 2	Cuivre	401	8933	385
Row 3	Or	317	19300	129
Row 4	Fer	80	7870	447
Row 5	Plomb	35	11340	129
Row 6	Platine	72	21450	133
Row 7	Argent	429	10500	235
Row 8	Titane	22	4500	522
Row 9	Zinc	116	7140	389

NB : Pour afficher du texte dans une colonne, il faut sélectionner celle-ci, appuyer sur le bouton de droite de la souris, cliquer sur « propriétés » et changer le format pour « string » dans le casier « Style ».

Il est possible de récupérer des valeurs des LOOKUP table et de les utiliser dans un programme EES. Pour effectuer une recherche dans une table, on utilise les commandes LOOKUP\$ et LOOKUP\$ROW pour du texte et INTERPOLATE, LOOKUP, LOOKUPROW et LOOKUPCOL pour les valeurs. La syntaxe de ces commandes est décrite dans l'aide du logiciel. Voici quelques utilisations des LOOKUP tables.

Equations Window

```
{ Exemple 6: Tables de données }

a=Lookup('Propriétés Physiques de Métaux',4,'K')
b$=Lookup$( 'Propriétés Physiques de Métaux',4,'Metal')
c=interpolate('Conductivité Thermique Huile',T,K,T=315)
```

Solution

```
a = 80          b$ = 'Fer'
c = 0.1441

Calculation time = .0 sec
```

	1	2	3	4
	Metal	K [(W/m²C)]	ρ [Kg/m³]	Cp [J/Kg.K]
Row 1	Aluminium	237	2702	903
Row 2	Cuivre	401	8933	385
Row 3	Or	317	19300	129
Row 4	Fer	80	7870	447
Row 5	Plomb	35	11340	129
Row 6	Platine	72	21450	133
Row 7	Argent	429	10500	235
Row 8	Titane	22	4500	522
Row 9	Zinc	116	7140	389

Equations Window

```
{ Exemple 6: Tables de données }

a=Lookup('Propriétés Physiques de Métaux',4,'K')
b$=Lookup$( 'Propriétés Physiques de Métaux',4,'Metal')
c=interpolate('Conductivité Thermique Huile',T,K,T=315)
```

Solution

```
a = 80          b$ = 'Fer'
c = 0.1441

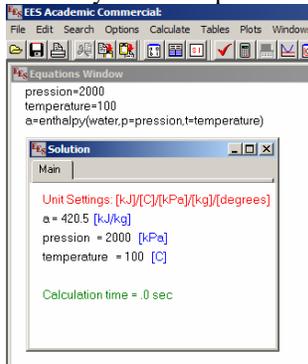
Calculation time = .0 sec
```

	T [K]	K [W/m.K]
Row 1	273	0.147
Row 2	280	0.144
Row 3	290	0.145
Row 4	300	0.145
Row 5	310	0.145
Row 6	320	0.143
Row 7	330	0.141
Row 8	340	0.139
Row 9	350	0.136
Row 10	360	0.138
Row 11	370	0.137
Row 12	380	0.136
Row 13	390	0.135

5-Tables Paramétriques

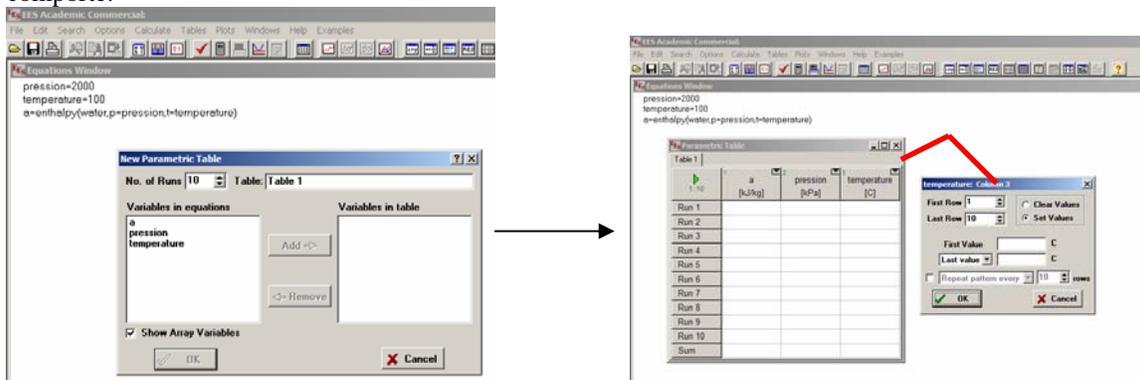
On peut générer des tables paramétriques c'est à dire des tables où on fait varier une ou plusieurs variables pour observer l'influence sur les résultats. En d'autres mots les tables paramétriques nous permettent de solutionner le même système d'équation en changeant la valeur de une (ou plusieurs) variables du programme.

Soit le système d'équation suivant :

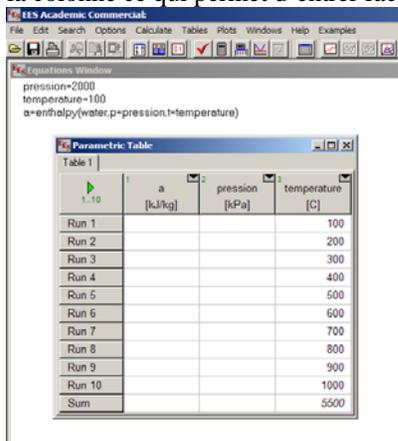


Supposons maintenant que l'on désire évaluer « a » pour différentes températures (de 100 à 1000 kPa par exemple) au moyen d'une table paramétrique.

Pour créer la table, on utilise la commande Tables/New Parametric Table et on obtient l'utilitaire suivant qui permet de sélectionner les variables qu'on désire dans la table et le nombre de lignes (runs) que la table comporte.



Il s'agit ensuite d'entrer les 10 valeurs de « a ». Pour ce faire on peut utiliser le triangle noir au sommet de la colonne ce qui permet d'entrer facilement une liste de nombres.



Avant de solutionner (F3 ou Solve Table) il est très important « d'enlever » le paramètre qui varie (la température dans l'exemple ci-haut). Deux options sont alors possibles:

- 1- mettre la variable en commentaire
- 2- utiliser l'option « \$IFNOT parametrictable \$ENDIF »

Equations Window

```

pression=2000
(temperature=100)
a=enthalpy(water,p=pression,t=temperature)
    
```

1..10	a [kJ/kg]	pression [kPa]	temperature [C]
Run 1	420.5	2000	100
Run 2	852.6	2000	200
Run 3	3023	2000	300
Run 4	3248	2000	400
Run 5	3468	2000	500
Run 6	3690	2000	600
Run 7	3918	2000	700
Run 8	4151	2000	800
Run 9	4390	2000	900
Run 10	4636	2000	1000
Sum	31797	20000	5500

Equations Window

```

pression=2000
$ifnot parametrictable
temperature=100
$endif
a=enthalpy(water,p=pression,t=temperature)
    
```

1..10	a [kJ/kg]	pression [kPa]	temperature [C]
Run 1	420.5	2000	100
Run 2	852.6	2000	200
Run 3	3023	2000	300
Run 4	3248	2000	400
Run 5	3468	2000	500
Run 6	3690	2000	600
Run 7	3918	2000	700
Run 8	4151	2000	800
Run 9	4390	2000	900
Run 10	4636	2000	1000
Sum	31797	20000	5500

L'option 2 fonctionne de la façon suivante. Si la solution par table paramétrique n'est pas activée alors les équations situées entre « \$IFNOT parametrictable » et « \$ENDIF » seront exécutées. En revanche, si la solution par table paramétrique est activée (F3) alors les équations situées entre « \$IFNOT parametrictable » et « \$ENDIF » ne seront pas exécutées.

À remarquer que les valeurs en bleu dans la table paramétrique sont des valeurs calculées.

Il est possible de lire des valeurs dans une table paramétrique à l'aide des commandes Tablevalue('nom',l,c) , Tablename\$ et Tablerun#. Tablename\$ retourne le nom de la table paramétrique actuellement utilisée dans les calculs. Tablerun# retourne le numéro de la ligne (run) présentement calculée. Tablevalue retourne la valeur se trouvant à la ligne 'l' et la colonne 'c' de la table 'nom'. L'exemple suivant montre quelques utilisations de ces commandes.

Equations Window

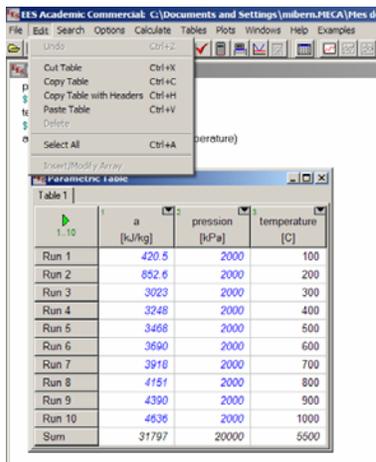
```

{ Exemple 7: Table Paramétrique }
FUNCTION enthalpie_prec(nom$,ligne)
IF (Tablerun#=1) THEN
enthalpie_prec:=0
ELSE
enthalpie_prec:=Tablevalue(nom$,ligne-1,'enthalpie')
ENDIF
END

enthalpie=ENTHALPY(R123,T=Température,P=Pression)
nom$=Tablename$
enthalpieprécédente=enthalpie_prec(nom$,Tablerun#)
    
```

1..10	Température [°C]	Pression [kPa]	enthalpie [kJ/kg]	enthalpieprécéd
Run 1	20	90	59.59	0
Run 2	20	110	59.6	59.59
Run 3	30	90	239.2	59.6
Run 4	30	110	69.89	239.2
Run 5	40	90	246.2	69.89
Run 6	40	110	245.8	246.2
Run 7	50	90	253.3	245.8
Run 8	50	110	252.9	253.3
Run 9	60	90	260.6	252.9
Run 10	60	110	260.2	260.6

Il est possible de copier/coller le contenu de la table (dans EXCEL par exemple). Pour ce faire il s'agit de d'aller dans le menu « Edition » faire « select all » et ensuite « copy.... »



A remarquer qu'il est également possible de faire l'opération inverse, c'est-à-dire de copier un tableau conçu dans EXCEL et de l'importer dans une table paramétrique de EES.

6-Graphiques

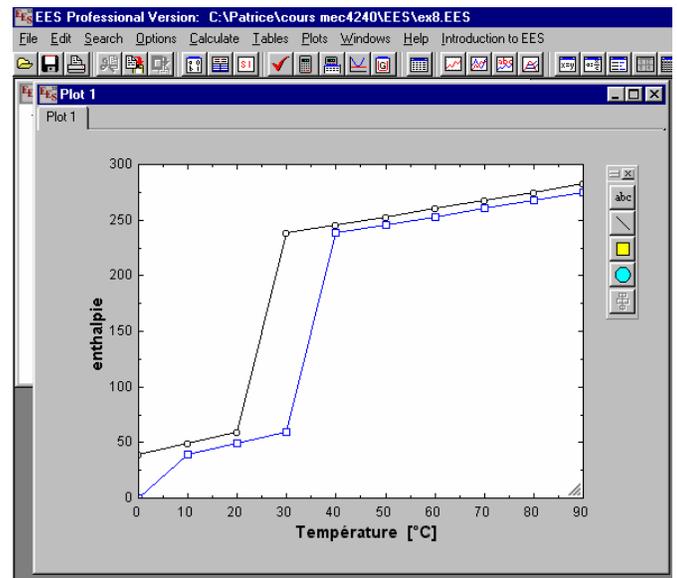
Il est possible de générer un graphique des données contenues dans une table paramétrique en sélectionnant l'option Plot/New Plot Window dans le menu.

The screenshot shows the EES Professional Version interface. The 'Plots' menu is open, showing options like 'New Plot Window', 'Overlay Plot', 'Modify Plot', 'Modify Axes', 'Show Tool Bar', 'Delete Plot Window', 'Property Plot', and 'Curve Fit'. The 'New Plot Window' option is selected. In the background, a 'Parametric Table' window is visible, displaying a table of enthalpy data.

Run	Température [°C]	enthalpie [kJ/kg]	enthalpieprécéd
Run 1	0	39.29	0
Run 2	10	49.39	39.29
Run 3	20	59.59	49.39
Run 4	30	238.9	59.59
Run 5	40	246	238.9
Run 6	50	253.1	246
Run 7	60	260.4	253.1
Run 8	70	267.7	260.4
Run 9	80	275.1	267.7
Run 10	90	282.6	275.1

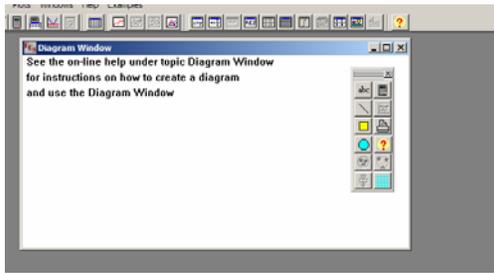
À partir de là, une boîte de dialogue apparaît où il faut sélectionner les variables représentées sur chaque axe et préciser le format. Une fois que c'est fait, on obtient le graphique.

The screenshot shows the 'New Plot Setup' dialog box. The title is 'Plot 1'. The X-Axis variables are 'Température', 'enthalpie', and 'enthalpieprécédente'. The Y-Axis variables are 'Température', 'enthalpie', and 'enthalpieprécédente'. The 'Table' dropdown is set to 'Parametric Table' and 'Table d'enthalpies'. The 'First Run' is 1 and the 'Last Run' is 10. The 'Format' is 'A 0'. The 'Minimum' is 0.00 and the 'Maximum' is 90.00. The 'Interval' is 10.00. The 'Line' is 'Linear' and 'Grid lines' are checked. The 'OK' button is highlighted.



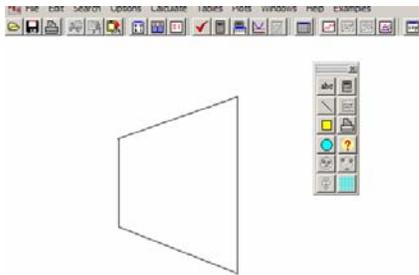
7-Diagram window

EES permet de tracer des dessins/objets pertinents au problème à solutionner. De plus, on peut y insérer des variables d'entrée « input » et des résultats de calculs « output ». Pour accéder au « diagram window » il suffit de faire « CTRL-D ». Cette opération mène à la fenêtre suivante :

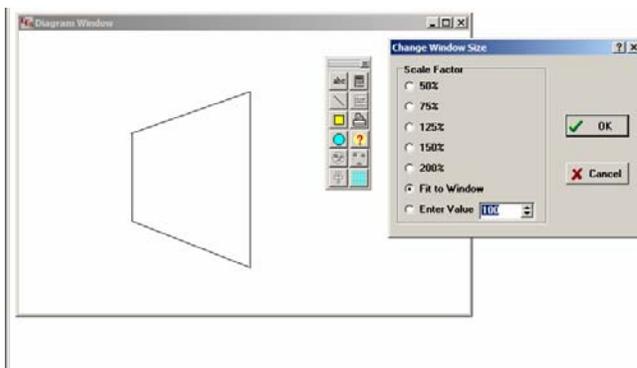


On remarque qu'un utilitaire de dessin (Tool bar) apparaît lors de l'ouverture du « diagram window ». Lorsque cet utilitaire est apparent, il est alors possible d'effectuer des opérations (dessiner par exemple) dans le « diagram window ». On est alors en mode « développement ». Sans cet outil il n'est pas possible de modifier le « diagram window ».

L'utilitaire permet de faire des dessins simples. La figure suivante montre le dessin d'une turbine. Il est également possible d'importer des dessins réalisés au moyen d'autres outils de dessins.

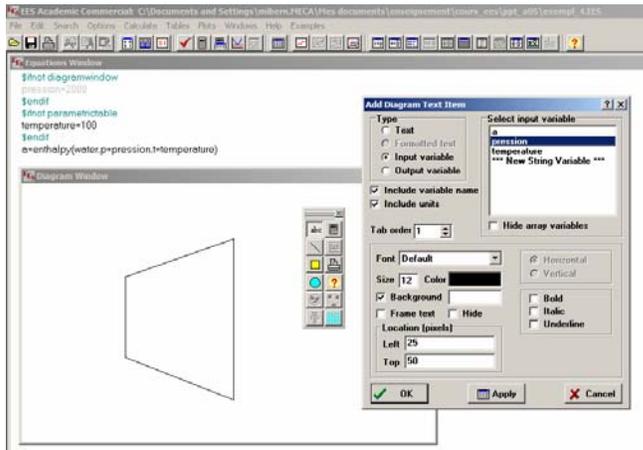


Tel que montré à la figure suivante, il est possible de redimensionner le dessin en cliquant sur le bouton droit de la souris.

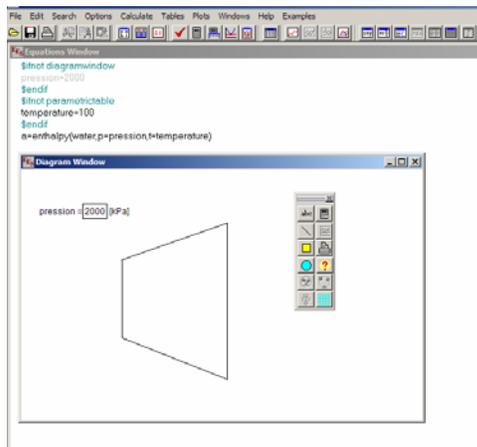


INPUT

Pour inclure une variable input dans le « diagram window » il s'agit d'effectuer l'opération suivante :

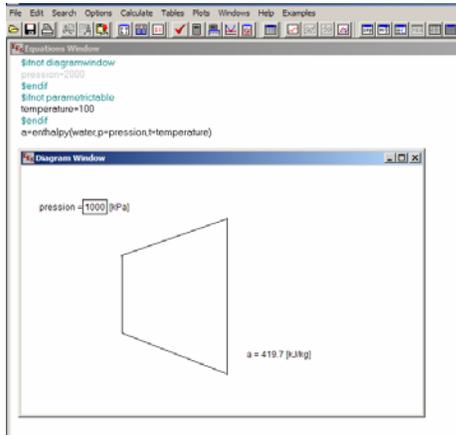


En insérant une variable « input » se trouvant dans la fenêtre équation à l'intérieur du diagram window il faut indiquer à EES quelle valeur sera utilisée. Il est recommandé d'utiliser les commandes « \$IFNOT diagramwindow » et « \$ENDIF » tel qu'indiqué ci-dessus en rapport avec les tables paramétriques. Avec cette approche si le « diagram window » n'est pas apparent (**i.e. qu'il aura été fermé avec le « x » du coin supérieur droit de la fenêtre « diagram window »**) alors c'est la valeur de la variable contenue dans la fenêtre « equation » qui est utilisé lorsque F2 (solve) est activé. En revanche, si le « diagram window » est apparent, alors les valeurs input du « diagram window » seront utilisées lorsque F2 (solve) est activé.



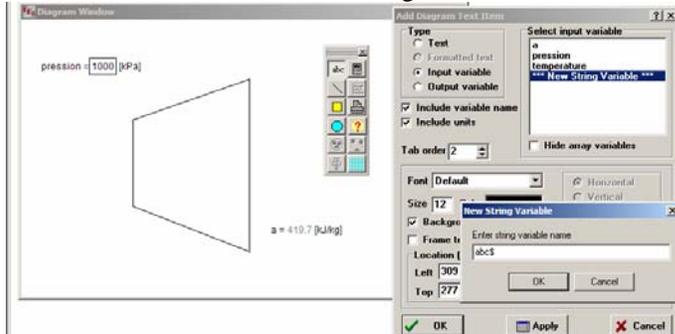
OUTPUT

Pour ce qui est des variables en output, il n'y a pas de restrictions. Elles peuvent apparaître à la fois dans le « diagram window » et dans la fenêtre « equation ». La figure suivante montre la variable « a » qui apparaît comme output dans le « diagram window ».



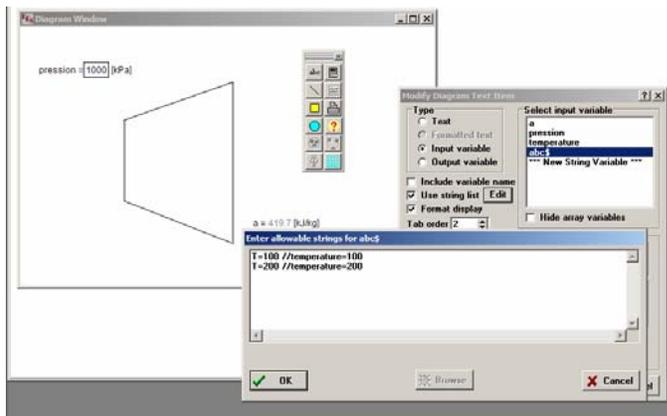
MENU DÉROULANT

Il est possible de construire des menus déroulants contenant des variable « input » dans EES. Pour ce faire, suivre les indications montrées à la figure suivante :



Il s'agit d'abord de créer une variable alphanumérique (string variable). Dans l'exemple ci-dessus la variable « abc\$ » est créée.

L'opération suivante permet de créer un menu déroulant avec 2 items soit deux choix de température (T=100, et T=200). Tel qu'indiqué ci-dessous, les variables apparaissent après les deux barres obliques (/).



Afin d'avoir le choix entre la température dans la fenêtre equation et la température dans le menu déroulant, il faut mettre la variable « température » entre « \$IFNOT diagramwindow » et « \$ENDIF ». Avec cette approche si le « diagram window » n'est pas apparent (i.e. qu'il aura été fermé avec le « x » du coin supérieur droit de la fenêtre « diagram window ») alors c'est la valeur de la variable contenue dans la fenêtre « equation » qui est utilisé lorsque F2 (solve) est activé. En revanche, si le « diagram window » est apparent, alors la valeur du menu déroulant du « diagram window » est utilisée lorsque F2 (solve) est activé.

