

E E S

Engineering Equation Solver

Pour Microsoft Windows

**F-Chart Software
4406 Fox Bluff Rd
Middleton, WI 53562
Tél. : 0 11 608-836-8531**

FAX: 0 11 608-836-8536

<http://www.fchart.com>

Copyright 1992-98 par S.A. Klein et F.L. Alvarado

Tout droits réservés.

The authors make no guarantee that the program is free from errors or that the results produced with it will be free of errors and assume no responsibility or liability for the accuracy of the program or for the results which may come from its use.

EES a été développé avec DELPHI (Borland)

Le présent document a été traduit de l'anglais par Jérôme ZATTONI

Registration Number _____

Toute correspondance doit inclure ce numéro.

E E S

Engineering Equation Solver

Pour Microsoft Windows

F-Chart Software
4406 Fox Bluff Rd
Middleton, WI 53562
Phone: 608-836-8531
FAX: 608-836-8536
<http://www.fchart.com>

Tables des matieres

TABLES DES MATIERES	IV
APERÇU GÉNÉRAL	1
PREMIERS PAS	4
INSTALLATION DE EES.....	4
DÉMARRER EES 	4
INFORMATIONS GÉNÉRALES	5
EXEMPLE DE RÉOLUTION D'UN PROBLÈME DE THERMODYNAMIQUE	9
LES FENÊTRES DE EES	19
INFORMATIONS GÉNÉRALES	19
LA FENÊTRE D'ÉQUATIONS	21
LA FENÊTRE DE MISE EN FORME D'ÉQUATIONS.....	23
LA FENÊTRE DE SOLUTION	25
LA FENÊTRE TABLEAUX.....	27
LA FENÊTRE RÉSIDUELLE (RESIDUAL WINDOW)	29
LA FENÊTRE DE LA TABLE PARAMÉTRIQUE.....	31
LA FENÊTRE 'LOOKUP TABLE'	34
LA FENÊTRE DE DIAGRAMME (DIAGRAM WINDOW).....	36
LA FENÊTRES DE TRACÉS (PLOT WINDOW).....	39
LA FENÊTRE DE DEBOGAGE (DEBUG WINDOW)	44
LES COMMANDES DU MENU	45
LE MENU FILE	45
LE MENU EDITION (EDIT)	49
LE MENU SEARCH	51
LE MENU OPTIONS	52
LE MENU TABLES.....	67
LE MENU PLOT	74
LE MENU WINDOWS	81
LE MENU HELP.....	83
LE MENU TEXTBOOK.....	84
FONCTIONS INCORPORÉES	86
FONCTIONS MATHÉMATIQUES	86
FONCTIONS ET PROPRIÉTÉS THERMODYNAMIQUES	93

UTILISATION DES FICHIERS ET TABLES 'LOOKUP'	99
FONCTIONS, PROCÉDURES ET MODULES	105
LES FONCTIONS EES	106
LES PROCÉDURES	109
LES INSTRUCTIONS IF THEN ELSE	111
LES STRUCTURES IF THEN ELSE MULTI-LIGNES	112
L'INSTRUCTION GOTO	113
LA STRUCTURE REPEAT UNTIL	113
LES MODULES	115
LES FICHIERS LIBRAIRIE	118
LA DIRECTIVE \$COMMON	121
LA DIRECTIVE \$INCLUDE	122
FONCTIONS ET PROCÉDURES COMPILÉES	123
LES FONCTIONS COMPILÉES (FICHIERS .DLF)	123
LES FONCTIONS COMPILÉES PWF	127
<i>Listing de la fonction PWF développée en Borland's Delphi 3.0</i>	128
PROCÉDURES COMPILÉES (FICHIERS .FDL ET .DLP)	130
<i>Procédure compilée, format .FDL : exemple en FORTRAN</i>	131
<i>Code du programme FORTRAN : MDASF</i>	133
<i>Les procédures compilées, format .DLP – un exemple en PASCAL</i>	134
<i>Exemple de procédure compilée (.DLP) en Borland's Delphi 2.0</i>	135
CARACTÉRISTIQUES AVANCÉES.....	138
CHAÎNES DE CARACTÈRES	138
VARIABLES COMPLEXES	139
LES VARIABLES 'TABLEAUX' (ARRAYS)	143
LA COMMANDE DUPLICATE	144
LES MATRICES	145
TRACÉS DES PROPRIÉTÉS THERMODYNAMIQUES (PROPERTY PLOT)	147
INTÉGRALES ET ÉQUATIONS DIFFÉRENTIELLES	148
<i>Méthode 1 : Résolution d'équations différentielles avec la Fonction INTEGRALE</i>	149
<i>Méthode 2 : Résolution d'équations différentielles avec la Fonction TableValue</i>	153
CONSEILS D'UTILISATION.....	157
SOLUTION D'ÉQUATIONS ALGÈBRIQUES	161
GROUPEMENT D'ÉQUATIONS DANS UN SYSTÈME	164
DÉTERMINATION DE LA VALEUR MINIMUM OU MAXIMUM D'UNE VARIABLE	166
INTÉGRATION NUMÉRIQUE	167
RÉFÉRENCES	169

AJOUT DE DONNÉES DE PROPRIÉTÉS	170
<i>Informations générales.....</i>	<i>170</i>
<i>Ajout de propriétés à un fluide.....</i>	<i>171</i>

Aperçu général

EES (prononcé ISE) est l'abréviation de Engineering Equation Solver ou Solveur d'équations pour l'ingénierie. EES permet de résoudre des systèmes d'équations algébriques, des équations différentielles, des équations à variables complexes. EES permet également d'optimiser les paramètres de modélisation d'un système, de calculer des régressions linéaires et non linéaires, de générer des courbes de grande qualité. EES fonctionne à la fois sur des plates-formes PC Compatible ou APPLE Macintosh.

Ce manuel décrit la version de EES développée pour les systèmes d'exploitation Windows 3.1, 95, et NT.

Il existe deux différences majeures entre EES et un solveur "classique" d'équations. Tout d'abord, EES identifie automatiquement et groupe des équations qui doivent être simultanément résolues. Cette caractéristique simplifie le procédé pour l'utilisateur et assure que le logiciel opérera toujours avec une efficacité optimale. De plus, de nombreuses fonctions mathématiques et thermodynamiques utilisées dans le milieu de l'ingénierie sont incorporées dans le logiciel. Par exemple, des tables "vapeur" sont implémentées tel que n'importe quelles propriétés thermodynamiques peuvent être obtenues à partir des fonctions incorporées. De même, cette fonctionnalité est fournie pour la plupart des réfrigérants (y compris une partie de nouveaux mélanges), l'ammoniac, le méthane, le dioxyde de carbone et beaucoup d'autres liquides. Les tables d'air sont incorporées, comme étant des fonctions psychrométriques ainsi que la table de JANAF applicable pour un grand nombre de gaz. Enfin, des propriétés de transport sont aussi disponibles pour la plupart de ces substances.

EES contient une vaste bibliothèque de fonctions mathématiques et thermodynamiques. Néanmoins, il est impossible de prévoir l'ensemble des besoins des utilisateurs. De fait, EES offre à l'utilisateur la possibilité d'entrer ses propres fonctions suivant trois méthodes :

1. Il est possible de saisir et d'interpoler des données numériques dans une table, pour ensuite les utiliser directement dans la résolution de système des équations.
2. EES comporte son propre langage de programmation. Il s'apparente au PASCAL et FORTRAN. Il permet à l'utilisateur de développer et sauvegarder dans des fichiers de librairie ses propres fonctions et procédures. Ces fichiers seront lus à chaque ouverture du logiciel et ainsi utilisables dans chaque session.

3. Enfin, les fonctions et les procédures compilées, écrites dans un langage de haut niveau tel le PASCAL, le C ou le FORTRAN, pourront être utilisées dans EES par le biais de bibliothèques dynamiques (DLL).

Il est ainsi possible d'étendre les capacités du logiciel.

Le développement de EES a été motivé par l'expérience dans l'enseignement des sciences mécaniques, thermodynamiques et de transfert de chaleur. Pour qu'un étudiant puisse apprendre et mieux appréhender les phénomènes physiques, il est nécessaire de travailler sur des problèmes concrets. Cependant, beaucoup de temps et d'effort sont exigés dans la recherche des propriétés des composants et la mise en équation appropriée du système. La connaissance des tables de propriétés, de l'algèbre ne contribue pas forcément à la compréhension du problème. De plus, le temps et l'effort exigés à résoudre des problèmes de manière conventionnelle, c'est-à-dire faire attention à l'ordre dans lequel les équations doivent être résolues (qui en fait importe peu) et/ou réaliser des études Paramétrique laborieuses peuvent pénaliser l'étudiant dans l'apprentissage de ces sciences.

Par exemple les problèmes combinant des phénomènes de thermodynamique et de transfert de chaleur sont difficilement assimilables de part leur complexité mathématique. EES permet à l'utilisateur de se focaliser beaucoup plus sur la conception en s'affranchissant de ces notions.

EES est particulièrement utile pour les problèmes de conception dans lequel les effets d'un ou de plusieurs paramètre(s) ont besoin d'être déterminés. Le programme offre cette possibilité avec sa Table Paramétrique, similaire à une feuille de tableur. L'utilisateur identifie les variables indépendantes en entrant leurs valeurs dans les cellules de table. EES calculera les valeurs des variables dépendantes dans la table. Le rapport des variables dans la table peut être visualisé sous forme de courbes, de diagrammes. EES offre également la possibilité de propager l'incertitude de données expérimentales afin de donner des résultats de variables calculées modulo les estimations d'incertitude.

Avec EES, il est également facile de résoudre des problèmes comportant un ensemble de variables indépendantes.

Grâce à son ensemble de commandes intuitives, compréhensibles, EES permet à un novice de résoudre rapidement n'importe quel problème algébrique.

En outre, les capacités de ce programme sont telles, qu'il peut être également utilisé par des " experts ". EES comporte une vaste banque de données de propriétés thermodynamiques et de transport. Il est ainsi possible de résoudre des problèmes de thermodynamique, de mécanique des fluides, de transfert de chaleur. EES s'applique également dans de nombreux domaines de l'ingénierie (telle la mécanique par exemple).

Le présent document s'articule autour de sept chapitres et de quatre annexes. Dans le premier chapitre, l'utilisateur trouvera la résolution à l'aide de EES d'un problème simple de thermodynamique.

Le chapitre 2 contient des informations spécifiques sur les diverses fonctions et contrôles. Le chapitre 3 est une section de référence contenant l'ensemble des commandes accessibles par la barre de menu.

Le chapitre 4 décrit les fonctions mathématiques ainsi que les fonctions et propriétés thermodynamiques incluses dans EES.

Dans le chapitre 5, le lecteur trouvera les informations nécessaires pour développer, archiver des fonctions, procédures.

Le chapitre 6 décrit comment des fonctions compilées dans des fichiers de bibliothèques dynamiques (DLL) peuvent être utilisées dans EES.

Le chapitre 7 comporte l'ensemble des informations relatives aux fonctions avancées de EES tel l'usage de chaînes de caractères, de variables complexes, de résolution d'équations simultanées, différentielles et algébriques, et la visualisation des tracés.

L'annexe A contient un ensemble de conseils, de suggestions d'utilisation.

L'annexe B décrit les méthodes numériques utilisées par EES.

L'annexe C décrit comment intégrer de nouvelles fonctions ou propriétés.

L'annexe D, décrit quelles sont les particularités utilisées par EES dans la résolution des problèmes, exemples fournis avec le logiciel.

Installation de EES

Deux versions de EES sont actuellement disponibles : EES et EES32. EES est conçu pour fonctionner sur n'importe quelle plate-forme Windows (3.1, 3.11, 95 et NT). EES32 est une version 32 bits fonctionnant uniquement sous Windows 95 et Windows NT. Cependant une partie des dernières nouvelles caractéristiques ES, e. g., les modules (cf. Chapitre 5) et les variables complexes (cf. Chapitre 7) sont disponibles uniquement dans la version 32 bits. Si vous utilisez une plate-forme Windows 95 ou NT, nous vous conseillons vivement d'installer EES32

EES est distribué sous forme de fichiers auto-extractibles, les fichiers SETUP16.EXE et SETUP32.EXE correspondent respectivement aux versions 16 et 32 bits du logiciel.

Pour installer EES ou EES32, double cliquer sur l'icône illustrant les fichiers SETUP16.EXE ou SETUP32. Sous Windows 95 et NT vous pouvez également installer le logiciel en cliquant sur le bouton 'Démarrer' puis 'Exécuter' et taper la ligne de commande suivante : A:\SETUP16.exe ou A:\SETUP32.exe.



Dans ce cas A: désigne le lecteur de disquette. Laissez vous ensuite guider par le programme d'installation.

Démarrer EES

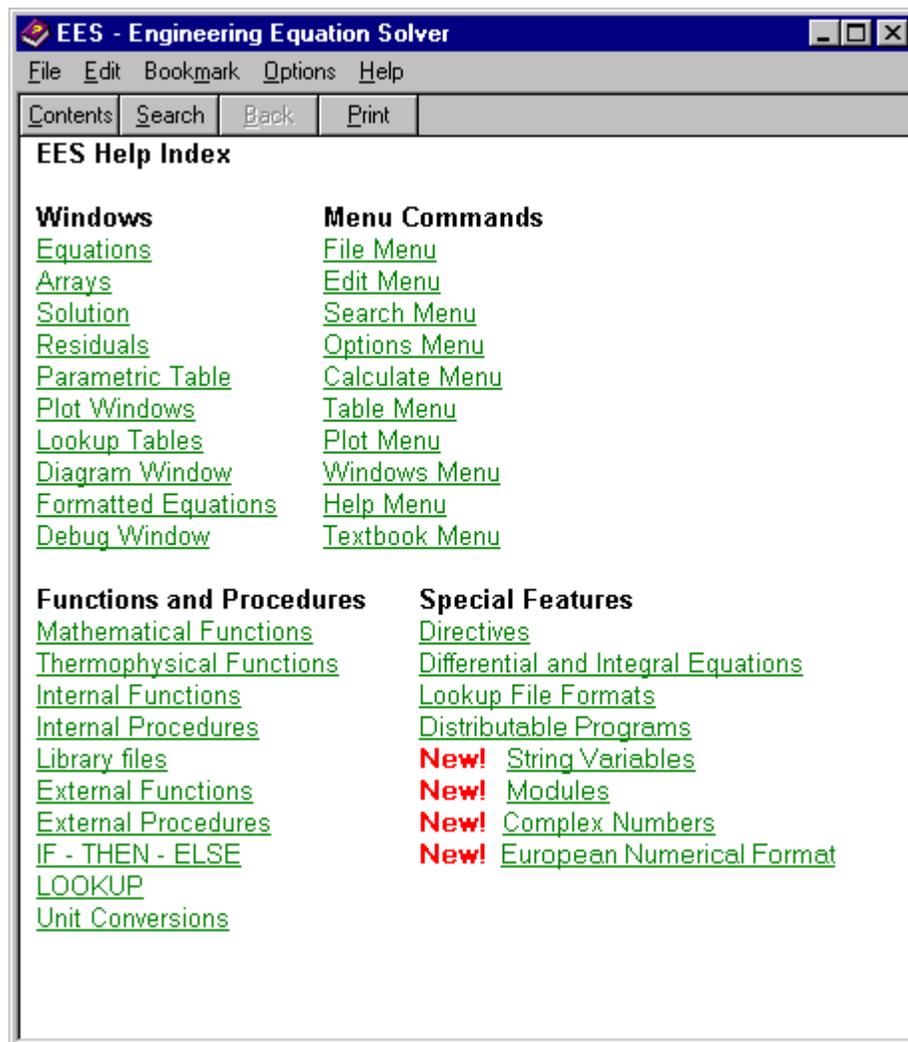


Le programme d'installation créera un dossier \EESW (16-bit version) ou \EES32 (32-bits version) contenant les fichiers nécessaires à l'exécution de EES. Pour lancer l'application, il suffit de double cliquer sur l'icône EES. EES charge automatiquement un fichier dénommé HELLO.EES. Ce fichier décrit les caractéristiques majeures de EES. Si vous désirez, vous pouvez supprimer ou modifier ce fichier.

Informations générales

Quand vous exécutez EES, une boîte de dialogue apparaît. Celle-ci contient le numéro d'enregistrement, le numéro de version. Ces informations vous seront demandées dans le cas où vous feriez appel au support technique. Cliquer sur le bouton 'OK' pour commencer une session EES.

Une aide en ligne est disponible en permanence. Pour activer l'aide, appuyer sur la touche 'F1'. Pour visualiser l'index de l'aide, cliquer sur le bouton 'Contents'. Ce fichier d'aide comporte de nombreux liens hypertexte (mots soulignés en vert).



Les commandes de EES sont réparties suivant 9 menus 'POP-UP' (une dixième commande peut être placée à droite de la commande 'Help', cf. chapitre 3). L'ensemble de ces commandes est détaillé dans le chapitre 3.



Au dessous de la barre de menu, se trouve une barre d'outils. Celle-ci comporte plusieurs icônes permettant d'accéder plus rapidement aux commandes de EES les plus fréquemment utilisées. Si vous placez le curseur sur un des boutons et patientez quelques secondes, une fenêtre décrivant la fonction du bouton apparaîtra.

La barre d'outil peut être masquée si vous le souhaitez. Pour ce faire, veuillez cliquer sur **Préférences** disponible via le menu **Options**

L'icône EES, disponible en haut à gauche de l'écran donne l'accès en cliquant avec le bouton droit de la souris aux caractéristiques de présentation de la fenêtre du programme. Il est possible de redimensionner la taille de la fenêtre, de mettre le logiciel en arrière plan ou de fermer l'application.

Le menu **Fichier** donne l'accès aux commandes d'ouverture, de fermeture, de sauvegarde, de fusion de fichiers, de chargement des bibliothèques, de configuration de l'imprimante, d'impression.

Le menu **Editer** donne l'accès aux commandes d'annulation de la dernière opération effectuée mais aussi aux commandes copier, couper, coller, suppression ou sélection de l'ensemble du document actif.

Le menu **Recherche** contient les commandes de rechercher/remplacer d'une expression dans le document actif.

Le menu **Options** permet de visualiser les informations relatives aux variables, fonctions. Il est possible de configurer le système d'unité, les limites de calculs, les plages de valeurs. La commande 'Preference' permet de configurer les formats de présentation des données, les échelles utilisées pour les tracés, etc. Une commande est aussi disponible pour configurer les fonctions utilisateurs.

Le menu **Calculer** donne l'accès aux commandes de vérification, de formatage des données et au lancement de la résolution de la série d'équations.

Le menu **Tables** donne l'accès aux commandes d'ajout, de modification, de suppression des Tables Paramétrique. Il est possible de faire des calculs de régression linéaire des données au sein de ces tables. Une table est similaire à une feuille de tableur. Les équations pourront être résolues à l'aide de celle-ci. L'utilisateur pourra faire varier la valeur des données pour de nouvelles résolutions. Ce menu offre la possibilité de visualiser, de configurer les tables saisies par les utilisateurs.

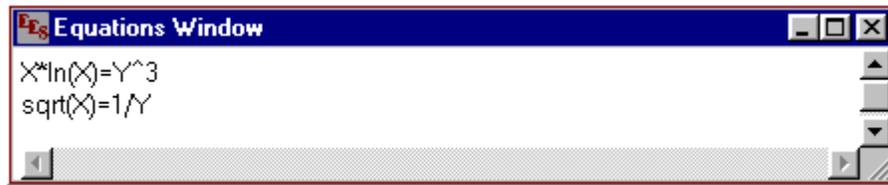
Le menu **PLOT** contient les commandes permettant d'afficher un tracé, d'en modifier son échelle. Il permet de préparer des traces pour les tables Paramétrique, les tables 'Lookup', les variables tableaux. Une fonction de 'Curve-fitting' est également disponible.

Le menu Fenêtres permet d'organiser la disposition de l'ensemble des fenêtres.

Le menu AIDE permet de visualiser l'aide en ligne.

Un des atouts fondamentaux de EES est sa capacité à résoudre des systèmes d'équations algébriques, non linéaires. Pour illustrer cela, le lecteur trouvera dans le paragraphe suivant la résolution d'un problème thermodynamique.

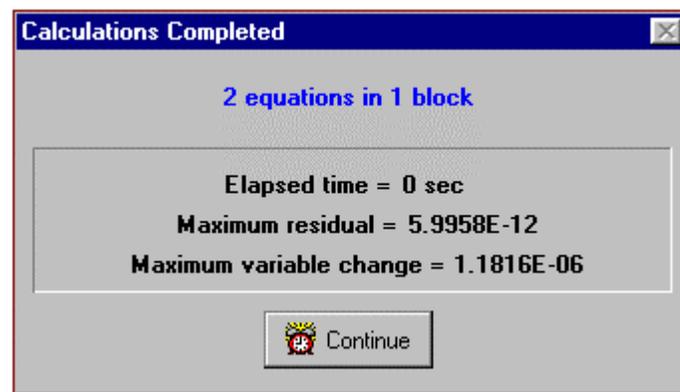
Noter, que EES ne fait pas la distinction entre les majuscules et minuscules. Les signes ^ et (**) sont indifféremment utilisés pour effectuer des puissances.



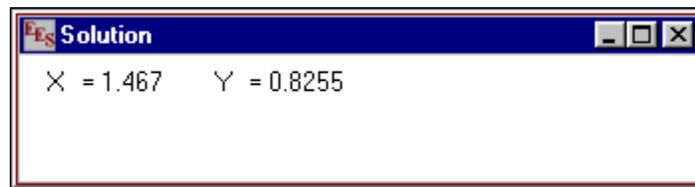
Le menu Fenêtre donne l'accès à la mise en forme mathématique d'équations.



En cliquant sur Résoudre dans le menu Calculer, une boîte de dialogue apparaît, elle indique l'avancement dans la résolution des calculs.



Cliquez sur le bouton 'Continue'. La solution du système d'équations sera alors affichée.



Où h est l'enthalpie et, $Vel^2/2$ l'énergie cinétique. Dans le système international (SI) l'enthalpie a comme unité : le kJ/kg. Nous serons donc amenés à convertir certains de nos paramètres. Par l'intermédiaire de la fonction ' CONVERTI ', EES offre la possibilité de convertir des variables dans n'importe quel système d'unité (cf. Chapitre 4).

D'autre part, nous savons de part les propriétés de notre fluide que :

$$v_1 = v(T_1, P_1) \quad (6)$$

$$h_1 = h(T_1, P_1) \quad (7)$$

$$v_2 = v(T_2, P_2) \quad (8)$$

$$h_2 = h(T_2, P_2) \quad (9)$$

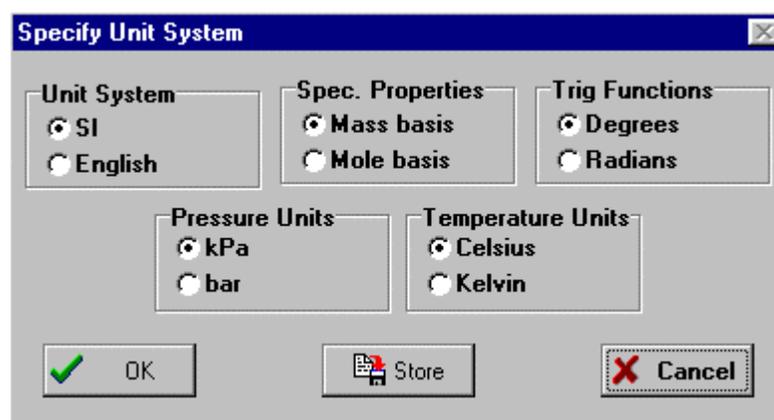
En temps normal, nous négligeons les termes contenant la vitesse car l'énergie cinétique de notre fluide est faible et donc négligeable. Par ailleurs, notre problème serait d'autant plus difficile à résoudre.

Toutefois, la difficulté de calcul n'est pas un facteur prépondérant pour EES. En effet, l'utilisateur peut résoudre le problème en tenant compte du paramètre d'énergie cinétique et juger de leur importance.

On connaît les valeurs de T_1 , P_1 , UN_1 , Vel_{11} et P_2 . Notre système comporte 9 inconnues : UN_2 , \dot{m}_1 , \dot{m}_2 , Vel_2 , h_1 , v_1 , h_2 , v_2 , T_2 et 9 équations, nous pouvons donc résoudre notre système.

Nous allons maintenant voir comment EES peut nous aider dans la résolution de notre système.

Exécuter EES, puis sélectionner Nouveau dans le menu Fichier. Une fenêtre d'Equations vierge apparaît. Avant d'entrer les équations, il convient de déterminer le système d'unités pour les fonctions et propriétés thermodynamiques incluses. Pour voir où changer le système d'unité, sélectionner Système d'unité dans le menu Options.



EES est configuré dès son initialisation pour travailler dans les unités du SI avec T en °C, P en kPa, et les valeurs de propriétés spécifiques dans leurs unités habituelles sur une base de la masse. Ces paramètres par défaut peuvent être changés. Sélectionner la commande 'Unit System' pour régler le système d'unités comme indiqué ci-dessus. Cliquer sur le bouton 'OK' pour sauver la configuration.

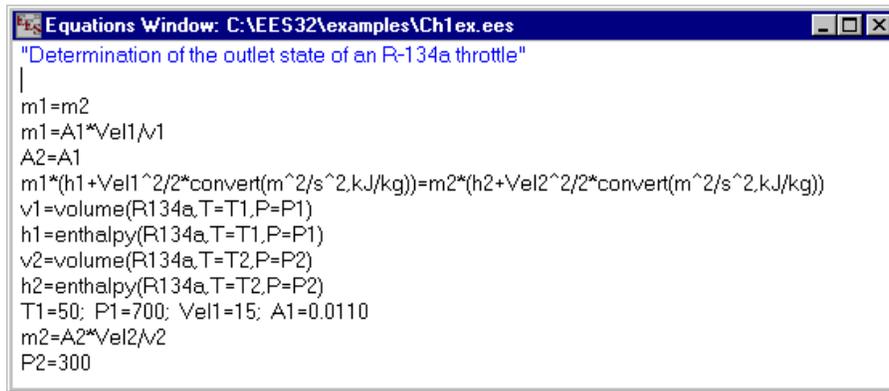
Les équations peuvent être saisies maintenant dans la fenêtre d'Equations. Le texte est entré de la même façon que dans n'importe quel traitement de texte. Voici quelques règles :

1. EES ne fait pas la distinction entre les majuscules et les minuscules.
2. Les 'retour chariot' et les espaces sont négligés.
3. Les commentaires doivent être saisis entre des crochets : { } ou entre des guillemets : " ". Il est possible d'entrer autant de lignes de commentaires que désirées ou bien d'imbriquer des commentaires entre des formules. Dans ce cas, seuls les paramètres extérieurs aux crochets seront analysés. Enfin, les commentaires apparaissent également dans la fenêtre d'Equations formatée.
4. Les noms de variables doivent commencer par une lettre et peuvent être composés de n'importe quel caractère, à l'exception de () | * / + - ^ ' { } : " ;
Une "variable tableau" (cf. chapitre 7) s'écrit de la façon suivante :
nom_variable[colonne, ligne] où colonne et ligne indiquent la taille maximum du tableau.
Les variables chaînes de caractères (cf. chapitre 7) se terminent toujours par un \$ et ont une longueur maximum de 30 caractères.
5. Les équations multiples peuvent être entrées sur une ligne si elles sont séparées par un point virgule (;)¹. La ligne doit comporter au maximum 255 caractères.
6. Les symboles ^ ou ** sont utilisés indifféremment pour indiquer une élévation à une puissance.
7. L'ordre dans lequel les équations sont entrées n'importe pas.
8. L'ordre dans lequel sont placées les variables connues ou inconnues n'importe pas.

Après avoir entré l'ensemble des équations de ce problème et (éventuellement) vérifié leur syntaxe en utilisant la commande **Vérifié / Format** dans le menu **Calculer**, la fenêtre d'Equations apparaît. Les commentaires sont normalement rédigés en bleu. Les autres

¹ EES utilise les paramètres de séparation des nombres (virgule ou point virgule) du système d'exploitation (Panneau de contrôle). Dès son ouverture, EES vérifie les paramètres régionaux, et convertira automatiquement si besoin est, les symboles de séparation qu'il utilise.

options de mise en forme des équations sont accessibles via la commande **Préférences** du menu **Options**.



```

Equations Window: C:\EES32\examples\Ch1ex.ees
"Determination of the outlet state of an R-134a throttle"
|
m1=m2
m1=A1*Vel1/v1
A2=A1
m1*(h1+Vel1^2/2*convert(m^2/s^2,kJ/kg))=m2*(h2+Vel2^2/2*convert(m^2/s^2,kJ/kg))
v1=volume(R134a,T=T1,P=P1)
h1=enthalpy(R134a,T=T1,P=P1)
v2=volume(R134a,T=T2,P=P2)
h2=enthalpy(R134a,T=T2,P=P2)
T1=50; P1=700; Vel1=15; A1=0.0110
m2=A2*Vel2/v2
P2=300

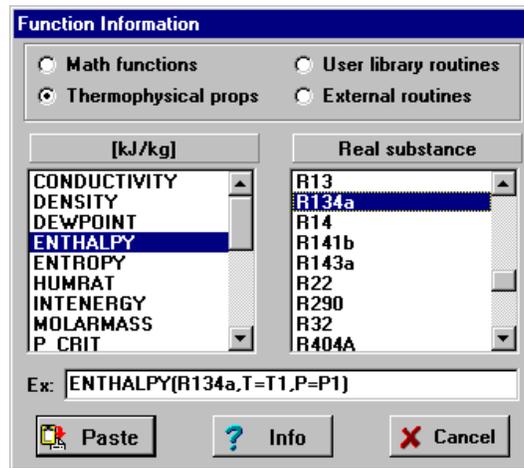
```

Noter ici, l'usage de la fonction **Convertir**. Cette fonction est utilisée dans notre exemple pour convertir les unités de l'énergie cinétique [m^2/s^2] en fonction des unités utilisées pour l'enthalpie [kJ/kg]. Cette fonction est fréquemment utilisée dans ce genre de problème (cf. Chapitre 4 pour voir sa description détaillée).

Les fonctions thermodynamiques telles l'**enthalpie** ou le **volume** nécessitent un format spécifique. Le premier argument de la fonction est le nom de substance, R134a dans ce cas. Les arguments suivants sont les variables indépendantes précédées d'une lettre unique les identifiant et d'un signe égal. Les lettres admises sont T, P, H, U, S, V, et X, correspondant à la température, la pression, l'enthalpie, l'énergie interne, l'entropie, le volume, et la qualité du fluide. (Pour les fonctions psychrométriques, les lettres admissibles sont W, R, D, et B, correspondant à la proportion d'humidité, l'humidité relative, la température de dewpoint, et la température de wetbulb.)

Pour saisir d'une façon simple les fonctions sans vous soucier de leur syntaxe, vous pouvez utiliser la commande **Information sur les fonctions** du menu **Options**. Une boîte de dialogue s'ouvrira alors, cliquer sur le bouton de fonctions thermodynamiques. Une liste de fonctions thermodynamiques apparaîtra sur la gauche de la boîte de dialogue et une liste de fluides, de composants sur la droite. Choisir alors la fonction désirée en cliquant sur son nom. Procéder de même pour le fluide.

Un exemple de fonction mise en forme apparaîtra dans la zone de texte d'exemple. Les informations présentes dans ce rectangle peuvent être changées, si besoin est. Cliquer ensuite sur le bouton 'Copy', la fonction sera immédiatement collée dans la fenêtre d'Equations au niveau de la position du curseur. Des informations supplémentaires sont disponibles en cliquant sur le bouton "Info" de cette même boîte de dialogue.



Il est possible d'indiquer à EES quelles sont les valeurs par défaut et les limites que peut prendre une variable. Pour ce faire, utiliser la commande **Information sur les variables** du menu **Options**. Avant d'afficher la fenêtre correspondante, EES vérifie la syntaxe de ou des équation(s) saisie(s) et résout toutes les équations comportant une inconnue.

La boîte de dialogue **Information sur les variables** contient une ligne pour chaque variable qui apparaît dans la fenêtre d'Equations. Par défaut, chaque variable a une valeur initiale de 1.0 bornée entre plus et moins l'infini. Notons que les valeurs des bornes sont indiquées en italique si EES a calculé précédemment la valeur des variables. Dans ce cas, la colonne de valeur initiale indique la valeur calculée de la variable. (Il est possible d'éditer ces valeurs indiquées en italique pour forcer EES à calculer de nouveau celles-ci).

Variable	Guess	Lower	Upper	Display	Units
A1	0.011	<i>-infinity</i>	<i>infinity</i>	A 3 N	m ²
A2	0.1	<i>-infinity</i>	<i>infinity</i>	A 3 N	m ²
h1	1	0.0000E+00	<i>infinity</i>	A 3 N	kJ/kg
h2	1	0.0000E+00	<i>infinity</i>	A 3 N	kJ/kg
m1	1	<i>-infinity</i>	<i>infinity</i>	A 3 N	kg/s
m2	1	<i>-infinity</i>	<i>infinity</i>	A 3 N	kg/s
P1	700	<i>-infinity</i>	<i>infinity</i>	A 3 N	kPa
P2	300	<i>-infinity</i>	<i>infinity</i>	A 3 N	kPa
T1	50	<i>-infinity</i>	<i>infinity</i>	A 3 N	C
T2	1	<i>-infinity</i>	<i>infinity</i>	A 3 X	C
v1	1	0.0000E+00	<i>infinity</i>	A 3 N	m ³ /kg

La première colonne identifiée par la lettre "A" de la colonne affichage indique que EES déterminera automatiquement le format de présentation de la valeur de la variable dans la

fenêtre de Solution. Dans ce cas, EES choisira le nombre de chiffres suffisant pour représenter la valeur (la colonne située à droite du “ A ” est inactive). Par défaut la mise en forme est automatique. Il est possible d’afficher sous forme décimale ou exponentielle (“ F ” ou “ E ”). Ces paramètres d’affichage peuvent être facilement changés via la commande **Information implicite** du menu **Options** (cf. Chapitre 3).

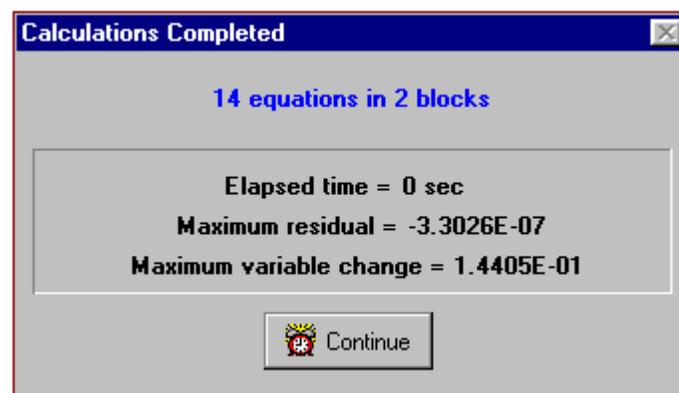
La troisième colonne d’options permet de mettre en forme (normal, gras, souligné ou encadré) la valeur de la variable dans la fenêtre de Solution.

Il est possible également de spécifier les unités pour chaque variable. Ces unités seront affichées dans la fenêtre Solution et/ou dans la table paramétrique. Notons que EES ne fait pas automatiquement les conversions d’unité, ceci est possible grâce à la fonction **converti** (cf. Chapitre 4) en effet les unités saisies ne servent que pour l’affichage.

Pour résoudre des équations de type non linéaires, il est parfois nécessaire de positionner des valeurs par défaut ainsi que des bornes afin de déterminer la solution désirée (Ceci n’est pas nécessaire dans le cas de notre exemple).

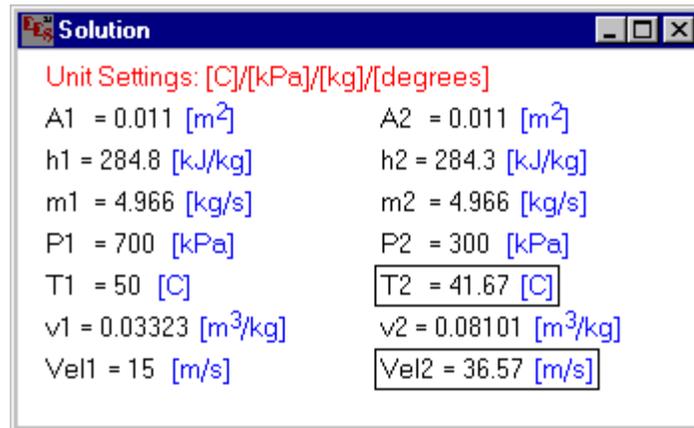
Dans notre problème, l’enthalpie à la sortie, h_2 , est pratiquement égale à celle de h_1 . Régler sa valeur initiale à 100 et sa limite inférieure à 0. Régler la valeur par défaut du volume v_2 à 0.1 et sa limite inférieure à 0. Faites défiler ensuite la liste d’information pour visualiser Vel_2 et positionner sa limite inférieure également à 0.

Pour résoudre notre système, cliquer sur la commande **Résoudre** du menu **Calculer**. Une boîte de dialogue apparaît et indique le temps de calcul, le maximum résiduel (c’est-à-dire, la différence entre les valeurs des paramètres droite et gauche d’une équation) et les variables dont la valeur a été modifiée depuis la dernière itération. Une fois les calculs terminés, EES indique le nombre d’équations qui composent le problème et le nombre de blocs. Un bloc (ou groupe) est un sous-ensemble d’équations qui peut être résolu indépendamment. EES utilise alors ces blocs quand cela est possible pour accroître l’efficacité de calcul, comme décrit dans l’Annexe B. Quand les calculs sont terminés, le bouton “Annuler” sera substitué par le bouton “Continuer”.



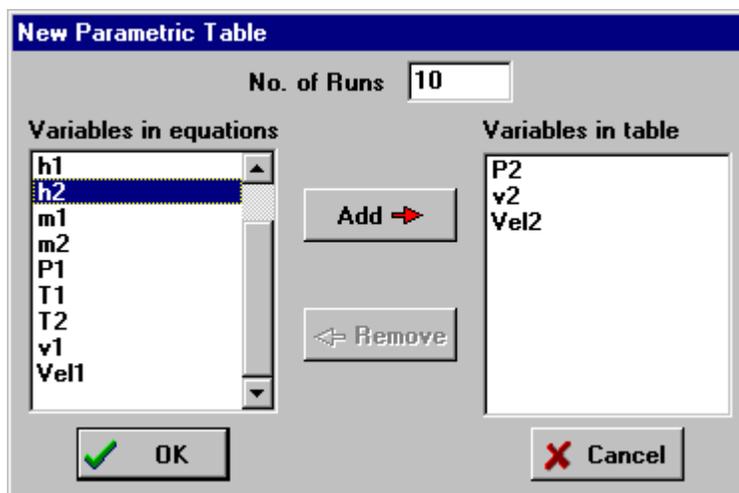
Par défaut, les calculs s’arrêtent quand 100 itérations ont été effectuées, que le temps de calcul a dépassé 60 secondes, que le maximum résiduel est inférieur à 10^{-6} ou que la valeur

de changement variable maximum est inférieur de 10^{-9} . Ces paramètres peuvent être modifiés par la commande **Critères d'arrêt** du menu **Options**. Si le maximum résiduel est supérieur à la valeur par défaut, il se peut que les équations n'aient pas été correctement résolues, probablement parce que les limites d'une ou de plusieurs variable(s) a/ont forcé la solution. Cliquer le bouton "Continuer", la boîte de dialogue se ferme et laisse apparaître la fenêtre **Solution**. Les valeurs de T2, m2, et Vel2 ont été déterminées, le problème est maintenant résolu.



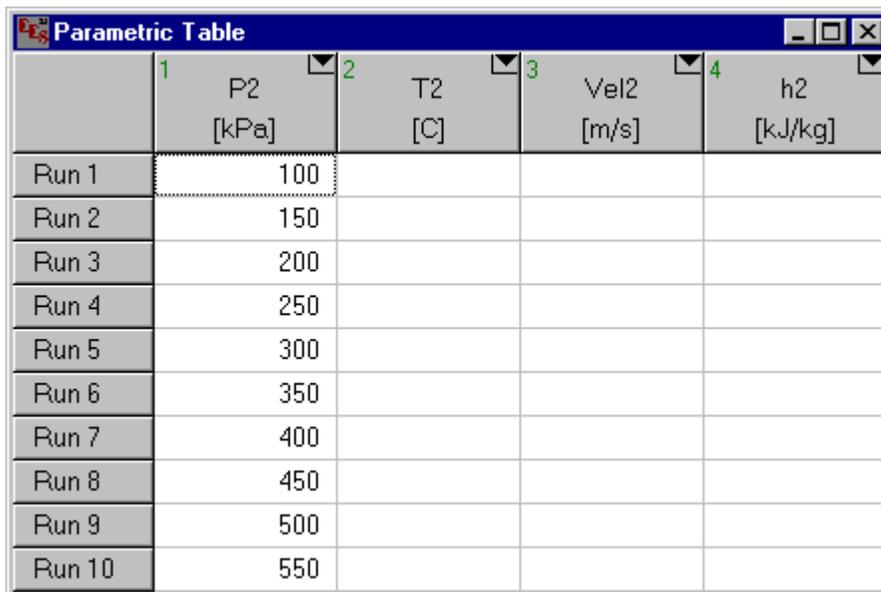
Une des caractéristiques la plus utilisée dans EES est la possibilité d'utiliser une table paramétrique. Par exemple, dans ce problème, il peut être intéressant de voir comment la vitesse et la température en sortie d'accélérateur varient selon la pression. Une série de calculs peut être automatiquement réalisée et matérialisée sous forme de graphe en utilisant les commandes du menu **Table**.

Cliquer sur la commande **Nouvelle Table**. Une boîte de dialogue apparaît. Elle contient la liste des variables utilisées dans la fenêtre d'Equations². Dans notre cas, nous allons construire une table qui contient les variables P2, T2, Vel2, et h2. Cliquer sur P2 dans la liste de variables visibles sur la gauche. P2 est maintenant souligné et le bouton "Ajouter" devient actif.



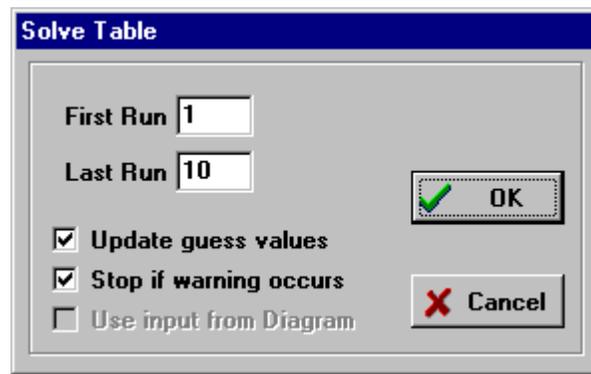
Maintenant, cliquer sur le bouton “Ajouter” pour transférer P2 de la gauche vers la droite. Faites de même pour T2, h2, et Vel2, (utiliser éventuellement la barre de défilement pour visualiser les variables non visibles dans la liste). Noter qu’un double-clic sur une variable a le même effet que les actions citées ci-dessus. La boîte de dialogue doit ressembler maintenant à l’image ci-dessus. Cliquer sur le bouton “Ok” pour continuer.

La Table Paramétrique fonctionne comme une feuille de tableur. Vous pouvez saisir des valeurs directement dans les cellules. Les valeurs saisies sont en noir et ont le même effet que si vous régliez sa valeur dans la fenêtre d’Equations. Effacer l’équation $P2 = 300$ (ou mettez la en commentaire à l’aide des crochets { }). Nous n’avons plus besoin de cette équation car nous avons initialisé sa valeur dans la table). Maintenant entrer les valeurs de P2 pour lesquelles les valeurs T2 seront déterminées. Pour cet exemple, nous choisirons des valeurs comprises entre 100 et 550. (Les valeurs peuvent être saisies automatiquement à l’aide de la commande **Itérer des Valeurs** dans le menu **Tables** ou en utilisant le bouton “Itérer” :  figurant à droite de chaque zone d’en-tête de colonnes de la table (cf. Chapitre 2)). La Table paramétrique devrait apparaître de la manière suivante :



	1 P2 [kPa]	2 T2 [C]	3 Vel2 [m/s]	4 h2 [kJ/kg]
Run 1	100			
Run 2	150			
Run 3	200			
Run 4	250			
Run 5	300			
Run 6	350			
Run 7	400			
Run 8	450			
Run 9	500			
Run 10	550			

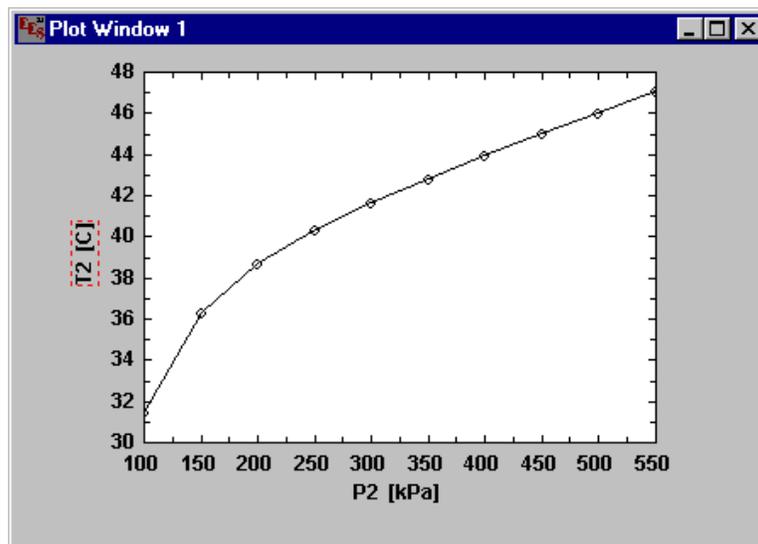
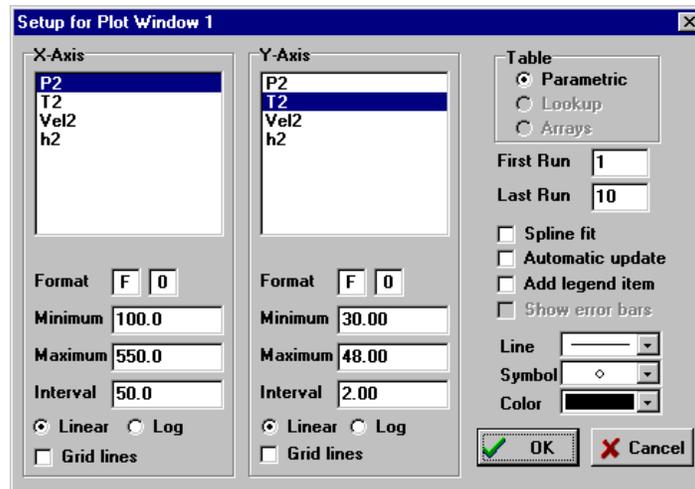
Maintenant, cliquer sur la commande **Résoudre la Table** du menu **Calculer**. La boîte de dialogue “Solve table” apparaîtra et vous permettra de choisir le nombre de pas de calcul à réaliser pour obtenir la solution.



Quand la case 'Update Guess Value' (mettre à jour les valeurs par défaut) est cochée, l'itération "n" utilisera les valeurs (Guess Value) de l'itération "n-1". Cliquer sur le bouton "Ok" pour continuer. Une fenêtre d'information sera alors affichée et indiquera l'état du processus de calcul. Une fois les calculs terminés, les valeurs de T2, Vel2, et h2 seront affichées dans la table. Les valeurs calculées par EES seront, suivant le mode d'affichage sélectionné, affichées en bleu, en gras ou en italique.

Le rapport entre les variables P2 et T2 est maintenant évident. Il est possible de le visualiser plus clairement à l'aide d'un graphique. Pour ce faire, cliquer sur la commande 'New Plot Window' du menu 'Plot'. Une boîte de dialogue apparaît comme montré ci-dessous. Sélectionner P2 en tant qu'axe des abscisses en cliquant sur 'P2' dans la liste d'axe des abscisses. Faites de même pour que T2 soit sur l'axe des ordonnées. Choisir les limites d'échelle pour P2 et T2, et régler le nombre de divisions pour l'échelle comme indiqué. Le quadrillage rend la lecture plus facile. Cliquer sur le contrôle de quadrillages pour les axes x et y. En cliquant sur le bouton "Ok", la fenêtre de Tracés apparaît. (Cf. voir ci-dessous).

	1 P2 [kPa]	2 T2 [C]	3 Vel2 [m/s]	4 h2 [kJ/kg]
Run 1	100	31.46	109.9	278.9
Run 2	150	36.32	73.79	282.2
Run 3	200	38.7	55.29	283.4
Run 4	250	40.34	44.08	284
Run 5	300	41.67	36.57	284.3
Run 6	350	42.86	31.19	284.4
Run 7	400	43.97	27.15	284.6
Run 8	450	45.03	24	284.6
Run 9	500	46.06	21.48	284.7
Run 10	550	47.07	19.42	284.7



Une fois créée, le tracé peut être représenté de différentes manières. Dans le chapitre 2, le lecteur trouvera une description détaillée de la fenêtre 'Plot' et une description du menu 'Plot' dans le chapitre 3.

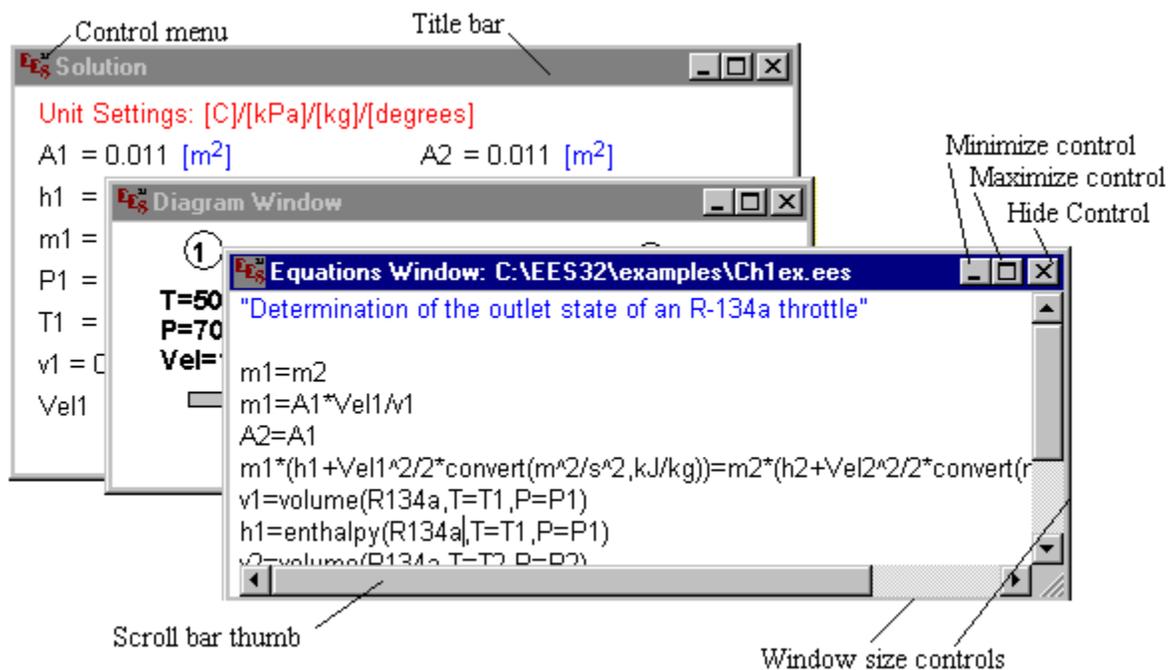
Ce petit exemple illustre une partie des capacités de EES. En suivant cet exemple, vous serez à même de résoudre beaucoup de problèmes similaires.

Toutefois, EES permet de faire encore beaucoup d'autres choses tels, tracés de courbes d'ajustement, calculs avec incertitudes, opérations avec des variables complexes, utilisation des tableaux, et bien d'autres.

Les fenêtres de EES

Informations générales

Toutes les données afférentes à un problème sont présentées dans une série de fenêtres. Les équations et les commentaires sont saisis dans la fenêtre d'Equations. Une fois les équations résolues, les valeurs des variables sont affichées dans les fenêtres de Solution et / ou de Tableaux. La valeur résiduelle des équations et l'ordre de calcul peuvent être visualisés dans la fenêtre Résiduelle. Il existe également des fenêtres pour : les tables Paramétrique, 'Lookup', jusqu'à 10 fenêtres de Tracés et une fenêtre de Debugage. Dans ce chapitre, le lecteur trouvera une description détaillée de chacune de ces fenêtres. Toutes les fenêtres peuvent être ouvertes (visibles) simultanément. La fenêtre d'avant plan est la fenêtre active, elle est identifiée par sa barre de titre soulignée (noir). La figure ci-dessous illustre l'apparence des diverses fenêtres de EES sous Windows 95 et NT 4.0 (l'apparence peut légèrement différer dans les autres versions de Windows).



Noter que, à la différence d'autres applications Microsoft, l'action de la commande "Fermer" ne fait que masquer la fenêtre. Pour la rendre visible de nouveau, utiliser le menu 'Window'.

Chaque fenêtre contient une série de commandes :

1. Pour déplacer une fenêtre, il vous suffit de cliquer avec l'un des boutons de la souris sur la barre de titre de la fenêtre. En maintenant le bouton appuyé, vous pouvez alors déplacer la fenêtre. Pour la positionner à l'endroit désiré, relâcher le bouton.
2. Pour masquer la fenêtre, cliquer sur l'icône EES dans le coin supérieur gauche de la fenêtre active. Un menu déroulement apparaît. Cliquer alors sur la commande "Fermer". (Sous Windows 95 et NT 4.0, vous pouvez cliquer sur la croix située dans le coin supérieur droit de la fenêtre active). Vous pouvez restaurer une fenêtre cachée en la sélectionnant dans le menu **Fenêtres**.
3. La commande "Maximiser" située dans le coin en haut à droite de chaque fenêtre permet de dimensionner la fenêtre à la taille de l'écran.
La commande **Restauration** permet de dimensionner de nouveau la fenêtre à sa taille originale (ces deux commandes figurent également dans le menu déroulant qui apparaît lorsque l'on clique sur l'icône EES situé en haut à gauche de la fenêtre).
4. Il est possible de modifier plus finement la taille des fenêtres: amener le curseur sur l'un des coins de la fenêtre (l'apparence du curseur change). Cliquer, puis, en maintenant le bouton appuyé, déplacer la souris.
5. Un double clique sur l'icône EES cache la fenêtre.
6. Utiliser la commande 'Cascade' dans le menu 'Window', pour afficher, déplacer, redimensionner chaque fenêtre.

La saisie des équations se fait dans une fenêtre dédiée d'une façon similaire à la saisie d'un document dans un traitement de texte classique. Les commandes d'édition, Copier, Couper, Coller sont disponibles dans le menu 'Edit'.

1. Les lignes de vide peuvent être utilisées pour améliorer la lisibilité. Pour mettre des commentaires, il suffit de les insérer entre des crochets { } ou entre des guillemets « » . Ceux ci apparaîtront dans la fenêtre de mise en forme des équations.
2. Les équations peuvent être saisies dans n'importe quel ordre. L'ordre des équations n'a pas d'effet sur la solution, puisque EES fixera les équations et les organisera de manière à optimiser le calcul (cf. Annexe B).
3. Les opérateurs mathématiques utilisés dans les équations sont conformes aux règles utilisées dans FORTRAN, C ou PASCAL. Par exemple, l'équation :

$$X = 3 + 4 * 5$$

Signifie que la variable X est égale à 23. Les symboles ^ ou ** sont utilisés pour élever un nombre, une variable à une puissance. Les arguments de fonctions doivent être entre parenthèses. A l'inverse d'autres langages, EES n'exige pas qu'une variable soit située à la gauche d'une équation. Par exemple, l'équation ci-dessous fonctionne tout à fait correctement :

$$(X - 3) / 4 = 5$$

4. EES ne fait pas de distinction entre les majuscules et les minuscules. Néanmoins, pour des raisons de présentation, EES changera la casse de toute fonction déjà compilée ou quand la commande *Verify/Format* du menu *Calculer* a été exécutée (la commande préférence du menu 'Option' permet de forcer EES à utiliser les majuscules ou minuscules).
5. Les noms de variables doivent commencer par une lettre et peuvent comporter n'importe quels caractères exceptés ' (|) * / + - ^ { } ": ;
La longueur maximum est de 30 caractères. Les variables String (chaînes de caractères) se terminent comme en langage Basic par le signe \$.
6. Les variables tableaux (arrays) sont désignées par un nom, suivi d'un ou de deux nombres entre crochets indiquant la taille maximum du tableau (en largeur et en longueur) ex : X[5,3], X est une variable tableau de largeur maximum de 5 et d'une longueur de 3 éléments. La quantité dans les crochets doit être un nombre, à l'exception des fonctions *sum*, *product* ou *Duplicate*.
7. En règle générale, les noms de variables doivent être différents de ceux utilisés pour des fonctions de EES (ex. : **pi**, **sinus**, **enthalpy**).

6. Dans la version 32 bits, le nombre total de variables ne peut excéder 5000.
7. Les équations sont, en règle générale, saisies sur une ligne et se terminent par l'appui de la touche « Entrée ». Néanmoins, il est possible de saisir une succession d'équations sur la même ligne en les séparant par un point virgule². Il est possible de visualiser celles-ci à l'aide des flèches de déplacement dans le cas où la fenêtre serait trop petite. Cependant, chaque équation doit comporter au maximum 255 caractères.
8. EES compile des équations sous la forme de pile. Une équation compilée est sauvegardée en mémoire. De fait, une équation n'a besoin d'être compilée que si elle est nouvelle, ou si elle a subi une modification. N'importe quelle erreur détectée pendant le processus de compilation ou solution sera affichée dans un message d'erreur comportant une série d'explications permettant de localiser et de comprendre l'erreur.
9. Les équations peuvent être importées ou exportées dans d'autres applications en utilisant les fonctions d'édition Couper, Copier, Coller du menu 'Edit'. Les commandes **Merge** et **Load Library** du menu **Fichier**, ou la directive **\$INCLUDE** peuvent être aussi utilisés pour importer les équations d'un fichier existant. La commande **Fusionner** importera les équations d'un fichier EES ou d'un fichier texte et les placera dans la fenêtre d'Equations à la position du curseur. Les équations importées avec la directive **\$INCLUDE** n'apparaîtront pas dans la fenêtre d'Equations.
10. En cliquant à l'aide du bouton droit de la souris sur une équation, vous pouvez mettre celle-ci en commentaire (dans le cas où elle serait déjà en commentaire, cette action permet des les enlever).
11. Si EES est configuré pour opérer en mode complexe, toute variable est supposée comprendre une partie réelle et une partie imaginaire. La configuration du mode complexe peut être changée via la commande **Preferences** du menu **Options** ou en utilisant la directive **\$Complex**.

² Si la virgule est choisie comme Symbole Décimal dans le panneau de contrôle du système d'exploitation (Réglages Régional), EES reconnaîtra la virgule (plutôt qu'un point) comme séparateur décimale, le point virgule (plutôt que la virgule) comme séparateur d'argument, et la barre verticale | (plutôt que le point virgule) comme séparateur d'équation.

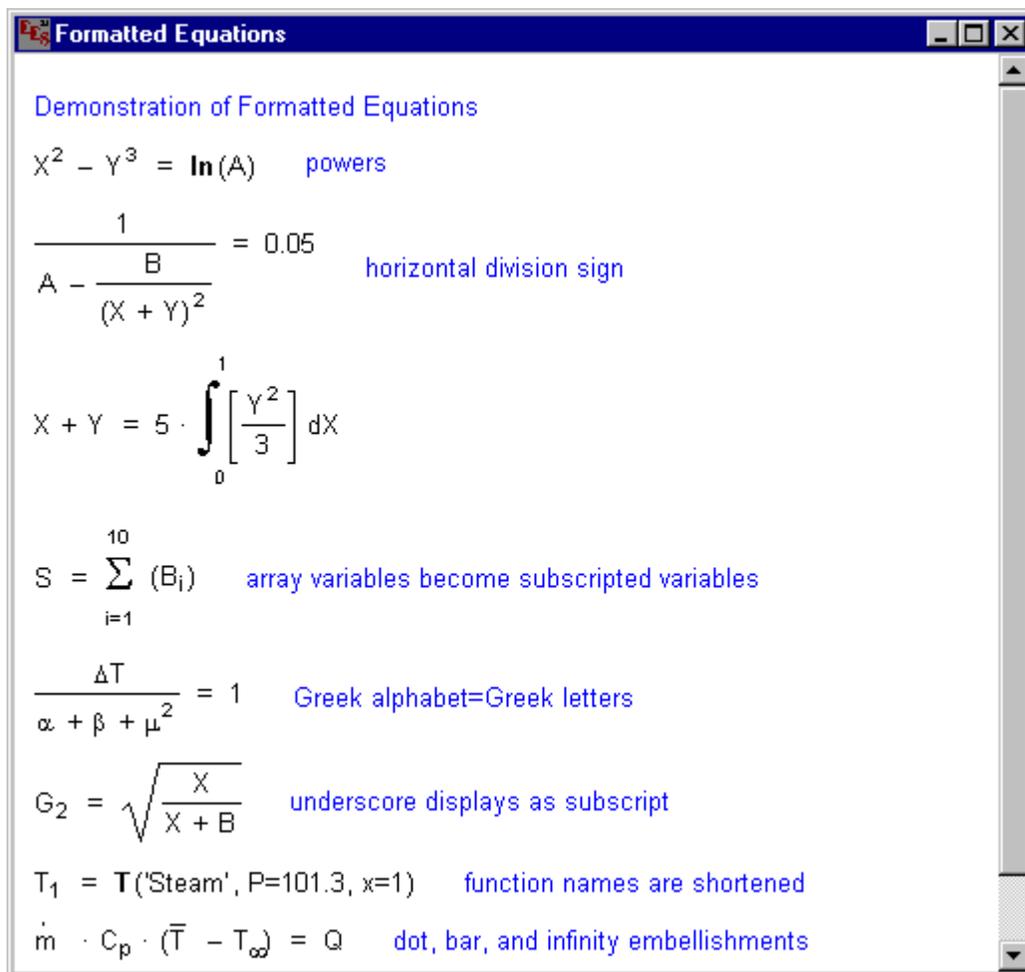
La fenêtre de mise en forme d'équations

Cette fenêtre affiche les équations saisies sous la forme mathématique usuelle.

```

Equations Window: C:\EES32\manual\format.ees
"Demonstration of Formatted Equations"
X^2-Y^3=ln(A) "powers"
1/(A-B/(X+Y)^2)=0.05 "horizontal division sign"
X+Y=5*integral(Y^2/3,X,0,1) {This comment will not appear}
S=sum(B[i],i=1,10) "array variables become subscripted variables"
DELTAT/(alpha+beta+mu^2)=1 "Greek alphabet=Greek letters"
G_2=sqrt(X/(X+B)) "underscore displays as subscript"
T[1]=temperature(Steam,P=101.3, x=1) "function names are shortened"
m_dot*C_p*(T_bar-T_infinity)=Q "dot, bar, and infinity embellishments"

```



Noter que seuls les commentaires placés entre guillemets sont affichés dans cette fenêtre. EES dispose d'un grand nombre de caractéristiques permettant d'améliorer la présentation de vos problèmes.

Les variables tableaux, tel que B[1] sont (facultativement) représentées ici comme les variables comportant un _ ; par exemple G[2] et G_2 sont deux variables différentes, mais seront représentées de la même façon.

Les signes de sommation et d'intégrale, etc. sont représentés suivant leur forme usuelle. Le symbole _ qui peut être ajouté à un nom de variable est considéré comme symbole d'indice et il sera affiché comme tel. Par exemple G_2 sera affiché G_2 .

Si vous écrivez _dot ou _bar à la suite du nom de la variable, celle-ci sera respectivement affichée avec le symbole de dérivé ou de barre. Un résultat infini (_infinity) sera affiché à l'aide du symbole ∞ .

Par ailleurs, les variables qui portent un nom provenant de l'alphabet grec seront présentées avec la lettre grecque correspondante. Par exemple, une variable bêta sera représentée comme β dans la fenêtre de mise en forme des équations. Si le nom d'une variable dans la fenêtre d'Equations est saisi entièrement en lettres capitales, et si la lettre grecque capitale est distincte de l'alphabet anglais, la lettre grecque capitale sera utilisée. Par exemple, la variable GAMMA apparaîtra Γ .

Une forme spéciale a été prévue pour les variables commençant par DELTA. Par exemple, DELTAT sera présenté ΔT .

Comme la représentation de BETA est identique à la lettre B, EES convertira BETA en β .

Il est possible de déplacer n'importe quel élément (commentaires, équations, etc.) au sein de cette fenêtre. Pour ce faire, sélectionner à l'aide de la souris l'élément que vous désirez déplacer, en maintenant le bouton gauche de la souris enfoncé, déplacer alors celui-ci à l'endroit désiré. Chaque entité (équations, commentaires) de cette fenêtre est considérée comme un élément MetaFilePict de Windows. En d'autre terme, vous pouvez copier tout ou partie du contenu de la fenêtre dans d'autres applications Windows tels Word, Excel, Paint. Pour copier une équation, il vous suffit de la sélectionner à l'aide du bouton gauche de la souris. L'équation ou le commentaire sélectionné(e) apparaîtra en inverse vidéo. Vous pouvez également choisir plusieurs commentaires, équations à l'aide de la commande SELECT du Menu EDITION. Copier, alors, la partie sélectionnée dans le presse-papiers en utilisant la commande "Copier". Les équations seront désélectionnées après cette opération. Les commentaires apparaissent normalement en bleu dans la fenêtre d'Equations mise en format et ils apparaîtront en couleur quand ils seront copiés au presse-papiers. Si vous souhaitez ne pas voir vos commentaires en couleur, appuyer sur la touche majuscule en effectuant la commande 'Copy'.

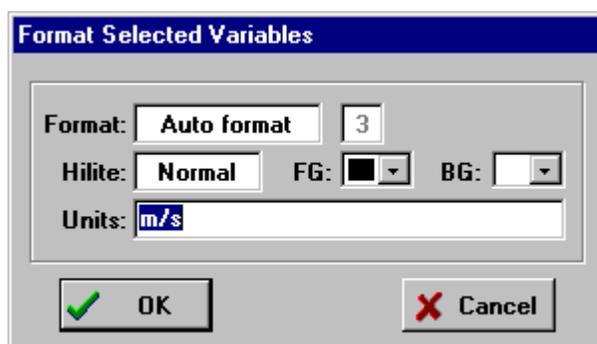
Le texte dans la fenêtre d'Equations mises en format ne peut pas être édité. Néanmoins, en cliquant avec le bouton droit de la souris sur la zone que vous désirez rééditer, vous aurez directement accès à la fenêtre d'Equations.

La fenêtre de Solution s'affichera automatiquement en avant plan une fois les calculs effectués. Les valeurs et les unités de chacune des variables de la fenêtre d'Equations apparaîtront dans l'ordre alphabétique. Par ailleurs, il est possible de réajuster la taille des colonnes dans cette fenêtre.

Le format des variables et leurs unités peuvent être modifiés à l'aide de la commande "Information" sur les variables du menu 'Option'.

Pour sélectionner ou désélectionner une variable, il vous suffit de cliquer avec le bouton gauche de la souris sur la variable à modifier (celle-ci apparaît alors en inverse vidéo).

En double cliquant sur une variable, la boîte de dialogue de mise en forme de variables s'affiche. Vous pouvez donc modifier l'ensemble des paramètres. Les changements opérés dans la boîte de dialogue seront appliqués à l'ensemble des variables sélectionnées. (Rem : l'appuie sur la touche « Entrée » entraînera l'ouverture de cette même boîte de dialogue).



Le format numérique (le style et le nombre de chiffres), les unités des variables sont modifiables dans la boîte de dialogue de mise en forme des variables. Si EES est configuré en mode complexe, une option supplémentaire est disponible pour afficher les variables complexes en coordonnées rectangulaires ou polaires. Il est possible également de cacher, de changer la graisse ou la couleur du premier et du dernier plan des variables sélectionnées.

D'autre part :

1. La fenêtre de Solution est accessible seulement si les calculs ont été effectués.
2. Le système d'équation utilisé pour les calculs est affiché dans la fenêtre Solution dans le cas où des fonctions ou des propriétés thermodynamiques, ou des fonctions trigonométriques auraient été utilisées.
3. La fenêtre de Solution sera automatiquement masquée si l'utilisateur modifie les équations initiales. Néanmoins, il est possible de laisser en permanence visible la fenêtre de Solution par l'intermédiaire de la commande **preference** du menu **Options**.

4. Si EES est incapable de résoudre le système d'équations et affiche une erreur, le titre de la fenêtre de Solution sera changé en 'Last Iteration Values' et seulement la valeur des variables calculées lors de la dernière itération sera affichée.
5. Quand la fenêtre de Solution est affichée, la commande 'Copy Solution' apparaît dans le menu 'Edit'. Celle-ci permet de copier l'ensemble des éléments de la solution ou uniquement les variables sélectionnées auparavant. Libre à vous, ensuite de la coller dans d'autres applications, soit sous forme de texte par le biais de la commande 'Paste', ou sous forme de graphique par le biais de la commande 'Collage Spécial'.

EES permet l'usage de variables de type tableaux. Ces variables possèdent un index figurant entre crochets [] ; par exemple, X [5] signifie que X est une variable de type tableaux à une dimension ; et Y[6,2] dans ce cas Y est une variable de type tableaux à deux dimensions. Ces variables se comportent comme des variables ordinaires. Elles comportent une valeur initiale, des valeurs limites hautes et basses et des propriétés d'affichage.

Il est possible (cf. Chapitre 7) de réaliser des opérations arithmétiques simples sur les indices de tableaux.

Les valeurs de toutes variables y compris les variables de type tableaux sont normalement affichées dans la fenêtre de Solution après les calculs. Néanmoins, il est possible de présenter ces variables dans une fenêtre Tableaux plutôt que dans la fenêtre Solution (cette option est paramétrable dans la boîte de dialogue Préférence) comme présenté ci-dessous :

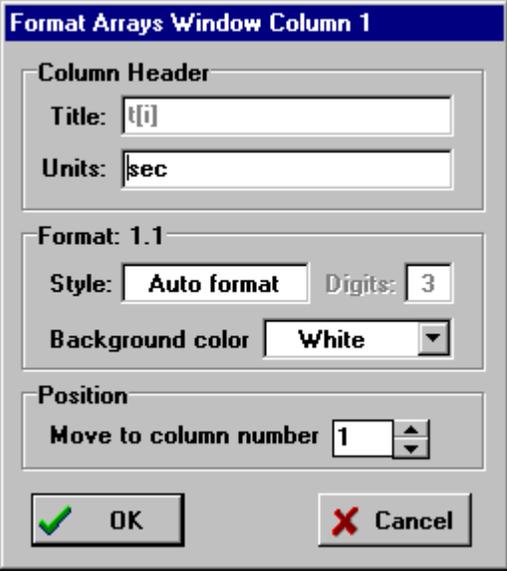
	1 t_i [sec]	2 y_i [observed]	3 $y_{p,i}$ [predicted]
[1]	1.1	3	2.998
[2]	1.2	2.9	3.16
[3]	1.3	3.6	3.322
[4]	1.9	4.2	4.297
[5]	2.3	5.1	4.948
[6]	3.1	5.9	6.257
[7]	3.3	7	6.586
[8]	4.1	7.8	7.903
[9]	4.4	8	8.399
[10]	4.6	9.1	8.73

Les valeurs dans la fenêtre d'Ensembles peuvent être visualisées sous forme de tracés en utilisant la commande 'New Plot' du menu 'Plot'.

Tout ou partie des données du tableau peut être utilisé(e) dans une autre application, par le biais des fonctions Copier/Coller. Pour ce faire, il vous suffit de sélectionner la zone à copier à l'aide de la souris, puis de copier celle-ci dans le presse-papiers. Si vous souhaitez inclure le nom de colonne et les unités dans chacune des colonnes, maintenir la touche Ctrl appuyée lors de la copie.

Le format des valeurs de n'importe quelle colonne peut être modifié. Cliquez sur le nom de la colonne à l'aide du bouton gauche de la souris : la boîte de dialogue suivante apparaît et permet de modifier la présentation, les unités des valeurs, etc.

Noter enfin, que vous avez la possibilité d'entrer des nombres dans un champ d'une colonne.



The image shows a dialog box titled "Format Arrays Window Column 1". It is divided into three sections: "Column Header", "Format: 1.1", and "Position".

- Column Header:** Contains two text input fields. The "Title" field contains the text "t[i]" and the "Units" field contains the text "sec".
- Format: 1.1:** Contains a "Style" dropdown menu set to "Auto format", a "Digits" spinner box set to "3", and a "Background color" dropdown menu set to "White".
- Position:** Contains a "Move to column number" spinner box set to "1".

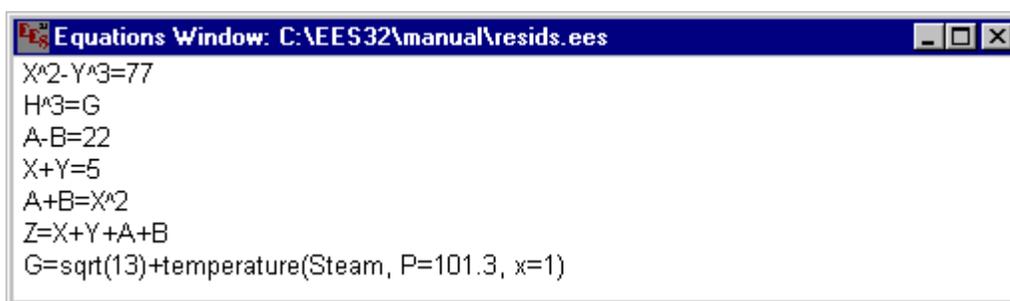
At the bottom of the dialog box are two buttons: "OK" with a green checkmark icon and "Cancel" with a red X icon.

La fenêtre Résiduelle (Residual Window)

La fenêtre Résiduelle indique le bloc d'équation, l'ordre de calcul utilisé par EES, ainsi que les valeurs relatives et absolues résiduelles. La valeur absolue résiduelle d'une équation est la différence entre les valeurs situées à gauche et à droite d'une équation. La valeur résiduelle relative est déterminée en faisant la division entre la valeur résiduelle absolue par la valeur de l'expression située à gauche d'une équation.

La valeur résiduelle relative sert à déterminer si la résolution d'une équation est terminée en fonction des critères d'arrêt spécifiés dans le menu **Options**.

Considérons, par exemple, le système suivant de six équations à six inconnues.

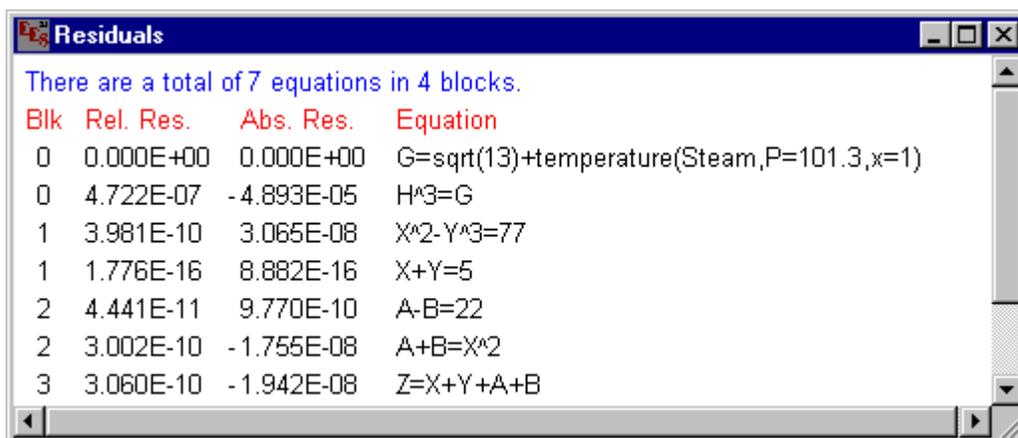


```

X^2-Y^3=77
H^3=G
A-B=22
X+Y=5
A+B=X^2
Z=X+Y+A+B
G=sqrt(13)+temperature(Steam, P=101.3, x=1)

```

EES reconnaîtra que ces équations peuvent être groupées, c'est-à-dire, combinées en deux ou plusieurs systèmes (cf. pour plus de détails se rapporter à l'annexe B). Ces informations apparaissent dans la fenêtre Résiduelle.



Blk	Rel. Res.	Abs. Res.	Equation
0	0.000E+00	0.000E+00	G=sqrt(13)+temperature(Steam,P=101.3,x=1)
0	4.722E-07	-4.893E-05	H^3=G
1	3.981E-10	3.065E-08	X^2-Y^3=77
1	1.776E-16	8.882E-16	X+Y=5
2	4.441E-11	9.770E-10	A-B=22
2	3.002E-10	-1.755E-08	A+B=X^2
3	3.060E-10	-1.942E-08	Z=X+Y+A+B

Les variables dont la valeur peut être déterminée directement, tel que G dans l'exemple ci-dessus, sont assignées en tant que Block 0³. Une fois G connu, H peut être déterminé.

³ Les variables spécifiées dans la fenêtre de Diagramme sont identifiées avec un D plutôt qu'un numéro de bloc. (Voir la section de Fenêtre de Diagramme). Dans le mode Complexe, chaque équation est affichée deux fois, une fois pour la partie réelle identifiée par (r) et une fois pour la partie imaginaire identifiée par (i)

L'ordre dans lequel apparaissent les équations individuelles résolues dans le Bloc 0 dans la fenêtre Résiduelle indique l'ordre dans lequel EES a fait ses calculs. Après avoir résolu toutes les équations du Bloc 0, EES résoudra simultanément les équations du Bloc 1, puis du Bloc 2, et ainsi de suite jusqu'à ce que toutes les équations soient résolues. La première et la troisième équation servant à déterminer X et Y peuvent se résoudre indépendamment des autres équations. Elles apparaissent ainsi dans le Bloc 1. De même, la seconde et la quatrième équation qui déterminent A et B sont placées dans le Bloc 2. Avec X, Y, A, et B maintenant connus, Z peut être déterminé, il apparaît donc dans le Bloc 3.

La fenêtre Résiduelle est normalement masquée si n'importe quel changement est effectué dans la fenêtre d'Equations. Cette option peut être modifiée dans le menu **Options**.

Il est possible de montrer la fenêtre Résiduelle lors d'un débogage. Si le nombre d'équations est inférieur au nombre d'inconnus, EES ne pourra pas résoudre le système. Toutefois, la fenêtre Résiduelle peut être rendue visible en la sélectionnant dans le menu 'Window'. Normalement, les numéros de bloc apparaissent dans l'ordre séquentiel. Quand une ou plusieurs équation(s) manque(nt), EES ne numérotera pas le bloc pour lequel il a rencontré ce problème. Les équations dans les blocs suivants devront être soigneusement réexaminées pour déterminer si elles sont correctement et complètement saisies.

Les informations figurants dans la fenêtre Résiduelle sont très utiles, lorsque nous sommes confrontés à des problèmes de convergence d'équations. L'examen de la valeur résiduelle indiquera quelle équation n'a pas été résolue. De cette façon, il sera possible d'identifier le bloc d'équations que EES n'a pas pu résoudre. Vérifier ces équations pour être sûr qu'elles possèdent une solution. Vous pouvez avoir besoin de changer les valeurs par défaut ou limites des variables du bloc erroné.

En cliquant deux fois sur le bouton gauche de la souris (ou sur le bon bouton de souris) sur une équation de la fenêtre Résiduelle vous pourrez positionner le curseur dans la fenêtre d'Equations l'emplacement de l'équation sélectionnée.

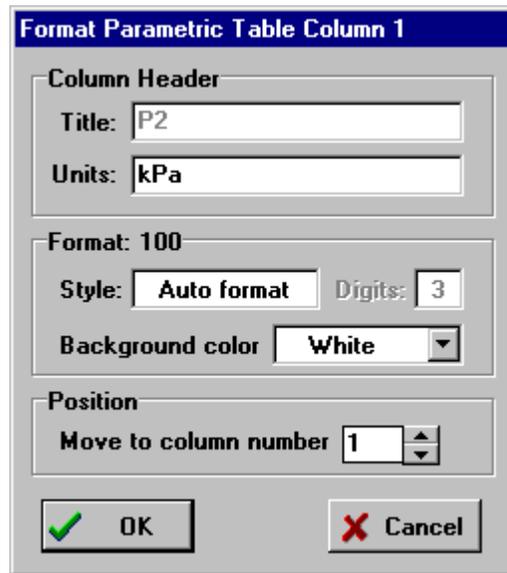
Enfin, le contenu de la fenêtre d'Equations peut être copié dans le presse-papiers en utilisant la commande 'Copy Residual Window' du menu 'Edit'.

La fenêtre de la table paramétrique

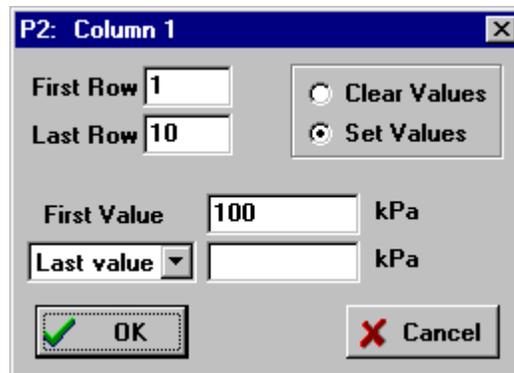
	1 P2 [kPa]	2 T2 [C]	3 Vel2 [m/s]	4 h2 [kJ/kg]
Run 1	100	31.46	109.9	278.9
Run 2	150	36.32	73.79	282.2
Run 3	200	38.7	55.29	283.4
Run 4	250	40.34	44.08	284
Run 5	300	41.67	36.57	284.3

La fenêtre “Parametric Table” contient une table paramétrique. Les valeurs numériques peuvent être saisies dans n’importe quelle cellule. Les valeurs entrées, ex., les valeurs dans la colonne P2 de la table mentionnée ci-dessus, sont supposées être des variables indépendantes et sont affichées. Ces valeurs sont affichées selon le jeu de caractères par défaut (il est possible de le modifier dans la commande **Preferences** du menu **Options**). Si vous entrez une valeur d’une variable dans la Table Paramétrique cela produira le même effet que si celle-ci avait été assignée dans la fenêtre d’Equations. Les variables dépendantes seront déterminées et seront affichées dans la table en bleu, en gras ou en italique (suivant les paramètres du menu **OPTIONS**) quand la commande **Resolve table** (résoudre la table) ou **Min/Max Table** du menu **Calculate** a été activée.

1. Une table est créée par la commande **New Parametric Table** du menu **Tables**. Les variables qui doivent apparaître dans la table sont choisies dans une liste des variables contenues dans la fenêtre d’Equations.
 2. Chaque ligne de la Table Paramétrique est issue d’un calcul séparé. Le nombre de lignes de la table est fixé lors de la création de celle-ci. Il est possible de modifier le nombre de lignes d’une table à l’aide de la commande **Insert/Delete Runs** (ajouter/supprimer des pas de calculs) du menu **Tables**.
 3. Des variables peuvent être ajoutées ou effacées en utilisant la commande **Insert/Delete Vars** (ajouter/supprimer des variables) du menu **Tables**.
 4. L’ordre initial dans lequel les colonnes de la Table Paramétrique apparaissent est déterminé par l’ordre dans lequel les variables de la table ont été choisies dans la boîte de dialogue **New Parametric Table**. Pour changer l’ordre de numérotation des colonnes, cliquer à l’aide du bouton gauche de la souris dans la cellule d’en-tête de colonne. Une boîte de dialogue apparaît comme indiqué ci-dessous vous permettant de modifier la disposition des colonnes (section **Position** de la boîte de dialogue).
- Le format d’exposition, les unités, et la couleur d’arrière-plan de colonne peuvent être changées dans cette même boîte de dialogue.



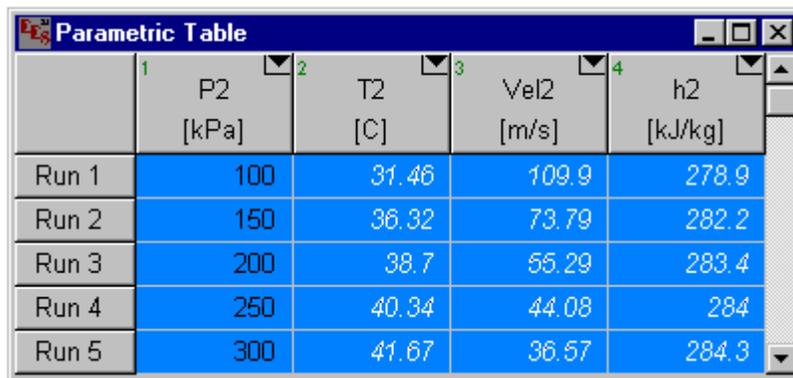
5. Les valeurs peuvent être entrées automatiquement grâce à la commande **Alter Values** (modifier les valeurs) du menu **Tables**. Comme alternative, vous pouvez cliquer sur le bouton  situé sur la zone d'en-tête de colonnes. La fenêtre suivante apparaît :



6. Une ligne résultante de la somme des colonnes peut être rendue visible ou invisible par la commande **Preferences**. Cocher ou décocher la case 'Include a Sum Row in Parametric table'
7. Une Table Parametrique peut être utilisée pour résoudre des équations différentielles ou des intégrales. Voir le Chapitre 7 pour de plus amples informations.
8. La fonction **TableValue** retourne la valeur d'une cellule de la table suivant son numéro de ligne et de colonne spécifié en paramètres.
9. La fonction **TableRun#** retourne le numéro de ligne de la table dans laquelle des calculs sont en train d'être effectués.

10. Des données en tableau peuvent être importées ou exportées de ou dans la Table Paramétrique via le presse-papiers en utilisant les commandes **Copy** et **Paste** (copier/coller) du menu **Édit.** Pour copier des données de n'importe quelles tables de EES, cliquer dans la cellule supérieure gauche. Maintenir la touche majuscule appuyée puis cliquer sur la cellule inférieure droite (utiliser la barre de défilement si nécessaire). Les cellules sélectionnées sont affichées en vidéo inverse. Vous pouvez également choisir la commande **SELECT ALL** du menu **EDITION** pour sélectionner directement l'ensemble de la table.

Les données sont placées dans le presse-papiers. Elles sont séparées par une tabulation et un retour chariot à chaque fin de ligne. Ainsi, il est possible de copier ces données dans n'importe quel tableur. Si vous souhaitez inclure le nom de colonne et les unités, maintenez la touche **Ctrl** enfoncée en exécutant la commande 'Copy'.



	1 P2 [kPa]	2 T2 [C]	3 Vel2 [m/s]	4 h2 [kJ/kg]
Run 1	100	31.46	109.9	278.9
Run 2	150	36.32	73.79	282.2
Run 3	200	38.7	55.29	283.4
Run 4	250	40.34	44.08	284
Run 5	300	41.67	36.57	284.3

La table 'Lookup' est en quelque sorte une base de données dans laquelle on peut extraire des informations et les utiliser dans des équations. Une table 'Lookup' est créée par la commande **New Lookup Table** du menu **Tables**. Le nombre de lignes et de colonnes est spécifié lors de la création de la table (il est possible de modifier le nombre de lignes à l'aide de la commande **Insert/Delete Rows** ou le nombre de colonnes à l'aide de la commande **Insert/Delete Cols**).

Une table 'Lookup' peut être sauvegardée (indépendamment du fichier EES) en utilisant la commande **SaveLookup**. Deux types de fichiers sont utilisables : un fichier d'extension **.LKT** (fichier binaire utilisé pour concevoir la table) et un fichier **.TXT** (de type ASCII contenant uniquement les données). Ces deux types de fichiers peuvent être ouverts indifféremment dans une session EES.

De nombreuses fonctions sont afférentes à la table 'Lookup'. La fonction **Interpolate** permet d'effectuer des interpolations ou des extrapolations linéaires, quadratiques ou cubiques (voir le Chapitre 4 pour plus de détails).

Les fonctions **Lookup**, **LookupCol**, et **LookupRow** permettent aux données d'être interpolées linéairement et utilisables dans des équations. La table 'Lookup' peut être contenue dans la fenêtre 'Lookup Table Window' ou dans un fichier de type **.LKT** (cf. Chapitre 4).

	1 Temp [C]	2 Time [sec]	3 Pos [m]
Row 1	100.0	0.00	5.50
Row 2	120.0	1.00	5.86
Row 3	140.0	2.00	6.11
Row 4	160.0	3.00	6.36
Row 5	180.0	4.00	6.58

Un exemple de table 'Lookup' est montré ci-dessus. Le numéro de colonne est affiché en petit caractère dans le coin supérieur gauche de chaque en-tête de colonnes. Ce numéro est utile pour utiliser les fonctions afférentes à la table 'Lookup'. Par ailleurs, ces fonctions acceptent le nom de la colonne ('ColumnName') à la place de leur numéro. Le nom de la colonne apparaît au milieu de chaque en-tête de colonnes (ex. T).

Initialement les noms de colonnes sont **Column1**, **Column2**, etc. Ils peuvent être naturellement modifiés en cliquant sur l'en-tête de colonnes.

The screenshot shows a dialog box titled "Format Lookup Table Column 1". It is organized into three main sections:

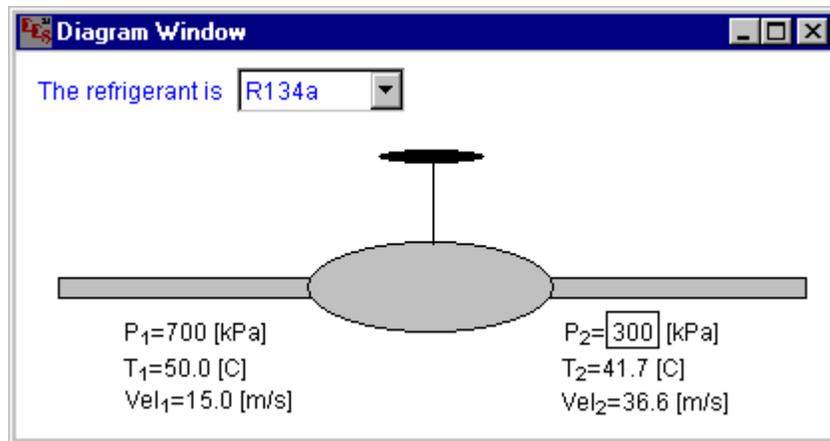
- Column Header:** Contains two text input fields. The first is labeled "Title:" and contains the text "Temp". The second is labeled "Units:" and contains the text "C".
- Format:** Shows the current format as "100.0". It includes a "Style:" dropdown menu set to "Fixed decimal", a "Digits:" input field set to "1", and a "Background color" dropdown menu set to "White".
- Position:** Contains a "Move to column number" input field with a spinner, currently set to "1".

At the bottom of the dialog are two buttons: "OK" (with a green checkmark icon) and "Cancel" (with a red X icon).

Il est possible de changer également les unités des valeurs dans chacune des colonnes. La section Format permet de fixer le format d’affichage des nombres ainsi que la couleur de fond des cellules. A l’instar de la table paramétrique il est possible de modifier la disposition des colonnes (section Position).

De même, il est possible d’ajouter, d’importer ou d’exporter des données (voir plus haut dans ce chapitre dans la section réservée à l’utilisation de la table paramétrique). Enfin, la table ‘Lookup’ peut supprimée de la mémoire en utilisant la commande **Delete Lookup** (effacer la table). Noter qu’il est impossible de supprimer des fichier .LKT ou .TXT dans EES.

La fenêtre de Diagramme peut être utilisée suivant deux façons. Premièrement, elle est utilisée pour afficher une représentation graphique ou textuelle du problème traité. Par exemple, un diagramme de chauffage. Par ailleurs, il est possible d'ajouter de l'interaction avec un diagramme. On peut saisir et afficher des valeurs. Ci-dessous est illustré un schéma dans lequel on peut saisir des valeurs et voir les résultats après résolution :



Le dessin de schéma ne peut se faire dans EES. Il convient d'utiliser des logiciels de dessin tels MSPaint, Microsoft Draw (disponible dans Word), Corel Draw, utilitaire de dessin de Power Point, etc.. il est possible également d'incorporer une image scannée dans la fenêtre de Diagramme.

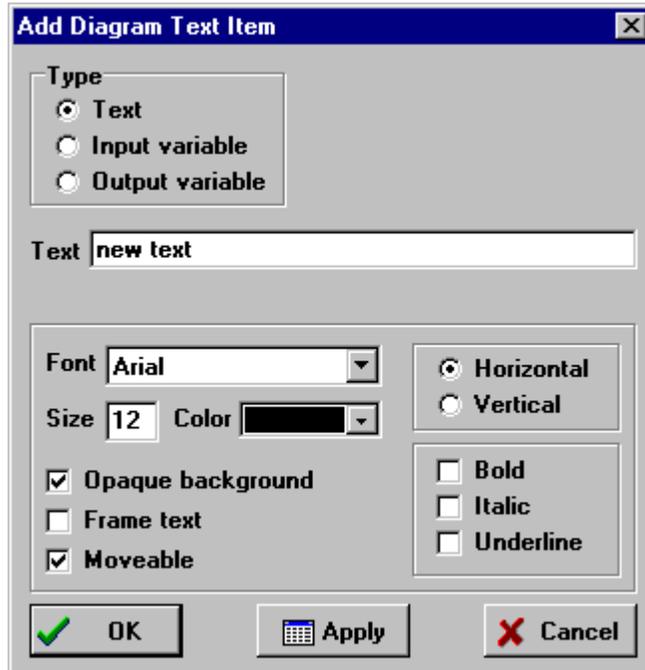
Pour ce faire, réaliser votre schéma à l'aide d'un outil quelconque, copier le dans le presse-papiers puis utiliser la commande PASTE de EES pour incorporer votre création. Noter que le diagramme sera sauvegardé en même temps que le fichier EES en cours d'utilisation.

Le diagramme peut être déplacé dans la fenêtre de Diagramme en cliquant sur celui-ci. N'importe quel texte positionné à côté du diagramme sera déplacé également.

Il est possible de redimensionner le diagramme à la taille de la fenêtre qui le contient. Pour ce faire, double cliquer dans une zone différente d'une zone de texte. Les proportions du diagramme sont conservées lors du redimensionnement.

La commande Add Diagram Text du menu Options permet d'ajouter une zone de texte n'importe où dans la fenêtre de Diagramme. Trois types de texte peuvent être affichés : commentaire, variable(s) d'entrée (Input variable), variable(s) de sortie (output variable). La boîte de dialogue "Add Diagram Text Item" permet de sélectionner les variables à afficher ainsi que leurs aspects (type et taille de police, couleur, etc.) La zone de texte sélectionnée

apparaîtra dans la fenêtre de Diagramme, il sera possible par la suite de la déplacer, d'en modifier ses paramètres (en double-cliquant sur celle-ci).

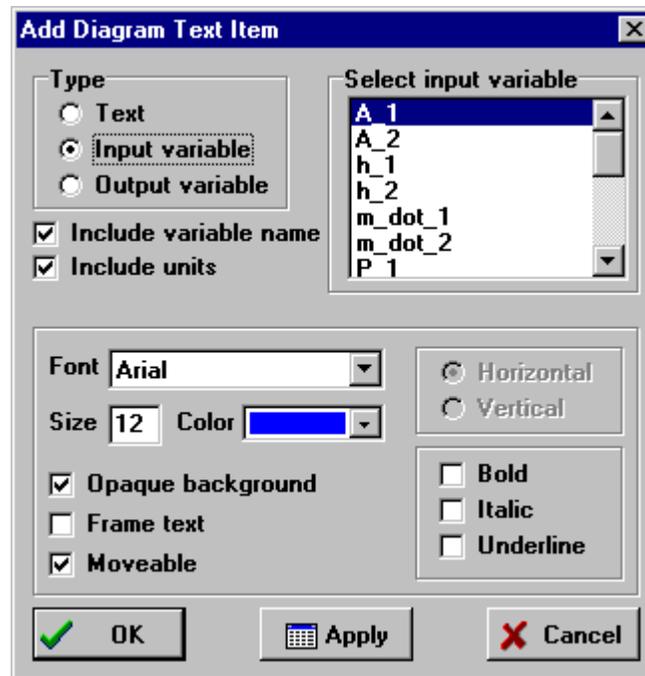


Boite de dialogue 'Add Diagram Text'

Pour sélectionner une variable d'entrée ou de sortie, il vous suffit de cliquer respectivement sur la case située à gauche du label 'Input Variable' et 'Output Variable'. Une liste déroulante dans laquelle vous pouvez choisir les variables apparaît. La valeur des variables d'entrée et de sortie s'afficheront dans la fenêtre de Diagramme dans leur format d'unités ainsi que leur nom. Une variable de sortie permet d'afficher la valeur d'une variable calculée précédemment. Une variable d'entrée est représentée par un rectangle dans la fenêtre de Diagramme, il est possible de la modifier directement dans cette fenêtre.

Si vous désirez insérer une chaîne de caractères et la sélectionner en tant que variable d'entrée, EES affichera une option 'Use String List' permettant d'afficher une liste contenant les variables et constantes chaînes de caractères présentes dans la fenêtre d'Equations .

Quand la commande **Solve** ou **Min/Max** est exécutée, EES examine d'abord le contenu de la fenêtre de Diagramme (variables d'entrées qui peuvent être éventuellement masquées). Noter qu'une valeur saisie dans la fenêtre de Diagramme ne peut être saisie également dans la fenêtre d'Equations . Une fois les calculs effectués, les résultats seront affichés (s'il existe des variables de sortie) dans la fenêtre de Diagramme. Remarquer que les variables de sorties contenant le champ suivant : **** indiquent que leurs valeurs n'ont pas été encore déterminées.



Les données de fenêtre de Diagramme sont ignorées si la fenêtre de Diagramme est masquée. La fenêtre de Diagramme peut être couplée avec la table Paramétrique (en exécutant la commande SOLVE TABLE) si l'option 'Use Input from Diagram' est cochée dans la boîte de dialogue 'Solve Table'

Pour effacer le contenu de la fenêtre, utiliser la commande **Clear** du menu **Edit**.

Les variables de la table Paramétrique, de la table 'Lookup', de la table tableaux peuvent être tracées grâce à la commande **New Plot Window** ou **Overlay Plot** du menu **Plot**. De plus, il est possible de tracer des fonctions ou des propriétés thermodynamiques via la commande **Property Plot**. Jusqu'à 10 fenêtres de Tracés peuvent être visibles à l'écran, chacune d'elles pouvant contenir un grand nombre de courbes superposées.

Il existe beaucoup d'options relatives aux tracés, tels le choix de l'échelle (linéaire ou logarithmique, l'affichage d'un quadrillage, l'ajout de zones de commentaires, l'affichage de courbe, etc.). Ces options peuvent être réglées dès la création d'un nouveau tracé ou en utilisant la commande **Modify Plot** et **Modify Axes** du menu **PLOT**.

L'apparence des tracés peut différer selon l'utilisation des commandes du menu **Plot** et des éléments de contrôle disponibles dans chacune des fenêtres :

1. Déplacement du tracé

Le graphique entier, y compris les échelles d'axe et toutes zones de texte, peut être déplacé à n'importe quel emplacement dans la fenêtre de Tracés. Pour ce faire, cliquer dans une zone du graphique, maintenez le bouton enfoncé pendant le déplacement, relâcher ensuite ce bouton à l'endroit désiré.

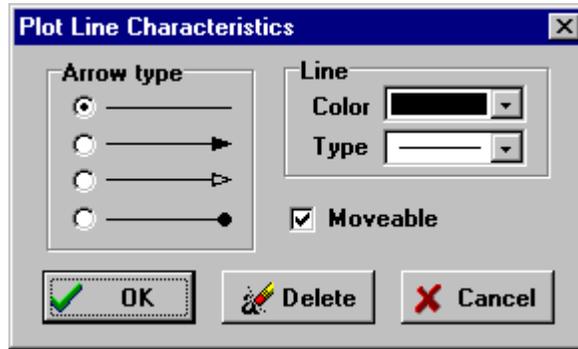
2. Déplacement d'une zone de texte

Les zones de textes telles que les titres d'axe et les zones de textes supplémentaires que vous avez ajoutées à l'aide de la commande **Add Text** du menu **PLOT** peuvent être déplacées à l'intérieur du graphique. Pour ce faire, cliquer sur une zone de texte, maintenez le bouton enfoncé pendant le déplacement, relâcher ensuite ce bouton à l'endroit désiré.

Il existe une option dans la boîte de dialogue **Préférence** dans l'onglet **PLOT** permettant d'ajuster automatiquement la position de la zone de texte au plus proche de la grille (snap to grid). Il est possible d'annuler cette option temporairement pour une zone de texte en maintenant la touche **Ctrl** enfoncée pendant le déplacement.

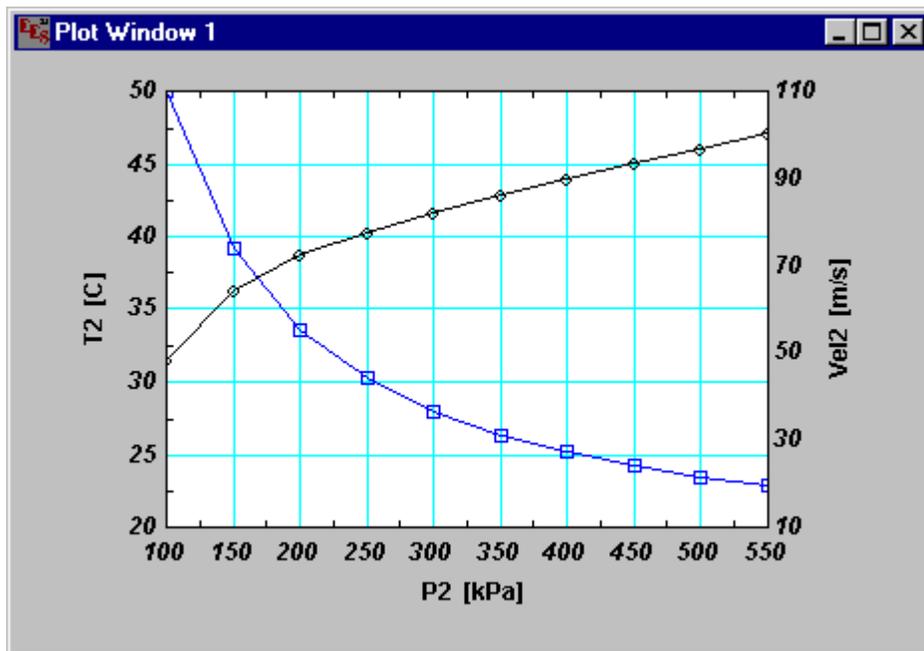
3. Déplacement des lignes et des flèches

Les lignes et les flèches peuvent être placées sur le graphique en utilisant la commande **ADD Ligne la Ligne** du menu **Plot**. Le choix du type de flèche s'effectue en double-cliquant sur celle-ci. Une boîte de dialogue apparaît alors. Sélectionner la tête de flèche désirée (arrow type) ainsi que sa couleur et son épaisseur. Une ligne peut être rotée, redimensionnée. Pour ce faire, cliquer sur l'une de ses extrémités et déplacer le curseur. Pour déplacer une ligne, procéder comme précédemment.



4. Redimensionner le graphique

La taille ainsi que la proportion du graphique peuvent être aisément modifiées. Cliquer sur l'une des extrémités du graphique (le curseur change alors d'aspect) puis tout en maintenant le bouton de la souris enfoncé, déplacer la souris (la nouvelle taille est matérialisée par un rectangle noir). Une fois la taille choisie, EES redimensionnera tous les éléments du graphique en respectant leur proportion.



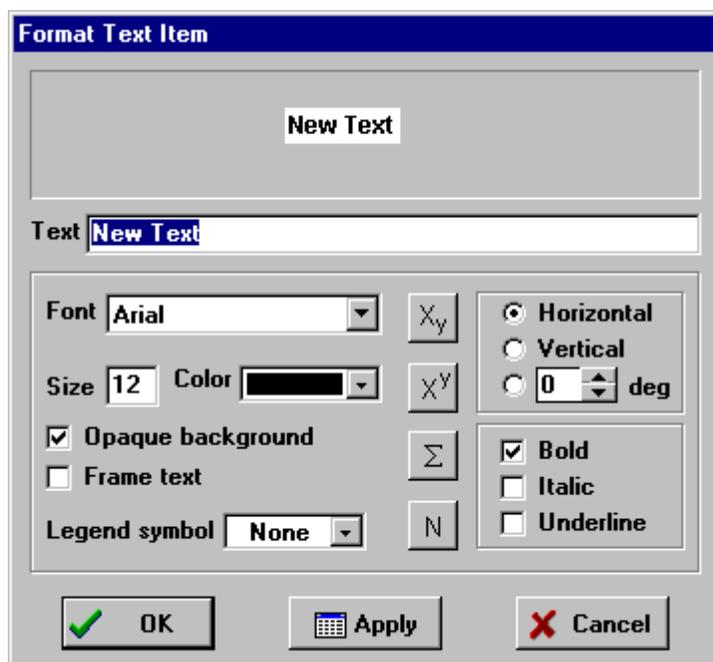
5. Changement des caractéristiques des zones de texte

Les caractéristiques (police, taille, style, couleur, orientation, etc.) de chaque zone de texte peuvent être changées individuellement en double-cliquant sur l'élément à modifier. La boîte de dialogue Format Text montrée ci-dessous apparaît et contient les caractéristiques actuelles de l'élément sélectionné. Il vous est possible d'éditer un texte, dans la zone 'Text'. Les indices, les exposants, les lettres grecs sont saisis de la manière suivante :

Tout d'abord, sélectionner l'élément à modifier. Ensuite, cliquer sur le bouton X_y (l'indice), X^y (exposant), Σ (grec), ou N (jeu de caractères normal) pour formater le

texte. Après l'appui sur le bouton 'Ok' le texte apparaîtra comme sélectionné précédemment.

EES permet à n'importe quelle zone de texte horizontale d'être associée à un symbole graphique pour faciliter la construction d'une légende. Cliquer dans la boîte de dialogue précédente dans la liste 'Symbol Legend'. Une liste déroulante contenant les symboles de graphiques existants apparaîtra. Une fois le symbole sélectionné, la zone de texte sera présentée dans la fenêtre de Graphique accompagnée à sa gauche du symbole précédemment sélectionné (ces deux articles sont dorénavant solidaires).



6. Modification des axes

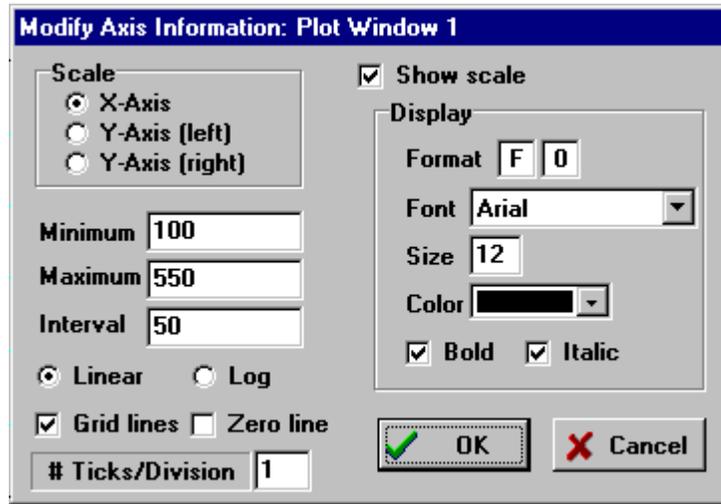
L'échelle et l'apparence peuvent être changées par l'intermédiaire de la boîte de dialogue 'Modify Axes Information' en double-cliquant sur l'axe des abscisses ou des ordonnées ou via la commande **Modify Axes** du menu **Plot**.

L'axe pour lequel les changements doivent s'opérer est indiqué dans la section 'Scale' située dans le coin supérieur gauche de la boîte de dialogue. Minimum, Maximum, et intervalle sont respectivement les valeurs minimum et maximum de l'axe ainsi que le pas d'intervalle. Celles-ci peuvent être modifiées ; le graphique sera redessiné à l'aide de ces nouveaux paramètres.

Si la case 'Zero line' est cochée, un trait vertical (pour les abscisses) et un trait horizontal (pour les ordonnées) d'origine 0 seront tracés.

La case contenant 'No. Ticks/Division' correspond au nombre de divisions entre chaque intervalle (sert uniquement pour améliorer la lisibilité du graphique). Si l'option 'Grid Lines' est sélectionnée, une grille sera superposée au graphique (les éléments de cette grille reprennent naissances aux points majeurs).

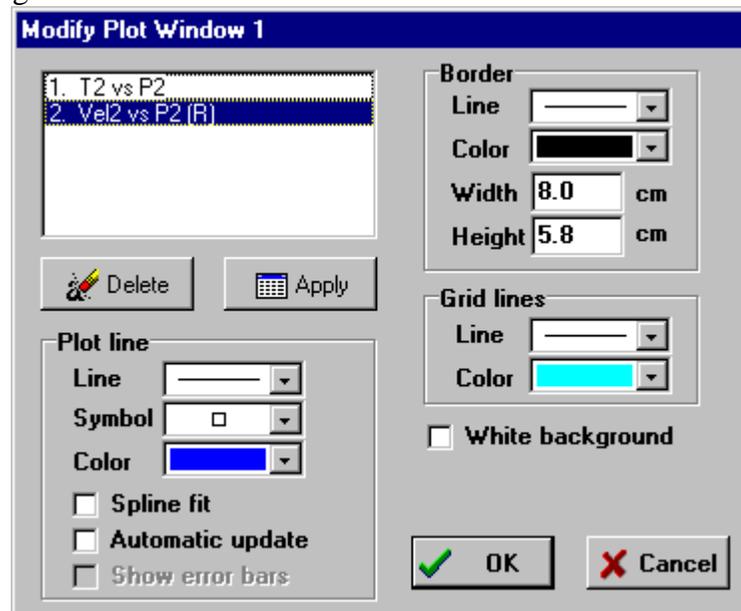
L'option 'Show Scale' permet d'afficher ou de masquer les éléments d'échelle qui sont paramétrables sous cette même option.



7. Modifier the Plot Information (modification du graphique)

Le type de ligne, la couleur, les symboles matérialisant les points, etc. peuvent être modifiés en double-cliquant à l'intérieur du graphique. La boîte de dialogue montrée ci-dessous permet de modifier ces paramètres (noter qu'il est possible d'y accéder par l'intermédiaire de la commande 'Modify Plot' dans le menu 'Plot') :

Tous les éléments présents sur le graphique sont visibles dans le rectangle supérieur gauche. Le symbole (R) qui peut figurer à droite d'un élément indique que celui-ci utilise l'axe des ordonnées gauche. Pour sélectionner un élément, il suffit de cliquer dessus dans ce même rectangle.



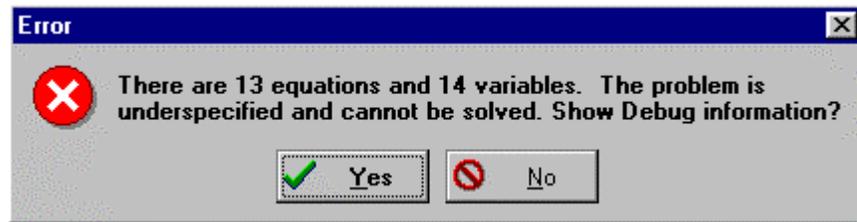
La case 'Spline fit control' permet, si elle est cochée, d'arrondir votre tracé.

La case 'Automatic sets up' permet de mettre à jour automatiquement votre tracé pour peu que vous modifiez ses valeurs supports (table paramétrique, etc.).

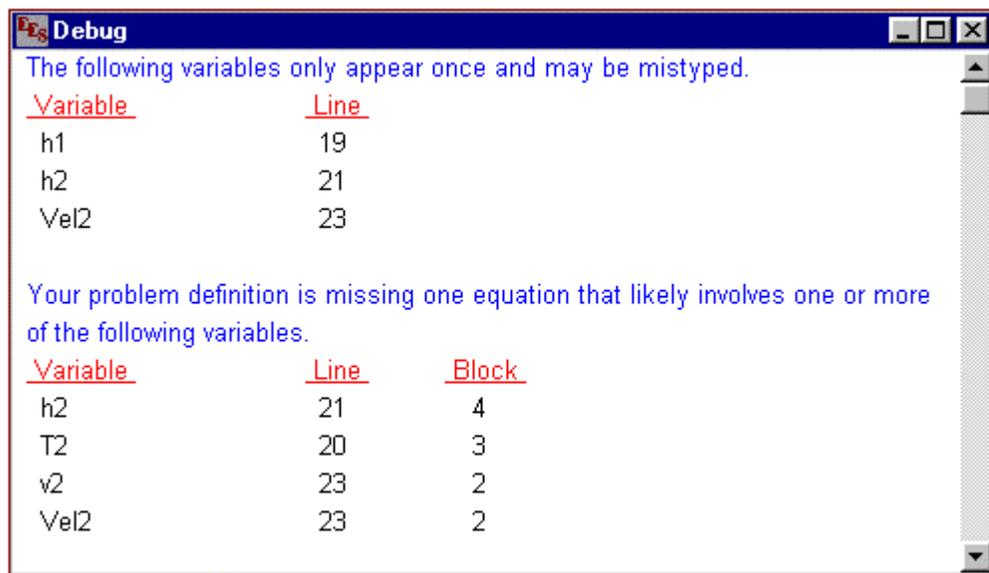
Enfin la case 'Show errors bar' n'est activée que si les données à tracer ont été obtenues à partir de la commande 'Uncertainly Propagation Error' du menu CALCULATE.

Pour mettre à jour votre graphique suivant les modifications effectuées, cliquer sur le bouton 'Apply'

La fenêtre Debug est un outil permettant de diagnostiquer, de localiser les erreurs dans vos équations. Si vous essayez de résoudre un système dans lequel il y a plus d'équations que d'inconnues, la boîte de message suivante apparaît :



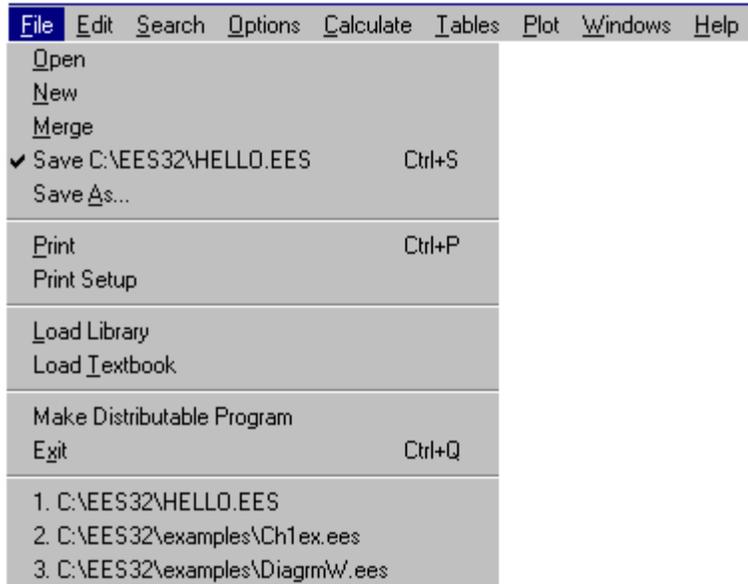
Cliquer alors sur « Yes », la fenêtre suivante s'affiche :



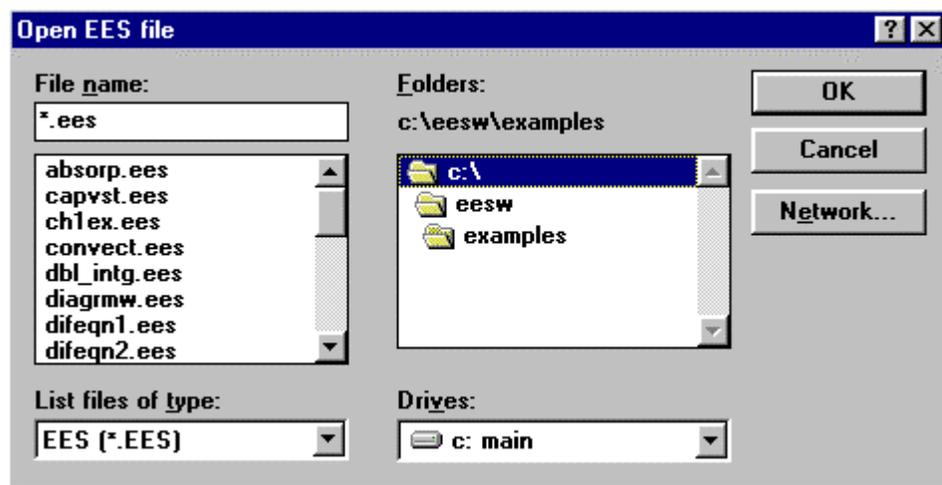
Cette fenêtre contient deux listes de variables. La première liste affiche toutes les variables qui sont référencées une fois dans la fenêtre d'Equations. Ces variables comportent peut être des erreurs de syntaxe, ou ne sont pas utilisées directement dans le problème mais servent uniquement à donner des informations. La deuxième liste montre les variables utilisées dans le système d'équation pour lesquelles il manque des informations. L'information à utiliser pour construire cette deuxième liste est déterminée par l'examen de l'ordre qui bloque des équations dans la fenêtre Résiduelle. Vous pouvez aussi trouver l'information dans la fenêtre Résiduelle servant à l'identification du problème avec votre série d'équation. En cliquant sur le bouton gauche de la souris sur le nom de la variable dans la fenêtre Debug, la fenêtre d'Equations apparaîtra au premier plan avec la variable sélectionnée.

Les commandes du menu

Le menu File



Open permet d'ouvrir un fichier EES précédemment sauvé à l'aide des commandes Save ou Save As.



Après vous avoir demandé de sauver le travail en cours, la boîte dialogue ci-dessus apparaît. Le répertoire courant est indiqué dans la section 'Folders'. Les fichiers EES sont visibles dans la section 'File name'. Pour ouvrir un fichier, cliquer sur le nom du fichier à ouvrir ou entrer directement votre nom dans le champ situé au-dessous du label 'File name' précédé éventuellement du nom du répertoire dans lequel il se trouve. Pour changer

de disque (disque dur, disquette, etc.), cliquer sur la liste déroulante 'Drive'. Cliquer sur le disque qui vous intéresse pour ouvrir son contenu. Cliquer sur 'Ok' pour ouvrir le fichier sélectionné.

EES peut lire 4 types de fichiers selon les formats : fichiers EES, Import File (Fichier importé), Text File (fichier textuel), et Library File (fichiers de librairie). Pour sélectionner un format de fichier, cliquer sur la liste déroulante 'List files of type'. Les fichiers EES possèdent l'extension .EES. Les fichiers d'importation créés à l'aide de la commande 'Export' sous un système d'exploitation différent (Macintosh par exemple) possèdent l'extension .XPT. Les fichiers textuels ASCII peuvent être lus dans la fenêtre d'Equations et ont l'extension .TXT. Enfin, les fichiers de librairie contiennent une ou plusieurs fonctions, procédures, modules (Cf. Chapitre 5).

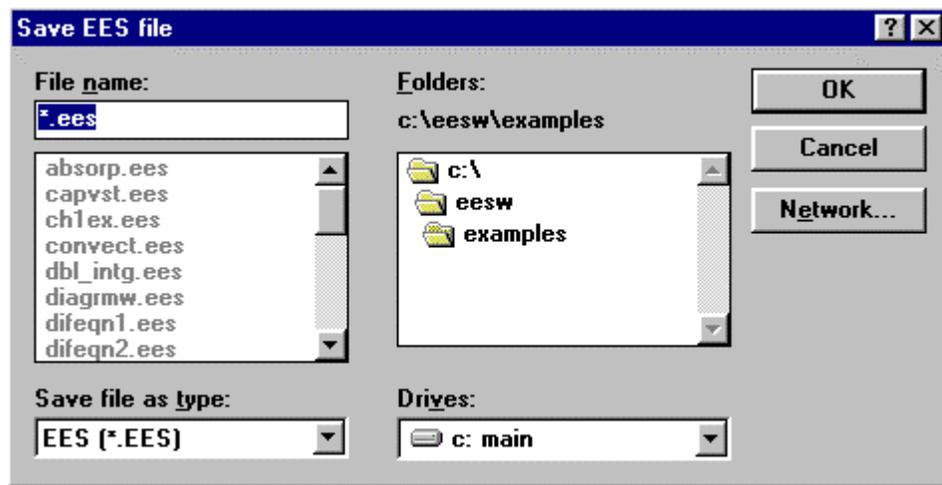
New initialise une nouvelle session EES. Toutes les variables et équations ont été supprimées de la mémoire. Le système d'unités est restauré à son état originel (si vous n'avez pas sauvegarder le travail en cours, EES vous demandera si vous voulez effectuer cette action).

Merge permet d'ajouter à la session en cours les équations d'un autre fichier EES. La boîte de dialogue 'Merge' est identique à celle utilisée par la commande 'Open'. Il est possible d'entrer ces équations dans un fichier texte et de les utiliser à l'aide de la directive \$Include (voir plus loin dans ce document).

Save permet de sauver le travail en cours sous le nom apparaissant dans la barre de titre. Dans le cas où aucune sauvegarde n'a été faite jusqu'à présent, EES vous demandera d'entrer un nom de fichier (comme pour la commande 'Save As'). Cette action permet de sauver l'ensemble de votre travail (équations, variables, tables, graphiques, diagramme, etc.). Par défaut, EES sauve ces fichiers avec l'extension .EES. Si vous désirez exporter un fichier, utiliser la liste déroulante 'Files type' de la boîte de dialogue 'Save As'. Noter que si vous n'avez sauvé que votre travail ou seulement si celui-ci a subi des modifications, il apparaît une petite marque à côté de la commande Save.

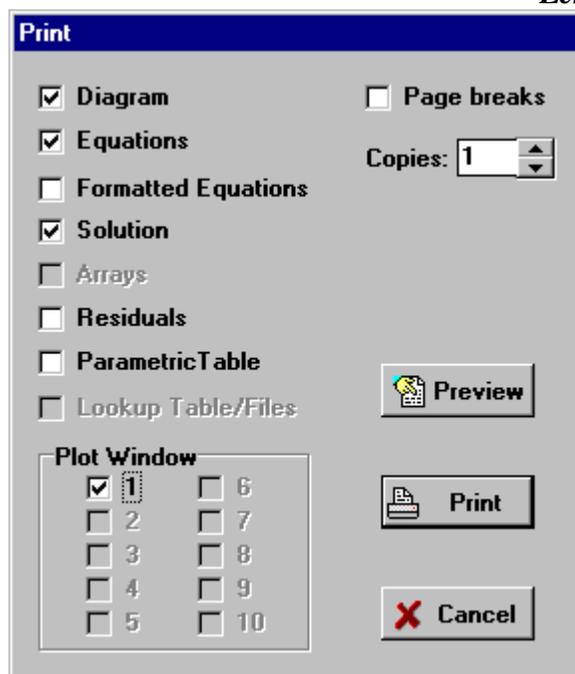
Save As... est similaire à la commande **Save** à la différence qu'un nom de fichier vous sera demandé. EES reconnaît 4 types de fichiers selon les formats suivants : fichiers EES, Import File (Fichier importé), Text File (fichier textuel), et Library File (fichiers de librairie). Pour sélectionner un format de fichier, cliquer sur la liste déroulante 'List files of type'. Les fichiers EES possèdent l'extension .EES. Les fichiers d'importation créés à l'aide de la commande 'Export' sous un système d'exploitation différent (Macintosh par exemple) possèdent l'extension .XPT. Les fichiers textuels ASCII peuvent être lus dans la fenêtre d'Equations et ont l'extension .TXT. Les fichiers de librairie possèdent l'extension .LIB. Chaque fois que EES démarre, il ouvre automatiquement l'ensemble des fichiers .LIB du répertoire USERLIB\ et charge les fonctions, procédures, modules présents dans ces fichiers. Cette caractéristique est l'une des plus puissantes de EES car

l'utilisateur peut rapidement développer des outils utilisables par la suite dans n'importe quelle session EES. (Cf. Chapitre 5).



Print permet d'imprimer tout ou partie de l'ensemble des fenêtres de EES. Quand la commande Print est invoquée, une boîte de dialogue apparaît. Elle offre la possibilité d'imprimer chaque fenêtre en cliquant sur la case située à droite des noms de celles-ci. (si la case est cochée, la fenêtre correspondante sera imprimée.) Si la case 'Page breaks' (saut de page) est sélectionnée, chaque fenêtre sera imprimée sur une page différente. Les imprimantes disponibles apparaissent dans la liste déroulante 'Printer' (pour sélectionner une imprimante, faire défiler la liste, et cliquer sur le nom de l'imprimante voulue).

Il est possible d'imprimer un document en EES dans un fichier. (voir le manuel de Windows pour de plus amples informations). Les options d'impression tels la police, l'espacement, la taille de police sont réglés par la commande **Préférences** du menu **Options**. Le bouton 'Preview' permet d'avoir un aperçu du document à imprimer avant son impression.



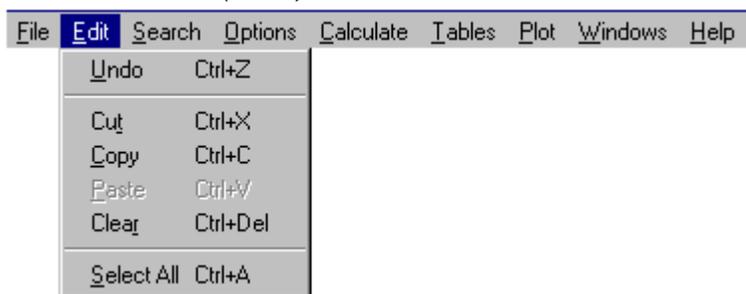
Printer Setup invoque une boîte de dialogue permettant de configurer les options d'imprimante telles la taille du papier, l'orientation, etc.

Load Library invoque la boîte de dialogue 'Open', et affiche les fichiers librairie (extension .LIB) disponibles dans le répertoire actuel. Une fois chargés, les fichiers restent dans la mémoire de EES jusqu'à sa fermeture. Noter, qu'à chaque ouverture, ESS charge en mémoire toutes les fonctions, procédures, modules présents dans les fichiers librairie du répertoire USERLIB\ (la commande Load Library n'est donc pas nécessaire pour ces fichiers). Cette commande permet également de charger des fichiers de type .DLF, .DLP et FDL (voir Chapitre 6).

Load Textbook permet de lire des fichiers de type .TXB créés par un utilisateur. Ces fichiers contiennent des informations, pour créer un nouvel élément de menu qui sera situé à droite du menu HELP. Ce type de fichier peut être lu automatiquement à condition de se trouver dans le répertoire USERLIB\

Quit permet de quitter 'proprement' le logiciel.

Les autres éléments du menu FICHIER constituent la liste des fichiers récemment ouverts. Il est possible de masquer cette liste au sein de la boîte de dialogue 'Preferences'.



Undo restaure la fenêtre d'Equations aux conditions dans laquelle elle se trouvait avant la dernière édition (Cette commande ne fonctionne que dans la fenêtre d'Equations).

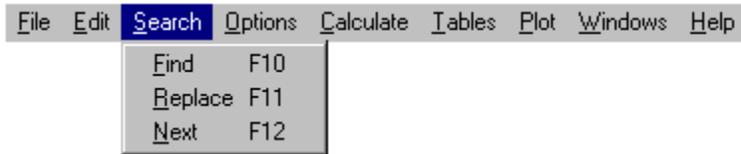
Cut efface le texte sélectionné et le copie dans le presse-papiers.

Copy: cette commande agit différemment selon son contexte d'application. 'Copy' permet de placer le texte sélectionné dans la fenêtre d'Equations dans le presse-papiers. Pour coller l'élément du presse-papiers, il suffit d'utiliser la commande Paste. Lors de l'utilisation de tables (table paramétrique, Lookup table et tableaux), la commande 'Copy' permet de copier les cellules sélectionnées (apparaissant en vidéo inverse). Les données copiées à partir d'une table sont placées dans le presse-papiers selon un format pour lequel chaque donnée de cellules est séparée par une marque de tabulation et se termine par un retour chariot. Ainsi, il est possible de copier ces données dans n'importe quelle table ou dans une autre application. (maintenir la touche Ctrl enfoncée en invoquant la commande 'Copy' si vous désirez copier également l'entête des colonnes et les unités). La commande 'Copy' permet de mettre dans le presse-papiers un graphique ou un diagramme dans le cadre des fenêtres de Tracés et Diagramme pour ensuite les coller dans d'autres applications (l'élément copié à le format MetaFilePict compréhensible par toute application Windows). 'Copy Solution' placera le contenu de la fenêtre Solution dans le presse-papiers en tant qu'élément textuel (ASCII) dans lequel chaque variable apparaîtra sur une ligne. La commande Copy permet de copier l'ensemble des informations figurante dans la fenêtre Résiduelle sous forme textuelle (chaque élément copié figure sur une ligne). Le contenu du presse-papiers pourra être collé dans d'autres applications.

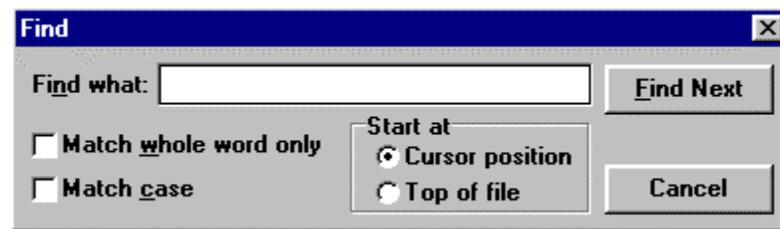
Paste est utilisable dans les fenêtres d'Equations, 'Parametric', 'Lookup' et 'Diagram'. Cette commande permet de coller le contenu du presse-papiers (un graphique pour la fenêtre Diagram ou un texte pour les autres fenêtres) précédemment copié à l'aide des commandes 'Copy' (copier), 'Cut' (couper) de EES ou toute autre application. Quand cette commande est invoquée dans la fenêtre Parametric ou 'Lookup Table', les valeurs du presse-papiers seront collées à partir de la position actuelle du curseur. Les commandes 'Copy' et 'Paste' (copier et coller) peuvent également s'utiliser dans la fenêtre de Tracés.

Clear efface le texte sélectionné ou le contenu de la fenêtre Diagram.

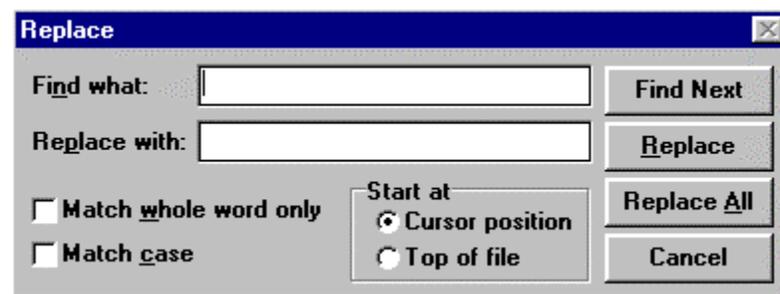
Select All permet de sélectionner l'ensemble des éléments composant la fenêtre d'Equations ou toutes les cellules d'une table. Dans la fenêtre Formated Equation (Equations mises en forme), cette commande est nommée 'Select Display'.



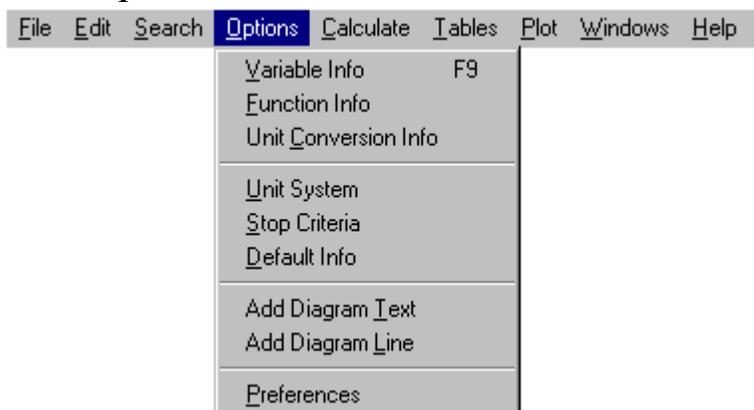
Find cherchera dans la fenêtre d'Equations la première occurrence du texte saisi dans le champ 'Find what'. La recherche tient compte de la casse si l'option 'Match case' est cochée. Si l'option 'Match whole word only' est cochée, le texte sera trouvé seulement s'il est délimité par des espaces ou des opérateurs mathématiques. La section 'Start at' permet de définir le point de départ de la recherche : l'option 'Cursor position' permet d'effectuer la recherche à partir de la position actuelle du curseur ; l'option 'Top of file' permet d'effectuer la recherche à partir du début de la fenêtre d'Equations. Le bouton 'Cancel' sera remplacé par le bouton 'Done' à l'issue du processus de recherche.



Replace cherchera dans la fenêtre d'Equations la première occurrence du texte saisi dans le champ 'Find what' et remplacera l'élément trouvé par le contenu du champ 'Replace with'. Le bouton 'Replace All' permet de remplacer tous les éléments trouvés par le contenu du champ 'Replace with'.



Next cherchera l'occurrence suivante d'un texte saisi dans le champ 'Find what' de la commande 'Find' ou 'Replace'.



Variable Info affiche une boîte de dialogue (voir ci-dessous) dans laquelle on peut trouver le nom des variables de la fenêtre d'Equations, suivi de leur valeur par défaut, des valeurs limites (basses et hautes), du format d'affichage ainsi que leurs unités.

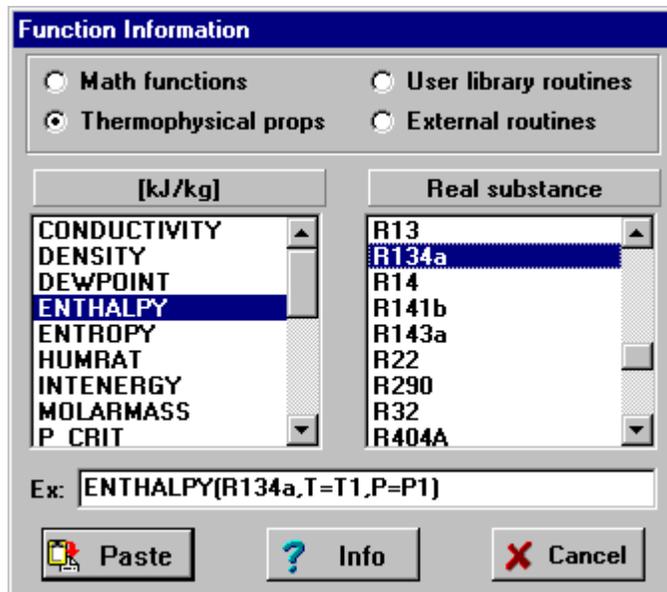
Variable Information						
Module Main						
Variable	Guess	Lower	Upper	Display	Units	
A1	0.011	<i>-infinity</i>	<i>infinity</i>	A 3 N	m^2	
A2	0.1	<i>-infinity</i>	<i>infinity</i>	A 3 N	m^2	
h1	1	0.0000E+00	<i>infinity</i>	A 3 N	kJ/kg	
h2	1	0.0000E+00	<i>infinity</i>	A 3 N	kJ/kg	
m1	1	<i>-infinity</i>	<i>infinity</i>	A 3 N	kg/s	
m2	1	<i>-infinity</i>	<i>infinity</i>	A 3 N	kg/s	
P1	700	<i>-infinity</i>	<i>infinity</i>	A 3 N	kPa	
P2	300	<i>-infinity</i>	<i>infinity</i>	A 3 N	kPa	
T1	50	<i>-infinity</i>	<i>infinity</i>	A 3 N	C	
T2	1	<i>-infinity</i>	<i>infinity</i>	A 3 X	C	
v1	1	0.0000E+00	<i>infinity</i>	A 3 N	m^3/kg	

Si le programme en cours contient plusieurs modules (voir Chapitre 5), une liste déroulante est accessible (Module) permettant de visualiser les variables relatives à chacun des modules. Les termes *-infinity* et *infinity* peuvent être utilisés pour indiquer des valeurs limites (lower and Upper) infinies. Les champs 'Guess' (valeur par défaut), 'Lower' et 'Upper' acceptent également un nom de variable. Si un nom de variable est fourni, EES utilise sa valeur courante.

EES essaye dans un premier temps de résoudre les équations comportant une seule inconnue avant d'afficher cette boîte de dialogue. Les variables pour lesquelles les valeurs ont déjà été calculées ont leurs limites affichées en italique. La valeur calculée apparaît dans la colonne 'Guess Value'. Ces valeurs peuvent être recalculées en double-cliquant sur celles-ci.

Le format d'affichage (Display Format) d'une variable dans la fenêtre de Solution ou dans une table est contrôlée par trois champs figurant dans la liste déroulante. Les unités des variables (ou toute autre information) peuvent être saisies dans la colonne Unit. Ces unités sont utilisées uniquement à des fins d'affichage. Une fois le bouton 'Ok' pressé, tous changements effectués dans la boîte de dialogue seront validés. Le bouton 'Update' (mise à jour) a pour effet de remplacer la valeur par défaut (Guess Value) de chaque valeur par la valeur précalculée. Le bouton 'Print' permet d'imprimer toutes les informations figurants dans cette table. Enfin, le bouton 'Cancel' (Annuler) permet de fermer la boîte de dialogue sans opérer de changement.

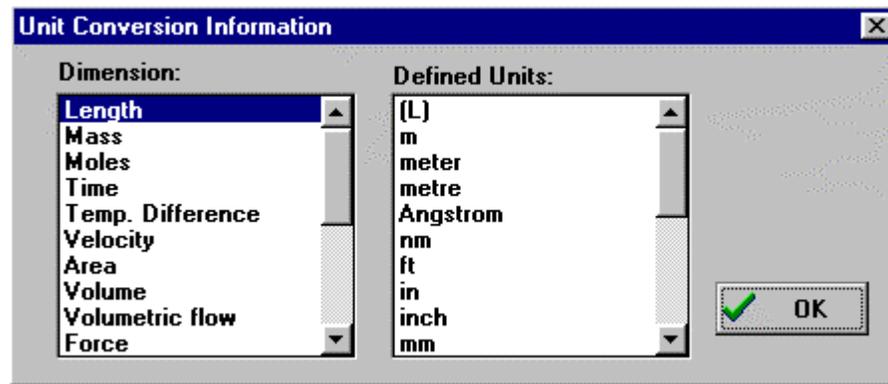
Function Info permet d'ouvrir la boîte de dialogue suivante :



Les 4 boutons situés en haut de celle-ci indiquent quel type d'information doit être fourni. 'Math Function' (fonctions mathématiques) et 'Thermophysical Props' (propriétés thermodynamiques) pointent vers les fonctions incluses dans EES. Le bouton 'User Library' permet d'afficher la liste des fonctions, procédures, modules utilisateurs chargés à partir de la librairie (cf. chapitre 5). Le bouton 'External Routine' (routines externes) pointe vers des fonctions déjà compilées qui peuvent être liées à EES comme décrit dans le chapitre 6. Les fonctions correspondantes au bouton sélectionné, apparaissent dans la liste déroulante située à gauche. Pour sélectionner une fonction, il vous suffit de cliquer sur le nom de celle-ci. Cliquer sur le bouton 'Info' pour obtenir des informations supplémentaires afférentes à la fonction choisie.

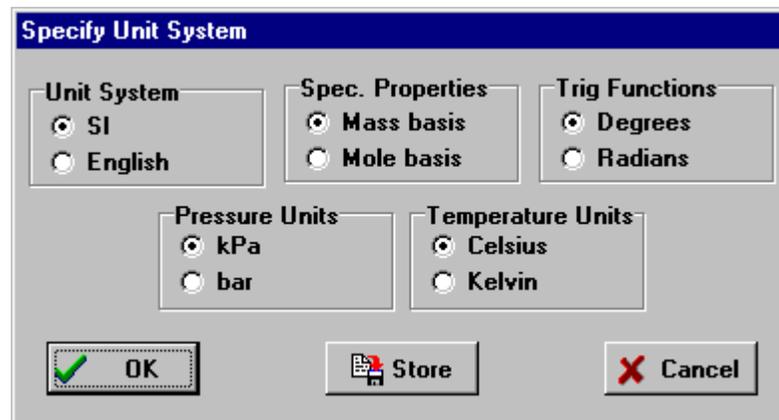
Les unités des fonctions de propriétés thermodynamiques sont affichées au-dessus de la liste déroulante. Ces fonctions requièrent la spécification d'une substance. Celles-ci apparaissent à droite de la liste de fonctions. Cliquer sur l'une d'entre elles pour la sélectionner. 'Ideal Gas' apparaîtra si la substance choisie est modélisée suivant la loi des gaz parfaits. 'Real substance' apparaîtra si celle-ci est liquide ou à l'état de vapeur. Les substances représentées par leur symbole chimique (ex. CO_2) sont modélisées suivant la loi des gaz parfaits et utilisent la table de référence JANAF pour leur valeur d'entropie et d'enthalpie. Les substances dont le nom figure en toutes lettres (ex. CarbonDioxide) sont modélisées en tant que fluides réels et n'utilisent pas la table JANAF. L'air est une exception à cette règle car il est modélisé en tant que gaz parfait. Les fonctions Psychométriques (psychrometric) sont applicables uniquement sur AirH2). (cf. chapitre 4 pour plus de détails sur ces fonctions). Un exemple de fonction avec des variables par défaut est affiché en-dessous de la liste de fonction. Vous pouvez éditer ces informations directement dans la zone d'exemple. Si vous cliquez sur la commande 'Paste' (coller), le contenu de la zone d'exemple sera collé dans la fenêtre d'Equations.

Unit Conversion Info permet d'afficher les unités afférentes à un ensemble de dimension :

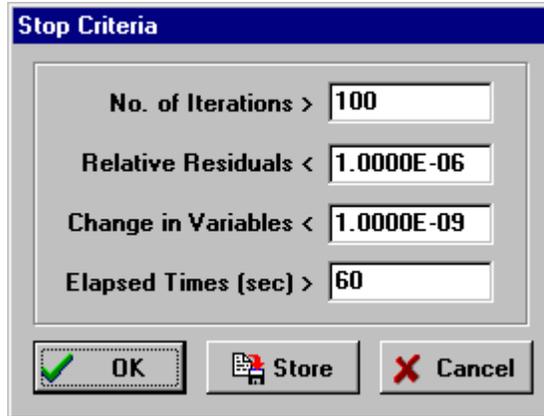


UNIT CONVERSION INFO permet de lister l'ensemble des unités utilisables dans EES. Cette commande sert de support à la fonction de conversion d'unités Convert. La fonction 'Convert' a le format suivant : Convert("De", "a") pour laquelle 'DE' et 'A' sont des chaînes de caractères identifiant une unité tel par exemple 'meter'. Pour visualiser les diverses unités relatives aux dimensions, utiliser la liste de droite ('Dimension'). Toute combinaison d'unités (visibles dans la liste de droite) et utilisable dans la fonction Convert. Vous pouvez rajouter vos propres unités. Celles-ci se trouvent dans le fichier UNITS.TXT (des indications sont fournies dans l'en-tête de ce fichier).

Unit System affiche la boîte de dialogue affichée ci-dessous. Celle-ci permet de fixer le système d'unités utilisé par les fonctions et propriétés mathématiques et thermodynamiques. EES ne permet pas de convertir automatiquement les unités. Les unités seront changées pour la suite de la session de travail si la touche 'Ok' est appuyée. Les unités sélectionnées seront sauveées avec les autres éléments du problème lors de l'invocation de la commande 'Save' du menu FILE. Les formats d'unités seront restaurés lors de l'ouverture du fichier dans lesquelles ils ont été précédemment sauveés. Si vous désirez que ces changements soient permanent, appuyer sur le bouton 'Store' de la boîte de dialogue.



Stop Criteria permet de spécifier les critères d'arrêt de calcul.

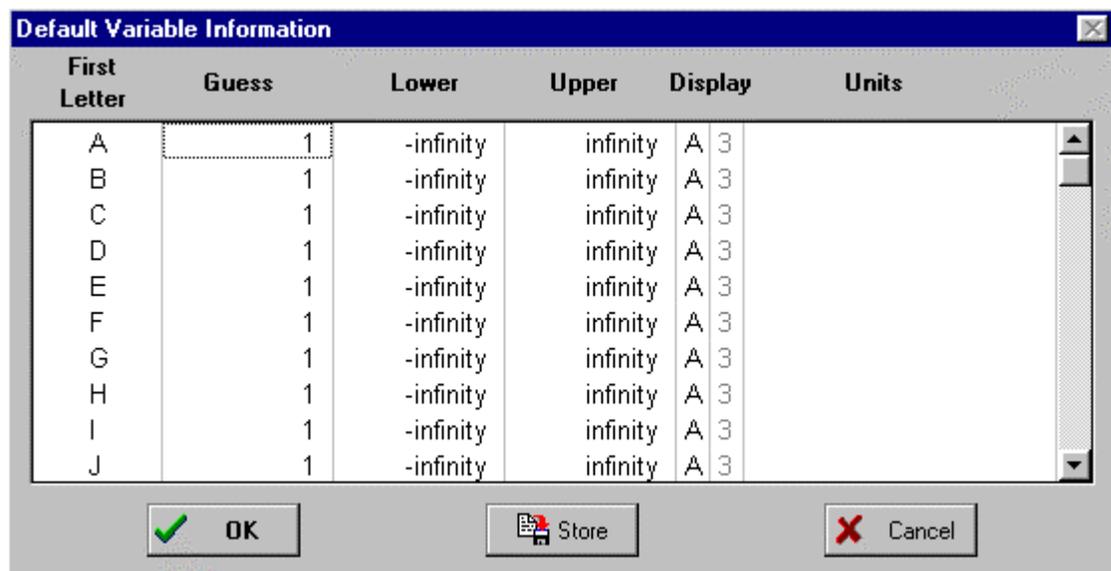


Ces critères sont le nombre d'itérations, la valeur résiduelle relative maximum, le changement maximum de la valeur d'une variable entre deux itérations, et le temps de calcul. Tous les calculs sont faits avec une précision de 21 chiffres significatifs. Il se peut que vous perdiez en précision de calcul si vous avez choisi des petites valeurs pour la valeur résiduelle relative maximum et le changement maximum de la valeur d'une variable.

Les critères d'arrêt seront pris en compte dans la suite de votre session en appuyant sur le bouton 'OK'. Pour que ces changements soient permanents, appuyer sur la touche 'Store'.

Default Info permet de spécifier la valeur par défaut, les limites, le format d'affichage et les unités pour l'ensemble des variables (représentées par la première lettre du nom de variables) présentes dans la fenêtre d'Equations. Dans le cas où les problèmes que vous avez l'habitude de traiter respectent la même nomenclature, il est bon d'utiliser le bouton 'Store' de la boîte de dialogue pour effectuer une sauvegarde permanente.

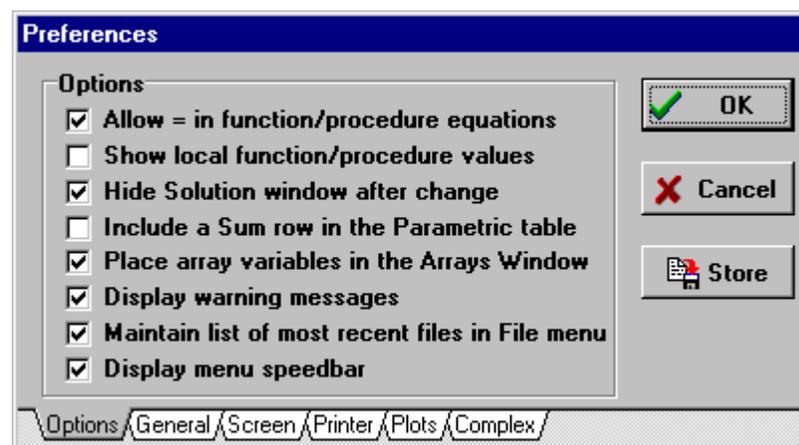
La commande 'Default Variable Information' (Information par défaut des variables) peut s'utiliser pour effectuer des changements sélectifs sur les variables existantes. Par exemple, si vous changez les unités pour les variables commençant par la lettre T en [K] et que vous appuyez sur la touche 'Ok', toutes les variables commençant par la lettre T utiliseront l'unité [K] (Kelvin) dans la session actuelle.



Add Diagram Text ouvre une boîte de dialogue dans laquelle il est possible de sélectionner 3 types de texte à afficher dans la fenêtre de Diagramme. Ces trois types sont : plain text (texte normal), input variables (variables d'entrée), output variables (variables de sortie). (voir le chapitre 2 pour plus d'informations).

Add Diagram Line permet de placer des lignes et des flèches dans la fenêtre de Diagramme de la même manière qu'à travers la commande Add Line (ajouter une ligne) du menu Plot.

Preferences ouvre une boîte de dialogue contenant 6 onglets, permettant de régler les options: du programme, d'affichage écran, d'affichage d'équations, d'imprimante, de la fenêtre de Tracés, pour l'utilisation des nombres complexes. Ces options sont détaillées ci-après. Si vous cliquez sur le bouton 'Ok' les options seront prises en compte dans la session de travail en cours; si vous cliquez sur Store, les options choisies seront permanentes.



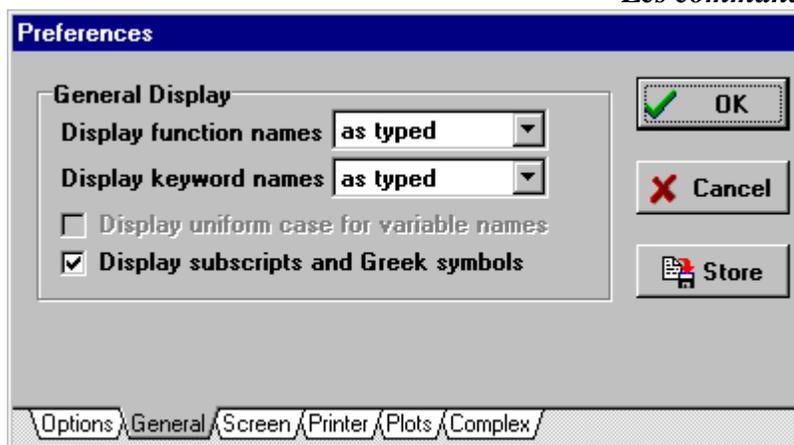
- ☒ **Allow = in function/procedure equations** a comme effet de supprimer le message d'erreur qui apparaît normalement si le symbole ($:=$) d'assignation n'est pas utilisé dans

les fonctions et procédures. En effet (voir chapitre 5) les fonctions et procédures de EES utilisent le même symbole d'assignation que dans FORTRAN ou PASCAL,

(les modules de EES utilisent le symbole d'égalité, par conséquent le symbole $:=$ n'est pas utilisable).

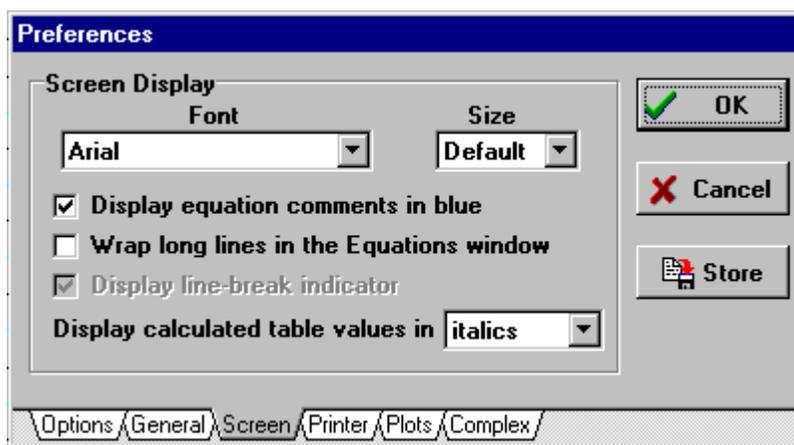
L'assignation d'une variable est identifiée par un nom de la variable à droite du symbole et une valeur numérique à gauche. (ex. $X := X + 1$, est une assignation mais pas une égalité).

- Show function/procedure/module values** permet d'afficher les valeurs récentes des variables locales de fonctions, procédures et modules de EES dans la fenêtre de Solution. Les équations présentes dans un module seront également affichées dans la fenêtre Résiduelle. Ces valeurs n'ont d'intérêt que dans une situation de débogage. Noter que les valeurs des variables locales de fonctions, procédures et modules issues des fichiers de librairie ne seront pas affichées.
- Hide Solution Window after change** permet de masquer automatiquement les fenêtres de Solution, Résiduelle, et de Tableaux dans le cas où un changement est effectué dans la fenêtre d'Equations . Si cette option n'est pas sélectionnée et qu'un changement est effectué dans la fenêtre d'Equations , la barre de titre de la fenêtre Solution sera changée en 'Last Solution' (correspondant aux résultats des derniers calculs effectués).
- Include a Sum row in Parametric Table** permet d'ajouter une ligne sommation de toutes les colonnes de la table paramétrique.
- Place array variables in the Arrays Window** cette option indique à EES d'afficher toutes les variables tableaux dans la fenêtre de Tableaux plutôt que dans la fenêtre de Solution après résolution. Ces valeurs peuvent être tracées et copiées comme dans une table paramétrique ou 'Lookup'. (voir Chapitre 2). Noter que le contenu de la fenêtre de Tableaux est effacé si cette option est désélectionnée.
- Display warning messages** permet d'afficher ou de masquer les messages d'avertissement pendant les calculs. Un message d'avertissement est affiché si par exemple des propriétés de corrélations thermophysiques sortent de leur domaine d'application. La directive \$Warnings On/Off a le même effet.
- Maintain a list of recent files in the File menu** permet d'afficher ou de masquer la liste des 8 derniers fichiers ouverts dans le menu FILE. Cette liste se trouve dans le fichier EES.FNL du répertoire EES.
- Display menu speedbar** permet d'afficher ou de masquer la barre d'icônes située au-dessous de la barre de titre.



Les deux premières options de l'onglet 'General' permet d'afficher en majuscules (upper case) ou minuscules (lower) respectivement les noms de fonctions (ENTHALPY, SIN, etc.) et les mots clefs (FUNCTION, DUPLICATE, fluid names, etc.).

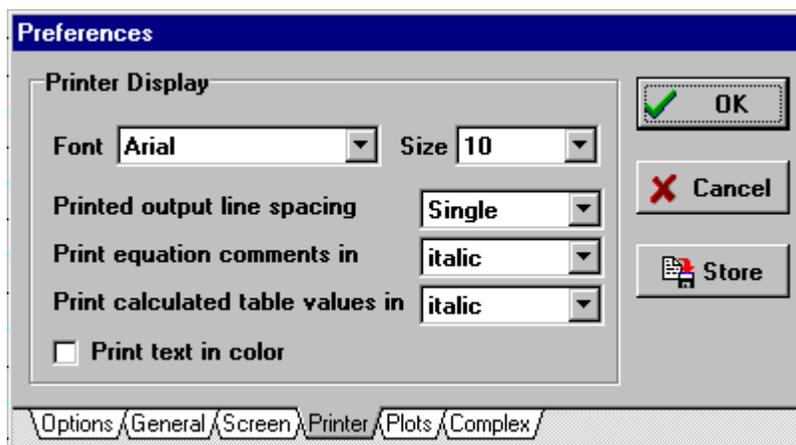
- ☒ **Display uniform case for variable names** affiche toutes les variables de la fenêtre d'Equations suivant la case utilisée pour le premier nom de variable saisi. Si celle-ci est modifiée, la commande **Check/Format** du menu **Calculate** remplacera toutes les variables suivant la case de cette première occurrence.
- ☒ **Display subscripts and Greek symbols** change l'apparence des variables dans la fenêtre de Solution ou d'Equations mises en forme. Si cette option est cochée, les variables tableaux seront affichées avec leur index en indice. Les caractères précédés d'un underscore () dans le nom d'une variable seront affichés en indice. Noter qu'un nom de variable suivi de **_dot**, ou **_bar** sera respectivement affiché avec le symbole de dérivé, de barre. Le symbole **_infinity** sera affiché (∞). Les noms de variables utilisant l'alphabet Grec (ex. alpha, beta, gamma, etc.) seront affichés à l'aide de la police Symbol. Si le nom entier de la variable est en majuscule, la police Greek Symbol en majuscule sera utilisée.



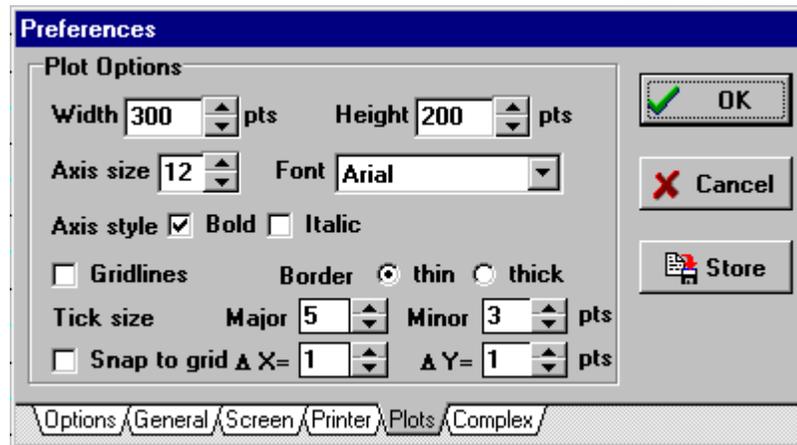
Changer la police et la taille dans l'onglet 'Screen' modifiera l'apparence de toutes les fenêtres EES exceptées les fenêtres de Tracés et de Diagramme qui possèdent leurs propres contrôles.

- Display equation comments in blue** permet d'afficher les commentaires (texte figurant entre crochets { } ou entre marques de citations ‘’) en bleu.
- Wrap long lines in the Equations window** masque la barre horizontale de déplacement. Les lignes trop longues seront coupées en un point approprié et continueront à la ligne suivante. Un symbole rouge > sera affiché dans la marge gauche si la case 'Display line-break indicator' est cochée.
- Display line-break indicator** est utilisable que si l'option 'Wrap long lines' est cochée. Cette option permet d'indiquer qu'une équation est écrite sur plusieurs lignes.

Display calculated table values in (same font, italics, blue, bold) contrôle l'apparence des cellules de la table paramétrique après calculs. Les valeurs saisies par l'utilisateur seront affichées en format normal suivant la police et la taille choisies.

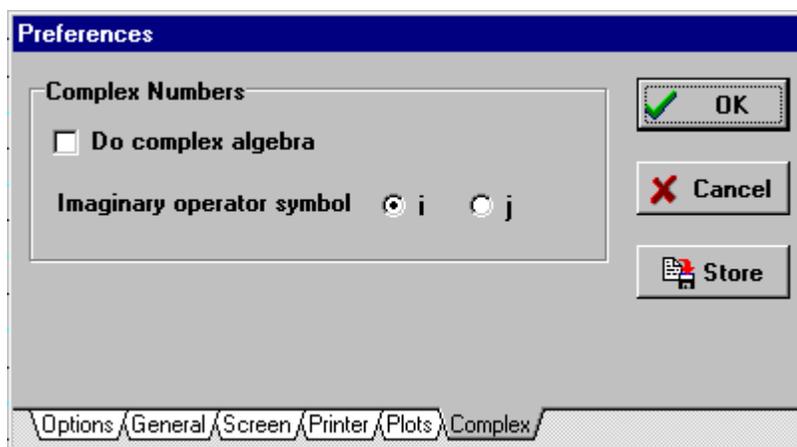


L'onglet 'Printer' permet de configurer les options d'apparence d'une sortie imprimante. Si l'option 'Print text in color' n'est pas cochée, l'ensemble des textes sera converti en couleur noire (un aperçu avant impression est disponible dans la boîte de dialogue FILE).



L'onglet 'Plots' permet de changer les paramètres par défaut de la fenêtre de Tracès : largeur (Width) et la hauteur (Height) de la zone de tracès, la police (Font), la taille de police (Axis size), le style : gras (Bold) *, italique (Italic). Par ailleurs, l'option 'Gridlines' permet d'afficher une grille dans la fenêtre de tracès. Les options thin (fin), thick (épais) permettent de définir la largeur de la bordure de la fenêtre de tracès. L'option Tick size permet de définir les points d'encrage des lignes de la grille.

L'option 'snap to grid' permet de créer une grille virtuelle servant de points d'attache aux zones de texte dans la fenêtre de Tracès. Les espaces entre une zone de texte et la grille sont saisis dans les cases ΔX et ΔY . Par exemple, si les cases ΔX et ΔY contiennent la valeur 4, la zone de texte sera alignée au pixel divisible par 4 le plus proche. Si cette option est cochée et que vous ne désirez pas aligner une zone de texte, appuyer sur la touche Ctrl pendant le déplacement de celle-ci.

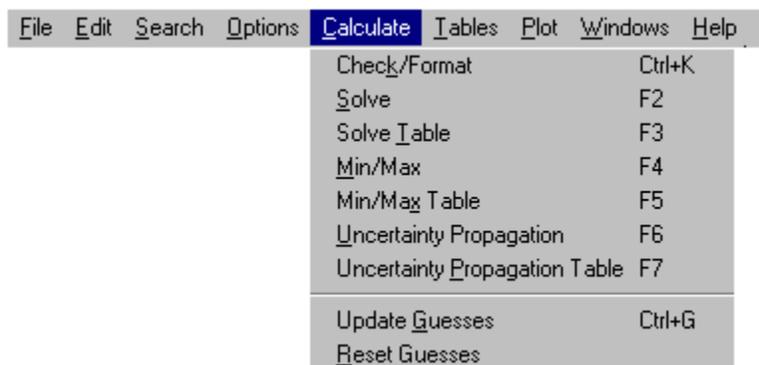


L'onglet 'Complex' indique à EES qu'il va travailler ou non en mode COMPLEXE. La directive \$COMPLEX ON/OFF placée dans la fenêtre d'Equations a le même effet.

Il est possible d'indiquer l'opérateur des imaginaires : I ou J en cochant une des 2 cases situées à droite de la ligne 'Imaginary operator symbol'.

Si vous appuyez sur le bouton 'OK', les changements s'appliqueront dans la session en cours. (Le bouton 'Store' rend ces changements permanents).

Le menu de calculs (Calculate)

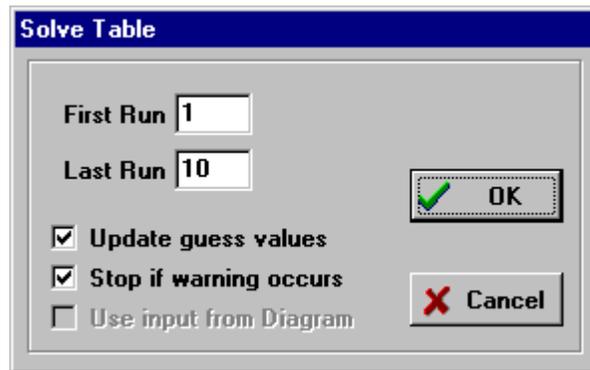


Check/Format vous permet de recompiler toutes les équations et d'appliquer les options de formatage mentionnées via la commande Preferences du menu Options. La première erreur de syntaxe trouvée sera affichée. Si aucune erreur n'est trouvée, EES indiquera le nombre d'équations et de variables du problème.

Solve permet de fournir la solution au problème posé. La commande 'SOLVE' vérifie tout d'abord la syntaxe des équations puis si aucune erreur n'est trouvée, EES affichera les résultats dans la fenêtre de Solution. Les méthodes de calculs utilisées par EES sont décrites dans l'Annexe B. Pendant les calculs, une boiter de dialogue est affiché afin d'en

indiquer la progression. Une fois les calculs effectués, une boîte d'information s'affiche. Elle mentionne le temps de calcul, le nombre de bloc, le maximum résiduel et le changement maximum de la valeur d'une variable depuis la précédente itération. Si le nombre d'inconnues est supérieur au nombre d'équations, EES affichera la fenêtre de Debugage afin de vous aider à localiser la source d'erreur. Si la fenêtre de Diagram est utilisée pour saisir des valeurs, celle-ci doit être visible lors de l'invocation de la commande Solve (cf. Chapitre 2).

Solve Table : cette commande permet de résoudre les calculs utilisant des données de la table paramétrique (voir plus loin dans ce chapitre). Lors de l'invocation de la commande SOLVE TABLE, la boîte de dialogue suivante apparaît :

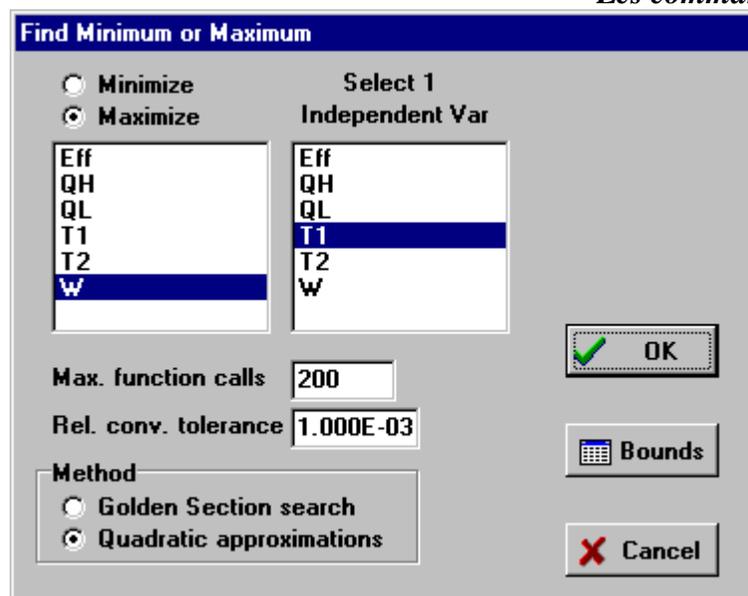


Chaque ligne de la table correspond à un problème différent. La valeur des variables indépendantes est affichée en format normal. La valeur des variables dépendantes est affichée en gras ou en italique. La valeur de l'ensemble des variables sera remplacée par les nouveaux résultats. Si l'option 'Update Guess Values' est cochée, la valeur par défaut des variables utilisera la valeur calculée lors de la précédente itération. Sinon, EES utilisera les valeurs par défaut paramétrées via la commande **Variable Info**.

Un message d'avertissement s'affiche si une corrélation sortant de son domaine d'application est utilisée. Si l'option 'Stop if warning occurs', EES arrêtera les calculs de table à l'endroit où l'avertissement a été signalé. Sinon, EES continuera les calculs et affichera à la fin du processus de résolution un message d'avertissement.

L'option 'Use input from Diagram' vous permet si la fenêtre de Diagramme est visible d'utiliser les variables d'entrée (si elles existent) du diagramme.

Min/Max est utilisé pour trouver le maximum et le minimum d'une variable indéterminée d'un système d'équations comportant au moins 1 degré de liberté (limité à 10). EES commencera par vérifier la syntaxe de vos équations. Si aucune erreur n'est trouvée, la boîte de dialogue suivante apparaît :



Les variables à minimiser (minimize) ou à maximiser (maximize) sont sélectionnées dans la liste située à gauche. La liste située à droite indique les variables indépendantes pour lesquelles la valeur sera modifiée lors de la recherche de l'optimum. Il est nécessaire de sélectionner autant de variables indépendantes qu'il y a de degrés de liberté dans la fenêtre d'Equations. Le nombre de variables indépendantes à sélectionner est indiqué au dessus de la liste de droite. Pour sélectionner ou désélectionner une variable, il suffit de cliquer sur son nom.

Dans le cas, où il n'y a qu'un degré de liberté, EES minimisera ou maximisera la variable sélectionnée en utilisant la méthode de la recherche de la section dorée (Golden Section search), ou la méthode d'approximation quadratique (Quadratic approximations) (Voir la description de ces algorithmes en annexe B).

EES nécessitent de disposer de valeurs finies de limites inférieures et supérieures pour chaque variable indépendante. EES. Noter que du choix des valeurs par défaut (Guess Value) ainsi que des limites (bounds) dépendra la probabilité de trouver la valeur optimum. Vous pouvez visualiser les valeurs par défaut, les limites des variables en cliquant sur le bouton 'Bounds'. Ainsi, une boîte de dialogue quasi similaire à la boîte de dialogue 'Variable Info' sera affichée.

Le label 'Number of function calls' correspond au nombre maximum de fois où les équations seront résolues. Les calculs s'arrêtent si :

- 1) la valeur relative de changement des variables indépendantes entre deux itérations est inférieure à la tolérance spécifiée.
- 2) Le nombre d'itérations dépasse celui fixé.

EES arrêtera également les calculs si les équations ne peuvent être résolues avec les valeurs par défaut des variables indépendantes (spécifiées via la commande Stopping Criteria du menu Options).

Min/Max Table : cette commande est similaire à la précédente, à la différence près que les calculs seront répétés pour chaque ligne de la table paramétrique (voir la description des commandes afférentes à la table paramétrique). Comme pour la commande **Min/Max**, une boîte de dialogue apparaît dans laquelle vous pouvez sélectionner les variables que vous souhaitez maximiser ou minimiser ainsi que les variables indépendantes. Néanmoins, dans ce cas, les variables à optimiser doivent figurer dans la table paramétrique. Il convient de spécifier l'itération de départ (start runs) et l'itération de fin (stop runs) de la table paramétrique entre lesquels vont s'effectuer les calculs. Les valeurs affichées en mode normal dans la table paramétrique sont fixées. Les variables à optimiser et les variables indépendantes doivent être identiques à chaque itération. Si aucune erreur n'est signalée, l'optimum est calculé.

Uncertainty Propagation détermine l'incertitude d'une variable calculée et sélectionnée en fonction de l'incertitude d'une ou de plusieurs valeur(s) mesurée(s) desquelles elle dépend. Dans la plupart des cas, une quantité telle l'énergie par exemple n'est pas mesurée directement, elle est fonction de variable(s) mesurée(s). (Ex. $Y=f(X_1, X_2, \dots)$). Les variables mesurées, X_1, X_2 , etc. ont une certaine variation appelée incertitude. Dans EES, cette incertitude est représentée avec le symbole \pm , Ex. : $X_1 = 300 \pm 2$.

Le but de cette commande est de calculer la propagation de l'incertitude des différentes valeurs mesurées sur la valeur finale. (Y dans notre cas). La méthode employée pour déterminer la propagation de l'incertitude est décrite dans le manuel technique « NIST » (réf. : NIST Technical Note 1297, Taylor B.N. and Kuyatt, C.E., Guidelines for Evaluating and Expressing the Uncertainty of NIST Measurement Results, National Institute of Standards and Technology Technical Note 1297, (1994)).

Partant du principe que les mesures individuelles sont aléatoires et qu'il est impossible de les corrélérer, l'incertitude d'une quantité calculée est estimée par :

$$U_Y = \sqrt{\sum_i \left(\frac{\partial Y}{\partial X_i} \right)^2 U_{X_i}^2}$$

U représente l'incertitude de la variable. Après avoir sélectionné cette commande, EES affiche deux listes de variables. La liste de gauche contient les variables dont l'incertitude doit être calculée. La liste de droite contient les variables mesurées. Noter que les variables mesurées doivent être des constantes numériques référencées dans une équation. Pour spécifier l'incertitude associée à une variable mesurée, cliquer sur le bouton 'Set Uncertainties'. Une seconde boîte de dialogue s'affiche. Elle contient un tableau rassemblant les variables sélectionnées ainsi que leurs unités, l'incertitude absolue et l'incertitude relative. Une valeur d'incertitude doit être spécifiée pour chacune des variables. Cliquer ensuite sur le bouton 'Ok' pour valider les changements. Cliquer sur le

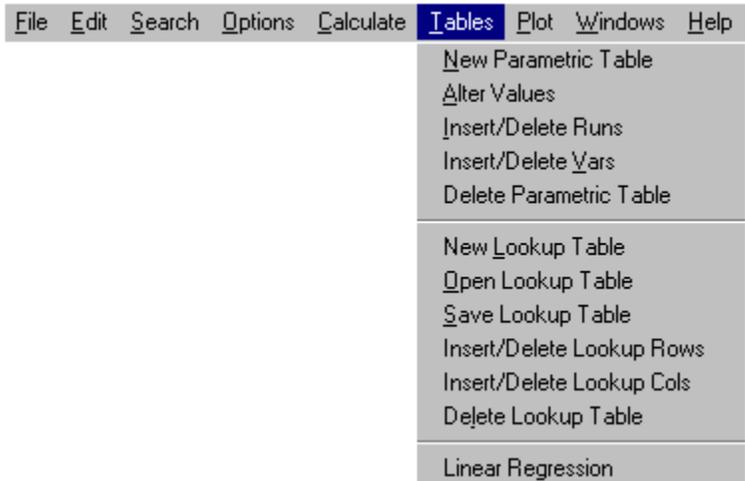
bouton 'Ok' de la boîte de dialogue 'Uncertainties Propagations' pour commencer les calculs.

Une fois les calculs effectués, EES affiche la fenêtre de Solution contenant les variables calculées et mesurées et leur incertitude relative. La dérivé partielle d'une variable calculée en fonction de variables mesurées peut être visualisée.

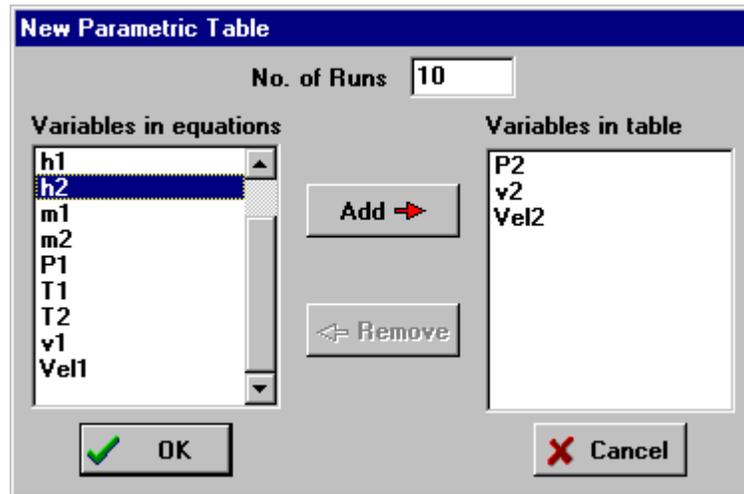
Uncertainty Propagation Table a le même effet que la commande précédente. La différence est que cette commande permet de répéter le calcul de propagation de l'incertitude d'une ou de plusieurs mesure(s) en utilisant la table paramétrique. Les variables calculées et mesurées doivent apparaître dans la table paramétrique avant d'invoquer cette commande. Une fois la commande 'Uncertainty Propagation Table' invoquée, une boîte de dialogue apparaît. La liste de gauche indique les variables dont l'incertitude doit être calculée. La liste de droite contient les variables mesurées. Les calculs de la table paramétrique seront effectués lorsque le bouton 'Ok' sera pressé. La valeur et l'incertitude de chaque variables ainsi que les variables mesurées apparaîtront dans la table paramétrique. Les variables calculées peuvent être représentées graphiquement à l'aide de bâtons permettant de visualiser l'erreur de propagation (menu PLOT).

Update Guesses remplace les valeurs initiales de chaque variable de la fenêtre d'Equations par la valeur déterminée lors des précédents calculs. Cette commande n'est accessible que si EES n'a pas rencontré d'erreur lors des calculs précédents. La commande 'Update Guesses' permet d'améliorer l'efficacité dans la résolution des calculs car EES va s'appuyer sur ces valeurs 'correctes' pour les calculs suivants. Noter que le bouton 'Update' de la boîte de dialogue **Variable Info** a le même effet.

Reset Guesses remplace la valeur initiale de chaque variable de la fenêtre d'Equations par la valeur par défaut associée à chacune de ces variables. Si rien n'a été précisé auparavant, EES remplacera la valeur initiale par : 1.0. (cette valeur peut être modifiée dans le menu **OPTIONS**, dans l'onglet 'Default Info'). N'utiliser cette commande que lorsque vous rencontrez des problèmes de convergence et que vous avez changé les valeurs initiales pour essayer de trouver une solution.



New Parametric Table permet de générer une nouvelle table paramétrique (cette action efface la table paramétrique existante). La table paramétrique permet de faire des calculs répétés pour résoudre par exemple des équations différentielles. Les résultats peuvent être représentés graphiquement par le biais du menu PLOT. Lors de l'invocation de cette commande, une boîte de dialogue s'affiche dans laquelle doivent être saisies des informations.



Le nombre d'itérations qui correspond au nombre de ligne de la table est saisi dans le champ 'No. of Runs'. Toutes les variables aussi bien dépendantes qu'indépendantes devant apparaître dans la table doivent être sélectionnées dans la liste déroulante de gauche (pour sélectionner une variable, il vous suffit de cliquer sur son nom; les noms de variables sont affichés suivant l'ordre alphabétique). Le bouton 'Add' permet de sélectionner également les variables. Noter que les variables qui figureront dans la table paramétrique apparaissent dans la liste de droite. Ces variables correspondront à des colonnes de la table. Il est possible de supprimer une variable déjà sélectionnée en cliquant sur le bouton 'Remove' ou en double cliquant sur son nom de la liste de droite.

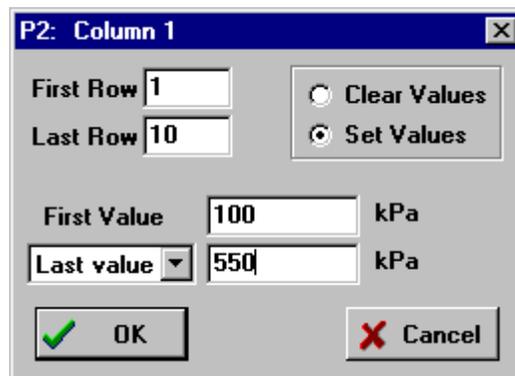
Le bouton 'OK' vous permet de valider les changements effectués et de générer la nouvelle table.

La table paramétrique fonctionne comme une feuille de calculs d'un tableur. Une valeur numérique peut être saisie dans chacune des cellules. Les valeurs numériques sont considérées comme des variables indépendantes et sont affichées en format normal. La saisie d'une valeur numérique dans la table paramétrique est similaire à l'affectation d'une valeur à une variable dans la fenêtre d'Equations. Les variables dépendantes déterminées et affichées dans la table apparaîtront en bleu, en gras ou en italique (le format d'affichage est configuré via la commande **Preferences**) quand l'utilisateur aura invoqué la commande **Solve Table**, **Min/Max Table**, ou **Uncertainty Propagation Table**.

Si une variable est déjà spécifiée dans la table paramétrique, elle ne peut l'être dans la fenêtre d'Equations.

Chaque ligne de la table correspond à un calcul séparé. La valeur des variables indépendantes peut différer d'une ligne à l'autre. Cependant, pour chaque ligne, le nombre de variables indépendantes plus le nombre d'équations doit être égal au nombre total de variables du problème.

Alter Values permet d'entrer ou d'effacer automatiquement les valeurs d'une variables sur plusieurs itérations. Il existe deux façons pour changer la valeur d'une donnée de la table paramétrique. Premièrement, les changements peuvent s'effectuer directement dans la cellule. Deuxièmement, un clic sur le bouton  situé dans le coin supérieur droit de chaque colonne invoque la boîte de dialogue ci-dessous, vous permettant de modifier la valeur des données de la colonne.:

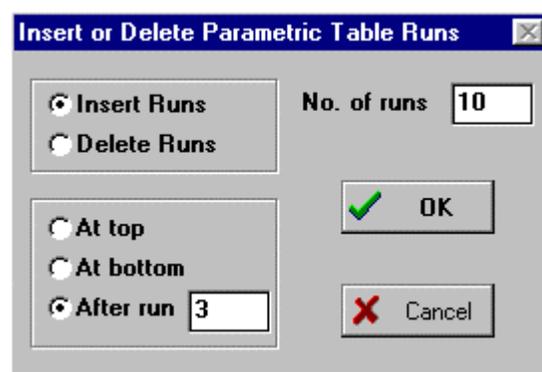


Les pas de calculs (correspondant aux lignes de la table : 'rows') qui doivent être modifiés sont saisis dans la cellule située en haut à gauche de la boîte de dialogue. La variable à modifier est sélectionnée dans la liste déroulante en cliquant sur son nom. Le contenu de la colonne relative à cette variable sera effacé si l'option 'Clear Value' est cochée. Si l'option 'Set Value' est cochée, les valeurs de la variable sélectionnée seront entrées automatiquement dans la table à partir de la valeur choisie dans le champ 'First value'. La liste déroulante située en dessous du champ 'First Value' permet de contrôler la façon

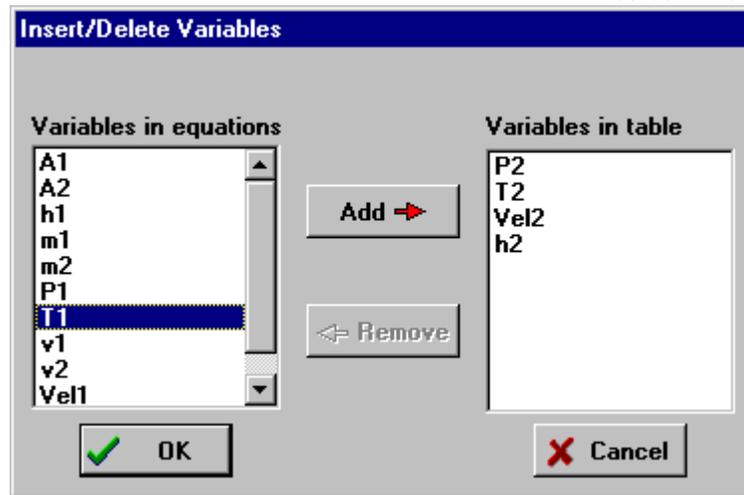
dont les valeurs vont être générées. Les choix possibles sont : ‘Last Value’ (choix de la dernière valeur de la colonne), ‘Increment’ (incrémentation), et ‘Multiplier’ (multiplication). ‘Increment’ et ‘Multiplier’ permettent de générer une série de valeurs calculée par l’ajout ou la multiplication d’un facteur à la valeur de la cellule précédente. Ce facteur est saisi dans le champ situé à droite du label ‘First Value’. Si l’option ‘Last Value’ est choisie, le facteur incrémental utilisé sera de telle sorte que la dernière valeur de la colonne sera celle spécifiée dans le champ de droite. Le bouton ‘Apply’ permet de valider les changements dans la table paramétrique en laissant affichée la boîte de dialogue pour permettre de saisir de nouveaux changements. Le bouton ‘Ok’ permet de valider les changements, et ferme la boîte de dialogue.

Des valeurs numériques peuvent être saisies directement dans la table ou en utilisant la commande **Alter Values** . Considérées comme des variables indépendantes, elles sont affichées en format normal. Celles-ci sont fixées en tant que constantes pour chaque pas de calculs. Les variables dépendantes sont affichées en bleu, en gras ou en italique suivant les préférences de l’utilisateur (cf. commande **Preferences**). Ces valeurs sont directement affichées lors de l’invocation des commandes **Solve Table** ou **Min/Max Table** . Noter que les valeurs des variables ne peuvent être assignées à la fois dans la fenêtre d’Equations et dans la table paramétrique (dans ce cas un message d’erreur apparaît).

Insert/Delete Runs permet d’ajouter ou de supprimer une ou plusieurs lignes de la table paramétrique à la position spécifiée.



Insert/Delete Variables permet d’ajouter ou de supprimer des variables dans la table paramétrique existante :



La liste de droite affiche les variables qui sont présentées dans la table paramétrique. Des variables pouvant être ajoutées apparaissent dans la liste de droite. Pour ajouter une ou plusieurs variable(s), il suffit de cliquer sur leur nom dans la liste de gauche (les variables sélectionnées apparaissent en inverse vidéo). Cliquer sur le bouton 'Add' pour ajouter ces variables (un double clic sur le nom de la variable produit le même effet). Les variables à supprimer de la table sont dans la liste de droite, pour les effacer, il vous suffit de les sélectionner et de cliquer sur le bouton 'Remove'.

Les variables apparaîtront dans les colonnes de la table paramétrique dans le même ordre d'apparition que dans la liste de droite. Cet ordre peut être modifié en cliquant sur le nom de variable à déplacer et tout en maintenant le bouton de la souris appuyé, déplacer la variable à la position désirée puis lâcher le bouton. L'ordre des colonnes peut lui être aussi modifié en cliquant sur l'en-tête de la colonne (voir Chapitre 2).

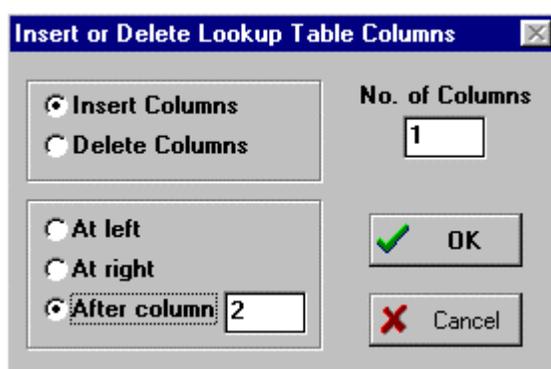
Delete Parametric Table, après confirmation, efface la table paramétrique et libère la mémoire correspondante allouée.

New Lookup permet de générer une table en spécifiant le nombre de lignes (row) et de colonnes (columns) dans laquelle on peut mettre des données (c'est en quelque sorte une base de données de nombres). Si une table 'Lookup' existe déjà, celle-ci sera effacée lors de l'invocation de cette commande. Les données tabulaires peuvent être automatiquement interpolées, dérivées et utilisables dans la solution d'un problème via les fonctions **Interpolate**, **Differentiate**, **Lookup**, **LookupRow**, et **LookupCol** (voir chapitre 4). Une seule table 'Lookup' apparaît dans la fenêtre 'Lookup' (Lookup Table Window). Ceci étant, les données de ce type de table peuvent être sauvegardées dans un fichier (extension .LKT) et appelées directement à l'aide des fonctions **Interpolate**, **Differentiate**, **Lookup**, **LookupRow**, et **LookupCol** dans la fenêtre d'Equations. Ainsi, il est possible en entrant et en utilisant des données tabulaires de déterminer la solution d'un problème qui utilise des relations entre fonctions de variables tabulaires.

Open Lookup permet de lire les informations présentes dans un fichier 'Lookup' à l'aide des commandes 'Save' et 'Save As Lookup' et les affiche dans la fenêtre 'Lookup'. (voir chapitre 4).

Save Lookup copie les données de la fenêtre 'Lookup' dans un fichier. Deux types de fichier sont acceptés : des fichiers de type binaire (.LKT) ou de type textuel (.TXT). Ces fichiers peuvent être ensuite lus à partir de la commande **Open Lookup Table** ou en utilisant les fonctions **Interpolate**, **Differentiate**, **Lookup**, **LookupRow**, et **LookupCol**. Noter que la table 'Lookup' est enregistrée lors de la sauvegarde du fichier EES en cours d'utilisation. (voir chapitre 4 pour de plus amples informations)

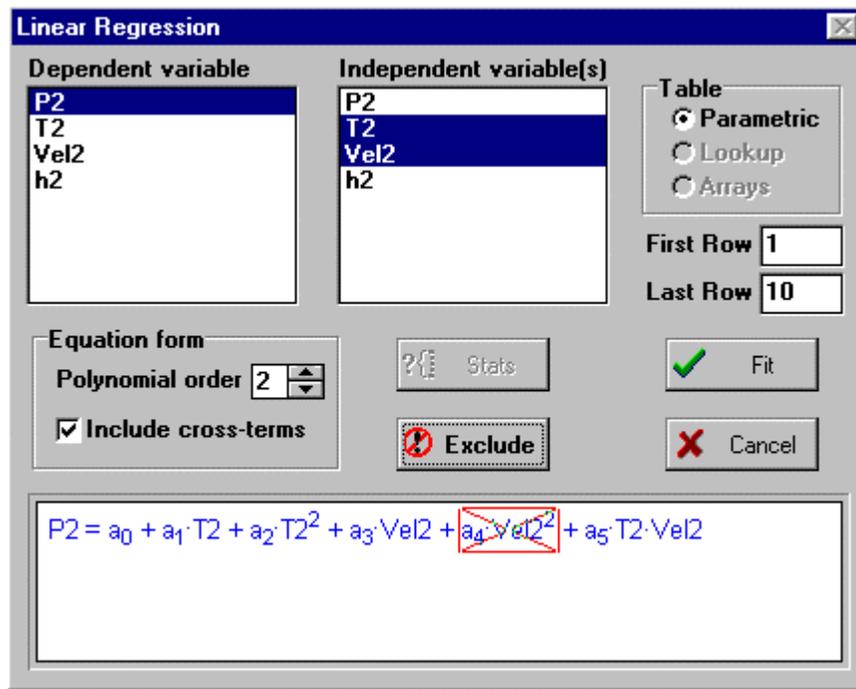
Insert/Delete Lookup Rows et **Insert/Delete Lookup Cols** permet d'ajouter ou de supprimer respectivement une ou plusieurs lignes ou colonnes de la table 'Lookup'



Delete Lookup après confirmation, efface la table 'Lookup' et libère la mémoire correspondante allouée.

Linear Regression permet de réaliser la régression linéaire de données issues de la table paramétrique, d'une table 'Lookup' ou d'un tableau. Noter que la commande 'Curve-Fit' du menu PLOT a le même effet mais uniquement pour une seule variable indépendante. Avec la commande 'Linear Regression' les données de n'importe quelle colonne peuvent être régressées comme une fonction de données issue de la table paramétrique jusqu'à 6 autres colonnes.

La boîte de dialogue montrée ci-dessous apparaît quand la commande est invoquée. Sélectionner la table que vous voulez utiliser en cochant l'une des cases de la section située en haut à droite de la boîte de dialogue. Vous devez également spécifier la ligne de début et de fin de la table (First Row et Last Row) en alimentant les champs situés au-dessus de la section. Spécifier les variables dépendantes figurant dans la liste de gauche en cliquant sur leur nom. Les variables dépendantes figurent dans la liste de droite, pour les sélectionner, procéder comme auparavant (pour désélectionner une variable il vous suffit de cliquer sur son nom).



Les variables dépendantes seront représentées comme des fonctions linéaires polynomiales de variables indépendantes. Le degré du polynôme est spécifié dans le champ 'Polynomial Order'. Si la case 'Include cross-term' est cochée Une fois toutes les informations entrées, une représentation de l'équation devant être fit est affichée en bas de la table de la boîte de dialogue.

Vous avez la possibilité d'exclure certains termes de l'équation en cliquant sur l'un d'entre eux. Cette action affichera un rectangle autour du terme à exclure sélectionné et rendra le bouton 'Exclude' actif. Cliquer alors sur ce bouton pour ôter le terme sélectionné de l'équation. Un rectangle de couleur rouge apparaît autour de celui-ci. Si, plus tard, vous désirez inclure ce terme, cliquer sur celui-ci. Ainsi le titre du bouton 'Exclude' se changera en 'Include'. Cliquer sur celui-ci pour inclure le terme sélectionné.

Quand la forme de l'équation est celle que vous désirez fiter, cliquer sur le bouton 'Fit'. Une fois le processus de **fitage** réalisé, l'équation fittée apparaîtra dans une fenêtre d'affichage. Le bouton 'Stats' devient actif.

Un clique sur ce bouton vous permet d'afficher les coefficients, leur valeur et l'erreur standard associées. D'autres statistiques telles que la 'root-mean-square', le biais, la valeur R^2 . Les coefficients exclus sont représentés dans la table par des étoiles. Ces coefficients peuvent être copiés dans le presse-papiers en cochant la case 'Copy to clipboard'

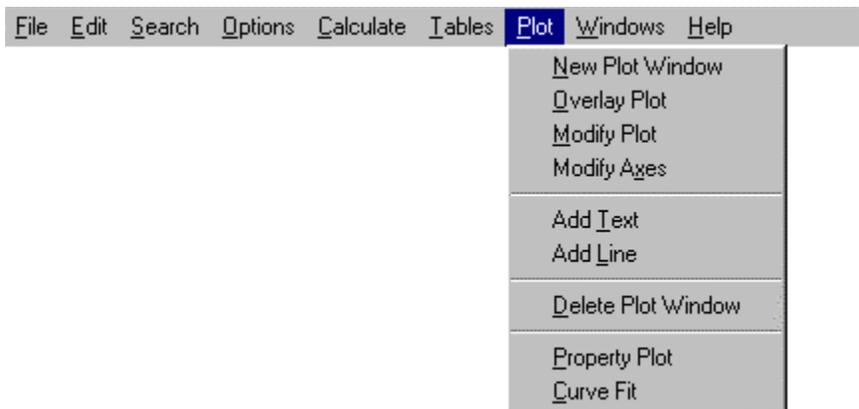
Une fois le processus de fitage réalisé avec succès, le bouton 'Fit' de la boîte de dialogue 'Linear Regression' ainsi que le bouton 'Cancel' seront respectivement changés en boutons 'Copy' et 'Done'. Le bouton vous permet de copier l'équations dans le presse-papiers. Ainsi il vous est possible de la copier dans la fenêtre d'Equations de EES ou dans tout autre application Windows.

	Value	Std. Error
a0	-3.341901E+03	1.235098E+02
a1	1.104265E+02	5.223082E+00
a2	-5.933022E-01	5.537634E-02
a3	1.671453E+01	4.459992E-01
a4	*****	*****
a5	-3.360885E-01	9.407243E-03

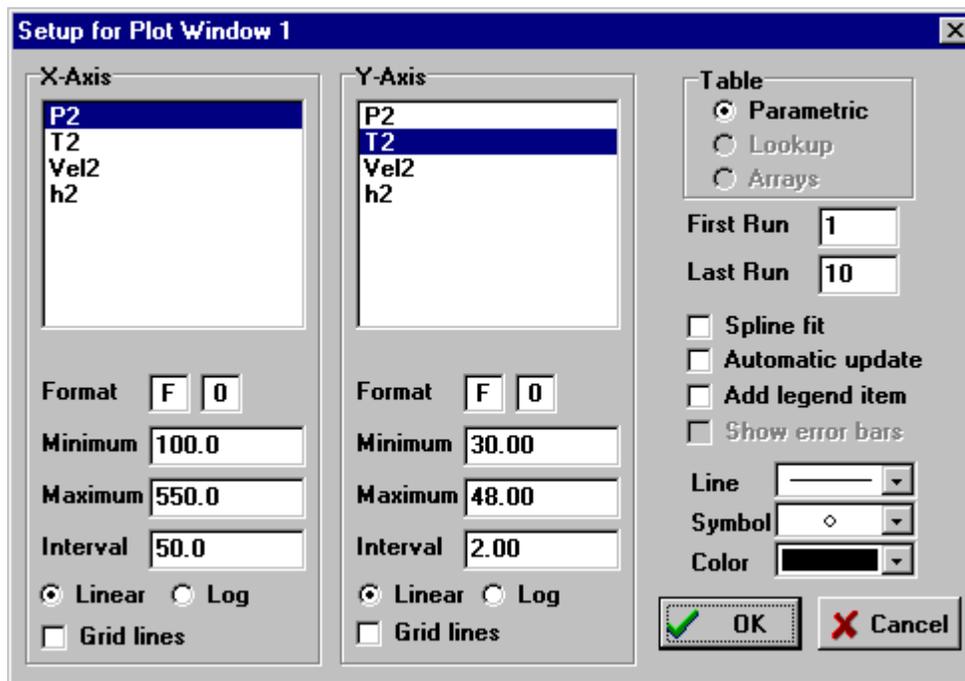
No. points = 10
rms = 2.3561E-01
bias = 5.3013E-16
R² = 100.00%

Copy to Clipboard

OK



New Plot Window permet de tracer toute variable issue d'une table paramétrique, 'Lookup' ou d'un tableau en fonction d'autres variables de cette table. Jusqu'à dix fenêtres de Tracés peuvent être visualisées simultanément. Utiliser la commande **Overlay Plot** si vous désirez superposer des courbes sur un même graphique. Les informations nécessaires aux tracés sont spécifiées dans la boîte de dialogue montrée ci-dessous. Toutes les informations fournies dans cette boîte de dialogue peuvent être modifiées via les commandes **Modify Axes** et **Modify Plot** (voir chapitre 2).



Tout d'abord, sélectionner la table à utiliser, en cochant l'une des cases de la section située en haut à droite (les tables non définies ne sont pas accessibles). Sélectionner l'intervalle dans lequel EES va puiser la valeurs des variables (champs : 'First Run' et 'Last Run'). Ensuite, sélectionner les variables devant être tracées sur les axes x- et y- en cliquant sur

leur nom dans la liste $-x$ et $-y$. EES sélectionnera automatiquement le nombre de chiffres significatifs, les valeurs minimum et maximum pour l'axe des abscisses et des ordonnées, ainsi que l'intervalle. Tous ces paramètres peuvent être changés par la suite. Deux types d'échelle sont possibles : linéaire (cocher la case 'Linear') ou logarithmique (cocher la case 'Log').

Les deux champs situés à droite du label format permettent d'accéder à un menu POPUP afin de paramétrer le format de représentation des nombres de l'échelle. Trois choix sont possibles : 'automatic' (mise en format automatique), 'Fixed décimal' (nombre fixe de digit) et 'Exponential' (format exponentiel). Le champ situé à droite permet de paramétrer le nombre de chiffres à afficher pour le format 'Fixed decimal' et 'Exponential'.

L'option 'Grid lines' trace un quadrillage sur le graphique. Le format est spécifié à l'aide de valeurs internes.

Le type de ligne 'line type', la représentation d'un point 'symbol' (ou 'bar type'), et la couleur du tracé 'color' d'une courbe sont paramétrés au travers des listes déroulantes respectives. L'option 'Spline fit' control will provide a spline-fit curve through the plotted points.

Quand l'option de mise à jour automatique : 'Automatic update' est cochée, le graphique sera redessiné à chaque changement de la table servant de support.

Si l'option 'Add legend item' est sélectionnée, une zone de texte possédant le même nom que la variable de l'axe des ordonnées sera placée dans le coin supérieur gauche du tracé, précédé du type de ligne et de point utilisés. Les zones de texte de légende peuvent être modifiées, déplacées comme toute autre zone de texte. (Cf. chapitre 2).

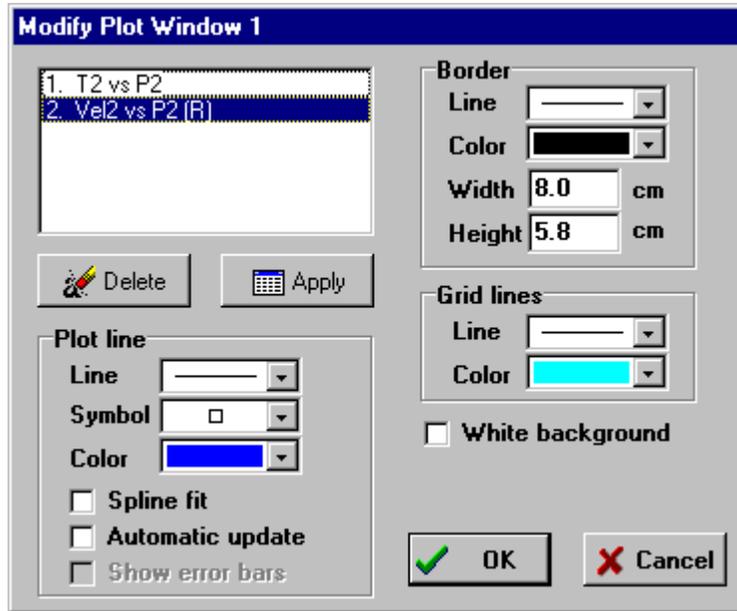
Le contrôle 'Show error bars' n'est accessible que si l'une des variables a été définie par la commande **Uncertainty Propagation Table**.

Overlay Plot permet de superposer des courbes sur un même graphique. L'utilisation de cette commande est identique à la précédente. Les tracés doivent partager la même échelle. Néanmoins, si l'échelle spécifiée est différente que celle utilisée dans le tracé existant, une boîte de dialogue vous permettra de garder l'échelle existante sur l'axe de ordonnées de gauche ou d'en créer une nouvelle qui apparaîtra sur l'axe des ordonnées de droite.

Modify Plot permet de modifier les caractéristiques du tracé existant.(Cette commande est accessible également en double-cliquant à l'intérieur de la zone de tracés).

Le graphique pour lequel des changements doivent être opérés, est sélectionné à l'aide de la liste déroulante situé en haut à gauche. Les tracés apparaissent dans la liste par ordre de création. Le commentaire (R) qui suit le tracé, indique que celui-ci a son échelle sur l'axe des ordonnées à droite.

Le type de ligne, de point, et la couleur peuvent être modifiés en utilisant les listes déroulantes situées en bas à gauche de la boîte de dialogue. Les options ‘Spline fit’ et ‘Automatic update’ peuvent être également changées (voir la commande New Plot Window).



Des champs sont fournis pour pouvoir modifier la taille et les caractéristiques de la bordure du graphique. Cliquer sur le bouton ‘Apply’ pour valider les changements avant de fermer cette boîte de dialogue ou de modifier un autre tracé.

Une courbe peut être effacée du graphique (les autres courbes restent intactes) en cliquant sur le bouton ‘Delete’. (la légende afférente au tracé effacé est, elle aussi, effacée). La commande **Delete Plot Window** décrite ci-dessous efface l’intégralité du graphique.

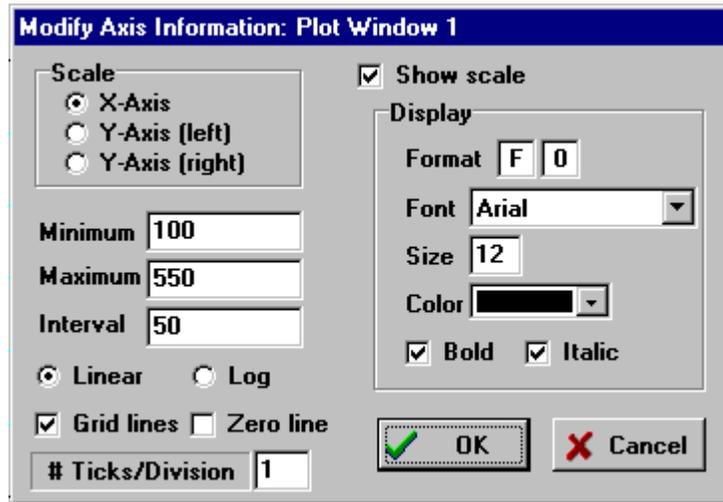
Modify Axes permet de modifier l’apparence des axes des abscisses et des ordonnées du tracé existant (cette commande peut être invoquée en cliquant sur l’axe pour lequel des changements doivent être opérés). La boîte de dialogue ci-dessous apparaît. L’axe à modifier est sélectionné dans la liste déroulante située en haut à gauche de la boîte de dialogue. Les valeurs min. et max., la valeur d’intervalle sont affichées. Ces valeurs peuvent être modifiées et le tracé sera dessiné de nouveau avec les valeurs choisies.

Le champ ‘No. Ticks/Division’ correspond au nombre de divisions mineures et majeures dans chaque intervalle. Si l’option ‘Grid lines’ est sélectionnée, les lignes de celle-ci prendront naissance à chaque division majeure. Noter que les lignes peuvent prendre naissance à des positions intermédiaires en sélectionnant une valeur supérieure à 0 dans le champ ‘No. Grids/Division’.

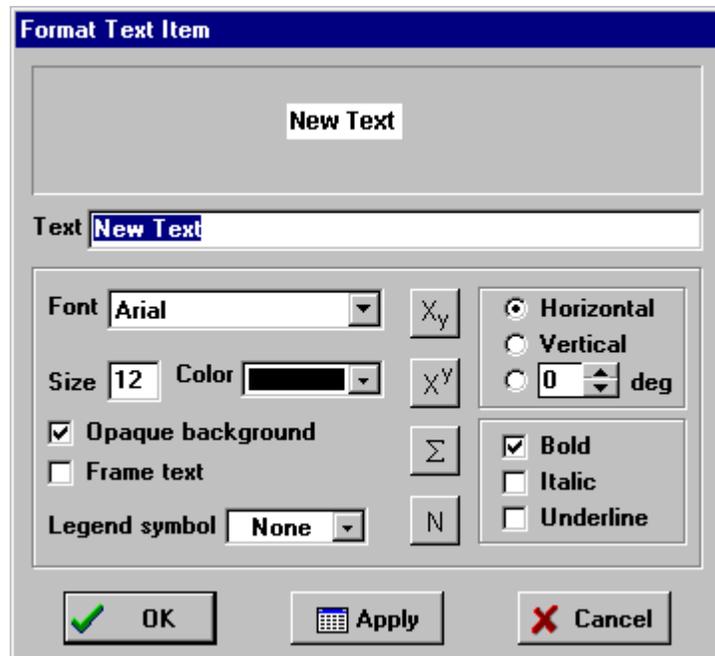
Le format d’affichage ‘Display Format’, la police ‘Font’, la taille de la police ‘Font Size’, son style ‘Font Style’ et la couleur ‘Color’ des nombres du repère peuvent être changés via

les listes déroulantes situées à gauche de la boîte de dialogue. Ces champs sont invisibles si l'option 'Show Scal' ('Afficher l'échelle') est cochée.

Le bouton 'Ok' permet de valider les changements et ferme la boîte de dialogue. La commande 'Cancel' ferme la boîte de dialogue sans tenir compte des changements effectués.



Add Text permet d'ajouter une zone de texte dans le graphique courant à l'aide de la boîte de dialogue ci-dessous. Le texte à afficher s'écrit dans le champ 'text'.



Le format d'affichage 'Display Format', la police 'Font', la taille de la police 'Font Size', son style 'Font Style' et la couleur 'Color' du texte peuvent être modifiés via les listes déroulantes. Le texte peut être orienté horizontalement, verticalement, ou selon un angle

spécifié (cliquer dans une des cases 'Horizontal', 'Vertical', ou sur les flèches du champ de rotation pour choisir l'orientation du texte). La case 'Opaque Control' affiche le texte sur fond blanc. L'option 'Frame Text' permet d'encadrer le texte.

D'autres possibilités de mise en forme de la zone sont offertes. Les 4 boutons (X^y) : exposant, (X_y) : indice, (Σ) : police symbol, (N) : normal permettent de modifier l'apparence du texte. Par exemple, pour créer un indice, sélectionner la partie du texte que vous desirez mettre en indice et cliquer sur le bouton (X_y). Noter que EES ajoute des caractères lors de cette action qui ne seront pas affichés dans le graphique.

EES permet d'associer des zones de texte horizontales avec un symbole de légende. La liste déroulante de légende contient les diverses options utilisables.

Toute zone de texte peut être déplacée à l'intérieur de la fenêtre de Tracés en cliquant sur celle-ci et en déplaçant la souris. Vous pouvez enfin modifier à tout moment les caractéristiques d'une zone de texte en double-cliquant sur la zone à modifier.

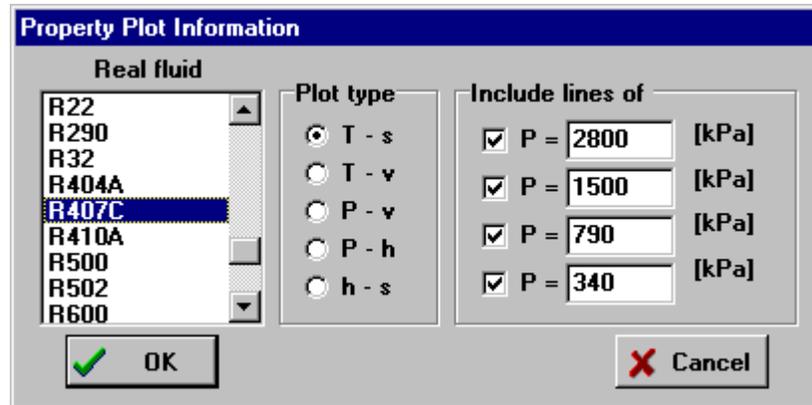
Add Line permet de placer une ligne ou une flèche à l'intérieur du graphique. Une fois la commande invoquée le curseur se change en croix. Cliquer à l'intérieur du graphique, et tout en maintenant le bouton de la souris appuyé, déplacer le curseur : une ligne apparaît. Relacher le bouton à l'endroit du point d'arrivée.

Vous pouvez par la suite déplacer, redimensionner, ou effectuer une rotation sur une ligne. Pour déplacer une ligne, cliquer sur celle-ci et, tout en maintenant le bouton enfoncé, déplacer le curseur. Pour effectuer une rotation ou changer la taille de la ligne, cliquer sur l'une de ses extrémités et, tout en maintenant le bouton enfoncé, déplacer le curseur. Vous pouvez enfin modifier les caractéristiques de la ligne en double-cliquant sur celle-ci pour faire apparaître la boîte de dialogue : 'Plot Line Characteristics'

Delete Plot Window permet d'effacer le contenu de la fenêtre de Tracés.

Property Plot permet de visualiser la représentation graphique d'une propriété thermodynamique pour une substance donnée. Une fois créée, il vous est possible de superposer le tracé de nouvelles propriétés thermodynamiques en utilisant la commande Overlay Plot. Les axes et le format de tracé peuvent être modifiés avec les commandes Modify Axes et Modify Plot.

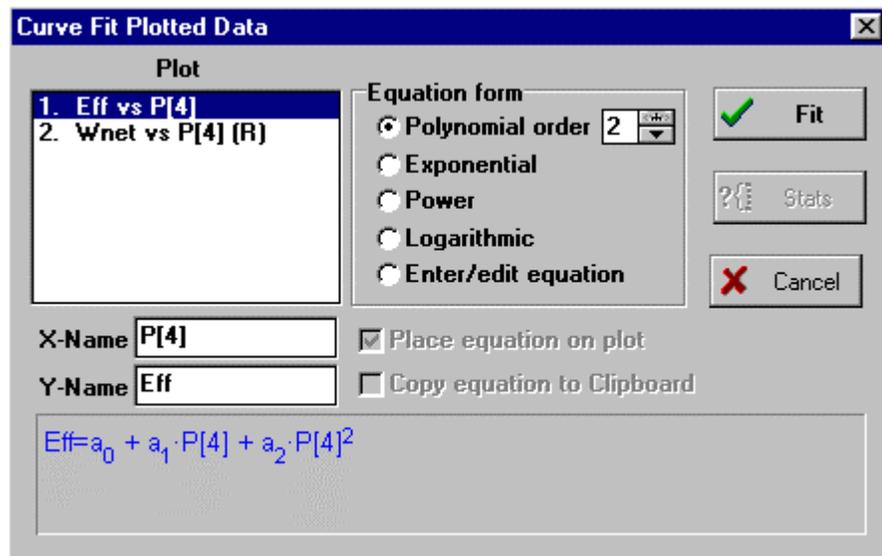
Sélectionner une substance dans la liste de droite. Pour chaque substance, excepté AIRH2O, 4 boutons sont disponibles pour spécifier les éléments à tracer : Temperature-entropy (T-s) : Temperature vs entropie, Temperature-volume (T-v) : Temperature vs volume, Pressure-volume (P-v) : Pression vs volume ou Pressure-enthalpy (P-h) : Pression vs entropie. AIRH2O possède un champ dans lequel la pression totale peut être spécifiée.



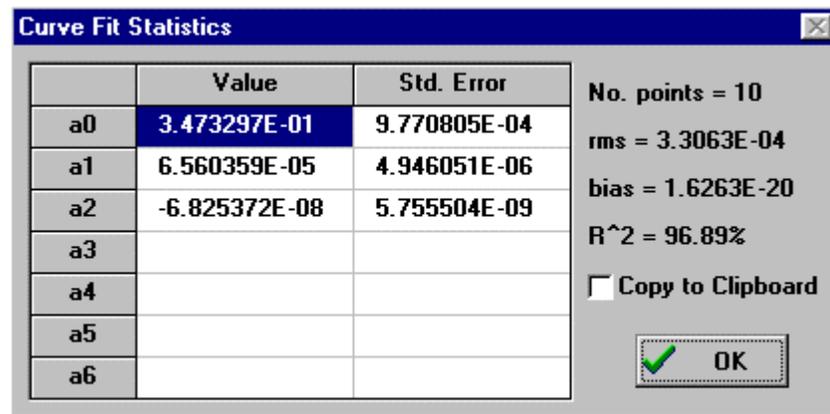
A droite sont placés des contrôles permettant de spécifier les valeurs de 4 isobares et isothermes. Ces isobares et isothermes seront affichées dans le graphique (pour ne pas les afficher, cocher la case relative à l'iso. considérée).

Curve Fit : will find the best fit of a smooth curve through a previously plotted set of data points using unweighted least squares. (The Curve fit dialog provides a fit with a single independent variable. The Linear Regression command in the table menu allows a variable to be fitted with as many as 6 independent variables.) The dialog window shown below will appear.

Chose the data to be fitted from the list of plots at the left. Note that data can be plotted from the Parametric Table, the Lookup Table, or the Arrays Table with the **New Plot** or **Overlay Plot** commands. Select the form of the curve fit by clicking the appropriate radio button. A sample of the equation form will appear in blue in the box at the bottom of the dialog window. The first four buttons correspond to commonly used equation forms for which linear least squares can be used to determine the unknown coefficients. The Enter/edit equation button allows you to enter any equation form or to edit a previously entered equation. The equation you enter may be linear or non-linear in the unknown parameters. You will be prompted to supply guess values bounds for the unknown parameters.

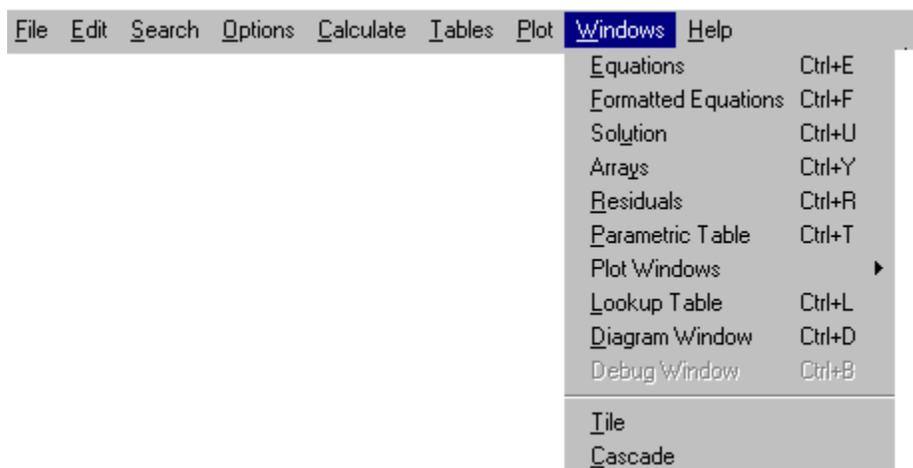


Click the Fit button (or press the Enter key). The fitted equation will be displayed in the box at the bottom of the dialog window. A Stats button will appear. Clicking the Stats button will display the following statistical information relating to the curve fit.



Std. Error is the standard error of the curve-fitted parameter value; rms is the root mean square error of the fit; bias is the bias error of the fit. R² is the ratio of the sum of squares due to regression to the sum of square about the mean of the data.

The Fit button will now have changed into the Plot button. Click the Plot button if you wish to have the curve fit equation overlaid on your plot. If the Plot Legend check box is selected, a legend containing the equation will be created and displayed on the plot. The curve fit equation will be copied to the clipboard if the To Clipboard checkbox is selected when either the Plot or Cancel button is selected.



Equations affiche la fenêtre d'Equations.

Formatted Equations vérifie la syntaxe des équations saisies et affiche la fenêtre de mise en forme usuelle des équations. (format mathématique).

Solution, Arrays and Residuals affichent respectivement les fenêtres de Solutions, de Tableaux et la fenêtre Résiduelle. Noter que ces fenêtres s'affichent d'elles-mêmes lors de l'invocation des commandes **Solve**, **Min/Max**, ou **Uncertainty Propagation**. Si un changement est effectué dans la fenêtre d'Equations, ces fenêtres seront automatiquement masquées. Dans le cas où EES est incapable de résoudre les équations posées, le nom de la fenêtre Solution sera changé en 'Iteration Value' (valeurs déterminées lors de la dernière itération de calcul). Les variables seront affichées dans la fenêtre Résiduelle.

Plot Windows affiche un sous menu pour chacun des 10 tracés possibles. L'invocation de l'un des ces sous menus affiche la fenêtre de Tracés correspondante. Ce menu sera 'grisé', si aucun tracé n'a été défini. Noter que tout graphique de la fenêtre de Tracé peut être copié dans le presse-papiers à l'aide de la commande 'Copy' du menu 'Edit'.

Parametric Table et Lookup Table affiche les fenêtres relatives à la table paramétrique et à la table 'Lookup' au premier plan.

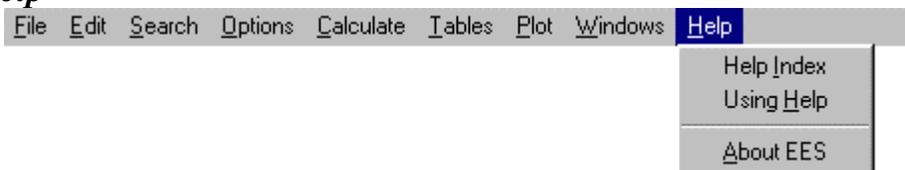
Diagram affiche la fenêtre de Diagramme. Un diagramme, ou un schéma de principe est saisi dans un utilitaire de dessin quelconque. Ensuite, via une opération de copier/coller, vous pouvez incorporer le schéma réalisé dans la fenêtre de Diagramme. Il est possible de saisir et d'afficher des valeurs à partir de cette fenêtre.

Debug Window : affiche la fenêtre de 'Débogage'. Cette commande est accessible

uniquement si le système d'équations soumis à EES comporte plus d'inconnues que d'équations.

Tile arrange les fenêtres actives de telle manière qu'il soit possible de visualiser une portion de celles-ci.

Cascade arrange les fenêtres actives de telle manière qu'il soit possible de visualiser la barre de titre de chacune d'entre elles.

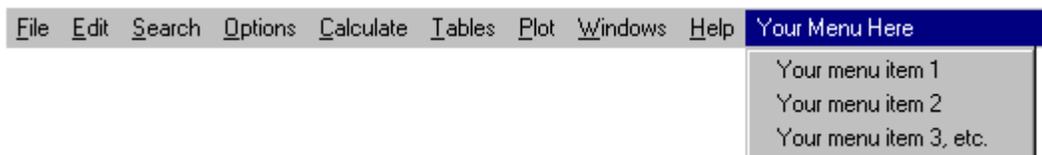


Help Index active l'aide. Une fenêtre contenant l'index des informations accessibles s'affiche.

Pour voir un sujet, cliquer sur celui-ci. L'aide est accessible également par l'appuie de la touche F1. L'aide affichée sera relative au contexte.

Using Help active une fenêtre décrivant comment utiliser l'aide.

About EES permet d'ouvrir la boîte 'A propos...' de EES. Elle contient entre autre, la version du programme, la quantité de mémoire disponible et le numéro d'enregistrement. Ces informations vous seront demandées dans le cas où vous feriez appel au support de F-Chart Software.



Le titre est peut être mal choisi. En fait, ce menu est paramétrable par l'utilisateur, conçu pour accéder rapidement aux fichiers problèmes EES. Ce menu peut être ouvert par l'intermédiaire de la commande 'Load Textbook' du menu 'File' ou en plaçant un fichier de type 'textbok' dans le répertoire USERLIB.

Un fichier 'Textbook' est un fichier ASCII avec l'extension (.TXB). Quand EES lit un fichier de ce type, il génère un menu 'Your Menu Here' comme montré ci-dessous. Ce menu a été généré à l'aide du fichier suivant :

```
Your Menu Here
1
Textbook information line 1
Textbook information line 2
Textbook information line 3
Reserved
>Your menu item 1
Descriptive problem name1 | FileName1.EES | HelpFile1.HLP | NO.BMP
Descriptive problem name2 | FileName2.EES | HelpFile2.HLP | NO.BMP
Descriptive problem name3 | FileName3.EES | HelpFile3.HLP | NO.BMP

>Your menu item 2
Descriptive problem name1 | FileName1.EES | HelpFile1.HLP | NO.BMP
Descriptive problem name2 | FileName2.EES | HelpFile2.HLP | NO.BMP
Descriptive problem name3 | FileName3.EES | NO.HLP | NO.BMP
>Your menu item 3, etc.
etc | FileName1.EES | HelpFile1.HLP | NO.BMP
```

La première ligne du fichier correspond au titre du menu qui apparaîtra dans la barre de menu à droite du menu 'Help'. La ligne qui suit indique un numéro de version utilisé de manière interne par EES. EES ignore cette ligne, mais le chiffre 1 doit être présent à cette ligne. Les 3 lignes suivantes, fournissent des informations relatives au 'Textbook' ou aux paramètres du problème. La quatrième ligne contenant le mot 'reserved' est fournie pour une utilisation ultérieure. EES ignore cette ligne, mais elle doit néanmoins être présente. Les lignes suivantes contiennent le nom du sous menu (précédé par le signe >) et par une ou plusieurs description(s) du problème.

Chaque description contient 4 types d'informations séparées par le caractère |. Le premier élément indique le nom du problème (128 caractères au maximum). Le second élément indique le nom du fichier EES. Le nom de fichier peut comporter également le nom du répertoire dans lequel il se trouve, par exemple : C:\myBook\Chapter1\Problem1.EES. Le troisième élément pointe vers un fichier aide optionnel. Ce fichier peut être issu d'un utilitaire de texte ou d'un programme de génération de fichier d'aide. Par convention, ce fichier doit comporter l'extension .HLP. Si aucun fichier d'aide n'est associé au problème, entrer l'instruction NO.HLP dans ce champ. Le dernier élément pointe vers un fichier dessin de type .BMP désignant une figure associée à votre problème. Si aucune figure n'est associée, entrer l'instruction NOBMP.BMP dans ce champ.

Le fichier 'Textbook' doit se trouver dans le même répertoire que tous les fichiers référencés dans ce fichier. Quand l'utilisateur invoque la commande 'Textbook' une boîte de dialogue affichant le nom des problèmes contenu dans le fichier 'Textbook' apparaît. Ainsi, il est possible d'en sélectionner un nom de problème et tous les fichiers relatifs à celui-ci seront ouverts.

Fonctions incorporées

EES dispose d'une grande bibliothèque de fonctions mathématiques. La plupart d'entre elles (Bessel, hyperbolique, etc.) sont particulièrement utiles dans des applications du niveau ingénieur. EES propose également une fonction de conversion des unités et des fonctions qui aident à manipuler les nombres complexes. EES se distingue d'autres solveurs d'équation sans aucun doute par sa librairie de fonctions et propriétés thermodynamiques incorporées. En effet, les propriétés thermodynamiques, de transport de la vapeur, de l'air, de l'ammoniac, du dioxyde de carbone et beaucoup d'autres sont utilisables par n'importe quels systèmes d'équations. EES offre la possibilité de déterminer une solution d'un système d'équation suivant des valeurs issues d'une table ('Lookup Table'). Dans la première partie de ce chapitre, le lecteur trouvera des informations sur les fonctions et propriétés incorporées. La deuxième partie de ce chapitre est consacré à l'utilisation de la table ('Lookup'). Noter que la plupart des informations présentes dans ce chapitre sont accessibles directement dans EES à l'aide de la commande 'Function info'.

Fonctions mathématiques

Toutes les fonctions mathématiques expliquées ci-dessous (exceptés **pi** et **tableRun#**) exigent un ou plusieurs argument(s) qui doivent être mis entre parenthèses. L'argument d'une fonction peut être une valeur numérique, un nom variable, ou une expression algébrique.

abs (X) : retourne la valeur absolue de l'argument. Dans le mode complexe, **abs** retourne l'angle de l'argument complexe. (Voir aussi **Magnitude(X)**);

angle (X), **angleDeg (X)** et **angleRad (X)** : retournent une valeur d'angle (appelé aussi argument) d'une variable complexe X. X est représenté comme égale à $X_r + i * X_i$.

La fonction **angle (X)** retourne $\arctan(X_i/X_r)$. Suivant le système d'unités, cette valeur sera exprimée en degré ou en radian. **angleDeg (X)** retournera toujours l'angle en degrés et **angleRad** en radians. Ces trois fonctions retournent l'angle dans le plan complexe exacte. Noter que ces fonctions sont utilisées uniquement pour extraire la valeur de l'argument d'un nombre complexe. Elles ne peuvent, en aucun cas, assigner une valeur d'argument à un nombre complexe. Par exemple, l'équation $\text{Angle}(X)=4$ produira une erreur.

arcCos (X) : retourne la valeur de l'arc cosinus de X. Les unités de l'angle (les degrés ou les radians) dépendront du choix du système d'unité pour les fonctions trigonométriques.

arcCosh (X) : retourne la valeur de l'arc cosinus hyperbolique de X.

arcSin (X) : retourne la valeur de l'arc cosinus de X. Les unités de l'angle (les degrés ou les radians) dépendront du choix du système d'unité pour les fonctions trigonométriques.

arcSinh (X) : retourne la valeur de l'arc du sinus hyperbolique de X.

arcTan (X) : retourne la valeur de l'arc tangente de X. Les unités de l'angle (les degrés ou les radians) dépendront du choix du système d'unité pour les fonctions trigonométriques.

arcTanh (X) : retourne la valeur de l'arc tangente hyperbolique de X.

bessel_I0 (X) retourne la valeur de zeroth-ordre fonction de Bessel Modifiée du premier genre pour la valeur d'argument X où $-3.75 \leq X < \infty$.

bessel_I1 (X) retourne la valeur de premier-ordre fonction de Bessel Modifiée du premier genre pour la valeur d'argument X où $-3.75 \leq X < \infty$.

bessel_J0 (X) retourne la valeur de fonction de Bessel de zeroth-ordre du premier genre pour la valeur d'argument X où $-3 \leq X < \infty$.

bessel_J1 (X) retourne la valeur de fonction de Bessel de premier-ordre du premier genre pour la valeur d'argument X où $-3 \leq X < \infty$.

bessel_K0 (X) retourne la valeur de zeroth-ordre fonction de Bessel Modifiée du deuxième genre pour la valeur d'argument X où $0 \leq X < \infty$.

bessel_K1 (X) retourne la valeur de premier-ordre fonction de Bessel Modifiée du deuxième genre pour la valeur d'argument X où $0 \leq X < \infty$.

bessel_Y0 (X) retourne la valeur de fonction de Bessel de zeroth-ordre du deuxième genre pour la valeur d'argument X où $0 < X < \infty$.

bessel_Y1 (X) retourne la valeur de fonction de Bessel de premier-ordre du deuxième genre pour la valeur d'argument X où $0 < X < \infty$.

cis (X) : est une fonction du mode complexe qui retourne $\cos(X)+i*\sin(X)$. Les unités exigées (les degrés ou les radians) de l'angle sont paramétrées grace à la commande Système d'unité pour les fonctions trigonométriques. Cependant, vous pouvez ajouter deg ou rad à l'angle pour forcer la fonction à opérer avec ce type d'unité. Par exemple, l'équation $V=3*cis(20deg)$ donnera à V la valeur tel que son module est égal à 3 et son argument à 20 degrés.

conj (X) : retourne le nombre complexe conjugué d'une variable complexe X. Pour une variable X tel que $X = X_r + i * X_i$, cette fonction retourne $X_r - i * X_i$. Noter que cette fonction retourne un résultat complexe.

Par exemple, l'équation $Y = \text{conj}(X)$, aura comme effet de donner à Y_r le module de X et à Y_I (argument de Y) la valeur opposée de la partie imaginaire de X.

converti ('From', 'A') : retourne le facteur de conversion a eu besoin de convertir des unités de la désignation d'unité spécifiée dans le 'De' la ficelle à que spécifié dans 'l'A' la ficelle. Les marques d'apostrophe sont facultatives. Par exemple, $FI = \text{converti}(\text{ft}^2, \text{dans}^2)$ régler FI à une valeur de 144 parce que 1 pied carré est 144 pouces carrés. La combinaison d'unités et les termes d'unité multiples peut être entrée. Dans une combinaison d'unités, tel que $\text{Btu/hr-ft}^2\text{-R}$, les unités individuelles sont séparées avec le tiret (c.-à-d., moins) ou les signes de division. Seulement un symbole de division peut être utilisé dans n'importe quel un terme. Toutes unités à la droite du symbole de division sont supposées pour être dans le dénominateur (c.-à-d., haussé à un pouvoir négatif.) Le ^ le symbole est facultatif si ft^2 et ft^2 sont équivalents. Le **converti** La fonction acceptera les termes d'unité multiples si chaque terme est enclos dans les parenthèses. Les termes sont séparés avec un facultatif * le symbole ou avec un/le symbole, comme dans l'exemple au dessous.

$$P = 15 * \text{Converti}((\text{lbm/ft}^3) * (\text{ft}) / (\text{s}^2/\text{ft}), \text{kPa})$$

Les symboles d'unité définis peuvent être montrés avec le **Info de Conversion d'unité** ordonner dans le **Options Menu**. Si vous trouvez qu'une unité vous avez besoin de n'est pas défini, vous pouvez l'entrer en éditant le fichier d'UNITÉS.TXT dans l'annuaire de EES.

cos(X) : retourne la valeur du cosinus de X. Les unités d'angle (degrés ou radians) dépendront du choix du système d'unité pour les fonctions trigonométriques.

cosh (X) : retourne la valeur du cosinus hyperbolique de X.

différentiate ('Filename', 'ColName1', 'ColName2', ColName2=Value) : retourne la dérivée déterminée à partir de deux colonnes d'un tableau. Voir le *L'utilisation des fichiers et tables 'Lookup' plus loin dans ce chapitre.*

erf (X) : retourne la fonction Gaussienne d'erreur de X.

erfc (X) : retourne le complément de la la fonction Gaussienne d'erreur de X : $1 - \text{erf}(X)$.

exp (X) : retourne l'exponentielle de X.

if (A, B, X, Y, Z) : permet de réaliser des structures conditionnelles. Si $A < B$; la fonction retourne la valeur de X. Si $A = B$, la fonction retourne la valeur de Y; si $A > B$, la fonction retournera la valeur de Z. Dans certains problèmes, l'usage de la fonction **si** peut causer une boucle sans fin. Il est donc préférable d'utiliser les fonctions **if then else**, **until while** et **goto** pour réaliser des structures conditionnelles. (cf. Chapitre 5 pour de plus amples informations).

imag (X) : retourne la partie imaginaire d'une variable complexe X. Noter que la fonction de **Imag** est utilisée pour extraire la partie imaginaire d'une variable complexe. Elle ne peut être utilisée pour assigner la partie imaginaire à un nombre complexe. Par exemple, l'équation $\text{Imag}(X)=4$ produira une erreur. Pour assigner une valeur à la partie imaginaire d'une variable complexe, taper : $X=4*i$ (X sera alors égal à $0 + 4*I$). Si vous souhaitez uniquement fixer la partie imaginaire de X, vous pouvez entrer $X_i=4.0$

intégral (Integrand, VarName) ou **intégral**(Integrand, VarName, le Début, l'Arrêt, l'Etape) retourne l'intégral de l'expression représentée par Integrand par rapport aux VarName variables, c.-à-d., $\int(\text{Integrand}) d(\text{VarName})$. Il y existe deux formes fondamentales de la fonction intégrale qui diffère dans leur confiance sur la table de Parametric. Si les valeurs de Début, l'Arrêt, et l'Etape ne sont pas fournies, le **intégral** La fonction est seulement utilisée conjointement avec la Table de Parametric. Dans ce cas, VarName doit être un nom variable légal qui à des valeurs définies dans une des colonnes de la table de Parametric et Integrand peut être une variable ou une expression impliquer VarName algébrique et les autres variables ou les valeurs. Si le Début, l'Arrêt, et (facultativement) l'Etape est fournie, EES intégrera numériquement toutes équations VarName variables qui impliquent, le montage de la valeur de VarName aux valeurs entre le Début et l'Arrêt comme approprié. Si l'Etape n'est pas fournie, le volonté de EES a choisi intérieurement une utilisation de taille d'étape un algorithme d'ajustement de stepsize automatique. Le **intégral** La fonction peut être utilisée pour résoudre la valeur initiale équations différentielles. Voir le Chapitre 7 pour l'information supplémentaire.

interpolate ('Filename', 'ColName1', 'ColName2', ColName2=Value) : retourne une valeur interpolée ou extrapolée d'un tableau de données ("Lookup file") ou d'un fichier de données (si le champ Filename est alimenté), ou des données de la table paramétrique utilisant une interpolation cubique. (cf. Voir les fonctions 'Lookup files' and 'Lookup table').

interpolate1 (Le "nom de fichier", 'ColName1', 'ColName2', ColName2=Value) se comporte comme la fonction **interpolate** mais utilise une interpolation linéaire.

interpolate2 (Le "nom de fichier", 'ColName1', 'ColName2', ColName2=Value) se comporte comme la fonction **interpolate** mais utilise une interpolation quadratique.

consulter ('*Filename*', Ligne, Colonne) : retourne la valeur située à la ligne : Ligne et à la colonne : Colonne d'un fichier *Filename* ou de la table 'Lookup table'. (l'argument *Filename* est facultatif). (cf. Voir les fonctions 'Lookup files' and 'Lookup table').

LookupCol ('*Filename*', Ligne, Valeur) : cette fonction retourne le numéro de colonne contenant la valeur 'Valeur' à la ligne Ligne d'un fichier ou d'une table 'Lookup table'. (cf. Voir les fonctions 'Lookup files' and 'Lookup table').

ln (X) : retourne le logarithme népérien de X.

log10 (X) : retourne le logarithme en base 10 de X.

magnitude (X) retourne la valeur du module d'une variable complexe X. Dans le mode complexe. Pour une variable complexe X de la forme $X_r + i*X_i$, cette fonction retourne $\sqrt{X_r^2 + X_i^2}$. Noter que cette fonction ne sert qu'à extraire le module d'une variable complexe et non pas à l'assigner.

max (X1, X2, X3, ...) retourne la valeur du plus grand de ces arguments.
(Le nombre d'argument doit être supérieur ou égal à 1).

min (X1, X2, X3, ...) : retourne la valeur du plus petit de ces arguments.
(Le nombre d'argument doit être supérieur ou égal à 1).

pi : variable réservée égale à 3.1415927.

produit (Arg, Series_info) : retourne le produit d'une serie de termes. Arg peut être une expression algébrique. Series_info contient le nom de la variable d'index de produit et les limites basses et supérieures (ces limites sont soit des nombres entiers soit des variables de type entier fixées). **produit** (j; j=1;4) retournera $1*2*3*4$ ou 24, qui est équivalent à $4!$. La fonction **produit** est très utile pour des opérations sur des variables de type tableau (X [j]). Par exemple, le produit du carré de 10 éléments d'un tableau X peut être obtenu en faisant :

produit (X[j]*X[j]; j=1;10).

real (X) : retourne la partie réelle d'une variable complexe X. A l'instar de la fonction magnitude, la fonction real (X) est utilisée uniquement pour extraire la partie réelle d'une variable complexe.

rond (X) : retourne une valeur égale à la valeur du nombre entier le plus proche de l'argument X.

sin(X) : retourne la valeur du sinus de X. Les unités d'angle (degrés ou radians) dépendront du choix du système d'unité pour les fonctions trigonométriques.

sinh (X) : retourne la valeur du sinus hyperbolique de X.

sqrt(X) : retourne la racine carrée de l'argument X (X doit être positif).

step(X) : retourne la valeur 1 si l'argument est plus grand ou égal à zéro; 0 sinon.

La fonction step peut être utilisée pour réaliser des tâches conditionnelles, similaires la fonction if. Cette fonction est disponible uniquement pour assurer la compatibilité avec des versions précédentes de EES. Il est plus commode d'utiliser les opérateurs if then else (cf. Chapitre 5)

sum (Arg, Series_info) retourne la somme d'un serie de termes, c'est-à-dire : $\sum \text{Arg}$. Arg peut être une expression algébrique. Series_info est le nom de la variable d'index de sommation, elle indique également les limites inférieures et supérieures. (ces limites doivent être des nombres entiers ou des variables de type entier). Par exemple : **sum**(J, j=1,4) retournera 1+2+3+4 ou 10. Cette fonction est très utile pour des variables de type tableau, X [j]. Par exemple, le produit de scalaire de deux vecteurs de coordonnées X et Y peut être obtenu par **sum**(X[j]*Y[j], j=1,2). (cf. voir le Chapitre 7 pour l'utilisation de la fonction somme pour des matrices).

tableRun# : retourne la Table de Parametric a couru le numéro, c.-à-d., le rang actuel dans la Table de Parametric ou zéro, si la Table de Parametric n'est pas utilisée dans les calculs. Cette fonction devrait être seulement utilisée avec le Résoudre la Table ou Table de Min/Max ordonner dans le Calculer Menu.

tableValue (Le rang, la Colonne) ou **tableValue** (Le rang, 'VariableName') Retourne la valeur emmagasinée dans un rang et une colonne spécifiés de la Table de Parametric. Le numéro de colonne peut être ou entré directement comme un numéro de nombre entier ou indirectement en fournissant le nom variable pour la colonne désirée, enclose par les apostrophes, l'e.g., **tableValue**(6,'ABC')⁴. Un message d'erreur sera engendré si le rang ou la colonne (ou le nom de variable correspondant) n'existe pas dans la Table de Parametric ou si la cellule de referenced n'a pas une valeur. Le **tableValue** La fonction est utile dans la solution de quelques-uns 'la marchant-solution' les problèmes de type dans lequel la valeur actuelle d'une variable dépendt de sa valeur dans les calculs précédents.

tan(X) : retourne la valeur de la tangente de X. Les unités d'angle (degrés ou radians) dépendront du choix du système d'unité pour les fonctions trigonométriques.

tanh(X) : retourne la tangente hyperbolique de X.

trunc(X) : retourne la partie entière d'un nombre décimal.

⁴ Pour la compatibilité avec plus tôt les versions, EES acceptera aussi le #précéder de symbole le nom variable au lieu d'enclore il dans les apostrophes, l'e.g., **tableValue** (6, #ABC).

UnitSystem ('Unittype') est une fonction qui permet à EES de savoir dans quel système d'unités (sélectionné dans le menu Unit System) il doit effectuer les calculs. Cette fonction requiert un seul argument. Les arguments autorisés sont "SI", "Eng", "Mass", "Molar", "Deg", "Rad", "kPa", "bar", "psia", "atm", "C", "K", "F", et "R". La fonction retourne 1 si le système d'unités correspond à l'argument de la fonction, 0 sinon. Par exemple :

$$g = \text{unitsystem}('SI') + 32.2 * \text{unitsystem}('Eng')$$

g sera égal à 1 si l'utilisateur a choisi précédemment le système d'unités International ou la valeur 32.2 si l'utilisateur a choisi le système d'unités anglais.

Fonctions et propriétés thermodynamiques

Le premier argument de toute fonction thermodynamique de EES est le nom de la substance. Les substances reconnues par EES sont les suivantes⁵:

Noms de substances utilisables dans les fonctions et propriétés thermodynamiques.			
Air	CH2	Oxygen	R123
AirH2O	H2O	Propane	R134a
Ammonia	Helium	R11	R141b
C2H6	Methane	R12	R500
C3H8	n-Butane	R13	R502
C4H10	N2	R14	R600a
CarbonDioxide	Nitrogen	R22	SO2
CH4	NO	R32	Steam (Water)
CO	NO2	R113	Steam_NBS
CO2	O2	R114	

Des substances supplémentaires (R113, R125, R143a, R404A, R407C, R410A, et R600a) sont fournies dans la version 32-bit de EES. Elles se trouvent dans les fichiers externes situés dans le répertoire USERLIB. Il est possible de rajouter vos propres propriétés en modifiant les fichiers d'extension .MHE (cf. annexe C). Le répertoire USERLIB contient également des routines externes contenant les données de propriétés thermodynamiques pour les mélanges suivant : lithium-bromide-eau (H_LIBR, T_LIBR, V_LIBR, Q_LIBR, P_LIBR, X_LIBR), ammoniac-eau (NH3H2O), et la chaleur spécifique, l'enthalpy et l'entropie pour les centaines de substances supplémentaires référencées dans la table de JANAF (JANAF). La commande 'Function Info' du menu 'Option', vous permet de visualiser la documentation de chacune de ces routines. (Cliquer sur le bouton 'external routine' pour sélectionner une routine de la liste, cliquer ensuite sur le bouton Info pour visualiser la documentation).

Noter que dans la liste mentionnée ci-dessous, certaines substances peuvent apparaître sous deux formes différentes : N2 et Azote, CO2 et CarbonDioxide, H2O et Vapeur (ou Eau). En fait, quand la notation de symbole chimique (N2, CO2, CH4, etc.) est utilisée, cela veut dire que la substance est modélisée comme un gaz idéal et les valeurs de l'enthalpy et de son entropie sont basées sur les références de table de JANAF (la référence de table de JANAF pour enthalpy est basée sur les éléments qui ont une valeur d'enthalpy pour une température comprise entre 0 et 298K (537R)). L'entropie de ces substances est basée sur le Troisième Principe de la thermodynamique. Quand le nom de substance est explicité (Vapeur (ou Eau), Azote, R12, CarbonDioxide, Méthane, etc.) la substance est modélisée comme fluide disposant

⁵ Votre version de EES peut avoir des liquides supplémentaires. Les données de propriété peuvent être ajoutées comme décrit dans l'annexe D.

de phase de subcooled, saturé, et superheated. Seul l'Air et AirH2O font exception à ses règles (toutes deux étant modélisées comme gaz idéal).

Des informations supplémentaires peuvent être ajoutées par l'utilisateur pour les quelques 150 types de fluide (cf. Annexe C).

Les mots réservés *Eau* et *Vapeur* sont traités identiquement. Ils permettent l'accès aux fonctions de propriétés de l'eau. Ces fonctions sont approximatives, elles sont basées sur des corrélations empiriques développées pour accroître la vitesse de calcul.

(En première approximation, le fluide est considéré comme incompressible dans la phase subcooled; pour des pressions supérieures à 350 atm ou pour des états proches du point critique, la précision est très moyenne). Le mot réservé *Steam_NBS* utilise les propriétés de corrélations publiées par Harr, Gallagher et Kell (Hemisphere, 1984). Celles-ci sont extrêmement précises dans n'importe quelle condition mais nécessitent un temps de calcul beaucoup plus long.

La plupart des fonctions thermodynamiques peuvent prendre des séries alternatives d'arguments. Par exemple, la fonction **enthalpy**, pour la vapeur peut être utilisée avec les arguments de température ou de pression; l'autre alternative est d'utiliser l'entropie et la qualité comme arguments. En général, des arguments indépendants peuvent être utilisés dans les fonctions thermodynamiques. Les fonctions de transport (**conductivité** et **viscosité**) nécessitent la température comme argument (pour les gaz idéaux) ou la température et la pression (pour les fluides).

Tout argument d'une fonction thermodynamique, excepté le nom de la substance, est identifié par une lettre (majuscule ou minuscule) suivi par un signe égal. Une valeur ou une expression algébrique doit suivre le signe égal. Les lettres reconnues en tant qu'arguments et leur signification sont les suivantes :

Property Indicators for Use in Thermophysical Functions	
B = Wetbulb Temperature	T = Temperature
D = Dewpoint Temperature	U = Specific Internal Energy
H = Specific Enthalpy	V = Specific Volume
P = Pressure	W = Humidity Ratio
R = Relative Humidity	X = Quality
S = Specific Entropy	

Les arguments doivent être séparés par des virgules. Le nom de la substance constitue le premier argument, le reste pouvant être saisi dans n'importe quel ordre. Le nom de la fonction ainsi que le nom de la substance apparaîtront suivant les paramètres fixés dans le menu 'Option'.

EES n'exige pas que tous les arguments de la fonction soient connus, par exemple :

$$h1 = \text{enthalpy}(\text{STEAM}, T=T1, P=P1)$$

retournera la valeur de h1 correspondant aux valeurs connus de T1 et P1. Cependant, si h1 est connu et que T1 est inconnu, cette même équation retournera la valeur de T1. Il est possible également de déterminer T1 de la manière suivante :

$$T1 = \text{temperature}(\text{STEAM}, h = h1, P=P1)$$

La dernière méthode est préférable dans des calculs itératifs pour s'affranchir de problèmes de convergence de solution.

Les fonctions et propriétés thermodynamiques incorporées dans EES sont énumérées ci-dessous dans l'ordre alphabétique. (les unités dépendent des choix effectués dans le système d'unité).

Conductivity [W/m-K, Btu/hr-ft-R] : retourne la conductivité thermique de la substance spécifiée. Pour un gaz supposé idéal, la fonction de conductivité prend la température comme argument unique en plus du nom de substance. Pour un fluide, la température et la pression doivent être saisies en arguments. La vapeur, l'eau et "Steam_NBS" acceptent le volume spécifique comme alternative à la pression. Pour AIRH2O (l'air humide), la température, la pression, et la proportion d'humidité (ou l'humidité relative) doivent être fournies.

Exemples: voir english version

Density [kg/m³, le kgmole/m³, lb/ft³, le lbmole/ft³] : retourne la densité d'une substance spécifiée. Deux arguments sont exigés pour toutes substances pures, trois pour l'air humide.

Exemple: voir english version

DewPoint °[F, °C, R, K] : retourne la température de pour les mélanges de gaz d'air-eau. Cette fonction peut être utilisée seulement avec AIRH2O comme le nom de substance. Trois arguments suivent le nom de substance dans n'importe quel ordre: La température, la pression totale, et l'humidité relative (ou la température de proportion d'humidité ou wetbulb).

Exemple: D1 = dewpoint(AIRH2O, T = 70, P=14.7, w=0.010)

D2 = dewpoint(AIRH2O, T = 70, P=14.7, R=0.5)

D3 = dewpoint(AIRH2O, T = 70, P=14.7, B = 50)

Enthalpy [le kJ/kg, le kJ/kgmole, le Btu/lbmole de Btu/livre] : retourne l'enthalpy spécifique d'une substance. La forme exacte de la fonction de enthalpy dépend de la substance et de la variable indépendante choisie. Les gaz qui obéissent aux lois des gaz idéaux, tel l'air,

exigent un seul argument, (la température ou l'énergie interne), en plus du nom de substance. Pour un fluide, (ex. VAPEUR et CARBONDIOXIDE), la fonction nécessite deux variables indépendantes (pour AIRH2O, trois arguments sont exigés).

Exemple: voir english version

Entropy [kJ/kg-K, kJ/kgmole-K, Btu/le livre-R, Btu/lbmole-R] : retourne l'entropie spécifique d'une substance. Pour toute substance pure, la fonction entropie exige toujours deux arguments, en plus du nom de substance. Pour AIRH2O, trois arguments sont exigés.

Exemple: $s1 = \text{entropy}(\text{O2}, T=400, P=100)$

$s2 = \text{entropy}(\text{AIRH2O}, T=70, P=14.7, R=0.50)$

MolarMass : retourne la masse molaire de l'argument (liquide).

Exemple: $M_{\text{CO2}} = \text{MolarMass}(\text{CarbonDioxide})$

Pression [kPa, la barre, psia, atm] : retourne la pression d'une substance spécifiée. La fonction pression exige toujours, le nom de substance suivi de deux arguments, (chaque argument doit être séparé par des points virgule). La fonction de pression ne fonctionne pas pour AIRH2O; cependant, une pression inconnue peut être déterminée par l'utilisation de n'importe quelle fonction applicable à l'air humide et qui prend la pression comme un argument.

Exemple: $u1 = \text{intEnergy}(\text{AIR}, T=300)$

$u2 = \text{intEnergy}(\text{STEAM}, T=1320, P=300)$

$u3 = \text{intEnergy}(\text{AIRH2O}, T=70, P=14.7, R=0.50)$

P_Crit [kPa, la barre, psia, atm] : retourne la pression critique du liquide spécifié. Le liquide peut être un fluide ou une variable de type chaîne de caractères. L'information de propriété critique n'est pas disponible pour les gaz considérés idéaux.

Exemple: $P_c = P_{\text{Crit}}(\text{R134a})$ "retourne la pression critique du réfrigérant : R134a"

Qualité [paramètres] : retourne la qualité (fraction de masse de vapeur) pour les substances modélisées comme vrais liquides tel que l'EAU et R12. Deux arguments indépendants sont exigés. La température et la pression ne sont pas indépendantes pour les états saturés. Si l'état de la substance est trouvé être subcooled, la fonction qualité retourne -100. S'il est superheated, la valeur 100 est retournée.

Exemple: $x1 = \text{quality}(\text{R12}, h = 50, T = 80)$

Relhum [paramètres] : retourne l'humidité relative pour les mélanges de gaz d'air-eau. Il y a trois arguments à cette fonction, en plus du nom de substance, AIRH2O. Les trois arguments sont des températures, la pression totale et n'importe laquelle des deux variables indépendantes restantes tel que la température, wetbulb, enthalpy, le point de dew, ou la proportion d'humidité.

Exemple: $R1 = \text{relhum}(\text{AIRH2O}, T = 70, P=14.7, w=0.01)$

R2 = relhum(AIRH2O, T = 70, P=14.7, h = 25)

R3 = relhum(AIRH2O, T = 70, P=14.7, B = 55)

Speheat [kJ/kg-K, kJ/kgmole-K, Btu/lb-R, Btu/lbmole-R] retourne la chaleur spécifique à pression constante d'une substance. Pour les substances qui obéissent à la loi des gaz parfaits, la température doit être fournie en plus du nom de la substance. La température et la pression doivent être fournies pour les fluides réels. La chaleur spécifique d'un liquide ou d'un gaz est retournée suivant les arguments fournis.

Exemple: Cp1 = speheat(AIR, T=350)

Cp2 = speheat (AMMONIA, T=100, P=30)

Temperature °[C, K, °F, R] : retourne la température de la substance. La forme exacte de la fonction dépend de la substance et l'argument choisi. Les substances qui sont supposées obéir à la loi des gaz idéaux, tel que l'air, nécessitent un ou deux arguments. Pour les fluides, tel que la vapeur, deux arguments sont indispensables.

Exemple: T1 = temperature(AIR, h=300)

T2 = temperature(AIR, s=1.75, P=100)

T_Crit °[C, K, ° F, R] : retourne la température critique du liquide spécifié. L'information de propriété critique n'est pas disponible pour un gaz suppose ideal.

Exemple: Tc=T_Crit(R134a)

Volume [m³/le kg, m³/kgmole, ft³/le livre, ft³/Lbmole] : retourne le volume spécifique d'une substance. Deux arguments sont exigés pour toute substance pure; trois pour l'air humide.

Exemple: v1 = **Volume**(AIR, T=300, P=100)

v2 = **Volume**(Steam, h=850, P=400)

v3 = **Volume**(AirH2O, T=70, R=0.5, P=14,7)

V_Crit [m³/le kg, m³/kgmole, ft³/le livre, ft³/Lbmole] retourne le volume critique du liquide spécifié. L'information de propriété critique n'est pas disponible pour un gaz idéal.

Exemple: vc=v_Crit(R134a) "retourne le volume critique de R134a"

Wetbulb °[C, K, °F, R] retourne la température de wetbulb pour les mélanges de gaz d'air-eau. Cette fonction est applicable seulement à la substance AIRH2O. Trois arguments sont nécessaires en plus du nom de la substance. Les trois arguments sont la température (ou enthalpy), la pression totale, et l'humidité relative (ou la proportion d'humidité ou dewpoint).

Exemple: B1 = wetbulb(AIRH2O, T = 70, P=14.7, w=0.01)

B2 = wetbulb(AIRH2O, h = 25, P=14.7, w=0.01)

B3 = wetbulb (AIRH2O, h = 25, P=14.7, D = 30)

Viscosity [N-sec/m², livre_m/Le ft-hr] : retourne la viscosité dynamique de la substance spécifiée. Pour un gaz idéal, la fonction viscosité prend la température comme son

argument unique en plus du nom de la substance. La température et la pression sont exigées pour les liquides. La VAPEUR et STEAM_NBS acceptent la température et le volume spécifique. Pour AIRH2O, l'argument est la proportion d' humidité ou l'humidité relative.

Exemple: v1 = la viscosity (AIR, T = 300)

v2 = viscosity(R12, T = 40, P = 30)

v3 = la viscosity (STEAM_NBS, T = 100, v=0.335)

v4 = viscosity(AIRH2O, T = 80, P=14.7, R=0.5)

Utilisation des fichiers et tables 'Lookup'

Un fichier 'Lookup' est une table de données à deux dimensions comportant un numéro de ligne et un numéro de colonne. Un fichier 'Lookup' est en quelque sorte une base de données dans laquelle il est possible de stocker des rapports fonctionnels avec des données tabulaires. Ces rapports peuvent être utilisés dans la solution d'équations. Les fichiers 'Lookup' peuvent être sauvegardés. Comme alternative, un seul fichier appelé 'Lookup Table' peut être utilisé dans la fenêtre 'Lookup Table'. Les six commandes afférentes à la fenêtre 'Lookup Table' sont décrites ci-dessous :

New Lookup : crée une nouvelle table 'Lookup' du nombre de lignes et de colonnes spécifiées par l'utilisateur avec un numéro spécifié de rangs et les colonnes dans la fenêtre 'Lookup table'. Si une table existe déjà, celle-ci sera remplacée par la nouvelle.

Open Lookup : lit un fichier du disque contenant un fichier 'Lookup' sur le disque. Si une table 'Lookup' est déjà ouverte, celle-ci sera remplacée par la nouvelle. Il existe deux types d'extension pour un fichier 'Lookup' : .LKT et .TXT. L'extension .LKT désigne un fichier binaire créé par la commande 'Save Lookup' de EES. L'extension .TXT désigne un fichier binaire créé par la commande 'Save Lookup' de EES ou par tout autre utilitaire.

Binary Lookup files (.LKT)

Un fichier 'Binary Lookup' conserve l'ensemble des informations (données, nom des colonnes, unités) qui apparaissent dans la fenêtre 'Lookup Table'. Ce type de fichier peut être généré à partir de la commande Save Lookup Table. Il requiert beaucoup moins d'espace disque qu'un fichier texte mais ne peut être visualisé dans une autre application.

ASCII Lookup files (.TXT)

Il existe plusieurs variantes pour les fichiers ASCII Lookup files.

Premier cas : la première ligne du fichier contient le nombre de lignes et de colonnes de la table. Les lignes suivantes contiennent les données relatives à chacune des colonnes séparées par un espace ou une tabulation. Noter qu'il n'est pas possible dans le cas présent de disposer du nom des colonnes, des unités des valeurs. EES assigne automatiquement les noms "COLUMN1", "COLUMN2", etc. pour chacune des colonnes. (noter que ces noms doivent être utilisés lors d'accès avec les fonctions **Interpolate**, **Differentiate**) Par ailleurs, EES formate les données automatiquement pour pouvoir les afficher dans la fenêtre Lookup Table.

L'exemple suivant montre les éléments requis pour une table de 5 lignes et de 3 colonnes :

5	3	
1	11	111
2	22	222
3	33	333
4	44	444
5	55	555

Si un nombre négatif est utilisé pour indiquer le nombre de lignes, EES déterminera automatiquement le nombre de lignes de la table. Par ailleurs, si le nombre de colonne est négatif, EES s'attend à trouver une forme spécifique (A3, F3 ou E4) suivie d'un espace ou d'une tabulation, puis du nom de chacune des colonnes et les unités utilisées (les unités doivent figurer entre crochets : []). L'exemple ci-dessous crée une table de 2 lignes et de 3 colonnes. Les colonnes sont mises en format à l'aide des indications de format E4, F0, et F3: les noms de colonnes sont ColA, ColB, et ColC.

2	-3	
E4	ColA	[Btu]
F0	ColB	
F3	Col	
1.23E-12	2	4.56
2.34E-11	4	7.89

Il est possible également d'effectuer des opérations sur ce type de table, à l'aide des fonctions **Interpolate**, **Differentiate**, **Lookup**, **LookupCol** et **LookupRow** documentées ci-dessous.

Save Lookup : permet de sauvegarder dans un fichier le contenu de la fenêtre Lookup Table. Il est possible de sauvegarder sous une forme binaire (extension .LKT) ou sous une forme ASCII (extension .TXT). Noter que la table est sauvegardée automatiquement lors de la sauvegarde du document EES actif.

Insert/Delete Rows : permet d'ajouter ou de supprimer une ligne de la table.

Insert/Delete Cols : permet d'ajouter ou de supprimer une colonne de la table.

Delete Lookup : permet de supprimer la table active.

Les données de la table Lookup peuvent être manipulées avec les fonctions **Interpolate**, **Differentiate**, **Lookup**, **LookupCol** et **LookupRow**. Ces fonctions opèrent soit directement sur les données de la fenêtre Lookup table, soit sur un fichier binaire ou texte du disque. Enfin, il est possible de fournir un nom de fichier en tant qu'argument d'une de ces fonctions. Le nom du fichier peut être une variable de type chaîne de caractères ou une constante chaîne de caractères (voir chapitre 7 pour de plus amples informations sur les chaînes de caractères). L'extension de nom de fichier peut être soit : .LKT (fichier binaire) soit .TXT (fichier ASCII). Si l'extension du nom de fichier n'est pas fournie, EES supposera que le fichier est de type binaire (EES ajoutera automatiquement l'extension .LKT).

Differentiate (*'Nom de fichier'*, 'ColName1', 'ColName2', ColName2=Value) : retourne la dérivée des données de deux colonnes ('ColName1' et 'ColName2') de la table déterminée à partir d'une interpolation cubique. Ces données peuvent être situées dans la table active, dans un fichier ou dans un table paramétrique.

'Nom de fichier' est optionnel. Dans le cas où un nom de fichier est donné, celui-ci doit correspondre à un fichier binaire ou ASCII précédemment généré à l'aide de la commande Save Lookup. Noter que 'Nom de fichier' peut être une variable de type chaîne de caractères, ou une constante chaîne de caractères. Si l'argument 'nom de fichier' n'est pas renseigné, EES effectuera la fonction 'differentiate' à partir de la table active.

ColName1 et ColName2 sont les noms d'en-tête de colonnes. Les apostrophes qui entourent ces noms d'en-tête de colonnes sont facultatives. Il est possible de passer une chaîne de caractères en tant qu'argument de la fonction. Le paramètre final est de la forme ColName2=Value où le texte à gauche du signe égal est le nom d'une des colonnes mentionnées précédemment. 'Value' doit être une expression numérique. EES retournera une expression de la dérivé : $d(\text{ColName1})/d(\text{ColName2})$ au point 'Value'.

La fonction **Differentiate** fonctionne également sur les données d'une table paramétrique si le 'Nom de fichier' est remplacé par 'Parametric'. Dans ce cas, les valeurs dans la table de Parametric doivent déjà exister. Noter que la fonction Differentiate ne peut utiliser les valeurs qui n'ont pas encore été calculées à l'aide de la commande 'Solve Table'

Exemples: $dXdY = \text{Differentiate}('X', 'Y', Y=2.34)$ {retourne dX/dY pour une valeur de $Y=2.34$ en utilisant les données de la table existante }
 $Y = \text{Differentiate}('C:\text{myFile}', T, X, T=100)$ { retourne dT/dX pour une valeur de $T = 100$ en utilisant les données du fichier myfile.lkt }

Interpolate (*'Nom de fichier'*, 'ColName1', 'ColName2', ColName2=Value) : retourne une valeur interpolée ou extrapolée à partir des données de la table active, d'un fichier ou d'une table paramétrique en utilisant une interpolation cubique. *'Nom de fichier'* est optionnel. Dans le cas où un nom de fichier est donné, celui-ci doit correspondre à un fichier binaire ou ASCII précédemment généré à l'aide de la commande 'Save Lookup'. Noter que 'Nom de fichier' peut être une variable de type chaîne de caractères, ou une constante chaîne de caractères. Si l'argument 'nom de fichier' n'est pas renseigné, EES effectuera la fonction

'interpolate' à partir de la table active. ColName1 et ColName2 sont les noms d'en-tête de colonnes. Les apostrophes qui entourent ces noms d'en-tête de colonnes sont facultatives. Il est possible de passer une chaîne de caractères en tant qu'argument de la fonction. Le paramètre final est de la forme ColName2=Value où le texte à gauche du signe égal est le nom d'une des colonnes mentionnées précédemment. 'Value' est une expression algébrique ou numérique. EES retournera la valeur interpolée à partir des données de la colonne 1 correspondant à la valeur spécifique de la colonne 2. Si la valeur de colonne 1 est donnée à la place, EES retournera la valeur interpolée de la colonne 2.

La fonction **Interpolate** fonctionne également sur les données d'une table paramétrique si le 'Nom de fichier' est remplacé par 'Parametric'. Dans ce cas, les valeurs dans la table de Parametric doivent déjà exister. Noter que la fonction Interpolate ne peut utiliser les valeurs qui n'ont pas encore été calculées à l'aide de la commande 'Solve Table'.

Exemples: Z = interpolate('Col1', 'Col2', Col1=2.3) {retourne une valeur de la colonne Col2 de la table qui correspond à une valeur de la Col1 égale à 2.3 en utilisant l'interpolation cubique.}

X = Interpolate(C:\myData,X,Y,Y=4.5) {retourne une valeur de la colonne X du fichier myData.LKT, correspondant à une valeur de la colonne Y égale à 4.5 en utilisant l'interpolation cubique }

interpolate1 (Le "nom de fichier", 'ColName1', 'ColName2', ColName2=Value) se comporte comme la fonction **interpolate** mais utilise l'interpolation linéaire.

interpolate2 (Le "nom de fichier", 'ColName1', 'ColName2', ColName2=Value) se comporte comme la fonction **interpolate** mais utilise l'interpolation quadratique.

Lookup ('Nom de fichier', ligne, colonne) : retourne la valeur d'une table 'Lookup' ou d'un fichier 'Lookup' correspondant au numéro de ligne et de colonne spécifiées en arguments. L'argument "colonne" peut être une valeur numérique (ou expression) ou le nom de celle-ci en utilisant une constante de chaîne de caractères (ex: 'ma colonne') ou une variable chaîne de caractères. Un ancien format pour lequel le nom de la colonne est précédé du caractère # est encore accepté. L'argument 'nom de fichier' est optionnel. Si cet argument est renseigné, EES vérifiera l'existence de ce fichier et le chargera en mémoire. Les arguments 'ligne' et 'colonne' ne sont pas nécessairement des nombres entiers. En effet, la valeur retournée dans le cas de nombres décimaux, sera l'interpolation entre les valeurs des lignes et colonnes (ex: Lookup(2.5;3) retournera la valeur médiane des valeurs situées entre la ligne 2 et la ligne 3 pour la colonne 3). Dans le cas où le numéro de ligne ou de colonne est inférieur à 1, la valeur de la première ligne ou colonne sera retournée. Il en est de même pour une valeur supérieure au nombre total de lignes ou de colonnes : la fonction 'Lookup' retournera la valeur de la dernière ligne ou de la dernière colonne. Cette fonction peut s'utiliser avec les fonctions 'LookupRow' ou 'LookupCol' pour donner des valeurs

interpolées. Cependant cette fonction est la plus adaptée pour effectuer ce genre d'opération.

Exemples:

X=Lookup(1,2) { fixe X à la valeur située à ligne 1 et à la colonne 2 d'une table 'Lookup'. }

X=Lookup(1,'macolonne') { fixe X à la valeur située à ligne 1 et à la colonne nommée macolonne d'une table 'Lookup'. }

X = Lookup(C: \abc\CopperK,R,'T') { fixe X à la valeur située à ligne R et de la colonne nommée T du fichier CopperK situe dans le répertoire C:\abc }

LookupCol ('Nom de fichier', Le Rang, la Value) : cette fonction utilise les données de la ligne spécifiée dans une table ou un fichier de type 'Lookup' pour déterminer la colonne qui contient la valeur fournie comme deuxième argument : 'Value'. Le but de la fonction **LookupCol** est de pouvoir rechercher des données dans une table ou dans un fichier.

Exemples:

C=LookupCol(2, 100) {C prend la valeur du numéro de colonne qui a pour valeur 100 à la ligne 2}

C = LookupCol('C:\abc\CopperK', R, X) {C prend la valeur du nom de colonne qui a pour valeur X à la ligne R du fichier C:\abc\CopperK}

LookupRow ('Nom de fichier', La colonne, la Valeur) : cette fonction utilise les données de la colonne spécifiée dans une table ou un fichier de type 'Lookup' pour déterminer la ligne qui contient la valeur fournie comme deuxième argument : 'Value'. Le but de la fonction **LookupRow** est de pouvoir rechercher des données dans une table ou dans un fichier.

Noter que les fonctions LookupCol et LookupRow acceptent toutes deux, des variables ou des constantes de type chaîne de caractères en tant que deuxième argument.

Exemples:

R=LookupRow(2, 100) {C prend la valeur du numéro de ligne qui a pour valeur 100 à la colonne 2}

R = LookupRow(C: \abc\ CopperK', C, X) {C prend la valeur du nom de ligne qui à pour valeur X à la colonne R du fichier C:\abc\CopperK}

Quand une nouvelle table est créée, les colonnes sont nommées implicitement **Column1**, **Column2**, Etc.

Il est possible, via le presse-papiers, de copier ou de coller des données d'une table de type 'Lookup'. Ainsi vous pouvez coller ces données dans une table paramétrique, une feuille de tableur, etc. Pour ce faire, sélectionner les cellules à copier à l'aide du bouton droit de la souris

en maintenant le bouton shift appuyé. Les cellules sélectionnées apparaissent en inverse vidéo. Ensuite, utiliser la commande 'Copy' pour copier les données dans le presse-papiers. (vous pouvez utiliser la commande 'Select all' du menu 'Edit' pour sélectionner l'ensemble des cellules).

De même, vous pouvez coller des données dans une table 'Lookup'. (au préalable, vous devez avoir copier un jeu de données). Positionner le curseur dans la cellule à partir de laquelle les données seront copiées.

Fonctions, Procédures et Modules

La plupart des langages de programmation de haut niveau permettent à leurs utilisateurs de développer leurs propres routines. EES possède également cette capacité et offre plusieurs façons de développer ses propres sous-programmes. Ainsi EES vous permet d'écrire vos propres fonctions, procédures ou modules. Une fonction est une 'portion de code' acceptant une ou plusieurs données en entrée et retourne un résultat unique. Une procédure peut retourner un ou plusieurs résultat(s).

Un module est similaire à une procédure car il peut également retourner un ou plusieurs résultat(s). Cependant un module n'emploie que des égalités à la place des déclarations d'assignation utilisées dans une procédure. EES peut accéder indifféremment à des sous-programmes écrits dans son propre langage ou à des sous-programmes externes écrits en PASCAL, C, C++, FORTRAN ou tout autre langage compilé. Le développement de routines externes est décrit dans le chapitre 6. Noter que les sous-programmes (internes et externes) doivent être localisés dans le répertoire USERLIB\, à partir duquel ils seront automatiquement chargés en mémoire au démarrage de EES.

Utiliser des sous-programmes offre de nombreux avantages. Tout d'abord, cela permet de segmenter un problème complexe en plusieurs ensembles, améliorant ainsi la lisibilité de votre code, facilitant sa compréhension. Par ailleurs, il est possible de stocker vos sous-programmes dans un fichier de librairie pour ensuite les réutiliser dans vos autres programmes. Enfin, les fonctions et procédures de EES permettent d'écrire des structures conditionnelles, des boucles : *if then else*, *repeat until* et *goto*. Les déclarations qui figurent dans les fonctions et procédures diffèrent de celles utilisées dans le corps (main) d'un programme EES. En effet, le corps du programme utilise des opérateurs d'égalité plutôt que d'assignation. Les fonctions et procédures sont exécutées suivant leur ordre d'apparition. Les modules utilisent des déclarations d'égalité comme le corps principal d'un programme EES. EES réorganise ces égalités pour améliorer l'efficacité des calculs. La combinaison de ces types de déclarations offre un grand niveau de flexibilité dans la formulation de vos problèmes dans EES.

Il existe plusieurs manières d'accéder à vos sous-programmes. La commande **Merge** du menu **File** peut être utilisée pour importer des sous programmes EES à partir d'un autre fichier EES. Il est possible également de sauvegarder vos routines dans des fichiers de librairie. Ces fichiers peuvent contenir plusieurs fonctions/procédures et/ou modules (un fichier de librairie possède l'extension .LIB). Les fichiers stockés dans le répertoire USERLIB\ sont automatiquement chargés en mémoire et de manière transparente pour l'utilisateur lors du démarrage de EES. La commande **Load Library** du menu **File** permet également de charger des fichiers de librairie.

Enfin, la directive \$INCLUDE est un moyen de charger vos routines. Les fonctions, procédures et modules contenus dans les fichiers de librairie agissent comme toute fonction interne de EES. Il est possible d'associer des fichiers d'aide à vos sous-programmes et des les invoquer à tout moment. Ce chapitre a pour objet de décrire les méthodes de développement des sous-programmes.

Les fonctions EES

EES permet à ses utilisateurs d'écrire des fonctions directement dans la fenêtre d'Equations. La syntaxe utilisée est similaire à celle employée en PASCAL. Les règles d'utilisation sont les suivantes :

1. Les fonctions utilisateurs doivent figurer au tout début de la fenêtre d'Equations avant tout module, équations dans le corps du programme principal de EES.
2. Toute fonction commence par l'instruction FUNCTION. Le nom de la fonction, ses arguments sont placés sur la même ligne et compris entre des parenthèses et séparés par des points virgules.
3. Une fonction se termine par l'instruction END.
4. Les équations qui figurent dans les fonctions et procédures sont fondamentalement différentes de celles rédigées dans la fenêtre d'Equations. Les équations utilisées dans les fonctions et procédures sont plus correctement appelées déclaration d'assignation, comme en FORTRAN ou PASCAL. Une déclaration d'assignation fixe la valeur d'une variable située à gauche de l'opérateur par la valeur située à gauche de celui-ci. $X:=X+1$ est une assignation valide mais ne peut être considérée comme une égalité similaire à toute équation EES rédigée dans le corps principale de EES. L'opérateur := (plutôt que =) est utilisé pour identifier une assignation. Cependant EES acceptera le signe = comme symbole d'assignation si la case **Allow = in Functions/Procedures** est cochée dans la boîte de dialogue **Display Options** accessible via la commande 'Preference' du menu **Options**.
5. EES lit les instructions dans une fonction ou une procédure suivant l'ordre dans lequel elles apparaissent. Ceci étant, les instructions *If Then Else*, *Repeat Until* et *goto* peuvent être utilisées pour modifier le séquençement des calculs. Ces instructions sont décrites plus loin dans ce chapitre.
6. Les fonctions sont appelées en utilisant leur nom dans une équation. Les arguments doivent suivre ce nom et doivent être compris entre parenthèses. Une fonction doit être appelée avec le même nombre d'arguments qui apparaît après l'instruction FUNCTION.
7. Les équations qui figurent dans les fonctions, peuvent être également appelées à partir de n'importe qu'elle autre fonction interne de EES. De plus, les fonctions peuvent appeler toute autre fonction utilisateur ou procédure chargée en mémoire en tant que fichier de librairie. Les fonctions récursives ne sont pas autorisées. Enfin, une fonction ne peut faire appel à un module.

8. Toutes les variables utilisées dans le corps principal d'un fichier EES sont locales exceptées celles définies à l'aide de la directive \$COMMON.
9. Les fonctions travaillent uniquement en mode réel même si vous avez demandé à EES de travailler en mode complexe.

Les fonctions peuvent être utilisées pour implémenter des relations analytiques entre plusieurs variables. Par exemple, la disponibilité spécifique d'un flux de vapeur communément appelée ψ , est donnée par la relation suivante :

$$\psi = (h - h_0) - T_0 (s - s_0) + V^2/2 + g z$$

où

h et s sont respectivement l'enthalpie et l'entropie

h_0 and s_0 sont respectivement l'enthalpie et l'entropie aux conditions extrêmes, T_0 et P_0

V la vitesse

g l'accélération gravitationnelle

z est la hauteur par rapport à un point 0

Une fois fixées les valeurs de la température et de la pression T_0 et P_0 , h_0 et s_0 sont constantes. Une fonction utilisateur permettant de calculer la disponibilité spécifique d'un flux de vapeur avec $T_0=530$ R et $P_0=1$ atm, peut s'implémenter comme ci-dessous. La référence $\psi(T_1, P_1, V_1, Z_1)$ à partir d'une équation retournera la valeur de la disponibilité spécifique d'un flux de vapeur en fonction des conditions extrêmes. (en Btu/lbm)

FUNCTION $\psi(T, P, V, Z)$

$T_0 := 530$ "R dead state temperature"

$h_0 := 38.05$ "Btu/lbm specific enthalpy at dead state conditions"

$s_0 := 0.0745$ "Btu/lbm-R specific entropy at dead state conditions"

$h := \text{enthalpy}(\text{STEAM}, T=T, P=P)$

$s := \text{entropy}(\text{STEAM}, T=T, P=P)$

$g = 32.17$ "ft/s² gravitational acceleration"

$\psi := (h-h_0) - T_0 * (s - s_0) + (V^2 / 2 + g * Z) * \text{Convert}(\text{ft}^2/\text{s}^2, \text{Btu}/\text{lbm})$

END

Les fonctions peuvent être utilisées également pour changer le nom d'une fonction EES ou pour éventuellement raccourcir la liste d'arguments. Par exemple, la fonction utilisateur suivante change le nom de la fonction **humrat** en **w**, évite à l'utilisateur de passer la substance AIRH2O en argument et enfin fixe la pression totale à 100 kPa dans chaque cas.

FUNCTION $w(T, RH)$

$w := \text{humrat}(\text{AIRH2O}, T=T, P=100, R=RH);$

END

Ces deux exemples utilisent des fonctions incluses de EES et retournent une valeur dépendante du système d'unités fixé. Utiliser la fonction UnitSystem (Chapitre 4) et les instructions IF THEN ELSE pour faire fonctionner correctement vos fonctions suivant le système d'unités courant.

Les procédures

Les procédures EES sont quasi similaires aux fonctions EES à la différence près que les procédures permettent de retourner plusieurs valeurs de sortie.

```
PROCEDURE test(A,B,C : X,Y)
...
...
X :=...
Y :=...
END
```

Les procédures doivent être placées en haut de la fenêtre d'Equations avant tout module, fonction ou équation. Le nom de la procédure : TEST dans cet exemple est un nom de variable valide pour EES. Dans l'exemple ci-dessus, A, B, et C sont des entrées et X et Y sont des sorties. Chaque procédure doit avoir au moins une variable d'entrée et une variable de sortie. Par ailleurs chaque variable de sortie doit être placée à droite de l'opérateur d'assignation. L'instruction END termine la procédure.

Pour utiliser une procédure, placer la commande CALL devant le nom de la procédure à invoquer n'importe où dans la fenêtre d'Equations :

```
...
CALL test(1,2,3 : X,Y)
...
```

Le nombre d'arguments invoqué via la commande CALL doit correspondre exactement aux nombres d'arguments définis lors de la création de la procédure. Ces arguments peuvent être des constantes, des chaînes de caractères des valeurs numériques ou des expressions algébriques. D'autres types d'arguments peuvent être passés en paramètres entre la procédure et le corps principal via la directive \$COMMON. Les procédures peuvent faire appel à des fonctions, procédures définies précédemment. Noter qu'une procédure ne peut appeler un module.

Les équations à l'intérieur d'une procédure diffèrent de celles du corps principal de la fenêtre d'Equations ou d'un module. Tout d'abord, toutes les variables, à l'exception des entrées et sorties sont locales à la procédure. Ensuite les équations sont considérées comme des déclarations d'assignation plutôt que des égalités. (Pour faire la distinction l'opérateur d'assignation suivant est utilisé : (:=)). Enfin les commandes, *if then else*, *repeat until* et *goto* peuvent être utilisées.

Les équations implicites ne peuvent pas être résolues directement dans une procédure ou une fonction à contrario d'un module ou dans le corps principal d'un programme EES.

En utilisant les instructions *If Then Else*, *Repeat Until* et *goto*, il est possible de créer des structures itératives. Cependant, il est possible de forcer EES à résoudre des équations implicites à l'intérieur d'une procédure. Par exemple, considérons les deux équations non linéaires suivantes :

$$X^3 + Y^2 = 66$$

$$X/Y = 1.23456$$

Pour résoudre X et Y au sein d'une procédure, vous devez soustraire les deux membres de chaque équation est de les égaliser respectivement à R1 et R2 qui sont nos variables de sortie considérées comme résiduelles. A présent, utiliser EES pour résoudre X et Y telles que les valeurs résiduelles soient nulles :

```
PROCEDURE Solve(X,Y:R1,R2)
```

```
  R1:=X^3+Y^2-66
```

```
  R2:=X/Y-1.23456
```

```
END
```

```
CALL Solve(X,Y:0,0)  {X = 3.834, Y = 3.106 when executed}
```

Vous venez de voir qu'il possible de résoudre des équations implicites au sein d'une procédure, mais il convient de signaler que les modules sont beaucoup mieux adaptés pour ce type d'opération.

Les procédures offrent un nombre important d'avantages pour les utilisateurs de EES. Les procédures fréquemment utilisées peuvent être sauvegardées dans des fichiers et fusionnées (commande Merge) dans la fenêtre d'Equations d'autres fichiers EES. Par ailleurs, il est possible de stocker vos procédures dans un fichier de librairie. Si celui-ci est placé dans le répertoire USERLIB\ il sera lu automatiquement par EES lors de son démarrage. (il est possible d'utiliser également la directive \$INCLUDE faisant référence à votre fichier).

EES supporte les procédures internes et externes quand celles-ci sont déjà compilées. Les procédures internes sont directement saisies dans la fenêtre d'Equations. Les procédures externes sont écrites et compilées à l'aide d'un langage de haut niveau tel C, PASCAL ou FORTRAN et peuvent être appelées par EES. L'instruction CALL est identique dans les deux cas de figures. (voir chapitre 6 pour de plus amples informations sur les fonctions compilées).

Les instructions If Then Else

Les fonctions et procédures de EES supportent plusieurs types de structures conditionnelles. Noter que celles-ci ne peuvent être utilisées dans les modules ou dans le corps principal d'un programme EES. La structure conditionnelle couramment employée est *If Then Else* (*Si Alors Sinon*). Cette structure peut s'écrire sur une ou plusieurs ligne(s).

If (Test conditionnel) *Then* Instruction 1 *Else* Instruction 2

Le champ 'Test conditionnel' produit un résultat booléen : *True* (vrai) ou *False* (faux). Le format est similaire à celui utilisé en PASCAL. Les opérateurs sont =, <, >, <=, >=, et <> (différent). Les parenthèses autour de ce champ sont optionnelles. Il est possible (voir Chapitre 7) d'utiliser des chaînes de caractères dans ce champ. Le mot clef *Then* et Instruction 1 doivent suivre impérativement le mot clef *If* 'Instruction 1' peut être une assignation ou une instruction de saut : *GoTo*. Le mot clef *Else* et 'Instruction 2' sont optionnels. Si la structure conditionnelle est écrite sur une seule et même ligne, elle ne peut comporter que 255 caractères au maximum. L'exemple suivant utilise une structure *If Then Else* pour déterminer le minimum de trois arguments au sein d'une fonction.

Function **MIN3**(x,y,z) { retourne la plus petite valeur des arguments }

If (x<y) *Then* m:=x *Else* m:=y

If (m>z) *Then* m:=z

MIN3:=m

End

Y = **MIN3**(5,4,6) { Y sera égal à 4 lors de l'utilisation de la fonction MIN3 }

Les opérateurs AND et OR (ET/OU) peuvent être utilisés dans le test conditionnel d'une structure *If Then Else*. A moins d'utiliser des parenthèses, EES lit les opérateurs logiques de la gauche vers la droite. Noter que les parenthèses (x>0) et (y<>3) sont nécessaires dans l'exemple suivant pour produire l'effet logique désiré.

If (x>y) *or* ((x<0) *and* (y<>3)) *Then* z:=x/y *Else* z:=x

Les structures If Then Else multi-lignes

Les structures If Then Else multi-lignes permettent d'exécuter des groupes d'instructions suivant certaines conditions.

```
If (Test conditionnel) Then  
    Instruction  
    Instruction  
    ...  
Else  
    Instruction  
    Instruction  
    ...  
EndIf
```

Les mots clefs *If* et *Then* doivent être sur la même ligne. Les parenthèses sont optionnelles. Les instructions qui doivent être effectuées si le Test conditionnel est vrai suivent le mot clef *Then*. Ces instructions peuvent contenir d'autres structures *If Then Else*. Le mot clef *Else* (ou *EndIf*) indique la fin du premier groupe d'instructions. Le mot clef *Else* doit être unique sur sa ligne et suivi de l'autre groupe d'instructions à exécuter si le test conditionnel est faux. Le mot clef *EndIf* indique la fin une structure *If Then Else* multi-lignes. L'exemple ci-dessous illustre ces propos. N'hésiter pas à utiliser des espaces pour améliorer la lisibilité de votre structure.

```
Function IFTest(X, Y)  
    If (X<Y) and (Y<>0) Then  
        A:=X/Y  
        B:=X*Y  
        If (X<0) Then           { imbrication }  
            A:=-A; B:=-B  
        EndIf  
    Else  
        A:=X*Y  
        B:=X/Y  
    EndIf  
    IFTest:=A+B  
End  
  
G=IFTest(-3,4) { G est fixé à 12.75 dans ce cas }
```

L'instruction GoTo

EES lit normalement les instructions d'une fonction, d'une procédure suivant l'ordre dans lesquelles elles apparaissent. Il est possible, néanmoins, de modifier le flux d'instruction à l'aide de l'instruction *Goto*. Le format est le suivant :

GoTo #

désigne un label numérique qui doit être un entier compris entre 1 et 30000. Un label précède une instruction et est séparé par le symbole (:). L'instruction *Goto* doit être combinée avec la structure *If Then Else* pour être utile. L'exemple suivant illustre cette instruction et permet de déterminer le factoriel d'un nombre.

Function **FACTORIAL**(N)

F:=1

i:=1

10: i:=i+1

F:=F*i

If (i<N) *Then GoTo* 10

FACTORIAL:=F

End

Y= **FACTORIAL**(5) { Y est égal à 120 }

La structure Repeat Until

Il est possible de créer des boucles à l'aide de la structure conditionnelle *If Then Else*. Néanmoins, la structure *Repeat Until* est beaucoup plus appropriée et commode pour réaliser des boucles. Cette structure a le format suivant. Noter que cette structure ne peut s'utiliser dans un module.

Repeat

Instruction

Instruction

...

Until (Test conditionnel)

Le champ 'Test conditionnel' produit un résultat booléen : *True* (vrai) ou *False* (faux). Le format est similaire à celui utilisé en PASCAL. Les opérateurs sont =, <, >, <=, >=, et <> (différent). Vous trouverez ci-dessous un exemple de calcul de factoriel utilisant la structure *Repeat Until* :

Function **Factorial**(N)

F:=1

Repeat

```

F:=F*N
N:=N-1;
Until (N=1)
Factorial:=F
End

Y= FACTORIAL(5) { Y est égal à 120 }

```

La procédure Error

Cette procédure permet à l'utilisateur de stopper le processus de calcul si la valeur fournie à une fonction ou une procédure dépasse une certaine limite. Le format de la procédure Error est le suivant :

Call Error('Message d'erreurs',X) or *Call Error*(X)

où 'Message d'erreurs' est une chaîne de caractères optionnelle comprise entre '' et X la valeur limite. Si la valeur X est dépassée, EES affichera le message contenu dans le champ compris entre '' si celui-ci est alimenté.

Les calculs ont du être arrêtés car la valeur XXX du paramètre est hors de la plage de valeurs tolérées.

La valeur de X fournie dans la procédure Error remplace XXX. Si un message d'erreur est fourni, EES affichera celui-ci en remplaçant la chaîne XXX par la valeur de X. Si une option de formatage tel F1 ou E4 est fournie comme dans l'exemple ci-dessous, la valeur de X sera affichée selon ces options (sinon, le mode normal est employé). La procédure ERROR est employée en grande partie avec la structure IF – THEN – ELSE.

Function bac(X,Y)

```

if (x<=0) then CALL ERROR('X doit être supérieur ou égal à 0. La valeur entrée était
XXXE4 !', X)

```

```

abc:=Y/X

```

```

end

```

```

g:=abc(-3,4)

```

Quand cette fonction est appelée, le message d'erreur suivant apparaît :

'X doit être supérieur ou égal à 0. La valeur entrée était -3.000E0 !

Les modules

Les modules peuvent être considérés comme des sous-programmes isolés pouvant être appelés à partir du corps principal d'un programme EES. Le format d'un module est quelque peu similaire à celui d'une procédure. Un module reçoit des paramètres et renvoi des valeurs calculées. Le format formel d'écriture d'un module est le suivant : Les variables d'entrée sont séparées par des points virgule et sont elles-mêmes séparées des variables de sortie par le symbole (:). Le nombre d'arguments fourni à gauche des deux-points correspond au nombre de degrés de liberté dans le module. Plus concrètement, un certain nombre d'arguments doit être passé au module pour que celui-ci contienne autant d'inconnues que d'équations.

```
MODULE Testme(A, B : X, Y)
```

Dans ce cas EES comprend qu'il y a deux entrées (A et B) et deux sorties (X et Y). Ceci étant, les modules utilisent des égalités plutôt que des assignations. Ainsi, dans la plupart des cas, l'ordre dans lequel on passe les arguments n'importe pas à condition bien sûr de respecter la logique de programmation du module. Par conséquent, le symbole (:) de séparation entre les variables d'entrée et de sortie est superflu et peut être remplacé par un point virgule. L'exemple suivant est équivalent à l'exemple précédent.

```
MODULE Testme(A, B, X, Y)
```

Un module est appelé à l'aide de l'instruction CALL, par exemple :

```
CALL Testme(77,1.5, X,Y) {Le module Testme est appelé}
```

Noter que le même symbole de séparation entre les variables d'entrée et de sortie, utilisé lors de la création du module doit être repris dans l'instruction CALL.

Quand EES rencontre une instruction CALL, il intègre, automatiquement et de manière transparente pour l'utilisateur, les équations du module dans le corps principal. Les étapes nécessaires pour ce processus sont les suivantes :

Tout d'abord, chaque variable du module (incluant les arguments) est renommée à l'aide d'un identifiant unique que seul EES reconnaît. Ensuite, EES ajoute une équation pour chaque entrée et sortie. Ces équations égalent les entrées et sorties par la valeur du paramètre remplacé par la valeur calculée dans le module. Enfin, toutes les équations du module avec leurs variables renommées sont intégrées dans le corps principal à l'endroit où se trouve l'instruction CALL. Si le module est de nouveau appelé, le processus est répété avec de nouveaux identifiants pour les variables. Le résultat final de ce processus est que EES copie toutes les équations d'un module dans le corps principal à chaque fois que l'instruction CALL est rencontrée. EES permet de traiter 6000 équations. Ainsi des problèmes complexes peuvent être formalisés. EES utilise sa technique de réorganisation efficace des équations pour optimiser le temps de calcul.

Par conséquent, les équations d'un module ne sont pas nécessairement appelées dans la même séquence. Vous pouvez contrôler la façon dont EES réorganise les équations dans la fenêtre Résiduelle. Les équations provenant de modules sont identifiées par le nom du module suivi du symbole (\) et le numéro d'index d'appel. Par exemple :

Turbine\2: $h_2=h_1+Q/m$

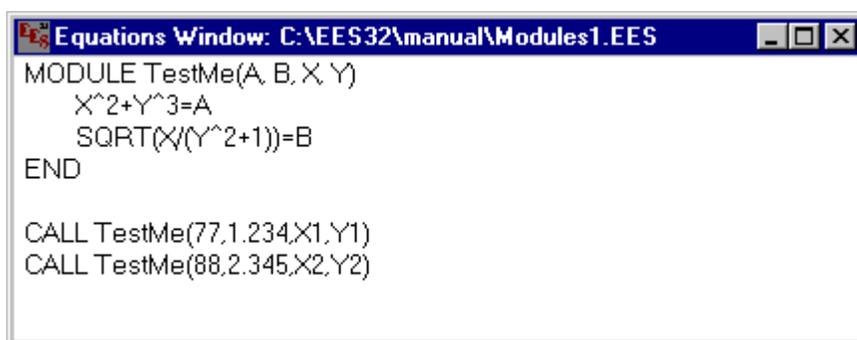
Indique que l'équation $h_2=h_1+Q/m$ provient du second appel du module Turbine.

Les variables locales d'un module ne sont normalement pas affichées dans la fenêtre de Solution. Vous pouvez cependant les voir en cochant la case 'Show function/procedure/module variables' dans l'onglet 'Options' de la commande 'Preferences'.

Les valeurs des variables sont normalement passées au module via la liste d'arguments. Néanmoins, la directive \$COMMON (Chapitre 7) peut être utilisée pour fixer la valeur de variables identifiées dans le corps principal.

Toutes les variables définies à l'intérieur d'un module ont les mêmes propriétés (valeur initiale, limites basse et haute, unités, option de formatage) que celles présentes dans le corps principal. Ces informations sont visibles par l'intermédiaire de la commande 'Variable Info'. Les variables locales sont toujours réelles même si EES travaille en mode complexe.

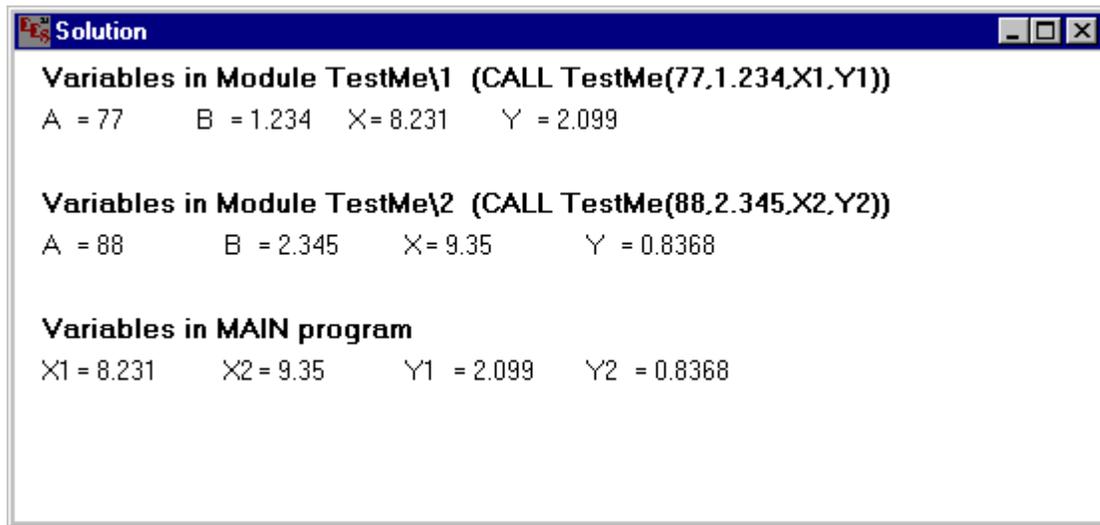
Un module se termine par l'instruction END. L'instruction CALL est utilisée pour appeler un module, comme pour les procédures. La différence majeure entre module et procédure, est qu'un module ne comprend que des égalités tandis que les procédures ne travaillent qu'avec des assignations. En conséquence, un module ne supporte pas les structures conditionnelles du type IF – THEN – ELSE. Par contre, un module peut fournir des solutions itératives pour des équations implicites. De plus, au sein d'un module, les entrées d'équations indépendantes sont réordonnées comme dans le corps principal de EES. Voici un exemple :



```
Equations Window: C:\EES32\manual\Modules1.EES
MODULE TestMe(A, B, X, Y)
  X^2+Y^3=A
  SQRT(X(Y^2+1))=B
END

CALL TestMe(77,1.234,X1,Y1)
CALL TestMe(88,2.345,X2,Y2)
```

Quand la commande 'Solve' est invoquée, ESS fusionnera automatiquement les équations du module TestMe, 2 fois dans le corps principal. La fenêtre de Solution qui apparaît est la suivante :



```
Solution
Variables in Module TestMe\1 (CALL TestMe(77,1.234,X1,Y1))
A = 77      B = 1.234   X= 8.231   Y = 2.099

Variables in Module TestMe\2 (CALL TestMe(88,2.345,X2,Y2))
A = 88      B = 2.345   X= 9.35    Y = 0.8368

Variables in MAIN program
X1 = 8.231  X2 = 9.35   Y1 = 2.099  Y2 = 0.8368
```

Les modules peuvent être sauvegardés dans un fichier de librairie comme les fonctions internes et les procédures. Un fichier librairie peut être automatiquement chargé lors du démarrage de EES à condition de se trouver dans le répertoire USERLIB. Des fichiers d'aide relatifs aux modules peuvent être incorporés. Ces fichiers doivent utiliser la même syntaxe que pour les procédures.

Les modules peuvent améliorer significativement le développement de vos programmes EES.

Les fichiers Librairie

EES offre la possibilité à des fichiers contenant une ou plusieurs fonctions, procédures ou modules d'être sauvés en tant que fichier de librairie. Ce type de fichier possède l'extension .LIB. Quand EES démarre, il charge automatiquement tous les fichiers librairie contenus dans le répertoire \USERLIB\. Les fichiers librairie peuvent être chargés manuellement par l'intermédiaire de la commande **Load Library** du menu **File** ou à l'aide de la directive \$INCLUDE. Les sous-programmes contenus dans la librairie ne sont pas affichés dans la fenêtre d'Equations comme les fonctions EES incluses.

Pour créer un fichier de librairie, entrer une ou plusieurs fonctions, procédures et/ou module dans la fenêtre d'Equations. Ensuite, compiler vos équations à l'aide des commandes **Check**, **Solve** ou **Solve Table**. Enfin, sauver votre fichier avec l'extension .LIB à l'aide de la commande **Save As**.

Des informations accessibles via la commande 'Function Info' à l'instar des fonctions déjà incluses dans EES peuvent être fournies avec un fichier de librairie. Le texte contenant ces informations est placé dans le fichier comme un commentaire (compris entre crochets). Le premier caractère suivant le premier crochet doit être le signe \$ suivi du nom de la fonction, de la procédure ou du module puis un retour chariot. Les lignes suivantes (chacune comprise entre crochets) seront affichées quand l'utilisateur appuiera sur le bouton 'Function Info' de la boîte de dialogue 'Info dialog window'

Le concept de fichier de librairie est une des grandes caractéristiques de EES. Il permet à un utilisateur de développer des programmes adaptés, réutilisables pour lui ou pour d'autres. L'exemple suivant utilise un fichier de la librairie pour ajouter une méthode numérique de résolution d'intégrales à EES. Cet algorithme appelé algorithme de Runge-Kutta permet de résoudre des équations différentielles de la forme suivante :

$$\frac{dY}{dX} = f(X,Y)$$

où $f(X,Y)$ est une fonction qui nécessite la variable dépendante Y et la variable indépendante X. Y doit avoir une valeur initiale connue, Y0, correspondante à la valeur initiale de X.

Cet algorithme a été implémenté sous le nom de RK4. RK4 requiert 4 paramètres : la valeur initiale de X (LowX), la valeur finale de X (HighX), l'intervalle de calcul (StepX), et la valeur de Y pour X=LowX (Y0). Cette fonction retourne la valeur de Y pour Y at X=HighX. La fonction RK4 appelle une autre fonction, fRK4(X,Y), pour obtenir la valeur de dY/dX pour les valeurs de X et Y. Cette fonction contient une équation que l'utilisateur peut remplacer suivant la fonction mathématique f à utiliser dans la fenêtre d'Equations. Les fonctions RK4 et fRK4 se trouvent dans le fichier RK4.LIB située dans le répertoire USERLIB\. Ce fichier contient les instructions ci-dessous. Noter comment fournir un fichier d'informations accessible à l'aide de

la commande 'Function Info' via la directive \$fRK4. Ces informations peuvent également provenir d'un fichier ASCII avec l'extension .HLP.

```
FUNCTION fRK4(X,Y)
{ $fRK4
fRK4 is a user-supplied function to evaluate dY/dX. This
function is used with the RK4 function to solve differential
equations with the Runge-Kutta method. Enter a fRK4(X,Y)
function in the Equations window to evaluate dY/dY for your
problem. See the RK4 function for additional information.}
fRK4:=(Y+X)^2
END
```

```
FUNCTION RK4(LowX,HighX,StepX,Y0)
{ $RK4
RK4 is a general purpose function which solves a first-order
differential equation of the form dY/dX=fRK4(X,Y) using the
Runge-Kutta 4th order algorithm. The RK4 function calls function
fRK4(X,Y) supplied by the user to evaluate dY/dX at specified values
of X and Y. The user must supply the fRK4 function.
```

RK4 requires four input parameters. LowX is the initial value of independent variable X. HighX is the final value of independent variable X and StepX is the step size. Y0 is the value of Y when X is equal to LowX. }

```

X := LowX
Y := Y0;
Tol := 0.1*StepX
10:
IF (X>HighX-Tol) THEN GOTO 20
k1 := fRK4(X,Y)*StepX
k2 := StepX*fRK4(X+0.5*StepX,Y+0.5*k1)
k3 := StepX*fRK4(X+0.5*StepX,Y+0.5*k2)
k4 := StepX*fRK4(X+StepX,Y+k3)
Y := Y+k1/6+(k2+k3)/3+k4/6
X := X+StepX
GOTO 10;
20:
RK4:=Y
END
```

Supposons que vous désirez évaluer l'équation $\int_0^2 X^2 dx$ utilisant la fonction RK4.

Dans la fenêtre d'Equation, rédiger la fonction fRK4 montrée ci-dessous pour évaluer l'opérande X^2 . Cette fonction prévaudra sur la fonction fRK4 existante dans le fichier RK4.LIB. Pour évaluer votre intégrale, taper les instructions suivantes dans la fenêtre d'Equations.

```
FUNCTION fRK4(X,Y)
```

```
fRK4:=X^2  
END  
V=RK4(0,2,0.1,0)
```

Une fois le processus de calcul terminé, EES affichera $V=2.667$ dans la fenêtre de Solution.

La directive \$COMMON

La directive \$COMMON est un moyen de faire passer des informations à partir du corps principal d'un programme EES à des fonctions internes, procédures, modules. \$COMMON permet également de faire passer des valeurs d'argument à vos fonctions, procédures, modules. Cette directive est basée sur le même concept que l'instruction FORTRAN COMMON. Elle diffère néanmoins car le flux d'information n'est pas bidirectionnel. Les valeurs de variables peuvent être passées à partir du corps principal aux fonctions et procédures mais une fonction ou une procédure ne peut assigner ou modifier ces valeurs.

La directive \$COMMON doit directement suivre les déclarations FUNCTION, PROCEDURE, ou MODULE. Cette directive doit être seule sur sa ligne. Les variables qui apparaissent dans \$COMMON doivent être séparées par des points virgule.

Exemple :

```
FUNCTION TESTCOMMON(X)
$COMMON B,C,D {les variables B,C, et D proviennent du corps principal }
    TESTCOMMON:=X+B+C+D
END
B=4; C=5; D=6
G=TESTCOMMON(3)
```

\$COMMON doit être uniquement utilisé avec des fonctions, procédures ou modules qui sont dans la fenêtre d'Equations. \$COMMON ne peut être utilisé avec des fichiers librairie.

La directive \$INCLUDE

La directive \$INCLUDE permet de charger automatiquement des fichiers de librairie ou des fichiers ASCII contenant des équations EES. Le format est le suivant :

```
$INCLUDE NOMDEFICHIER
```

NOMDEFICHIER est le nom du fichier (nom et extension) à appeler. Les extensions possibles sont : .TXT, .LIB, .FDL, .DLF, or .DLP. NOMDEFICHIER doit aussi comprendre le chemin complet dans lequel il se trouve, ex. C:\EESW\myDefn.TXT. Si le chemin n'est pas spécifié, EES cherchera le fichier invoqué dans le répertoire courant. L'instruction \$INCLUDE doit être la seule sur sa ligne et doit commencer à la colonne 1. Il est conseillé de la placer tout en haut de la fenêtre d'Equations pour être sur que celle-ci sera lue avant la compilation des équations.

Fichiers .TXT

Si le nom de votre fichier possède l'extension .TXT, EES s'attend à ce que ce type de fichier soit ASCII contenant des équations EES. Les erreurs de syntaxe ne peuvent pas être mises en évidence dans un fichier de la librairie, en conséquence, vérifier la syntaxe de vos équations avant de les sauver dans un fichier librairie,. EES intégrera ces équations ainsi que les variables avec celles de la fenêtre d'Equations pendant la compilation. Cependant les informations contenues dans ce type de fichier ne peuvent être lues dans la fenêtre d'Equations. Il n'est pas possible d'imbriquer les directives \$INCLUDE, c'est-à-dire que la directive \$INCLUDE ne peut être dans un fichier de librairie.

Fichiers Librairie

Si le nom du fichier à inclure possède l'extension .LIB, .FDL, .DLF, ou .DLP, EES s'attend à ce que le contenu de ce fichier corresponde à son extension. Les fonctions internes, procédures et modules sont reconnus à l'aide de l'extension .LIB. Les fonctions externes utilisent l'extension .FDL ou .DLP.

Fonctions et procédures compilées

EES possède une vaste librairie de fonctions incorporées (mathématiques, propriétés thermodynamiques). Cependant il est impossible d'anticiper l'ensemble des besoins des utilisateurs. Une caractéristique remarquable de EES est qu'il permet à son utilisateur d'ajouter (et bientôt de supprimer) des fonctions et procédures écrites dans un langage compilé tel PASCAL, C, C++ ou FORTRAN. Ces routines compilées peuvent comprendre n'importe quel nombre d'arguments et peuvent retourner une ou plusieurs valeur(s). Les fonctions retournent une seule valeur, tandis que les procédures peuvent retourner plusieurs valeurs. Les routines compilées sont utilisées exactement comme des sous-programme EES. Cette caractéristique donne à EES un grand degré de flexibilité.

Les fonctions et procédures compilées doivent être écrites dans un fichier de librairie dynamique (DLL) dans le cas de l'utilisation de EES sous Windows⁶. Les fonctions compilées sont identifiées à l'aide de l'extension .DLF. Il existe deux formats de fichier pour les procédures compilées : .DLP et .FDL. Quand EES démarre, il examine tous les fichiers du répertoire USERLIB\. Tous les fichiers .DLF, .DLP, ou .FDL sont considérés comme des fonctions et procédures compilées et sont chargés automatiquement en mémoire. Les routines externes peuvent être également chargées en utilisant la commande **Load Library** du menu **File** ou en utilisant la directive \$INCLUDE. Le nom des fonctions externes à référencer dans EES est celui du fichier externe sans son extension.

Les fonctions et procédures compilées peuvent (optionnel) être configurées de telle manière qu'elles peuvent travailler avec la commande **Function Info** du menu **Options**. Ainsi, il est possible d'illustrer ces fonctions par des exemples, ajouter des fichiers d'aide. Le présent chapitre a pour objet de décrire comment intégrer des fonctions et procédures.

Les fonctions compilées (fichiers .DLF)

Les fonctions compilées peuvent être écrites en C, C++, PASCAL, ou tout autre langage pouvant générer des fichiers DLL. L'en-tête de fonction doit cependant respecter un format spécifique. Pour éviter de fixer une limite maximale d'entrées, les informations relatives aux entrées de la fonction sont implémentées en temps que liste liée. Cette liste liée consiste en une

⁶ Noter qu'il existe deux versions de fichiers DLL suivant l'environnement Windows : 16 et 32 bits. Ces fichiers ne sont pas interchangeables. La version 16 bit de EES ne peut intégrer que des fichiers DLL 16 bits (la version 32 bits ne comprend que des DLL 32 bits.). La version 32 bits de EES ne fonctionne qu'à partir des versions Windows 95, NT. Des instructions pour générer les deux types de fichier (16 et 32 bits) sont fournies dans ce chapitre.

valeur (précision étendue) et un pointeur sur l'entrée suivante. La dernière entrée pointe sur un élément nul. Certains langages comme FORTRAN 77 ne supportent pas les pointeurs. Les fonctions compilées ne peuvent être écrites à l'aide de ce langage. Le format .FDL peut être utilisé et est décrit dans ce chapitre.

Une fonction compilée doit vérifier que le nombre d'entrées fourni dans la liste liée est égal au nombre de fonctions attendues. (la fonction PWF décrite dans la section suivante illustre comment réaliser ce processus). Les valeurs des entrées peuvent changer dans la fonction, mais ceci reste purement local à la fonction et ne regarde pas EES. Seul le résultat est utilisé. Un squelette de fonction écrite en Borland's Delphi 1.0 (16-bits) ou Delphi 3.0 (32-bits) est présenté ci-dessous :

```
library XTRNFUNC;
{$N+}

type
  ParamRecPtr = ^ParamRec;
  ParamRec = record { défini la structure de liste d'entrées liée }
    Value: extended;
    next: ParamRecPtr;
  end;

function FuncName (var S:string; Mode:integer; Inputs:ParamRecPtr): extended; export; stdCall;7
  begin
    ...
    FuncName:=Value; { Funcname doit avoir une valeur (précision étendue) }
  end;

exports FuncName;

begin
end.
```

Pour que la fonction soit reconnue par EES, celle-ci doit avoir le même nom que le fichier qui la contient. Dans notre exemple la fonction 'FuncName' doit appartenir au fichier Funcname. L'instruction Function dans ce cas requiert 3 arguments.

S est une chaîne de caractères standard. Le premier caractère contient la taille actuelle de la chaîne. S peut être aussi bien utilisé en tant qu'entrée ou sortie. Si le premier argument fourni à la fonction EES est une chaîne de caractères (entre ''), EES passera cette chaîne à la routine externe. Si une erreur est rencontrée, S doit être couplée dans la routine à un message d'erreur. Si la taille de S est nulle, EES terminera les calculs et affichera S en tant que message d'erreur.

⁷ Le mot clef **stdCall** est requis par Delphi 3.0 (32-bit). Il n'est pas utile dans la version Delphi 1.0 (16-bit)

Mode est un entier fixé par EES. Si Mode=-1, EES demande à la fonction que S retourne un exemple de l'appel de fonction. Si Mode>=0, la fonction doit retourner sa valeur. Pour l'instant EES n'utilise pas la valeur retour du paramètre Mode.

Inputs est un pointeur vers l'en-tête de la liste liée d'entrée fournie par EES. Chaque entrée consiste en une valeur (précision étendue) et un pointeur sur l'entrée suivante (voir la structure ParamRec). La fonction peut avoir une ou plusieurs entrées. Le champ qui suit la dernière entrée est un pointeur nul (nil). La fonction doit vérifier que le nombre d'arguments passé par EES correspond au nombre attendu, afin d'afficher un message d'erreur S le cas échéant.

Un squelette de fonction écrite en Borland's C est présenté ci-dessous :

```
#include <windows.h>
#include <stdlib.h>
#include <string.h>
#define EXAMPLE=-1;
// Use extern "C" to prevent C++ name mangling
extern "C"
{
    long double far PASCAL _stdcall _export FUNCNAME
        (char* S, int Mode, struct ParamRec *FirstInput);
}
int far PASCAL LibMain //DLL entry point for initiation
    (HINSTANCE hInstance, WORD wDataSeg, WORD cbHeapSize, LPSTR lpstrCmdLine)
{
    if (cbHeapSize) UnlockData(0);
    return TRUE;
}
int far PASCAL WEP(int nParam) //Windows exit procedure - not needed in Borland C++
{
    return TRUE;
}
struct ParamRec
{
    long double          value;
    struct ParamRec     *next;
};
long double far _export _stdcall PASCAL FUNCNAME(char* S, int Mode, struct ParamRec *FirstInput)
{
    ....
    ....
    return (v);
}
```

Noter que le mot clef 'PASCAL' doit être fourni pour être sur que les paramètres d'appel sont ordonnés et seront compris par EES.

Les fonctions compilées PWF

EES ne possède pas de fonction économique interne. Une fonction économique appelée 'present worth factor (PWF)⁸ a été ajoutée en tant que fonction compilée. PWF est la valeur actuelle d'une série de N futurs paiements qui augmentent à un taux i par période. Une période est comptabilisée suivant la valeur de la monnaie fonction du temps avec une escompte du marché d par période. L'équation est la suivante :

$$PWF(N,i,d) = \sum_{j=1}^N \frac{(1+i)^{j-1}}{(1+d)^j} = \begin{cases} \frac{1}{d-i} \left[1 - \left(\frac{1+i}{1+d} \right)^N \right] & \text{if } i \neq d \\ \frac{N}{1+i} & \text{if } i = d \end{cases}$$

où

N est le nombre de périodes (années)

i est le taux d'intéressement par période (exprimé en fraction)

d est l'escompte du marché par période (exprimé en fraction)

Une fonction compilée appelée PWF a été développée pour faire des calculs économiques. Cette fonction se trouve dans le fichier PWF.DLF. Les pages suivantes contiennent le listing complet de cette fonction écrite en Borland's Delphi 3.0 (32-bit version).

Quatre autres fonctions sont fournies avec EES. Ces fonctions implémentent une équation d'état général utilisant l'équation de Redlich-Kwong-Soave⁹.

Compressibility(Tr, Pr, w) retourne la compressibilité d'un gaz, c'est-à-dire le ratio entre le volume de gaz spécifiée et le même volume d'un gaz parfait. Tr est la température réduite, Pr est la pression réduite, et w est le facteur acentrique. Ce facteur est optionnel.

EnthalpyDep(Tr, Pr, w) et **EntropyDep**(Tr, Pr, w) retourne respectivement l'enthalpie (sans dimension) et l'entropie (sans dimension). L'enthalpie est définie comme $(h[\text{ideal}]-h)/(R T_c)$. $h[\text{ideal}]-h$ est la différence d'enthalpie entre un gaz parfait et un gaz réel pour les mêmes température et pression. R est la constante du gaz, T_c est la température critique. L'enthalpie est donnée par la formule : $(s[\text{ideal}]-s)/(R)$.

FugCoef(Tr, Pr, w) retourne le coefficient de fugacité (fugacité / pression).

Vous avez besoin d'une fonction que EES ne contient pas. Ecrivez-la vous même c'est EESY ! (~facile)

⁸ Duffie, J.A. and Beckman, W.A., *Solar Engineering of Thermal Processes*, 2nd edition, J. Wiley and Sons, 1992, Chapter 11.

⁹ G. Soave, *Chem. Eng. Science*, Vol. 27, pp. 1197-1203, 1972

Listing de la fonction PWF développée en Borland's Delphi 3.0

```
library PWF;
uses
  SysUtils, Classes;
{$N+}

const doExample = -1;

type
  ParamRecPtr=^ParamRec;
  ParamRec=record
    Value:extended;
    next:ParamRecPtr;
  end;

function CountValues (P: ParamRecPtr): integer;
var N: integer;
begin
  N := 0;
  while (P <> nil) do begin
    N := N + 1;
    P := P^.next
  end;
  CountValues := N;
end; {CountValues}

function PWF(var S:Shortstring; Mode:integer; Inputs:ParamRecPtr):extended; export; stdcall;
var P: ParamRecPtr; V: extended;

function CountValues (P: ParamRecPtr): integer;
var
  N: integer;
begin
  N := 0;
  while (P <> nil) do begin
    N := N + 1;
    P := P^.next
  end;
  CountValues := N;
end; {CountValues}

function PWFCalc: extended;
var
  Periods, NArgs: integer;
  interest, discount: extended;
begin
  PWFCalc:=0; {in case of error exit}
  S := '';
  P := Inputs;
  Periods := round(P^.value);
  if (Periods < 1) then begin
    S := 'The number of periods for the PWF function must be >0.';
    exit;
  end;
  P := P^.next;
  interest := P^.value;
```

```

if (interest >= 1) or (interest < 0) then begin
  S := 'The interest rate is a fraction and must be between 0 and 1.';
  exit;
end;
P := P^.next;
discount := P^.value;
if (discount >= 1) or (discount < 0) then begin
  S := 'The discount rate is a fraction and must be between 0 and 1.';
  exit;
end;
if (interest <> discount) then
  PWFCalc := 1 / (discount - interest) * (1 - exp(Periods * ln((1 + interest) / (1 + discount))))
else
  PWFCalc := Periods / (1 + interest);
end; {PWF}

begin
PWF:=1;
if (Mode = doExample) then begin
  S := 'PWF(Periods,Interest,Discount)';
  exit;
end;
if (CountValues(Inputs)<>3) then
  S := 'Wrong number of arguments for PWF function.'
else begin
  PWF:=PWFCalc;
end;
end; {PWF}

exports PWF;

begin
{no initiation code needed}
end.

```

Quand le code PASCAL est compilé avec Borland Delphi 3.0, un fichier de librairie dynamique (DLL) est généré. EES doit faire la distinction entre une fonction compilée et une procédure compilée. Pour ce faire, renommer votre DLL avec l'extension .DLF par exemple pour une fonction compilée.

L'accès à la fonction PWF externe se fait dans EES de la manière suivante :

$$P = \text{PWF}(\text{Periods}, \text{Interest}, \text{Discount})$$

Procédures Compilées (fichiers .FDL et .DLP)

Les procédures compilées sont pratiquement similaires aux fonctions compilées. Dans les deux cas, l'utilisateur écrit une fonction ou une procédure au travers d'une DLL. La différence majeure entre une fonction et une procédure, est qu'une procédure peut retourner une ou plusieurs valeur(s). Les procédures sont utiles par exemple, pour des évaluations de propriétés thermodynamiques où de multiples facteurs (volume, enthalpie, entropie, etc.) sont déterminés à partir d'un ensemble de variables indépendantes (ex. température, pression).

Les procédures externes sont écrites dans une DLL. Faites seulement attention de développer une routine 16-bits pour la version 16-bits de EES, et 32-bits pour la version 32-bits de EES. Il existe deux formats de procédures, ceux-ci sont identifiés par l'extension du fichier. Ces deux formats diffèrent dans la façon dont ESS échange des données avec les procédures. Le format .FDL permet d'échanger des données d'entrée et sortie au travers d'un tableau de nombres flottants (double précision) qui peut contenir jusqu'à 50 éléments. Le format .DLP utilise une liste liée (comme les fonctions .DLF) ; ainsi il n'y a pas de limite pour le nombre d'entrées/sorties. Les procédures externes qui utilisent des tableaux pour stocker les entrées/sortie doivent avoir l'extension .FDL. Ce format est utile si vous désirez générer votre procédure à l'aide de FORTRAN. Les langages C, C++ et PASCAL peuvent accepter les deux formats.

Les procédures compilées sont accédées dans EES de la manière suivante :

CALL procname('text', A, B : X, Y, Z)

où

procname est le nom de la procédure.

'text' est une chaîne de caractères (optionnel) qui peut être passée à la procédure. Ce texte peut être soit une chaîne de caractères, soit une constante chaîne de caractères (compris entre "). (Voir Chapitre 7).

A et B sont des entrées. Il peut y avoir une ou plusieurs entrée(s). Celles-ci sont séparées par des virgules. Les entrées peuvent être des constantes numériques, des noms de variables EES ou des expressions algébriques (les chaînes de caractères sont interdites ici).

X, Y, et Z sont les sorties déterminées par la procédure. Il peut y avoir une ou plusieurs sorties. Celles-ci sont séparées par des virgules. Les sorties doivent être des variables numériques reconnues par EES (les chaînes de caractères sont interdites ici).

Noter que l'instruction CALL utilisée pour les procédures compilées est identique à celle utilisée pour appeler les procédures EES.

Les deux sections suivantes décrivent les 2 types de procédures externes : .FDL et .DLP.

Procédure compilée, format .FDL : exemple en FORTRAN

Le format .FDL est illustré par l'exemple FORTRAN suivant. Le code diffère légèrement suivant les versions 16 ou 32-bits.

Version 16-bits d'un fichier .FDL écrit à l'aide du compilateur Microsoft FORTRAN 5.1

```
SUBROUTINE MYPROC(S,MODE,NINPUTS,INPUTS,NOUTPUTS,OUTPUTS)
INTEGER*2 MODE,NINPUTS,NOUTPUTS
REAL*8 INPUTS(50),OUTPUTS(50)
CHARACTER*255 S
...
OUTPUTS(1)=...
...
RETURN
END
```

Version 32-bits d'un fichier .FDL écrit à l'aide du compilateur Digital Visual FORTRAN

```
SUBROUTINE MYPROC(S,MODE,NINPUTS,INPUTS,NOUTPUTS,OUTPUTS)
!DEC$ATTRIBUTES ALIAS:'MYPROC' :: MYPROC
!DEC$ATTRIBUTES DLLEXPORT :: MYPROC
INTEGER(4) MODE,NINPUTS,NOUTPUTS
REAL(8) INPUTS(50),OUTPUTS(50)
CHARACTER(255) S
...
OUTPUTS(1)=...
...
RETURN
END
```

S est une chaîne de caractères de terminaison contenant 255 caractères au maximum. Si le premier paramètre dans l'appel de la procédure (instruction CALL) est une chaîne de caractères, EES passera ce texte à la chaîne S. Quand EES appelle la routine avec comme paramètre MODE = -1, un exemple de l'utilisation de la fonction doit être placé dans la variable S pour être ensuite affiché dans la fenêtre 'Function Info Dialog'. S peut retourner également un message d'erreur utilisateur si cela s'avère nécessaire. Si une erreur est détectée dans la routine, MODE doit avoir une valeur supérieure à 0 pour indiquer à EES de terminer les calculs. Si S est défini, un message d'erreur sera affiché. Dans un fonctionnement normal MODE = 0 et S n'a pas besoin d'être identifié.

NINPUTS et NOUTPUTS sont les nombres d'entrées et sorties fournis par EES. La routine doit vérifier si ces nombres correspondent bien aux nombres de paramètres attendus. Sinon, MODE > 0. INPUTS et OUTPUTS sont des tableaux de 50 valeurs avec une double précision (Reel*8) values. EES devra fournir la valeur du tableau des données d'entrées. Les résultats des calculs seront placés dans le tableau OUTPUTS.

Le programme externe doit être compilé et ‘linké’ en un fichier DLL. La procédure de compilation dépend du langage utilisé. Pour compiler et ‘linker’ une procédure externe FORTRAN appelée MYPROC sous forme de DLL 16-bits à l’aide de Microsoft FORTRAN 5.1, la procédure est la suivante :

```
f1 /c /Aw /Gw MYPROC.FOR
link MYPROC, MYPROC.FDL,NUL,/NOD LDLLFEW, MYPROC.DEF
```

Les options du compilateur et du ‘linker’ sont décrites dans le manuel Microsoft FORTRAN 5.1. Un fichier de définition : MYPROC.DEF est requis par le linker, son format est le suivant :

```
LIBRARY MYPROC
DESCRIPTION MYPROC TEST FDL'
APPLoader '__MSLANGLOAD'
EXETYPE WINDOWS 3.0
CODE PRELOAD MOVEABLE DISCARDABLE
DATA PRELOAD MOVEABLE SINGLE
HEAPSIZE 1024
EXPORTS MYPROC @1
        WEP @2 RESIDENTNAME
```

Générer un fichier DLL 32-bits avec Digital Visual FORTRAN 5.0 est plus facilement réalisé sous l’environnement Microsoft Developer Studio. Un nouveau projet est sélectionné en tant que DLL. Les fichiers sources FORTRAN sont insérés et compilés avec les options standards. Noter que les deux directives !DEC\$ATTRIBUTES doivent être incluses dans le corps principal du programme comme noté ci-dessus. Le nom du fichier de sortie dans les paramètres de ‘linkage’ doit être MYPROC.FDL où MYPROC est le nom utilisé par EES dans l’instruction CALL (une autre méthode consiste à renommer votre DLL en .FDL).

Le programme FORTRAN ci-dessous permet de calculer le produit, le quotient, la somme et la différence de deux variables d’entrée. Ce programme peut être utilisé comme modèle de développement de procédures FORTRAN pour EES.

Code du programme FORTRAN : MDASF

```
      SUBROUTINE MDASF(S,MODE,NINPUTS,INPUTS,NOOUTPUTS,OUTPUTS)
C. The following two lines are specific to Microsoft Power Station 4.0
      !MS$ATTRIBUTES ALIAS:'MDASF' :: MDASF
      !MS$ATTRIBUTES DLLEXPORT :: MDASF
C. Replace INTEGER(4) with INTEGER*2 for a 16 bit DLL in the following line
      INTEGER(4) MODE, NINPUTS, NOOUTPUTS
      REAL(8) INPUTS(25), OUTPUTS(25)
      CHARACTER(255) S
C.
      IF (MODE.EQ.-1) GOTO 900
      IF (NINPUTS.NE.2) GOTO 100
      IF (NOOUTPUTS.NE.4) GOTO 200
C. DO CALCULATIONS
      X=INPUTS(1)
      Y=INPUTS(2)
      IF (ABS(Y).LE.1E-9) GOTO 300
      OUTPUTS(1)=X*Y
      OUTPUTS(2)=X/Y
      OUTPUTS(3)=X+Y
      OUTPUTS(4)=X-Y
      MODE=0
      S=' 'C
      RETURN
100    CONTINUE
C. ERROR: THE NUMBER OF INPUTS ISN'T WHAT THIS SUBROUTINE EXPECTS
C. NOTE: SET MODE>0 IF AN ERROR IS DETECTED. IF S IS EQUAL TO A
C. NULL STRING, THEN EES WILL DISPLAY THE MODE NUMBER IN AN ERROR
C. MESSAGE. IF S IS DEFINED, EES WILL DISPLAY THE STRING IN THE
C. ERROR MESSAGE. THE C AT THE END OF THE STRING INDICATES C-STYLE
C. S='MDASF REQUIRES 2 INPUTS'C
      MODE=1
      RETURN
200    CONTINUE
      S='MDASF EXPECTS TO PROVIDE 4 OUTPUTS'C
      MODE=2
      RETURN
300    CONTINUE
      S='DIVISION BY ZERO IN MDASF'C
      MODE=3
      RETURN
900    CONTINUE
C. PROVIDE AN EXAMPLE OF THE CALLING FORMAT WHEN MODE=-1
      S='CALL MDASF(X,Y:A,B,C,D)'C
      RETURN
      END
```

Les procédures compilées, format. DLP – un exemple en PASCAL

Le format .FDL décrit précédemment a été illustré à l'aide d'un exemple développé en FORTRAN. Celui-ci aurait pu être développé avec n'importe quel langage de haut niveau. Le format .DLP décrit dans cette section fait appel à la notion de pointeur. En effet, il utilise des listes liées pour les variables d'entrées et sorties et donc n'est pas adapté à la programmation FORTRAN. Il n'existe pas de différence majeure au niveau efficacité entre ces deux formats. Les deux sont tout à fait compatibles.

Les procédures compilées utilisant le format .DLP sont quasi similaires aux fonctions compilées décrites précédemment (fichier .DLF). La seule différence est qu'une procédure doit être 'liée' à une liste de variables d'entrées et une liste de variables de sorties. La séquence d'appel d'une procédure en PASCAL est la suivante :

```
procedure procname (var S: string; Mode: integer; Inputs, Outputs: ParamRecPtr);
```

S, Mode, et Inputs sont identiques aux fonctions compilées. Outputs est liée à une liste de valeurs qui seront fournies à EES dans l'ordre dans lequel elles apparaissent dans l'instruction CALL de EES.

La page suivante contient un listing de procédures compilées pour EES. Ce fichier appelé MDAS contient des procédures permettant de calculer le produit, le quotient, la somme et la différence de deux variables d'entrées (ce programme est identique au précédent). Le code vérifie que le nombre d'arguments fourni à la fonction correspond bien à ce qu'elle attend et affiche, via la chaîne de caractères S, un message d'erreur.

Exemple de procédure compilée (.DLP) en Borland's Delphi 2.0

```
library MDASP;

uses
  SysUtils, Classes;
{$N+}
const Example = -1;
type
  ParamRecPtr=^ParamRec;
  ParamRec=record
    Value:extended;
    next:ParamRecPtr;
  end;

function CountValues (P: ParamRecPtr): integer;
var N: integer;
begin
  N := 0;
  while (P <> nil) do begin
    N := N + 1;
    P := P^.next
  end;
  CountValues := N;
end; {CountValues}

procedure MDAS(var S:Shortstring; Mode:integer;
  Inputs,Outputs:ParamRecPtr); export; stdcall;

procedure MyDearAuntSally;
var
  P1, P2: extended;
  P: ParamRecPtr;
begin
  P := Inputs;
  P1 := P^.Value;
  P := P^.next;
  P2 := P^.value;
  P := Outputs;
  P^.Value := P1 * P2;
  P := P^.next;
  P^.Value := P1 / P2;
  P := P^.next;
  P^.Value := P1 + P2;
  P := P^.next;
  P^.Value := P1 - P2;
end; {doCall}

begin {MDAS}
  if (Mode = -1) then
    S := 'CALL MDAS(In1,In2:Out1,Out2,Out3,Out4)'
  else begin
    if (CountValues(Inputs) <> 2) then begin
      S := 'Wrong number of inputs for MDAS.';
      exit;
    end;
  end;
end;
```

```
end;
if (CountValues(Outputs) <> 4) then begin
  S := 'Wrong number of outputs for MDAS.';
  exit;
end;
MyDearAuntSally;
S:='';
end;
end; {MDAS}

exports
  MDAS;

begin
  {no initiation code needed}
end.
```

Aide pour les fonctions et procédures compilées

La boîte de dialogue 'Fonction Info' accessible via le menu **Options** possède un bouton **INFO** utilisable pour afficher une série d'informations sur l'utilisation de la fonction sélectionnée. Quand l'utilisateur clique sur ce bouton, EES cherche un fichier .HLP possédant le même nom que la fonction sélectionnée. Ce fichier peut être soit de type ASCII soit un fichier généré à l'aide d'un éditeur de fichier d'aide Windows. L'aide sera affichée si le fichier est trouvé dans le répertoire contenant également le fichier auquel la fonction sélectionnée appartient. Si EES ne trouve pas le fichier, le message indiquant qu'il n'y a pas d'aide pour cette fonction.

Si un fichier ASCII est fourni, il doit être avoir un format spécifique : chaque fin de paragraphe doit se terminer par un retour chariot. Les lignes trop longues pour la fenêtre d'aide seront cassées. Les lignes vierges et les espaces peuvent être utilisés pour améliorer la lisibilité de votre texte.

Noter qu'un fichier d'aide Windows autorise les schémas ainsi que des options de formatage de texte. Les fichiers .HLP Windows peuvent être générés à l'aide d'un traitement de texte permettant de sauvegarder des fichiers au format RFT (Rich Text Format) ou à l'aide d'un outil permettant de générer des fichiers d'aide.

Caractéristiques avancées

Les caractéristiques avancées dans EES permettent au programme de travailler avec des variables de type chaînes de caractères, complexes ou de type tableaux ou bien de résoudre simultanément des équations algébriques et différentielles. L'ensemble de ces fonctions et caractéristiques est décrit dans ce chapitre.

Chaînes de caractères

EES dispose de deux types de variables : les variables numériques et les chaînes de caractères. Une variable de type chaîne de caractères est identifiée par un nom de variable suivie du signe \$ comme en langage Basic. Ces variables doivent commencer par une lettre et ont une longueur maximum de 30 caractères (signe \$ compris).

Les variables de type chaînes de caractères peuvent être également des constantes. Une constante chaîne de caractères comporte 255 caractères incluant les signes " (ex : A\$='dioxide de carbone').

Une chaîne de caractères peut prendre la valeur d'une autre, par exemple :

A\$=B\$

Vous pouvez utiliser ces variables en tant qu'arguments au sein de fonctions, procédures ou modules ou fonctions externes (cf. Chapitre 6).

En règle générale, les chaînes de caractères peuvent être utilisées n'importe où dans les équations. Par exemple,

h=enthalpy(R\$,T=T,P=P)

La variable H\$ a été précédemment initialisée comme étant un fluide tiré des fonctions et propriétés thermodynamiques de EES.

Les chaînes de caractères peuvent être également utilisées dans les tables Paramétrique. Dans l'exemple suivant, une table paramétrique est utilisée pour représenter dans un tableau l'enthalpie de 4 réfrigérants. Noter que les signes " normalement utilisés pour les constantes chaînes de caractères ne sont pas requis dans la table paramétrique.

	1 R\$	2 h [kJ/kg]
Run 1	R134a	342.2
Run 2	R22	326.7
Run 3	R404A	320.7
Run 4	R407C	359.4

Une chaîne de caractères peut être utilisée dans les fonctions relatives à la table ou fichier 'Lookup' :

$m = \text{Interpolate}(\text{File}\$, \text{Col1}\$, \text{Col2}\$, \text{Col1}\$ = x)$

$k = \text{Lookup}(\text{File}\$, \text{Row}\$, \text{Col}\$)$

Enfin, les variables chaînes de caractères utilisées dans la fenêtre de Diagramme (cf. Chapitre 2) peuvent être transmises dans la fenêtre d'Equations.

Variables complexes

Pour résoudre des équations à base de nombres complexes de la forme $a + b \cdot I$, l'option 'Do Complex Algebra' doit être cochée dans la boîte de dialogue 'Preference'. L'opérateur des imaginaires¹⁰ peut être soit i soit j , suivant l'option sélectionnée.

Quand EES travaille en mode complexe, il considère que toutes variables (exceptées les chaînes de caractères) sont divisées en deux : une variable pour les nombres réels, une pour les nombres complexes. La partie réelle est identifiée par le nom de la variable suivi de $_r$ (ex : X_r). De même, la partie imaginaire est représentée par le nom de la variable suivi de $_i$ (ex : X_i) (lors de la création de vos variables, éviter de nommer vos variables nom_i ou nom_r à moins de faire référence aux parties réelles et imaginaires d'une variable complexe).

L'exemple suivant montre comment opère EES en mode complexe :

$$X = Y$$

EES créera automatiquement des variables X_r , X_i , Y_r et Y_i qui correspondent aux parties réelles et imaginaires des variables complexes. Normalement, vous n'aurez pas à utiliser ces variables, bien que les variables complexes apparaissent sous cette forme dans la boîte de dialogue 'Variable Info' ou dans une table paramétrique. Ceci étant, vous pouvez modifier la valeur de la partie réelle ou imaginaire d'un complexe lors de la création de la variable. Dans

¹⁰ Dans le reste du paragraphe, ' i ' sera utilisé comme opérateur des imaginaires.

l'exemple suivant omega est une variable complexe pour laquelle nous voulons rendre nulle sa partie imaginaire :

$\omega_i=0$

Si vous désirez modifier la valeur de la partie imaginaire de omega, EES retourne un message d'erreur. Par exemple, l'équation $\omega = 3$ entraînera un message d'erreur car les valeurs des parties réelles et imaginaires seront égales à 3 alors que vous avez initialement fixé la partie imaginaire à 0.

Les nombres complexes peuvent être saisis sous la forme rectangulaire ou polaire. Dans la forme rectangulaire, un nombre complexe se représente de la façon suivante :

Exemple : $Y = 2 + 3 * i$

Dans la forme polaire, un nombre complexe s'écrit à l'aide de son module et de sa phase séparés par le signe <. La phase peut être en degrés ou en radians. Si rien n'est spécifié, EES considère que l'angle saisi est dans les mêmes unités que celles paramétrées au travers de la commande 'Unit system'. Cependant, vous pouvez forcer EES à utiliser un angle en degrés ou en radians, en ajoutant à la valeur de la phase (sans espace) deg ou rad. Par exemple, la variable complexe Y peut s'écrire sous 2 autres formes :

$Y = 3.606 < 56.31 \text{deg}$

$Y = 3.606 < 0.9828 \text{rad}$

L'usage des paramètres deg ou rad pour indiquer les unités d'angle est fort recommandé. Tout d'abord, la valeur saisie ne sera pas modifiée par des opérations sur le système d'unités. Ensuite, vous pouvez écrire un nombre complexe sous forme polaire avec des angles en degrés et effectuer les calculs en radians (si l'option radian est cochée dans la boîte de dialogue Unit System) ce qui est plus efficace.

Enfin, si vous spécifiez des angles en degrés, le signe des degrés sera affiché dans la fenêtre d'Equations mise en forme.

En mode complexe, EES crée deux équations (interne) pour chaque équation saisie dans la fenêtre d'Equations. La première est utilisée pour opérer sur la partie réelle du nombre complexe, l'autre sur la partie imaginaire. Il est possible de visualiser ces équations dans la fenêtre 'Residuals', qui affiche le résidu et l'ordre de traitement lors de la résolution d'équations. Les équations utilisant la partie réelle ou imaginaire sont identifiées respectivement par (r) ou (i).

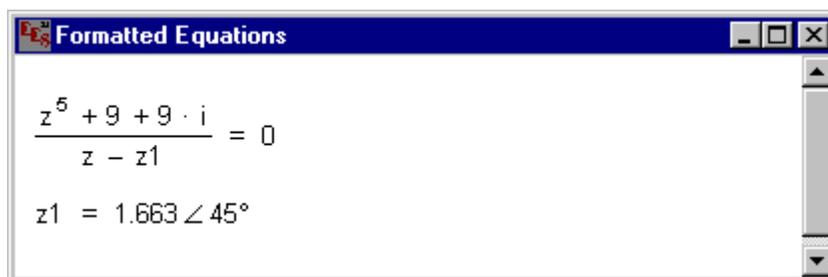
En mode complexe, certaines fonctions telles **Min** et **Max**, ne sont pas accessibles. Ceci étant, la plupart des fonctions incorporées (y compris les fonctions de propriété de thermodynamique) peuvent travailler en mode complexe. Par exemple, **sin**, **cos**, **ln**, **exp**, et **tanh** acceptent et retournent des nombres complexes correspondants. Les fonctions utilisateur, les procédures, et les routines externes peuvent être utilisées mais acceptent et retournent les nombres réels seulement. (les modules ne sont pas actuellement utilisables en mode complexe). Seule la partie réelle d'une variable complexe sera placée dans la liste d'argument de fonctions, de procédures, et de modules internes ou externes.

Il existe également des fonctions qui ne travaillent qu'en mode complexe : **Real**, **Imag**, **Cis**, **Magnitude**, **Angle**, **AngleDeg**, **AngleRad**, et **Conj**. Ces fonctions utilisent un argument complexe.

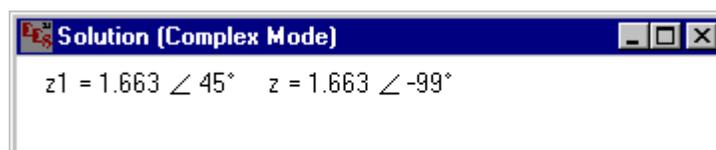
En mode complexe, EES ne peut retourner qu'une solution d'une équation même s'il en existe plusieurs. Cependant, il est possible de contourner ce problème et forcer EES à donner plus d'une solution. Considérons le problème de détermination de 5 racines complexes de l'équation : $z^5 + 9 + 9i = 0$. Entrer cette équation dans la fenêtre d'Equations et lancer le calcul (EES utilisera des valeurs par défaut), la solution de cette équation est $z=1.176+1.176i$:



Ceci est une solution exacte de l'équation, mais il en existe 4 autres. Pour trouver les solutions suivantes, il suffit de diviser notre équation de départ par la différence entre z et l'une des racines :

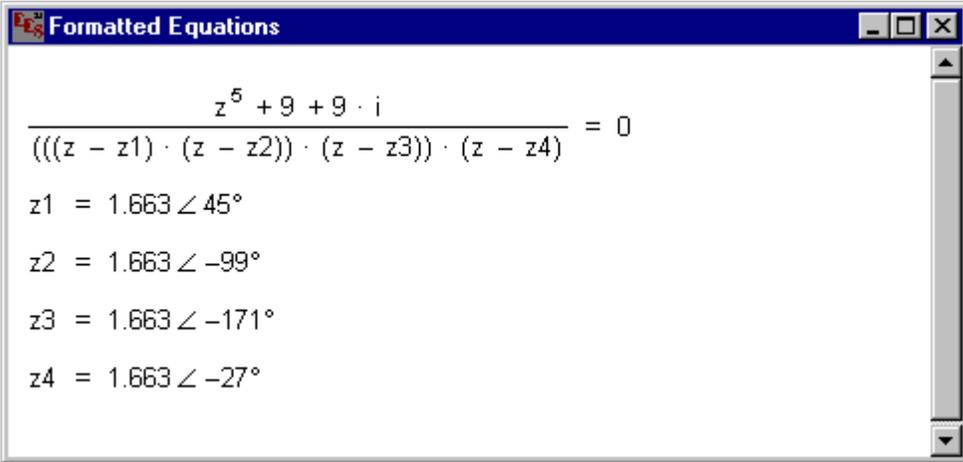


ainsi, EES trouve une deuxième solution.

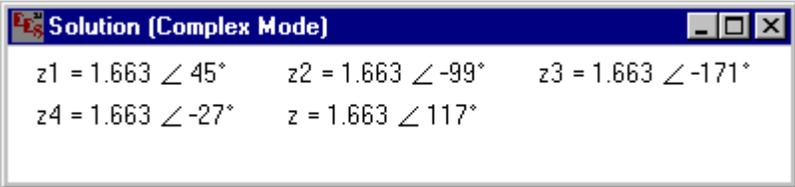


Répetons ce procédé pour trouver les racines suivantes.

Le résultat final est le suivant (fenêtre d'Equations mise en forme) :



The screenshot shows a window titled "Formatted Equations" with a blue header. The main content displays the equation
$$\frac{z^5 + 9 + 9 \cdot i}{((z - z1) \cdot (z - z2)) \cdot (z - z3) \cdot (z - z4)} = 0$$
 followed by four lines of roots: $z1 = 1.663 \angle 45^\circ$, $z2 = 1.663 \angle -99^\circ$, $z3 = 1.663 \angle -171^\circ$, and $z4 = 1.663 \angle -27^\circ$. The window has standard Windows-style window controls (minimize, maximize, close) in the top right corner and a vertical scrollbar on the right side.



The screenshot shows a window titled "Solution (Complex Mode)" with a blue header. The main content displays five roots of the equation: $z1 = 1.663 \angle 45^\circ$, $z2 = 1.663 \angle -99^\circ$, $z3 = 1.663 \angle -171^\circ$, $z4 = 1.663 \angle -27^\circ$, and $z = 1.663 \angle 117^\circ$. The window has standard Windows-style window controls (minimize, maximize, close) in the top right corner.

Les variables 'tableaux' (Arrays)

Une variable de type tableau (array), s'écrit de la façon suivante : `mavariante[index]` ; `mavariante` est le nom d'une variable de type tableau, et `index` désigne l'index du tableau. Il est possible d'utiliser des tableaux a plus d'une dimension, en séparant les index de tableau par le signe ; (ex : `Z[1 ;2 ;3]`).

Remarques :

1. L'index d'un tableau doit être un nombre entier. Il est également possible d'utiliser des constantes, la fonction `TableRun#`, une expression algébrique comprenant les opérateurs¹¹ `+`, `-`, `*`, `/`. Par exemple, `X[2*3+1]` est une variable tableau valide, qui sera transformée par EES en `X[7]`. `X[1+2*3]` sera transformé en `X[9]`. Les variables d'index pour la commande `DUPLICATE` ou les fonctions **sum** ou **product** peuvent être également utilisées en tant qu'index de tableau.
2. L'index doit être compris entre -32760 et $+32760$, 0 compris.
3. Le crochet droit doit être le dernier caractère de la variable tableau.
4. La longueur totale du nom variable (y compris les crochets et la valeur de l'index, ne doit pas dépasser 30 caractères.

EES traite les variables tableau d'une façon très différente de FORTRAN ou PASCAL. Dans EES, chaque variable, tel que `X[99]`, est une **variable unique**. Par exemple, `X[99]` se comporte comme toute autre variable telle `ZZZ`. Les valeurs par défaut, les limites minimales et maximales peuvent être spécifiées en utilisant la commande `Variable Info`. Il est possible (mais non conseillé) d'avoir des variables nommées `X`, `X[1]`, `X[2,3]` dans un même système d'équation. Noter, que l'utilisation de la variable `X[99]` ne veut pas dire que EES réservera 99 zones de la mémoire. Une zone mémoire est allouée seulement quand une variable apparaît dans une équation.

L'utilisation des variables est très utile. Cela permet de grouper des variables de type similaire. Par exemple, les températures à différent état d'un système peuvent être écrites `T[1]`, `T[2]`, etc. Par ailleurs, les variables tableau peuvent être tracées. Par exemple, la température et l'entropie pour chaque état d'un système peuvent être superposées sur un même diagramme (voir `Property Plot` dans le chapitre 3). Enfin, les variables de type tableau peuvent être utilisées avec la commande `DUPLICATE` et les fonctions `sum` et `product` pour pouvoir travailler avec des matrices et réduire significativement la quantité de données à saisir pour résoudre certains problèmes.

¹¹ Le calcul se fait de la gauche vers la droite sans tenir compte des règles de calcul.

La commande DUPLICATE

La commande DUPLICATE est un moyen rapide et commode pour saisir des équations dans EES. Les fonctions qui doivent être dupliquées doivent être comprises entre les commandes DUPLICATE et END. DUPLICATE est une commande utile pour les variables tableaux. Par exemple :

```
N = 5
X[1]=1
DUPLICATE j=2,N
    X[j]=X[j-1 ] + j
FIN
```

Est équivalent à :

```
X[1]=1
X[2]=X[1]+2
X[3]=X[2]+3
X[4]=X[3]+4
X[5]=X[4]+5
```

Les conditions d'utilisation de la commande DUPLICATE sont les suivantes :

1. Les limites plus basses et supérieures spécifiées pour la variable d'index dans l'ordre de DUPLICATE doivent être des nombres entiers, les variables de EES précédemment assignées aux valeurs constantes ou par la fonction **TableRun#**.
2. Il est possible d'imbriquer des commandes DUPLICATE autant que désiré. Ceci étant, chaque commande DUPLICATE doit faire référence à des index différents, et chacune d'entre elles doit se terminer par la commande END. La limite inférieure ou supérieure d'une commande DUPLICATE interne peut être égale à la valeur d'index d'une commande DUPLICATE externe, par exemple :

```
DUPLICATE i=1 ;5, DUPLICATE j=I ;6, X [i, j] = i*j, END, END
```

3. La commande END termine toujours une section DUPLICATE.

Les matrices

Beaucoup de problèmes d'ingénierie peuvent être formalisés au travers de système d'équations algébriques de la forme :

$$[A] [X] = [B]$$

Où $[A]$ est une matrice carrée de coefficients, $[X]$ et $[B]$ des vecteurs. L'équation de cette matrice est résolue pour déterminer les coordonnées du vecteur $[X]$ connaissant $[A]$ et $[B]$; pour ce faire :

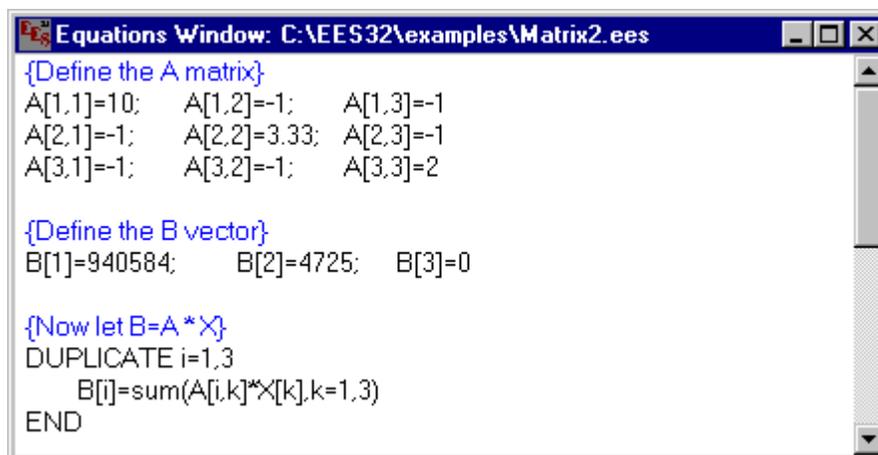
$$[X] = [A]^{-1} [B]$$

EES peut résoudre directement les équations représentées par $[A] [X] = [B]$ en entrant chaque équation directement dans la fenêtre d'Equations dans n'importe quel format et dans n'importe quel ordre. Cependant, une méthode plus élégante et commode pour résoudre ces équations dans EES est d'utiliser les fonctionnalités matricielles. EES peut résoudre les équations de la matrice ci-dessus à l'aide de variables de type tableaux, de la commande DUPLICATE et de la fonction **sum**.

Par exemple, considérons le problème de transfert de chaleur avec $[A]$ et $[B]$ et le vecteur de radiosité donnés ci-dessous, pour lequel il faut déterminer¹² $[X]$:

$$[A] = \begin{bmatrix} 10 & -1 & -1 \\ -1 & 3.33 & -1 \\ -1 & -1 & 2 \end{bmatrix} \quad [B] = \begin{bmatrix} 940584 \\ 4725 \\ 0 \end{bmatrix}$$

Les équations sont les suivantes :



```

EES Equations Window: C:\EES32\examples\Matrix2.ees
{Define the A matrix}
A[1,1]=10;  A[1,2]=-1;  A[1,3]=-1
A[2,1]=-1;  A[2,2]=3.33;  A[2,3]=-1
A[3,1]=-1;  A[3,2]=-1;  A[3,3]=2

{Define the B vector}
B[1]=940584;  B[2]=4725;  B[3]=0

{Now let B=A*X}
DUPLICATE i=1,3
  B[i]=sum(A[i,k]*X[k],k=1,3)
END
  
```

¹² Incropera, F.P. et DeWitt, le D.P., *Les principes de base de Transfert de Chaleur et Masse*, 2nd édition, John Wiley et Fils, 1985, Chapitre 13

Les coordonnées du vecteur $[X]$ apparaissent dans la fenêtre Array (tableau) :

	1	2	3	4	5
	$A_{i,1}$	$A_{i,2}$	$A_{i,3}$	B_i	X_i
[1]	10.00	-1.00	-1.00	940584	108339
[2]	-1.00	3.33	-1.00	4725	59093
[3]	-1.00	-1.00	2.00	0	83716

Noter qu'il n'a pas été nécessaire de déterminer l'inverse de $[A]$ pour résoudre ce problème. En effet, EES a calculé intérieurement et simultanément la matrice inverse ainsi que les autres équations. Cependant, la matrice inverse $[A]^{-1}$ peut être déterminée en écrivant le $[A] [A]^{-1}$ égale à la matrice identité :

```

{Set up identity matrix using Step function}
N=3
DUPLICATE i=1,N
  DUPLICATE j=1,N
    Identity[i,j]=1-step(abs(i-j)-1)
  END
END
{Set identity matrix to the product of A and Ainv}
DUPLICATE i=1,N
  DUPLICATE j=1,N
    Identity[i,j]=sum(A[i,k]*Ainv[k,j],k=1,N)
  END
END
  
```

La matrice inverse apparaîtra dans la fenêtre Array (tableau) :

	4	5	6
	$Ainv_{i,1}$	$Ainv_{i,2}$	$Ainv_{i,3}$
[1]	0.11	0.06	0.09
[2]	0.06	0.39	0.22
[3]	0.09	0.22	0.66

Les deux exemples ci-dessus ont montré une procédure générale pour déterminer le produit d'une matrice par un vecteur ou le produit de deux matrices. Remarquer que l'utilisation de la commande DUPLICATE et les variables tableaux ne sont pas plus efficaces que si vous entriez chaque équation séparément avec des variables normales; cependant, les fonctionnalités des matrices de EES peuvent réduire significativement la quantité d'information à saisir pour un problème donné ou bien rédiger des équations d'une façon beaucoup plus lisibles.

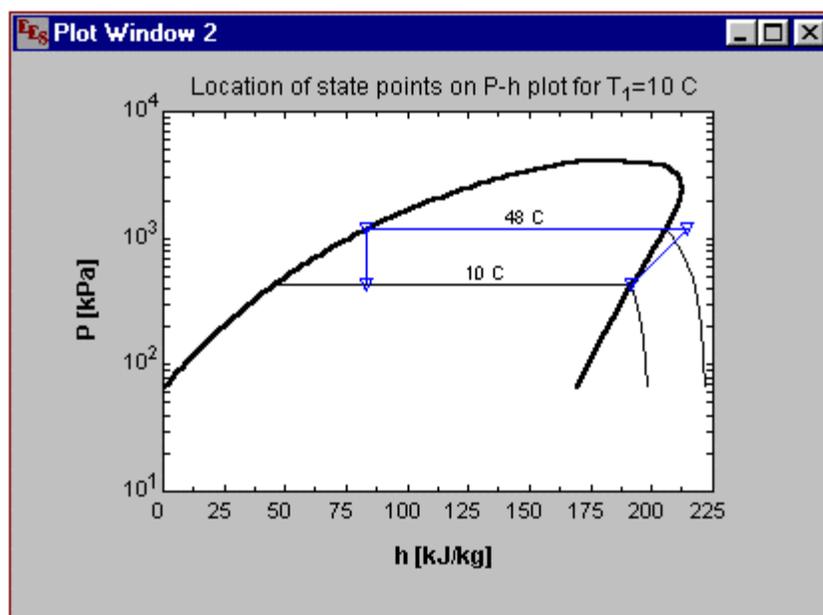
Tracés des propriétés thermodynamiques (property plot)

La commande **Property Plot** du menu **Plot** permet de générer les tracés T-s, T-v, P-v ou P-h de l'ensemble des fluides contenus dans la base de données interne de EES. Un graphique psychrométrique peut être généré si la substance AirH2O est choisie.

Des propriétés thermodynamiques additionnelles peuvent être visualisées sur un même diagramme en utilisant la commande **Overlay Plot** du menu **Plot**. Ceci est fort agréable si des variables de type tableaux sont utilisées comme variables thermodynamiques.

Le tracé P-h ci-dessous montre les différents points correspondant aux états d'un cycle simple de réfrigération entre une température d'évaporation de 10°C et une température de condensation de 48°C avec une efficacité isentropique du compresseur de 0.70. Dans un premier temps, le tracé du P-h pour un réfrigérant R12 a été réalisé avec un isotherme de 10°C et 48°C avec la commande **Property Plot**.

Puis, la valeur des variables P[I] et h[I] issues d'une table Arrays a été tracé sur le même graphique pour chacun des 4 états du cycle de réfrigération. (les équations se trouvent dans le fichier REFRIG.EES dans le répertoire Examples).



Intégrales et équations Différentielles

La fonction INTEGRAL est utilisée pour évaluer une intégrale et/ou pour trouver une solution à une équation différentielle. Le format de la fonction **Integral** est le suivant :

$$\int_{t_1}^{t_2} f dt = \mathbf{Integral} (f,t)$$

La fonction INTEGRAL peut s'écrire suivant deux façons, suivant si l'on utilise ou pas la table paramétrique.

Utilisation de la fonction INTEGRAL avec la table paramétrique

La table paramétrique est utilisée pour fournir à la fonction INTEGRALE les bornes d'intégration, ainsi que le nombre de pas total d'intégration. La variable d'intégration t doit être une variable dont les valeurs ont été définies dans une des colonnes de la table paramétrique. Les bornes d'intégration t1 et t2 son respectivement la première et la dernière valeur de t. La fonction à intégrer f, peut être une variable ou une fonction algébrique faisant appel à des variables, à des valeurs, et à la variable d'intégration.

Utilisation de la fonction INTEGRALE en tant qu'équation

Dans ce cas, la fonction INTEGRALE a le même objectif mais ne nécessite pas l'aide de la table paramétrique. Le format de cette équation est le suivant :

F = INTEGRAL (f, t, t1, t2, tStep)

ou

F = INTEGRAL (f, t, t1, t2) {nombre total de pas d'intégration calculé automatiquement}

T1 et t2 désignent les bornes hautes et basses d'intégration. Ces variables peuvent être spécifiées en tant que constante ou toute expression. Ceci étant, ces variables ne doivent pas dépendre de la variable d'intégration ou bien utiliser des variables dont la valeur varie au cours de l'intégration.

Tstep est une variable d'incrémentation utilisée par EES pendant l'intégration spécifiant le pas de calcul entre les deux bornes de l'intégrale.

EES utilise un algorithme basé sur un indicateur/correcteur du 2nd ordre pour évaluer une intégrale d'une expression. Cet algorithme est conçu pour résoudre des équations algébriques

ou différentielles qui résultent quand l'opérande est une fonction complexe composée d'autres variables. Cet algorithme est parfaitement adapté pour les équations complexes.

EES utilise la fonction **INTEGRALE** pour résoudre la valeur initiale d'une équation différentielle. Par exemple, l'équation différentielle, le $dy/dx = f(x, y)$ peut s'écrire :

$$y = y_0 + \int f(x, y) dx$$

où y_0 est la valeur initiale de y . Cette équation peut être résolue suivant les deux modes de calculs cités précédemment.

- Dans le cas de l'utilisation de la table paramétrique, l'équation sera résolue de la façon suivante :

$$y = y_0 + \text{INTEGRAL}(fxy, x)$$

Où fxy est une variable ou une expression. Pour résoudre l'équation il est nécessaire de créer une colonne pour la variable x dans une table paramétrique. Les valeurs de x sont entrées dans la table paramétrique avec comme première valeur la borne basse d'intégration et comme dernière valeur, la borne haute d'intégration. Le nombre d'itération est déterminé selon le nombre de valeurs de x et non pas selon une valeur fixée. L'intégral est évalué quand la commande **Solve Table** (résolution de la table) est invoquée.

- Dans le cas de l'utilisation de l'équation (sans utiliser la table paramétrique) , l'intégrale apparaîtra de la façon suivante dans la fenêtre d'équation :

$$y = y_0 + \text{INTEGRAL}(fxy, x, \text{low}, \text{high})$$

y_0 et fxy ont été définis ci-dessus. low et high sont les bornes d'intégration. 'stepsize' n'est pas spécifique, EES déterminera sa valeur automatiquement.

Résolution d'une équation différentielle du premier ordre connaissant la valeur initiale

La résolution d'équations différentielles du premier ordre peut s'effectuer avec EES suivant plusieurs méthodes. Le chapitre 5 décrit une fonction de la bibliothèque située dans le répertoire USERLIB qui utilise l'algorithme Runge-Kutta. Cette méthode est utilisable seulement si la dérivée peut être exprimée explicitement comme une fonction de variables dépendantes et indépendantes.

Le paragraphe suivant expose deux façons de résoudre simultanément des équations algébriques et différentielles en utilisant la fonction **Intégral** et la fonction **TableValue** utilisée conjointement avec la table paramétrique.

Méthode 1 : Résolution d'équations différentielles avec la Fonction INTEGRALE

Considérons le problème suivant : étudions la variation de température d'une sphère de rayon $r=5\text{mm}$ au cours du temps (initialement la température de la sphère est de 400°C). La sphère est exposée à l'air à une température de 20°C avec un coefficient de convection $h=10\text{ W/m}^2\text{-K}$. Les propriétés de la sphère sont les suivantes :

ρ = densité = 3000 kg/m^3

k = conductivité thermique = 20 W/m-K

c = chaleur spécifique = 1000 J/le kg-K

Le calcul du nombre de Biot indique que la sphère peut être considérée comme un système avec une conductivité thermique fort élevée. On peut donc supposer que sa température est uniforme au cours du temps.¹³ La relation entre la température de sphère et le temps est donnée par l'équilibre énergétique de la sphère. Ce qui se traduit par l'équation différentielle suivante :

$$-h A (T - T_\infty) = \rho c V \frac{dT}{dt}$$

où

h : le coefficient de transfert de chaleur

T : la température de la sphère (uniforme au cours du temps)

T_∞ : la température du faisceau d'air = 20°C

Un : l'air de la sphère = $4 \pi r^2$

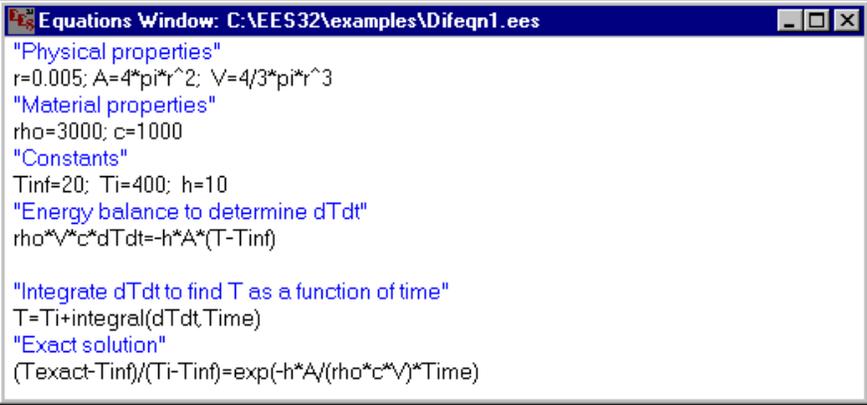
V : le volume de la sphère = $4/3 \pi r^3$

t : le temps

La solution analytique de cette équation est la suivante :

$$\frac{T - T_\infty}{T_i - T_\infty} = \exp\left(\frac{-h A}{\rho c V} t\right)$$

Pour résoudre l'équation différentielle numériquement dans EES, entrer les équations suivantes :



```

Equations Window: C:\EES32\examples\Difeqn1.ees
"Physical properties"
r=0.005; A=4*pi*r^2; V=4/3*pi*r^3
"Material properties"
rho=3000; c=1000
"Constants"
Tinf=20; Ti=400; h=10
"Energy balance to determine dTdt"
rho*V*c*dTdt=-h*A*(T-Tinf)

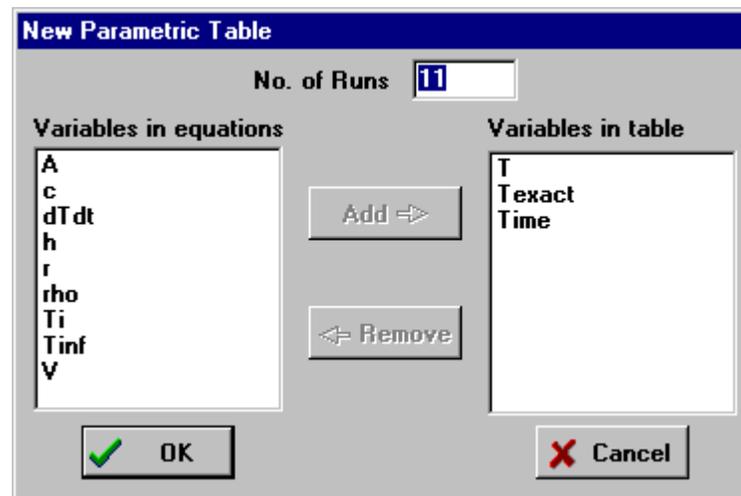
"Integrate dTdt to find T as a function of time"
T=Ti+integral(dTdt,Time)
"Exact solution"
(Texact-Tinf)/(Ti-Tinf)=exp(-h*A/(rho*c*V)*Time)
    
```

Ensuite, créer une table paramétrique à l'aide de la commande **New Table** du menu **Parametrics**. (Noter que si les troisième et quatrième arguments correspondant au début et à la fin du pas

¹³ Incropera, le F.P. et DeWitt, le D.P., *Les principes de base de Transfert de Chaleur et Masse*, deuxième édition, John Wiley et Fils, 1985, Chapitre 5

d'intégration sont fournis, l'usage de la table paramétrique n'est pas nécessaire. Cependant, les résultats intermédiaires ne seraient pas visualisables dans le tracé.

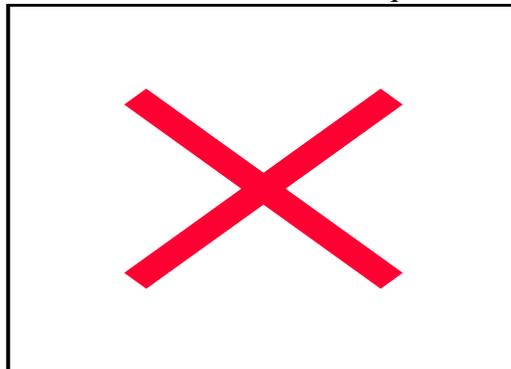
Sélectionner T, Time, et Texact comme les trois variables à inclure dans la table paramétrique. Onze itérations, permettront d'avoir un historique de température de 1000 secondes en commençant par 0 avec 100 secondes d'intervalles. La boîte de dialogue **New Table** est montrée ci dessous :



Cliquer sur le bouton 'OK'. Ensuite, il est nécessaire d'entrer les valeurs de la variable Time dans la table pour laquelle la température T sera calculée.

Un pas de 100 seconds a été choisi. Avec un pas fixe, les valeurs de la variable Time peuvent être saisies simplement en cliquant sur le bouton . Entrer la valeur 0 dans la case First Value (valeur initiale). Sélectionner le paramètre 'Last Value' dans la liste déroulante située au-dessous du label First Value. Entrer alors la valeur 1000. (voir figure ci-dessous).

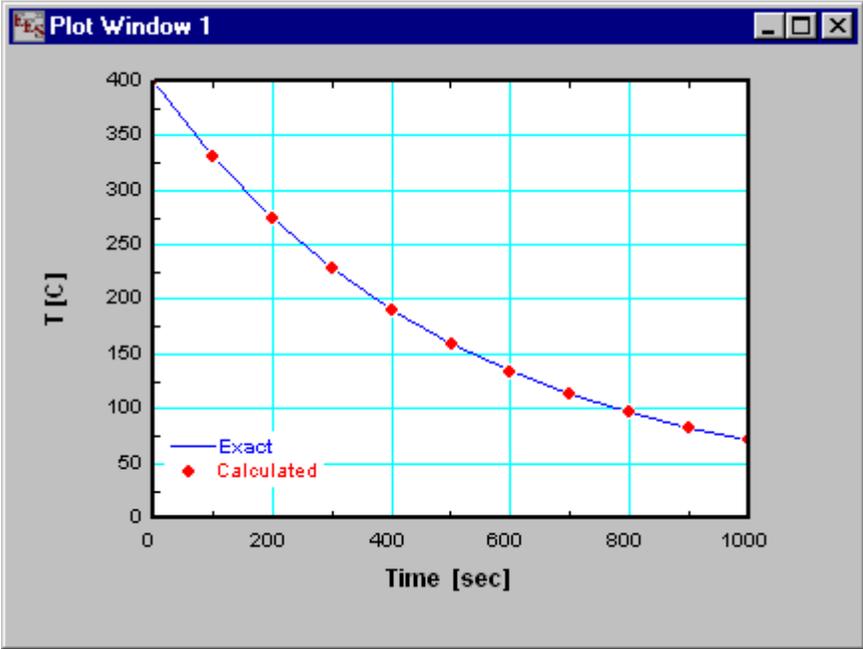
Les valeurs comprises entre 0 et 1000 seront automatiquement entrées dans la table en cliquant



sur le bouton 'OK'. A présent, sélectionner la commande **Solve Table** du menu **Calculate** déterminer les valeurs numériques et analytiques de la température T correspondant à chaque pas de temps. Une fois les calculs réalisés, la fenêtre Parametric Table de résultats sera affichée.

Les valeurs calculées apparaissent en gras dans une des colonnes. Le diagramme affiché à l'écran représente le tracé des valeurs numériques de la température en fonction du temps, et ce, proche de la solution analytique.

	1	2	3
	T [C]	Texact [C]	Time [sec]
Run 1	400.00	400.00	0
Run 2	330.91	331.12	100
Run 3	274.38	274.72	200
Run 4	228.13	228.55	300
Run 5	190.29	190.75	400
Run 6	159.33	159.79	500
Run 7	133.99	134.45	600
Run 8	113.27	113.71	700
Run 9	96.31	96.72	800
Run 10	82.44	82.81	900
Run 11	71.08	71.43	1000



Méthode 2 : Résolution d'équations différentielles avec la Fonction TableValue

Le problème posé ici est identique au précédent. L'idée est de résoudre l'équation différentielle suivante :

$$-h A (T - T_{\infty}) = \rho c V \frac{dT}{dt}$$

qui peut s'écrire approximativement de la façon suivante :

$$\frac{dT}{dt} \approx \frac{T^{new} - T^{old}}{\Delta t}$$

T^{new} est la température à calculer au temps présent. T^{old} est la température au temps précédent (T-1) qui peut être trouvée grâce à la fonction **Table Value**. La fonction **TableValue** retourne la valeur d'une ligne et d'une colonne spécifiée de la Table Paramétrique, comme décrit dans le Chapitre 4. Il est ainsi possible, à condition de spécifier le numéro de ligne et de colonne, d'accéder à des valeurs de variables calculées précédemment pendant la résolution de la table. ΔT est le pas de temps qui, dans les équations montrées ci-dessous, est la différence T^{new} et T^{old} .

Deux méthodes vont être utilisées pour résoudre l'équation différentielle et pour comparer le résultat à la solution exacte.

- Une méthode explicite (méthode d'Euler) pour laquelle seulement les températures précédentes sont utilisées pour évaluer la partie droite de l'équation.
- Une méthode implicite (méthode Crank-Nicolson) pour laquelle la moyenne entre la température actuelle et les températures précédentes est utilisée.

La méthode de Crank-Nicolson est implicite car la température actuelle n'est pas encore évaluée. Noter que ce type de méthode n'est pas difficile à implémenter car EES a été conçu pour résoudre des équations implicites. Les équations utilisées pour résoudre ce problème sont visibles plus loin dans ce paragraphe.

La plupart des équations utilisées dans la Méthode 1 sont ici réutilisées. T_Euler est la température calculée par la méthode Euler. T_CN est la température calculée par la méthode de Crank-Nicolson. (Dans les fenêtres d'Equations et de solution mise en forme, ces variables apparaîtront respectivement comme T_{Euler} et T_{CN}).

Pour effectuer les calculs, il convient de créer une table paramétrique à l'instar de celles de la méthode 1. T_Euler, T_CN et T_exact figurant dans la première ligne de la table (correspondant au temps 0) sont les conditions initiales et leurs valeurs (400 °C) doivent être saisies. Ensuite, exécuter la commande **Solve Table** pour résoudre le problème (noter que les calculs ne commencent qu'à partir de la deuxième itération). Remarquer, l'utilisation de la fonction TableValue permettant de retourner d'une valeur de la table paramétrique en spécifiant sa position (ligne, colonne).

```

Equations Window: C:\EES32\examples\Difeqn2.ees
"Physical properties"
r=0.005; A=4*pi*r^2; V=4/3*pi*r^3
"Material properties"
rho=3000; c=1000
"Constants"
Tinf=20 {C}; Ti=400 {C}; h=10 {W/m2-K}; delta=100 {s}
"Finite difference energy balance"
"Euler Method"
T_Euler_old=tablevalue(TableRun#1,#T_Euler) "retrieves previous T_Euler"
rho*V*c*(T_Euler-T_Euler_old)/delta=-h*A*(T_Euler_old-Tinf)
"Crank-Nicolson Method"
T_CN_old=tablevalue(TableRun#1,#T_CN) "retrieves previous T_CN"
rho*V*c*(T_CN-T_CN_old)/delta=-h*A*((T_CN_old+T_CN)/2-Tinf)
"Exact solution"
(T_exact-Tinf)/(Ti-Tinf)=exp(-h*A/(rho*c*V)*Time)

```

Les résultats numériques du problème posé sont montrés ci-dessous (les valeurs calculées sont affichées en italique). Il est évident que la méthode d'Euler ne donne pas des résultats aussi précis que ceux obtenus par la fonction Integral (méthode 1) ou par le biais de la méthode Crank-Nicolson. Il serait possible d'améliorer le degré de précision en réduisant le pas de temps, mais ceci entraînerait un temps de calcul beaucoup plus long.

	1 Time [sec]	2 <i>T_{Euler}</i> [C]	3 <i>T_{CN}</i> [C]	4 <i>T_{exact}</i> [C]
Run 1	0	400.0	400.0	400.0
Run 2	100	324.0	330.9	331.1
Run 3	200	263.2	274.4	274.7
Run 4	300	214.6	228.1	228.5
Run 5	400	175.6	190.3	190.7
Run 6	500	144.5	159.3	159.8
Run 7	600	119.6	134.0	134.5
Run 8	700	99.7	113.3	113.7
Run 9	800	83.8	96.3	96.7
Run 10	900	71.0	82.4	82.8
Run 11	1000	60.8	71.1	71.4

Résolution d'équations du 2nd ordre et supérieur

Il est possible de résoudre des équations de degré supérieur à 1, en utilisant la fonction Integral. Le lecteur trouvera ci-dessous, un exemple de résolution d'équation différentielle du 2nd ordre dans le but de déterminer la vitesse et la position d'un objet qui tombe librement est montré ci-dessous

```

Equations Window: C:\EES32\examples\drag.ees
"This program demonstrates the use of the Integral function. Here it is used to
calculate the velocity and position of a freely falling object, subject to aerodynamic
drag."

F=M*g*Convert(lbm-ft/s^2, lbf) "Newton's Law"
M*a*Convert(lbm-ft/s^2, lbf)=F-F_d "force balance"
F_d=C_d*(1/2*rho*v^2)*Convert(lbm-ft/s^2, lbf) "definition of drag coefficient"
C_d=0.2
M=1.0 "mass of object"
rho=density(Air, T=70, P=14.7)
g=32.17 "ft/s^2"

v=0+integral(a, t, 0, 5) "velocity after 5 seconds"
z=0+integral(v, t, 0, 5) "vertical position after 5 seconds"

```

```

Solution
Unit Settings: [F]/[psia]/[lbm]/[degrees]
a = 0.9194 [ft/s^2]      C_d = 0.2              F = 0.9999 [lbf]
F_d = 0.9713 [lbf]      g = 32.17 [ft/s^2]     M = 1 [lbm]
rho = 0.07488 [lbm/ft^3] t = 5 [s]              v = 64.6 [ft/s]
z = 236.2 [ft]

```

Intégration de variables multiples

L'intégration de variables multiples est possible en imbriquant les variables au sein de la fonction **Integral**, (6 niveaux d'imbrication sont possibles), par exemple :

```

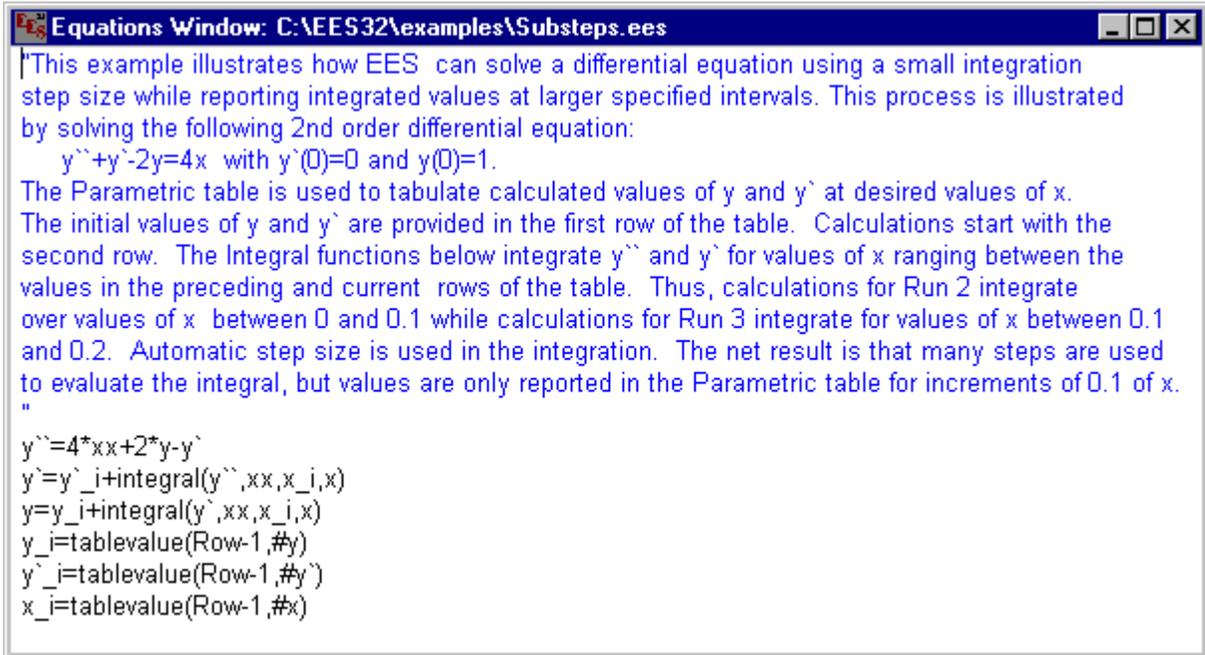
Equations Window: C:\EES32\examples\Dbl_intg.ees
F=integral(integral(xy, y, 0, x), x, 0, 3, 0.06)
xy=x^3*y^2

```

Formatted Equations	Solution
$F = \int_0^3 \left[\int_0^x (xy) dy \right] dx$ $xy = x^3 \cdot y^2$	F = 104.3 x = 3 xy = 243 y = 3

Usage combiné de la fonction Integral basée sur une équation et une table paramétrique

L'avantage d'utiliser une table paramétrique couplée avec la fonction Integral est qu'il est possible de visualiser les valeurs de calculs intermédiaires et ainsi observer l'évolution des calculs. Un des inconvénients est que la variable à intégrer doit apparaître dans la table pour chaque itération de calcul. Il est néanmoins possible de combiner l'utilisation de la table et de l'équation, et ainsi permettre de visualiser directement le résultat final comme dans l'exemple ci-dessous :



Equations Window: C:\EES32\examples\Substeps.ees

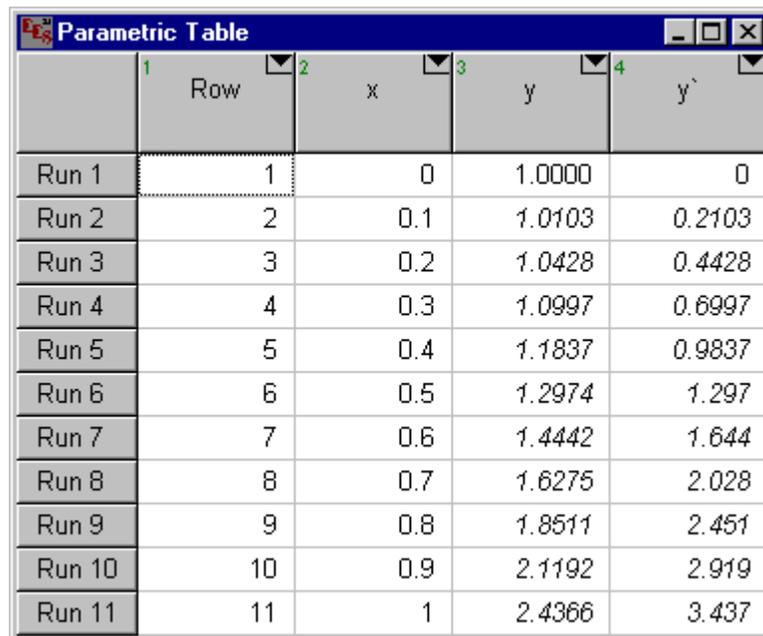
This example illustrates how EES can solve a differential equation using a small integration step size while reporting integrated values at larger specified intervals. This process is illustrated by solving the following 2nd order differential equation:

$$y'' + y' - 2y = 4x \quad \text{with } y(0) = 0 \text{ and } y'(0) = 1.$$

The Parametric table is used to tabulate calculated values of y and y' at desired values of x . The initial values of y and y' are provided in the first row of the table. Calculations start with the second row. The Integral functions below integrate y'' and y' for values of x ranging between the values in the preceding and current rows of the table. Thus, calculations for Run 2 integrate over values of x between 0 and 0.1 while calculations for Run 3 integrate for values of x between 0.1 and 0.2. Automatic step size is used in the integration. The net result is that many steps are used to evaluate the integral, but values are only reported in the Parametric table for increments of 0.1 of x .

"

```
y''=4*x+2*y-y'  
y'=y'_i+integral(y'',xx,x_i,x)  
y=y_i+integral(y',xx,x_i,x)  
y_i=tablevalue(Row-1,#y)  
y'_i=tablevalue(Row-1,#y')  
x_i=tablevalue(Row-1,#x)
```



	1 Row	2 x	3 y	4 y'
Run 1	1	0	1.0000	0
Run 2	2	0.1	1.0103	0.2103
Run 3	3	0.2	1.0428	0.4428
Run 4	4	0.3	1.0997	0.6997
Run 5	5	0.4	1.1837	0.9837
Run 6	6	0.5	1.2974	1.297
Run 7	7	0.6	1.4442	1.644
Run 8	8	0.7	1.6275	2.028
Run 9	9	0.8	1.8511	2.451
Run 10	10	0.9	2.1192	2.919
Run 11	11	1	2.4366	3.437

Annexe A

Conseils d'utilisation

1. La commande 'Variable Information' permet de visualiser par ordre alphabétique l'ensemble des variables présentes dans la fenêtre d'Equations. Il est ainsi possible de vérifier la syntaxe de chacune des variables.
2. La fenêtre Residuelle fournit une indication sur la précision de résolution de chacune des équations et leur ordre de résolution. L'examen de ces indicateurs permet de voir quelles équations n'ont pu être résolues.
3. Si vos équations ne sont pas convergentes, il est possible que les valeurs par défaut n'ont pas été ajustées correctement. Dans ce cas, le problème peut être fréquemment résolu en modifiant la valeur par défaut de certaines inconnues ou en modifiant les équations de telle façon que le nombre d'inconnues est égal au nombre d'équations. Si vous obtenez une solution, utiliser la commande 'Update Guesses' du menu 'Calculate' pour modifier la valeur par défaut de l'ensemble des variables à leur valeur courante. Ensuite, retourner dans la fenêtre d'Equations et relancer les calculs.
4. Dans le cas où EES est incapable de résoudre un système non linéaire d'équations, essayer d'échanger certaines variables dépendantes et indépendantes pour produire un système d'équations plus facile à résoudre. Par exemple, EES est incapable de résoudre les équations suivantes à partir des valeurs limites et initiales.

$$\text{Eff} = 0.9$$

$$C_{\text{max}} = 432$$

$$C_{\text{min}} = 251$$

$$\text{eff} = (1 - \exp(-\text{NTU} * (1 - (C_{\text{min}}/C_{\text{max}})))) / (1 - (\exp(-\text{NTU} * (1 - (C_{\text{min}}/C_{\text{max}}))))))$$

Ceci étant, le problème peut se résoudre aisément en spécifiant la valeur de NTU à la place de Eff :

$$\text{NTU} = 5$$

$$C_{\text{max}} = 432$$

$$C_{\text{min}} = 251$$

$$\text{eff} = (1 - \exp(-\text{NTU} * (1 - (C_{\text{min}}/C_{\text{max}})))) / (1 - (\exp(-\text{NTU} * (1 - (C_{\text{min}}/C_{\text{max}}))))))$$

Quelques essais indiqueront que la valeur de NTU doit être comprise entre 3 et 5 pour une valeur de $Eff = 0.9$. Modifier la valeur par défaut de NTU à 4, permettant à EES de déterminer rapidement la valeur de la solution finale à 3.729.

5. Une façon sûre et efficace pour résoudre des problèmes complexes avec EES est d'ajouter une variable supplémentaire pour que le problème ait un degré de liberté supplémentaire. Ainsi, utiliser la table paramétrique pour faire varier la valeur d'une ou de plusieurs variable(s) implicite(s) dans le but de trouver une solution de valeur 0. Par exemple, considérons le calcul de rayonnement suivant dans lequel la valeur de T doit être déterminée. Les trois premières équations doivent être résolues simultanément. Ceci étant, celles-ci sont non linéaires, car T est élevé à la puissance 4. EES va rencontrer des difficultés pour déterminer la solution finale, en fonction des valeurs initiales :

$$\begin{aligned}
 QL &= AL * \text{Sigma} * (T^4 - TL^4) \\
 QB &= AH * \text{Sigma} * (TH^4 - T^4) \\
 QL &= QB \\
 \text{Sigma} &= 0.1718E-8 \\
 AL &= .5; AH=1; LE TL=300; LE TH=1000
 \end{aligned}$$

Comme alternative, ajouter une variable Delta comme tel :

$$\begin{aligned}
 QL &= AL * \text{Sigma} * (T^4 - TL^4) \\
 QB &= AH * \text{Sigma} * (TH^4 - T^4) + \text{Delta} \\
 QL &= QB \\
 \text{Sigma} &= 0.1718E-8 \\
 AL &= .5; AH=1; LE TL=300; LE TH=1000
 \end{aligned}$$

Maintenant, créer une table paramétrique contenant les variables T et Delta. Utiliser la commande **Alter Value** pour modifier la plage de valeurs que peut prendre la variable T. Puis lancer la résolution de la table pour calculer les valeurs correspondantes de Delta. La (es) valeur(s) de T pour laquelle (lesquelles) la valeur de Delta est nulle constitue un solution de notre système d'équations. La fenêtre **New Plot** (nouveau tracé) est commode pour visualiser la relation entre T et Delta. Si Delta ne prend jamais la valeur 0, cela veut dire que notre système n'a pas de solution pour une plage de valeurs de T donnée. Ceci est sans doute la méthode la plus appropriée pour résoudre des systèmes d'équations non linéaires.

6. Le bouton 'Store' (enregistrer) de la boîte de dialogue **Info implicite** peut être particulièrement commode si vous avez l'habitude de nommer vos variables d'une certaine façon. Par exemple, si les variables commençant par le lettre T désignent des variables de température :

ajuster les valeurs limites, le format d'unités et d'affichage pour la lettre T et enregistrer en cliquant sur le bouton 'Store' pour sauver cette configuration.

7. Les touches 'flèches' facilite la navigation au sein des fenêtres d'Equations de table 'Lookup' et paramétrique. Dans la fenêtre d'Equations, les flèches haut et bas déplace le curseur au début de la ligne. Les flèches droite et gauche permettent de déplacer le curseur d'un caractère. Les touches Debut et Fin du clavier permettent d'atteindre le début ou la fin de la ligne sur laquelle se trouve le curseur. Dans les tables les touches de flèches permettent se déplacer de cellule en cellule. La touche 'Entrer' et la touche de tabulation produisent respectivement le même effet que les flèches bas et droite.
8. Utiliser la touche de tabulation dans la fenêtre d'Equations pour indenter les équations afin d'améliorer la lisibilité de votre document..
9. A l'exception de la substance Steam_NBS, les corrélations de propriété de EES ne sont pas applicables dans la plage de compression d'une fluide (sub_cooled). Il est supposé que le liquide de subcooled est incompressible et que les propriétés sont issues de ce cas ci pour le liquide saturé. Ainsi, dans la région de subcooled, $v(T,P)=v(T,Passis)$, $u(T,P)=u(T,Passis)$ et $s(T,P)=s(T,Passis)$. Pour calculer le travail idéal d'une pompe, par exemple, le rappel que $h_2-h_1 = -W_{\text{pump}} = \int V dP = v*(P_2-P_1)$, puisque pour une substance de incompressible, v est indépendant de P .
10. La fenêtre Arrays (tableaux) peut être tout à fait utile pour organiser les informations de propriété dans un problème de thermodynamique comportant des états multiples. Utiliser les variables d'ensemble, tel que $T[1]$, $P[1]$, et $h[1]$ (plutôt que T_1 , P_1 , et h_1) pour les propriétés afférentes à chaque état. Ainsi, les propriétés apparaîtront dans une table plutôt que mélangées dans la fenêtre de Solution. (vérifier que l'option Use Array (utilisation des tableaux) a été cochée dans la boîte de dialogue Display Options).
11. Un effort considérable à été fourni pour éviter que EES ne 'plante' en certaines circonstances. Néanmoins, ceci peut toujours arriver. Dans ce cas, EES sauvera votre travail dans un fichier EESERROR. Ainsi, si la résolution d'un calcul a provoqué l'arrêt soudain de EES, il vous sera possible de récupérer votre travail à partir de ce fichier.
12. Utiliser la directive \$INCLUDE pour charger les constantes, les formats d'unités ou toute autre équation dans la fenêtre d'Equations . Vous pouvez également charger des fichiers de librairie en utilisant la directive \$INCLUDE.
13. Si vous écrivez une fonction destinée a être rangée dans la bibliothèque de EES, qui appelle n'importe quelles fonctions thermodynamiques ou trigonométriques, utiliser la commande

'UnitSystem' pour déterminer les unités utilisées. Ainsi, vous pouvez utiliser les déclarations If Then Else pour vérifier que les paramètres passés à votre fonction sont dans le bon système d'unités.

14. Utiliser l'option de couleurs d'arrière-plan de colonnes d'une Table de Paramétrique si vous désirez que d'autres utilisateurs saisissent des valeurs dans ces colonnes.

15. Si vous travaillez en Mode Complexe, écrire la directive \$COMPLEXE en haut de la fenêtre d'Equations. (ce qui est plus commode que de modifier le mode de calcul à partir de la boîte de dialogue de préférences.

Méthodes numériques utilisées dans EES

EES utilise une variante de la méthode de Newton [1-4] pour résoudre des systèmes d'équations algébriques non linéaires. Le Jacobien nécessaire dans la méthode de Newton est numériquement évalué à chaque itération. Les techniques de matrices éparées [5-7] sont employées pour améliorer l'efficacité de calcul et ainsi résoudre des problèmes complexes dans la limite de la mémoire d'un micro-ordinateur. Les propriétés d'efficacité et de convergence sont également améliorées par le changement de la taille de pas de calcul et par l'implémentation de l'algorithme de Tarjan [8] (cet algorithme permet de segmenter un problème en plusieurs ensembles). Enfin, plusieurs algorithmes sont utilisés pour déterminer la valeur minimum ou maximum d'une variable [9,10]. Le lecteur trouvera ci-dessous un résumé des méthodes employées par EES :

Solution d'équations algébriques

Considérons l'équation suivante à une inconnue :

$$x^3 - 3.5 x^2 + 2 x = 10$$

Pour appliquer la méthode de Newton à la solution de cette équation, il faudrait écrire cette équation en utilisant la notion d'approximation :

$$\varepsilon = x^3 - 3.5 x^2 + 2 x - 10$$

La fonction décrite par cette équation est représentée dans la figure 1. Il y existe une seule solution réelle (c.-à-d., la valeur de x pour laquelle $\varepsilon = 0$) pour $x = 3.69193$.

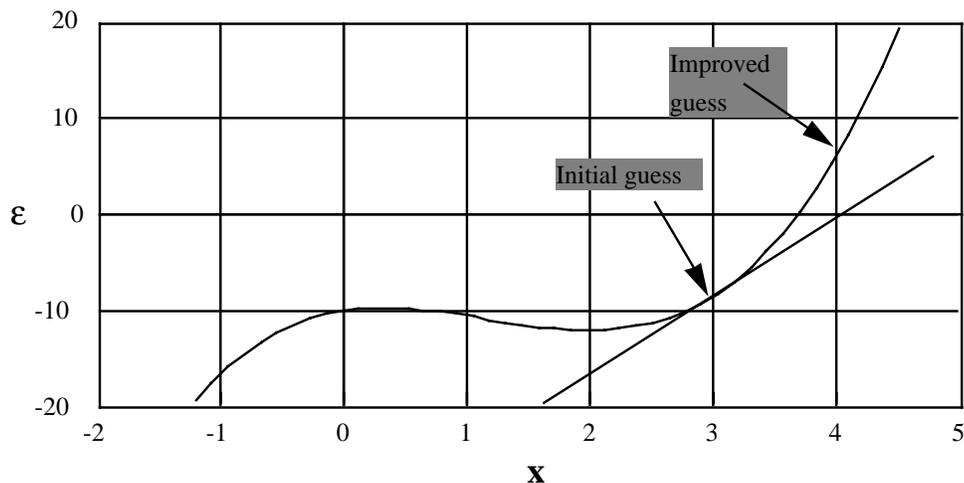


Figure 1: Résiduel de $x^3 - 3.5 x^2 + 2 x = 10$ en tant que fonction de x

La méthode de Newton exige une estimation de la dérivée totale du résiduel, J . Pour cette équation, la dérivée est:

$$J = \frac{d\varepsilon}{dx} = 3x^2 - 7x + 2$$

Pour résoudre l'équation, les étapes de la méthode de Newton sont les suivantes :

1. Une supposition initiale est faite pour la valeur de x (ex : 3).
2. La valeur de ε est évaluée en utilisant la valeur de supposition de x . Pour $x = 3$, $\varepsilon = -8.5$.
3. La dérivée J est évaluée. Pour $x = 3$, $J = 8$.
4. Le changement de la valeur de supposition de x , c'est-à-dire Δx est calculé en résolvant $J \Delta x = \varepsilon$. Dans cet exemple, Δx égal -1.0625 .
5. Une valeur plus proche de la solution est en générale obtenue en calculant : $x - \Delta x$. Dans l'exemple, la valeur de x est 4.0625 ($\varepsilon = 7.4084$).

Les étapes 2 à 5 sont répétées jusqu'à ce que la valeur absolue de ε ou Δx soit la plus petite suivant les critères de tolérance spécifiés dans la boîte de dialogue **Stop Criteria** (critères d'arrêt). La méthode, quand elle converge, converge tout à fait rapidement. Cependant, une mauvaise supposition initiale peut causer la divergence ou une convergence très lente. Essayer, par exemple, une supposition initiale de 2.

La méthode de Newton peut être étendue pour résoudre simultanément des équations non linéaires. Dans ce cas, le concept de "dérivée" est généralisé au concept de "la matrice de Jacobien." Considérons les deux équations suivantes comportant deux inconnues:

$$x_1^2 + x_2^2 - 18 = 0$$

$$x_1 - x_2 = 0$$

Les équations peuvent être réécrites de la manière suivante :

$$\varepsilon_1 = x_1^2 + x_2^2 - 18 = 0$$

$$\varepsilon_2 = x_1 - x_2 = 0$$

Le Jacobien pour cette matrice est une matrice 2×2 . La première ligne contient les dérivées de la première équation par rapport à chaque variable. Dans l'exemple ci-dessus, la dérivée de ε_1 par rapport à x_2 est $2x_2$. La matrice de Jacobien pour cet exemple est:

$$J = \begin{bmatrix} 2 \cdot x_1 & 2 \cdot x_2 \\ 1 & -1 \end{bmatrix}$$

La méthode de Newton, comme expliquée, ci-dessus s'applique à des systèmes linéaires et non linéaires d'équations. Si les équations sont linéaires, la convergence est assurée lors de la première itération, même si une "mauvaise" supposition initiale a été faite. Les équations non linéaires exigent des calculs itératifs. Considérons la supposition d'initiale suivante:

$$x = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$$

$$\varepsilon = \begin{bmatrix} -10 \\ 0 \end{bmatrix}$$

$$J = \begin{bmatrix} 4 & 4 \\ 1 & -1 \end{bmatrix}$$

Les valeurs de ε et J pour cette supposition initiale sont:

Les valeurs plus proches de la solutions pour le vecteur x sont obtenues en résolvant la matrice suivante utilisant le Jacobien et le vecteur résiduel.

$$\begin{bmatrix} 4 & 4 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \end{bmatrix} = \begin{bmatrix} -10 \\ 0 \end{bmatrix}$$

La résolution de ces équations donne :

$$\begin{bmatrix} \Delta x_1 \\ \Delta x_2 \end{bmatrix} = \begin{bmatrix} -1.25 \\ -1.25 \end{bmatrix}$$

Les estimations plus précises de x_1 et x_2 sont obtenues en soustrayant respectivement Δx_1 et Δx_2 à la valeur de supposition déterminée précédemment :

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 3.25 \\ 3.25 \end{bmatrix}$$

La solution exacte à ce problème est $x_1 = x_2 = 3.0$. Les valeurs calculées de x_1 et x_2 sont plus proches de la solution exacte que les valeurs de supposition. Les calculs sont répétés jusqu'à l'obtention d'une solution convergente.

La matrice du Jacobien joue un rôle primordial dans la résolution d'équations algébriques. La matrice du Jacobien peut être obtenue symboliquement ou numériquement. L'évaluation symbolique du Jacobien est plus précise, mais exige un traitement plus long. La précision du Jacobien, cependant, ne mène pas nécessairement à plus de précision dans la solution finale, seulement (parfois) à moins d'itérations. EES évalue le Jacobien numériquement et effectue tous les calculs sur 96 bits (20 décimales) ce qui permet d'avoir des résultats convergents sans trop de perte de précision.

Dans la plupart des systèmes d'équations, beaucoup d'éléments de la matrice sont nuls. Une matrice contenant beaucoup de zéro est appelée une matrice éparsée. Des techniques spéciales permettent de composer de manière efficace avec ce type de matrice. En fait, sans ces techniques, le nombre d'équations simultanées qui devraient être résolues par EES serait approximativement égal à 6000 (limite actuelle de EES). (Pour plus d'information sur la manipulation de matrices éparsées, voir [5, 6] ; un ensemble de routines conçues pour contrôler de très grandes matrices éparsées sont décrites dans [7]).

La méthode de Newton ne fonctionne pas toujours, surtout si la supposition initiale du vecteur X est « mauvaise ». La solution obtenue après avoir appliqué la correction ΔX au vecteur x précédent doit se rapprocher d'avantage de la solution finale (c'est-à-dire, un résultat inférieur au maximum résiduel). EES vérifie toujours cette condition. Si celle-ci n'est pas vérifiée, EES divise en deux le pas ΔX et réévalue le résiduel. Si ceci n'améliore pas la solution, cette étape dichotomique est répétée (jusqu'à 20 fois). Si la solution obtenue n'est pas satisfaisante, EES réévaluera le Jacobien et répétera le processus précédent jusqu'à ce que l'un des critères d'arrêt forcent EES à stopper les calculs. Procéder par dichotomie est très utile surtout si une mauvaise supposition initiale a été faite. La figure 2 illustre le processus de résolution de l'équation du premier exemple. La supposition initiale est $x=2.5$. Dans ce cas, la méthode dichotomique fonctionne tout à fait correctement.

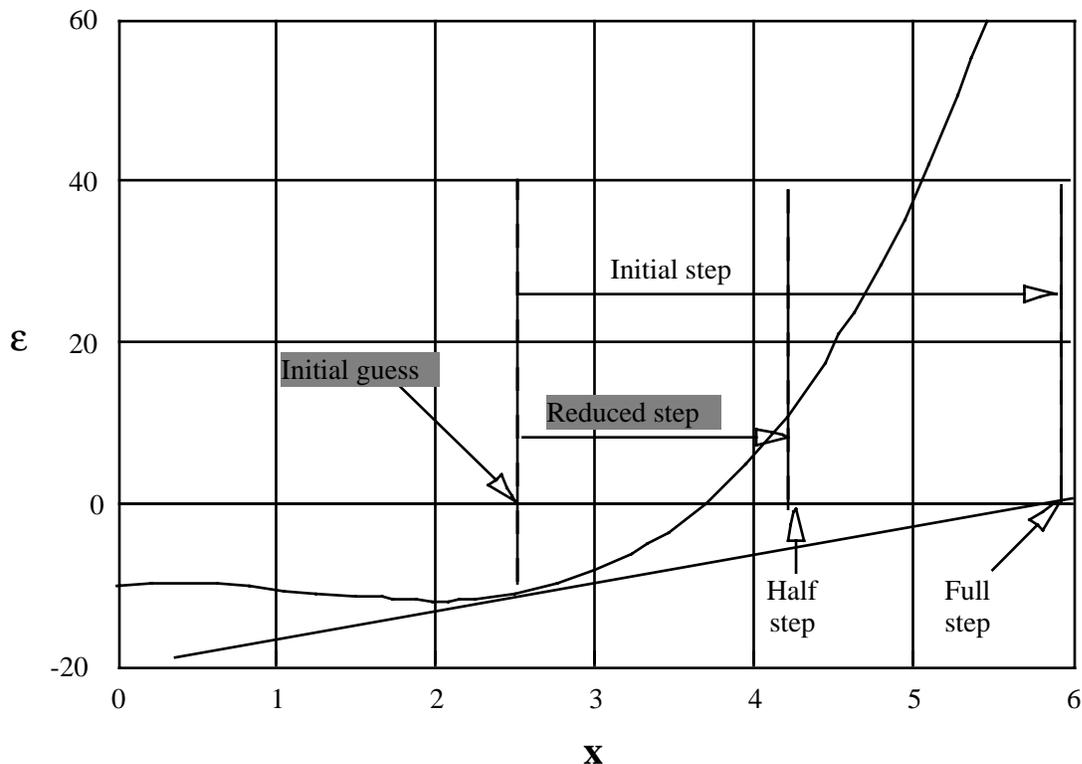


Figure 2: Utilisation de la dichotomie pour améliorer la convergence

Groupement d'équations dans un système

Bien que vous puissiez avoir un système d'équations à résoudre simultanément, il est souvent possible de grouper et de résoudre celles-ci plutôt que le système tout entier. La résolution par groupe d'équations accroît significativement la fiabilité de la méthode de Newton. Pour cette raison, EES organise les équations en groupes (ou blocs) avant de résoudre un système. Considérons, par exemple, le système d'équations suivant:

$$\begin{aligned}x_1 + 2x_2 + 3x_3 &= 11 \\5x_3 &= 10 \\3x_2 + 2x_3 &= 7\end{aligned}$$

Ces équations peuvent être résolues simultanément. Cependant, il est plus facile de résoudre le système si ses équations sont réorganisées et groupées. EES reconnaît automatiquement que la valeur de x_3 peut être déterminée à partir de l'équation 2. Ainsi, la valeur de x_2 peut être trouvée dans l'équation 3. Enfin, à partir des valeurs trouvées précédemment, l'équation 1 nous donne la valeur de x_1 . Ainsi, le système initial a été décomposé en trois équations contenant une variable pour lesquelles la solution est évidente. Le cas présent est trivial. Ceci étant, les choses peuvent devenir beaucoup plus intéressantes si les blocs sont moins évident à cerner. Considérons l'exemple suivant avec 8 équations linéaires et 8 inconnues:

$$\begin{array}{rcccccccl} & & x_3 & & & & + x_8 & = & 11 \\ & & & & & & & & \\ & & & & & & x_7 & = & 7 \\ & & & & x_5 & - x_6 & - x_7 & = & -8 \\ x_1 & & & & & & - x_6 & = & -1 \\ & & & + x_4 & & & & & \\ & & x_2 & & & & + x_8 & = & 10 \\ & & x_3 & & - x_5 & & + x_8 & = & 6 \\ & & & x_4 & & & & = & 4 \\ x_1 & & & & & + x_6 & + x_7 & = & 14\end{array}$$

Ces équations peuvent être réorganisées et groupées. Chaque bloc est résolu séparément. Dans le cas présent, le système est découpé en 6 blocs :

Groupe 1: équation 7

$$x_4 = 4$$

Groupe 2: équation 2

$$x_7 = 7$$

Groupe 3: équations 4 et 8

$$x_1 + x_4 - x_6 = -1$$

$$\text{D'où} \quad x_1 = 1$$

$$x_1 + x_6 + x_7 = 14$$

$$\text{et:} \quad x_6 = 6$$

Groupe 4: équation 3

$$x_5 - x_6 - x_7 = -8$$

$$\text{D'où:} \quad x_5 = 5$$

Groupe 5: équations 1 et 6

$$x_3 + x_8 = 11 \quad \text{D'où:}$$

$$x_3 = 3$$

$$x_3 - x_5 + x_8 = 6$$

$$\text{et:} \quad x_8 = 8$$

Groupe 6: équation 5:

$$x_2 + x_8 = 10 \quad \text{D'où:}$$

$$x_2 = 2$$

Le premier des deux blocs (ou groupes) contient une équation avec une seule variable. Ces blocs définissent des constantes. EES reconnaît que les équations qui dépendent d'une variable seule sont en réalité des paramètres ou des constantes. Ces paramètres sont déterminés avant n'importe quelle solution d'équations. Les valeurs par défaut, ou valeurs limites ne sont pas utilisées puisque les valeurs de ces paramètres sont immédiatement déterminées. La solution des équations restante est maintenant très simple, bien que cela semblait fort compliqué initialement.

Ce mécanisme de groupement d'équations est utile quand les équations sont linéaires, mais il n'est pas essentiel. Quand les équations sont non linéaires, il est quasi indispensable. En effet, sans l'utilisation de ce mécanisme, le processus de résolution devient itératif et s'effectue à partir de valeurs erronées. Le résultat obtenu est souvent divergent. EES peut reconnaître des groupes d'équations avant la solution en créant et en inspectant la matrice du Jacobien selon l'algorithme de Tarjan [8] (voir la référence [6] pour plus de détails sur cet algorithme).

Détermination de la valeur minimum ou maximum d'une variable

EES a la capacité de trouver le minimum ou le maximum de la valeur d'une variable quand il y a entre 1 et 10 degrés de liberté (c'est-à-dire, le nombre de variables moins le nombre d'équations). Pour les problèmes avec un seul degré de liberté, EES utilise un des deux algorithmes fondamentaux de détermination du minimum ou maximum : une approximation quadratique récursive connue sous le nom de méthode de Brent ou de la méthode Golden Section search [9]. L'utilisateur spécifie la méthode, la variable à optimiser et une variable indépendante dont la valeur sera comprise entre les bornes basses et hautes à spécifier. Pour des problèmes à plusieurs degrés de liberté, EES utilise la méthode de Brent à maintes reprises pour déterminer le minimum ou le maximum suivant un cheminement particulier. Ce cheminement est déterminé par un algorithme de recherche direct connu sous le nom de méthode de Powell ou par la méthode dite de conjugate gradient method¹⁴ [9,10].

L'algorithme d'approximation quadratique récursive détermine la valeur de la variable qui doit être optimisée selon trois valeurs différentes de la variable indépendante. Une fonction quadratique est ajustée à partir de ces trois points. Ensuite, cette fonction est analytiquement différenciée pour localiser une estimation du point extremum. Si le rapport entre la variable à optimiser et la variable indépendante est véritablement quadratique, l'optimum est directement trouvé. Dans le cas contraire, l'algorithme utilisera l'estimation de l'extremum récemment obtenue et deux (des trois) points proches afin de répéter l'ajustement de la fonction. Ce

¹⁴ Recherche de la Section Dorée

procédé est répété jusqu'à ce que les critères de convergence de minimisation/maximisation soient satisfaits.

La méthode Golden Section search (recherche de la Section Dorée) est une méthode de région-élimination dans laquelle les limites inférieures et supérieures spécifiées par l'utilisateur se rapprochent à chaque itération. La région comprise entre les deux bornes est divisée en deux sections (Figure 3). La valeur de la variable dépendante est déterminée dans chaque section. Les limites de section contenant la plus petite (cas de minimisation) ou la plus grande (cas de maximisation) variable indépendante remplaceront les bornes de l'intervalle pour l'itération suivante. A chaque itération, la distance entre les bornes est réduite par le facteur $(1-\tau)$ où $\tau=0.61803$ connu comme la proportion de la Section Dorée.

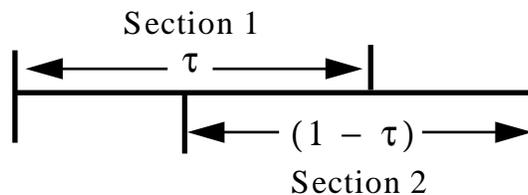


Figure 3: Région/Élimination utilisant la méthode 'Golden Section'

Intégration numérique

EES intègre des fonctions et résout des équations différentielles en utilisant une variante de la règle du trapèze avec un algorithme de prédiction/correction.

Considérons le problème d'estimation de l'intégrale suivante graphiquement

$$\text{Soit } f = 5 - 5X + 10X^2$$

Pour X compris entre 0 et 1. Pour intégrer graphiquement f , un tracé de f en fonction de X doit être réalisé. L'axe des abscisses sera divisé en un certain nombre de sections comme indiqué ci-dessous. L'aire sous la courbe dans chaque section est estimée comme l'aire d'un rectangle de largeur égale à la largeur de la section et de hauteur égale à la valeur de l'ordonnée moyenne de la section. Par exemple, les points d'abscisse 0 et 0.2 du graphique ci-dessous ont respectivement comme valeurs d'ordonnée 5 et 4.4. L'aire de la première section est alors $0.2 * (5+4.4)/2$ soit 0.94. L'estimation de la valeur de l'intégrale entre 0 et 1 est la somme des domaines des 5 sections. La précision est d'autant plus forte qu'il y a de sections.

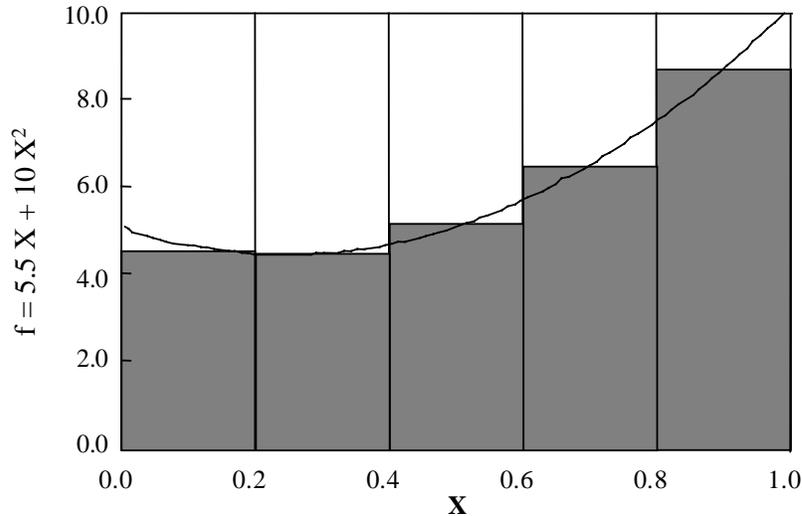


Figure 4: Approximation numérique d'une intégrale

EES intègre quasiment de la même façon les fonctions. La variable d'abscisse, par exemple X est placé dans la table paramétrique. Les valeurs de X entrées dans la table identifient la largeur de chaque section. EES n'exige pas d'avoir des sections de largeur identique. La valeur numérique de la fonction f est évaluée pour chaque valeurs de X fournie au travers de la fonction Integral (ex : Integral(f,X)).

Dans certaines situations, telle la résolution d'équations différentielles, la valeur de f peut ne pas être explicitement connue pour une valeur de X donnée. La valeur de f peut être dépendante de la solution d'équations algébriques non linéaires non encore convergentes.

De même, la valeur de f peut dépendre de la valeur de l'intégrale en ce point. Dans ce cas, une itération est nécessaire. EES évaluera à maintes reprises l'aire de la section en utilisant la dernière estimation de f, jusqu'à obtenir un résultat convergent. La procédure pour laquelle l'estimation d'une intégrale est corrigée à partir des informations obtenues lors de l'itération suivante se réfère à un algorithme de prédiction/correction.

Références

1. A. W. Al-Khafaji and J. R. Tooley, *Numerical Methods in Engineering Practice*, Holt, Rinehart and Winston, 1986, pp. 190 & ff.
2. C. F. Gerald and P. O. Wheatley, *Applied Numerical Analysis*, Addison-Wesley 1984, pp. 135 & ff.
3. J. H. Ferziger, *Numerical Methods for Engineering Application*, Wiley-Interscience 1981, Appendix B.
4. F. S. Acton, *Numerical Methods that Usually Work*, Harper and Row 1970.
5. I. S. Duff, A. M. Erisman and J. K. Reid, *Direct Methods for Sparse Matrices*, 1986 Oxford Science Publications, Clarendon Press.
6. S. Pissanetsky, "Sparse Matrix Technology," Academic Press 1984.
7. F. L. Alvarado, "The Sparse Matrix Manipulation System," *Report ECE-89-1*, Department of Electrical and Computer Engineering, The University of Wisconsin, Madison, Wisconsin, January 1989.
8. Tarjan, R. "Depth-First Search and Linear Graph Algorithms," *SIAM J. Comput.* 1, 146-160, (1972)
9. Powell's Method of Successive Quadratic Approximations, Reklaitis, Ravindran and Radsdell, *Engineering Optimization*, John Wiley, New York, (1983)
10. W. H. Press, B. P. Flannery and S. A. Teukolsky, and Vetterling, W.T., *Numerical Recipes in PASCAL*, Cambridge University Press, Chapter 10, (1989)

Ajout de données de propriétés

Informations générales

EES utilise une équation d'état approchée plutôt qu'un tableau de données internes pour calculer les propriétés des fluides. Pour quelques substances et dans certaines conditions, la loi des gaz parfaits est applicable. EES utilise une convention de nomination pour distinguer les gaz idéaux des fluides. Les substances qui sont représentées par leur symbole chimique (par exemple, N₂) sont modélisées à partir de la loi des gaz parfaits tandis que les substances pour lesquelles le nom est défini explicitement (par exemple, azote) sont considérées comme de fluides réels (Air et AirH₂O sont des exceptions à cette convention).

La table de JANAF [Stull, 1971] est utilisée pour fournir l'enthalpie de formation et l'entropie absolue à un état de référence de 298 K pour 1 atmosphère des substances gazeuses obéissant à la loi des gaz parfaits. Des corrélations spécifiques de chaleur pour ces substances sont calculées pour déterminer les propriétés thermodynamiques à des conditions différentes de l'état de référence. Un certain nombre de substances est inclus dans EES. Le programme externe de JANAF fournit des informations sur les propriétés thermodynamiques de centaines de substances supplémentaires. Par ailleurs, il est possible d'ajouter de nouvelles substances à l'aide des fichiers d'extension .IDG du répertoire Userlib, comme expliqué ci-dessous.

Les équations d'état de Martin-Hou [1955] (ou ses variantes) sont utilisées pour tous les fluides réels excepté l'eau (plusieurs équations d'état sont données pour l'eau, les plus précises étant celles éditées par Harr, Gallagher, et Kell [1984]). Les propriétés de la glace sont déterminées à partir de corrélations développées par Hyland et Wexler [1983].

Des relations de propriétés thermodynamiques sont employées pour déterminer l'enthalpie, l'énergie interne et les valeurs d'entropie. Celles-ci sont basées sur l'équation d'état et des corrélations supplémentaires pour la densité liquide, la pression de la vapeur et la chaleur spécifique pour une pression nulle en fonction de la température. Une modification de l'équation d'état de Martin-Hou proposée par Bivens et al. [1996] permet à cette équation d'être appliquée à des mélanges, tels que les réfrigérants (ex. R400).

La viscosité et la conductivité thermique des liquides et des gaz à basse pression sont corrélées (utilisation de polynômes fonctions de la température). Seule la température détermine les propriétés de transport pour les gaz parfaits. Pour les fluides réels, l'effet de la pression sur les propriétés de transport de gaz est estimé en utilisant les corrélations de Reid et al. [1977].

Ajout de propriétés à un fluide

EES a été conçu pour permettre à son utilisateur d'ajouter des fluides (parfaits et réels) supplémentaires dans la base de données de propriétés. L'utilisateur doit fournir les paramètres nécessaires pour les corrélations thermodynamiques et de propriétés transport. Les paramètres sont placés dans un fichier ASCII qui doit être localisé dans le sous-répertoire EES\USERLIB. EES chargera tous les fichiers liquides trouvés dans ce sous répertoire dès son ouverture. Les fluides supplémentaires apparaîtront totalement comme les fluides déjà intégrés. Les sections suivantes décrivent le format exigé pour les fichiers de données de propriétés.

Fichiers de gaz parfaits

Les fichiers de gaz de parfaits doivent avoir une extension .IDG. Une équation d'état n'est pas nécessaire puisqu'on suppose que le fluide obéit à l'équation d'état des gaz parfaits. Cependant, une attention particulière doit être prêtée aux états de référence si le gaz doit être utilisé dans les calculs impliquant des réactions chimiques. L'enthalpie de formation et la troisième loi de l'entropie à 298 K et à une pression de 1 bar (ou 1 atmosphère) doit être assurée.

Un fichier d'exemple fournissant le paramètre pour le CO₂ est montré ci-dessous.

Fichier d'exemple TESTCO2.IDG

```

TestCO2
44.01      {Molar mass of fluid}
100.0     {Tn Normalizing value in K}
250       {Lower temperature limit of Cp correlation in K}
1500      {Upper temperature limit of Cp correlation in K}
-3.7357 0  {a0, b0 Cp=sum(a[i]*(T/Tn)^b[i], i=0,9 in kJ/kgmole-K}
30.529 0.5 {a1, b1}
-4.1034 1.0 {a2, b2}
0.02420 2.0 {a3, b3}
0 0        {a4, b4}
0 0        {a5, b5}
0 0        {a6, b6}
0 0        {a7, b7}
0 0        {a8, b8}
0 0        {a9, b9}
298.15    {TRef in K}
100       {Pref in kPa}
-393520   {hform - enthalpy of formation in kJ/kgmole at TRef}
213.685   {s0 - Third law entropy in kJ/kgmole-K at Tref and PRef}
0         {reserved - set to 0}
0         {reserved - set to 0}

```

```

200      {Lower temperature limit of gas phase viscosity correlation
in K}
1000     {Upper temperature limit of gas phase viscosity correlation
in K}
-8.09519E-7 {v0 Viscosity = sum(v[i]*T^(i-1)) for i=0 to 5 in Pa/m^2}
6.039533E-8 {v1}
-2.8249E-11 {v2}
9.84378E-15 {v3}
-1.4732E-18 {v4}
0        {v5}
200     {Lower temperature limit of gas phase thermal conductivity
correlation in K}
1000    {Upper temperature limit of gas phase thermal conductivity
correlation in K}
-1.1582E-3 {t0 Thermal Conductivity = sum(t[i]*T^(i-1)) for i=0 to 5
in W/m-K}
3.9174E-5  {t1}
8.2396E-8  {t2}
-5.3105E-11 {t3}
3.1368E-16 {t4}
0          {t5}
0          {Terminator - set to 0}

```

Fichier de fluides réels

Un fichier de fluide réel pur est identifié avec l'extension .MHE (comme équation de Martin-Hou). Un fichier d'exemple nommé XFLUID.MHE est montré dans les pages suivantes pour illustrer le format exigé. (ce contient les paramètres utilisés pour le n-butane). Le fichier se compose de 75 lignes. La première ligne fournit le nom du fluide qu'EES identifiera dans les déclarations de fonctions de propriétés. Par exemple, la première ligne du fichier exemple contient UserFluid. L'enthalpie pour cette substance serait alors obtenue comme suit :

$$h = \mathbf{Enthalpy}(\text{UserFluid}, T=T1, P=P1)$$

Le nom du liquide apparaîtra dans l'ordre alphabétique avec les autres noms de fluide dans la boîte de dialogue 'Function Information'. Les 74 lignes suivantes contiennent un nombre. Un commentaire est ajouté sur la même ligne (après un ou plusieurs espace(s)) pour identifier le nombre. Les formes de toutes les corrélations excepté la relation pression-volume-température sont dans le fichier de XFLUID.MHE. La pression, le volume et la température sont associés par l'équation d'état de Martin-Hou sous la forme suivante. Une méthode pour obtenir les coefficients est décrite par Martin et Hou [1955].

Equation d'état de Martin-Hou (paramètres en lignes 18-36)

$$P = \frac{RT}{v-b} + \frac{A_2 + B_2T + C_2e^{-\beta T/T_c}}{(v-b)^2} + \frac{A_3 + B_3T + C_3e^{-\beta T/T_c}}{(v-b)^3} + \frac{A_4 + B_4T + C_4e^{-\beta T/T_c}}{(v-b)^4} + \frac{A_5 + B_5T + C_5e^{-\beta T/T_c}}{(v-b)^5} + \frac{A_6 + B_6T + C_6e^{-\beta T/T_c}}{e^{\alpha v}(1 + C'e^{\alpha v})}$$

où

$$P [=] \text{psia}, T [=] \text{R}, \text{ and } v [=] \text{ft}^3/\text{lb}_m$$

Vous pouvez avoir besoin d'utiliser des données convenables de propriétés ou des données corrélées afin obtenir les paramètres appropriés. La plupart des corrélations concernant les paramètres sont linéaires de sorte qu'elles puissent être déterminées par régression linéaire. Un ajustement de paramètres qui améliore le résultat de la méthode de Martin-Hou peut être déterminé par régression non linéaire.

EES peut être utilisé pour faire ces deux types de régression.

Fichier pour les fluides purs SAMPLE XFLUID.MHE

```
UserFluid
58.1      { molecular weight}
0         { not used}
12.84149  { a}  Liquid
Density=a+b*Tz^(1/3)+c*Tz^(2/3)+d*Tz+e*Tz^(4/3)+f*sqrt(Tz)+g*(Tz)^2}
33.02582  { b}      where Tz=(1-T/Tc) and Liquid
Density[=]lbm/ft3
-2.53317  { c}
-0.07982  { d}
9.89109   { e}
0         { f}
0         { g}
-6481.15338 { a}  Vapor pressure fit: lnP=a/T+b+cT+d(1-
T/Tc)^1.5+eT^2
15.31880  { b}      where T[=]R and P[=]psia
-0.0006874 { c}
4.28739   { d}
0         { e}
0         { not used}
0.184697  { Gas constant in psia-ft3/lbm-R}
1.5259e-2 { b}  Constants for Martin-Hou EOS/English_units
-20.589   { A2}
9.6163e-3 { B2}
-314.538  { C2}
0.935527  { A3}
-3.4550e-4 { B3}
19.0974   { C3}
-1.9478e-2 { A4}
```

```

0          { B4}
0          { C4}
0          { A5}
2.9368e-7 { B5}
-5.1463e-3 { C5}
0          { A6}
0          { B6}
0          { C6}
5.475     { Beta}
0         { alpha}
0         { C'}
-7.39053E-3 { a}      Cv(0 pressure) = a + b T + c T^2 + d T^3 +
e/T^2
6.4925e-4  { b}          where T[=]R and Cv[=]Btu/lb-R
9.0466e-8  { c}
-1.1273e-10 { d}
5.2005e3   { e}
124.19551  { href offset}
0.0956305  { sref offset}
550.6      { Pc [=] psia}
765.3      { Tc [=] R}
0.07064    { vc [=] ft3/lbm}
0          { not used}
0          { not used}
2          { Viscosity correlation type: set to 2: do not change}
260        { Lower limit of gas viscosity correlation in K}
535        { Upper limit of gas viscosity correlation in K}
-3.790619e6 { A}      GasViscosity*1E12=A+B*T+C*T^2+D*T^3
5.42356586e4 { B}      where T[=]K and GasViscosity[=]N-s/m2
-7.09216279e1 { C}
5.33070354e-2 { D}
115        { Lower limit of liquid viscosity correlation in K}
235        { Upper limit of liquid viscosity correlation in K}
2.79677345e3 { A}      LiquidViscosity*1E6=A+B*T+C*T^2+D*T^3
-2.05162697e1 { B}      where T[=]K and LiquidViscosity[=]N-s/m2
5.3698529e-2 { C}
-4.88512807e-5 { D}
2          { Conductivity correlation type: set to 2: do not
change}
250        { Lower limit of gas conductivity correlation in K}
535        { Upper limit of gas conductivity correlation in K}
7.5931e-3   { A}      GasConductivity=A+B*T+C*T^2+D*T^3
-6.3846e-5  { B}      where T[=]K and GasConductivity[=]W/m-K
3.95367e-7  { C}
-2.9508e-10 { D}
115        { Lower limit of liquid conductivity correlation in K}
235        { Upper limit of liquid conductivity correlation in K}
2.776919161e-1 { A}      LiquidConductivity=A+B*T+C*T^2+D*T^3
-8.45278149e-4 { B}      where T[=]K and LiquidConductivity[=]W/m-K
1.57860101e-6 { C}

```

```
-1.8381151e-9 { D}
0 { not used: terminator}
```

Propriétés des mélanges

L'équation d'état de Martin-Hou peut être adaptée pour des mélanges comme proposé par Bivens. Les modifications principales requises pour rendre cette équation d'état applicable aux mélanges sont de fournir des corrélations séparées for the bubble and dew point vapor pressure et une corrélation pour l'enthalpie de la vaporisation, puisque l'équation d'état ne peut pas fournir ces informations. Ci-dessous, vous trouverez le fichier R410A.MHE utilisé pour fournir les propriétés du réfrigérant R410A :

```
R410A
72.584 {molecular weight Bivens and Yokozeki}
400 {Indicator for blend}
30.5148 {a} Liquid density = a+b*Tz^(1/3)+c*Tz^(2/3)+d*Tz
60.5637 {b} +e*Tz^(4/3)+f*sqrt(Tz)+g*(Tz)^2}
-5.39377 {c} where Tz=(1-T/Tc) and Liquid Density[=]lbm/ft3
55.5360815 {d}
-21.88425 {e}
0 {f}
0 {g}
-5.9789E+03 -5.9940E+03 {a} Bubble and Dew Pt Vapor pressure fit:
24.06932 24.04507 {b} lnP=a/T+b+cT+d(1-T/Tc)^1.5+eT^2
-2.1192E-02 -2.1084E-02 {c} where T[=]R and P[=]psia fit
-5.5841E-01 -4.4382E-01 {d}
1.3718E-05 1.3668E-05 {e}
0 0 {not used}
0.1478 {Gas constant in psia-ft3/lbm-R}
0.006976 {b} Constants for Martin-Hou EOS/English_units from Bivens
-6.40764E+00 {A2}
3.40372E-03 {B2}
-2.34220E+02 {C2}
1.41972E-01 {A3}
4.84456E-06 {B3}
9.13546E+00 {C3}
-4.13400E-03 {A4}
0 {B4}
0 {C4}
-9.54645E-05 {A5}
1.17310E-07 {B5}
2.45370E-02 {C5}
0 {A6}
0 {B6}
0 {C6}
5.75 {Beta}
0 {alpha}
```

```

0          {C'}
0.036582  {a}   Cv(0 pressure) = a + b T + c T^2 + d T^3 + e/T^2
2.808787E-4 {b}           where T[=]R and Cv[=]Btu/lb-R from Bivens
-7.264730E-8 {c}
2.6612670E-12 {d}
0          {e}
65.831547  {href offset}
-0.082942  {sref offset}
714.5      {Pc [=] psia}
621.5      {Tc [=] R}
0.03276    {vc [=] ft3/lbm}
0          {not used}
7          {# of coefficients which follow - used for blends}
1          {DeltaH Correlation type}
0.5541498  {Xo}
87.50197   {A}   DeltaH_vap=A+B*X+C*X^2+D*X^3+E*X^4 Bivens
185.3407   {B}   where X =(1-T/Tc)^.333-X0, T in R and enthalpy in Btu/lb
13.75282   {C}
0          {D}
0          {E}
2          {Viscosity correlation type: set to 2: do not change}
200        {Lower limit of gas viscosity correlation in K}
500        {Upper limit of gas viscosity correlation in K}
-1.300419E6 {A}   GasViscosity*1E12=A+B*T+C*T^2+D*T^3
5.39552e4   {B}   where T[=]K and GasViscosity[=]N-s/m2
-1.550729e1 {C}
0          {D}
-999       {Lower limit of liquid viscosity correlation in K}
-999       {Upper limit of liquid viscosity correlation in K}
0          {A}   Liquid Viscosity*1E6=A+B*T+C*T^2+D*T^3
0          {B}   where T[=]K and Liquid Viscosity[=]N-s/m2
0          {C}
0          {D}
2          {Conductivity correlation type: set to 2: do not change}
200        {Lower limit of gas conductivity correlation in K}
500        {Upper limit of gas conductivity correlation in K}
-8.643088e-3 {A}   GasConductivity=A+B*T+C*T^2+D*T^3
7.652083e-5 {B}   where T[=]K and GasConductivity[=]W/m-K
2.144608e-9 {C}
0          {D}
-999       {Lower limit of liquid conductivity correlation in K}
-999       {Upper limit of liquid conductivity correlation in K}
0          {A}   LiquidConductivity=A+B*T+C*T^2+D*T^3
0          {B}   where T[=]K and LiquidConductivity[=]W/m-K
0          {C}
0          {D}
0 {terminator}

```

{The forms of the correlations and in some cases the coefficients have been adapted from D.B. Bivens and A. Yokozeki, "Thermodynamics and Performance Potential of R-410a," 1996 Intl. Conference on Ozone Protection Technologies

Oct, 21-23, Washington, DC.}

Références

ASHRAE Handbook of Fundamentals, (1989, 1993, 1997), American Society of Heating, Refrigerating and Air Conditioning Engineers, Atlanta, GA

ASHRAE, **Thermophysical Properties of Refrigerants**, American Society of Heating, Refrigerating, and Air-Conditioning Engineers, Atlanta, GA, (1976)

D.B. Bivens and A. Yokozeki, "Thermodynamics and Performance Potential of R-410a," 1996 Intl. Conference on Ozone Protection Technologies Oct, 21-23, Washington, DC.

Downing, R.C. and Knight, B.W., "Computer Program for Calculating Properties for the "FREON" Refrigerants," DuPont Technical Bulletin RT-52, (1971); Downing, R.C., "Refrigerant Equations", ASHRAE Transactions, Paper No. 2313, Vol. 80, pt.2, pp. 158-169, (1974)

Gallagher, J., McLinden, M, Morrison, G., and Huber, M., REFPROP - NIST Thermodynamic Properties of Refrigerants and Refrigerant Mixtures, Versions 4, 5, and 6, NIST Standard Reference Database 23, NIST, Gaithersburg. MD 20899, (1989)

Harr, L. Gallagher, J.S., and Kell, G.S (Hemisphere, 1984). **NBS/NRC Steam Tables**, Hemisphere Publishing Company, Washington, (1984)

Howell, J.R., and Buckius, R.O., **Fundamentals of Engineering Thermodynamics**, McGraw-Hill, New York, (1987)

Hyland and Wexler, "Formulations for the Thermodynamic Properties of the Saturated Phases of H₂O from 173.15 K to 473.15 K, ASHRAE Transactions, Part 2A, Paper 2793 (RP-216), (1983)

Keenan, J.H., Chao, J., and Kaye, J., **Gas Tables**, Second Edition, John Wiley, New York, (1980)

Keenan, J.H. et al., **Steam Tables**, John Wiley, New York, (1969)

Irvine, T.F. Jr., and Liley, P.E., **Steam and Gas Tables with Computer Equations**, Academic Press Inc., (1984)

Martin, J.J. and Hou, Y.C., "Development of an Equation of State for Gases," A.I.Ch.E Journal, 1:142, (1955)

McLinden, M.O. et al., "Measurement and Formulation of the Thermodynamic Properties of Refrigerants 134a and 123, ASHRAE Trans., Vol. 95, No. 2, (1989)

Reid, R.C.Prausnitz, J.M. and Sherwood, T.K., **The Properties of Gases and Liquids**, McGraw-Hill, 3rd edition, (1977)

Shankland, I.R., Basu, R.S., and Wilson, D.P., "Thermal Conductivity and Viscosity of a New Stratospherically Sate Refrigerant - 1,1,1,2 Tetrafluoroethane (R-134a), published in **CFCs**:

Time of Transition, American Society of Heating, Refrigeration and Air-Conditioning Engineers, Inc., (1989)

Shankland, I.R., "Transport Properties of CFC Alternatives", AIChE Spring Meeting, Symposium on Global Climate Change and Refrigerant Properties, Orlando, FL, March, (1990)

Stull, D.R., and Prophet, H., **JANAF Thermochemical Tables**, Second Edition, U.S. National Bureau of Standards, Washington, (1971)

Van Wylen, G.J., and Sonntag, R.E., **Fundamentals of Classical Thermodynamics**, Third Edition, John Wiley, New York, (1986)

Wilson, D.P. and Basu, R.S., "Thermodynamic Properties of a New Stratospherically Safe Working Fluid - Refrigerant 134a", paper presented at the ASHRAE meeting, Ottawa, Ontario, Canada, June, (1988), published in **CFCs: Time of Transition**, American Society of Heating, Refrigeration and Air-Conditioning Engineers, Inc., (1989)

Example Problem Information

The EXAMPLES subdirectory within the EES directory contains many worked-out example problems. Each example problem illustrates one or more EES features, as indicated in the information below.

<u>Feature</u>	<u>EES Example Files</u>
Arrays	MATRIX.EES, MATRIX2.EES, RANKINE.EES, REFRIG.EES, REGEN.EES
Complex numbers	COMPLEXROOTS.EES
Comments	HEATEX.EES
Curve-fitting	COPPER.EES
Diagram window	DIAGRMW.EES
Differential equations	DRAG.EES, RK4_TEST.EES, SUBSTEPS.EES, DIFEQN1.EES, DIFEQN1.EES
Differentiate function	COPPER.EES
DUPLICATE command	MATRIX.EES, MATRIX2.EES, NLINRG.EES, REGEN.EES
Formatted Equations	HEATEX.EES, DRAG.EES
Functions, user-written	CONVECT.EES, MOODY.EES, RK4_TEST.EES
Greek symbols	HEATEX.EES, NLINRG.EES
Integration	DBL_INTEG.EES, DIFEQN1.EES, DIFEQN2.EES, SUBSTEPS.EES, DRAG.EES, RK4_TEST.EES
Interpolate function	COPPER.EES
JANAF table	FLAMET.EES
LOOKUP table	NLINRG.EES, COPPER.EES
Minimize or maximize	MAXPOWER.EES, NLINRG.EES, RANKINE.EES
Modules	MOODY.EES
Overlay Plot	CH1EX.EES RANKINE.EES
Parametric table	CAPVST.EES, CH1EX.EES, DIFEQN1.EES, FLAMET.EES, SUBSTEPS.EES
Plotting	CAPVST.EES, DIFEQN2
Procedures, user-written	REGEN.EES
Procedures, external	ABSORP.EES
Properties, thermodynamic	REFRIG.EES, CATVST.EES, CH1EX.EES, FLAMET.EES, REGEN.EES
Property Plot	RANKINE.EES, REFRIG.EES
Psychrometric functions	SUPERMKT.EES
Regression	NLINRG.EES
Subscripted variables	MATRIX.EES, MATRIX2.EES, HEATEX.EES
SUM function	MATRIX.EES, MATRIX2.EES, NLINRG.EES
Systems of equations	HEATEX.EES, CH1EX.EES
TABLEVALUE	DIFEQN2.EES
Transport properties	CONVECT.EES
Unit conversion	DRAG.EES