CHAM

InForm

Extending PHOENICS

Summary

- In-Form enables users of PHOENICS to greatly extend its capabilities, without any need to write FORTRAN coding.
- Express their requirements by using formulae.
- These are read by the Input Module (Satellite), which transmits them to the Solver Module (EARTH); this then interprets them and performs the implied computations.
- In-Form does **not** require use of a re-compilable version of PHOENICS.

Summary

- This lecture provides a brief introduction to In-Form.
- The complete documentation can be found in POLIS.
- In-Form can be used to:
  - Set sources
  - Set initial values
  - Set physical properties
  - Define and calculate new derived quantities
  - Calculate total or average values and print them
  - Generate additional monitoring tables
  - Many others...

Syntax

CHAM

- The syntax of In-Form is relatively simple.
- A typical In-Form command will have some if not all of these elements:

(**KEYWORD** of **VARIABLE** at **LOCATION** is **FORMULA** with **CONDITION**)

  - KEYWORD defines what the In-Form command is setting.
  - VARIABLE defines which SOLVEd or STOREd variable it is to apply to.
  - LOCATION defines where
  - FORMULA defines what
  - CONDITION sets the conditions under which it will happen.

- Such lines are inserted into the Q1 as required.

Syntax

CHAM

- It should be noted that:
  - The opening and closing brackets of the statement must always be present. Brackets must match!
  - The opening bracket must start in the first or second column.
  - A dollar sign, $, at the end of a line will be taken as an append-next-line instruction. The total maximum length of a line is 1024 characters.
  - On output, lines longer than 68 characters will be ‘folded’ with a $ sign at column 68.
  - Characters may be upper- or lower-case without consequences.
  - Blank spaces separate the items, several successive spaces having the same effect as one.

How to create In-Form coding

CHAM

- In-Form coding may be typed into the Q1 file directly. In VR, click “File” / “Open file for Editing” / “Q1”.
- This simple example specifies that the density in all cells should be obtained from the ideal gas law.

```
************************************************************
 Group 9. Properties
PRESS0 =1.01325E+05 ;TEMP0 =273.
   * Domain material index is   0 signifying:
   * Air at 20 deg C, 1 atm, treated as incompressible
SETPRPS(1, 0)
DVO1DT =3.41E-03
   * Non-default SOLUTN setting for KE
SOLUTN(KE  ,Y,Y,Y,N,N,N)
   * Non-default SOLUTN setting for EP
SOLUTN(EP  ,Y,Y,Y,N,N,N)
PRT(EP)=1.314
************************************************************
 Echo save-block settings for Group  9
 save9begin
(property RHO1 is (p1+press0)/(287*(tem1+273)))
 save9end
************************************************************
```

- “press0” is the pressure reference, defined in the “Properties” menu.

How to create In-Form coding

- In this example, note the importance of the statements

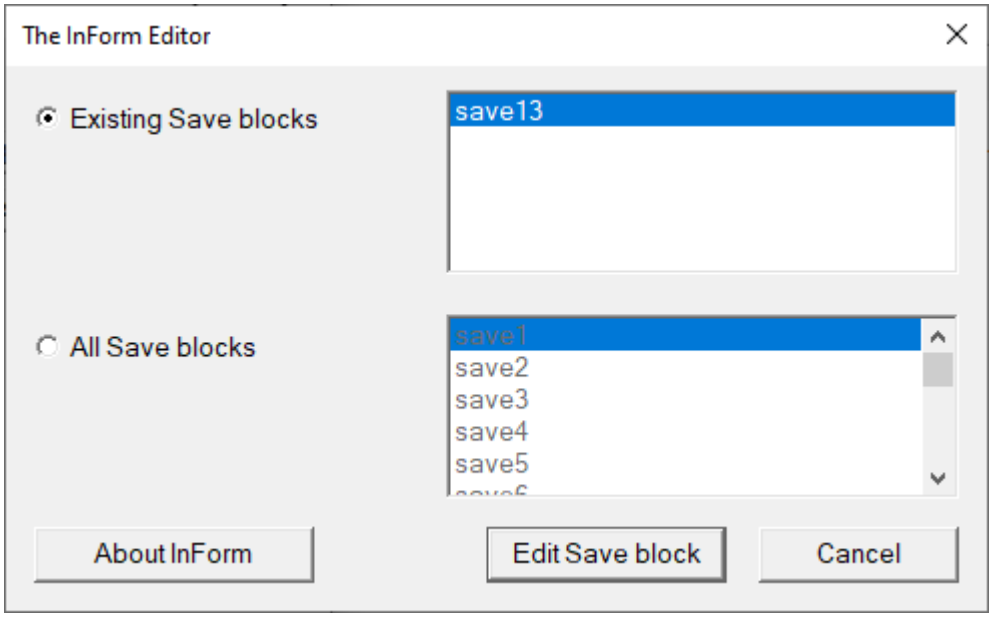
  save9begin
  save9end

  which “bracket” the In-Form coding, and distinguish it from the PIL coding in the rest of the Q1.
- The “save” lines must begin with two spaces.
- The “9” above is the number of the Q1 Group.
- Note that it does not matter which Group of the Q1 you use for the In-Form, but it is conventional to locate properties in Group 9 and sources in Group 13.
- The single In-Form line operates over the whole 3D space. No loops are necessary in the coding.

How to create In-Form coding - the In-Form Editor

CHAM

- There is an ‘INFORM' button in the Menu. Clicking it starts the In-Form Editor:



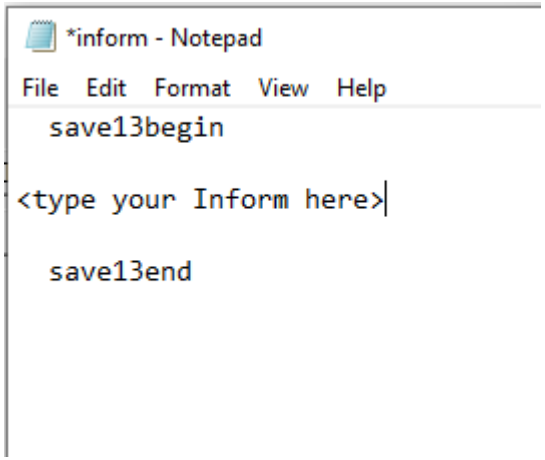
- In this example some InForm coding has already been created in Group 13.
- Click the relevant block and type your InForm…

How to create In-Form coding - the In-Form Editor

Lecture

CHAM

- Type your InForm between the “save” markers…



- Be sure to close the In-Form Editor when finished, by clicking:

“File” / ”Save”, and then
“File” / ”Exit”.

Common Keywords - PROPERTY

- **PROPERTY** - This keyword sets a physical property.

  (**PROPERTY** of **VARIABLE** at **LOCATION** is **FORMULA** with **CONDITION**)
- The VARIABLE keyword is the property to be set. It can be one of these:

  RHO1, DRH1DP, RHO2, DRH2DP, ENUT, ENUL, PRNDTL, PHINT, TMP1, TMP2, EL1, EL2, CP1, CP2, DVO1DT, DVO2DT, CFIPS, CMDOT, CVM

  (A full description of each of these is given in POLIS.
- The FORMULA keyword defines the value of the property.
- The LOCATION or CONDITION keywords can be used to limit the region of applicability of the FORMULA.

Common Keywords - PROPERTY

- We have already seen how to compute the density DEN1 from the pressure P1 and the temperature TEM1, using the gas law:

```
save9begin
(property DEN1 is (p1+press0)/(287*(tem1+273)))
save9end
```

(Note: in this context DEN1 and RHO1 are synonymous.)

- We can add some PIL to get the density for a mixture of air and water vapour:

```
save9begin
real(wair,wh2o); wair=29.0; wh2o=18.0; gascon=8314.43
(property DEN1 is (p1+press0)/$
 (gascon*((1.-MH2O)/wair+MH2O/wh2o)*(TEM1+temp0)) )
save9end
```

Common Keywords - PROPERTY

- **EXAMPLE** - Make density a quadratic function of temperature (TEM1) within object B10:

```
Save9begin
(property RHO1 at B10 is 1.0+0.1*TEM1+1.E-6*TEM1^2)
Save9end
```

- The ‘at B10’ limits the formula to those cells occupied by B10.
- Everywhere else, the density takes its set value, assigned from the property marker PRPS.

Common Keywords – STORED

- **STORED** - This keyword is used for the creation of auxiliary variables, which can have distinct values for each cell in the domain.

  (**STORED** of **VARIABLE** at **LOCATION** is **FORMULA** with **CONDITION**)

- The VARIABLE keyword is any name of the user's choice, up to 4 characters long.

- A list of names already used by PHOENICS, and which should therefore be avoided, is given in POLIS under ‘Reserved Names’.

Common Keywords – STORED

- **EXAMPLE** - create a new 3D variable which contains the temperature, in fluid cells only.
- (This shows the geometry more clearly if you do not need to see the temperature in solids.)

```
save7begin
(stored of TEMF is TEM1 with imat<100)
save7end
```

- TEMF will be set to the temperature in all cells where PRPS is less than 100, i.e. in fluids (but not in solids).
- (Note: Group 7 is for Stored and Solved variables.)

Common Keywords – INITIAL

CHAM

- **INITIAL** - This keyword sets the initial value of a STOREd or SOLVEd variable.

  (**INITIAL** of **VARIABLE** at **LOCATION** is **FORMULA** with **CONDITION**)

Common Keywords – SOURCE

- **SOURCE** - This keyword is used for introducing formulae defining the sources of mass, momentum, energy and other conserved properties.
  (**SOURCE** of **VARIABLE** at **LOCATION** is **FORMULA** with **CONDITION**)

- The ability to set general sources of mass, momentum, heat, concentration etc is one of the greatest benefits of InForm.

Common Keywords – SOURCE

- A CONDITION keyword can be used to specify the "type" of source - i.e. whether it is
  - a fixed flux (the default, no flag)
  - a fixed value (flag FIXVAL)
  - total for the object (flag WHOL)
  - per unit area or volume (flags AREA or VOLU)

- Example - Set a heat source of 100W/m^3 in object HOT.

```
save13begin
(source of tem1 at HOT is 100 with VOLU)
save13end
```

Common Keywords – SOURCE

- Example - Set object HOT to be 100°C.

```
save13begin
(source of tem1 at HOT is 100 with FIXVAL)
save13end
```

- Example - Set a heat source of 100W in object HOT. (Note: this is for the whole object, not per m^3.)

```
save13begin
(source of tem1 at HOT is 100 with WHOL)
save13end
```

- Note: Group 13 is traditionally used for sources.

Common Keywords – SOURCE

- **EXAMPLE** - Set a time-dependent heat source:
  - Q = min (10000*t, 3.E6) for t < 720
  - Q = max (3.0e6-13000*(t-720), 5000) for t > 720
- The heat source increases linearly with time for the first 720 seconds, to a peak of 3MW which is reached after 300 seconds and maintained until 720 seconds.
- After 720 seconds, the heat source reduces linearly with time until it reaches a minimum value of 5kW at 950 seconds.
- This is a typical car fire curve.
- Suppose the fire object is called FIRE1; the required InForm coding would look like this:

Common Keywords – SOURCE

CHAM

- Q = min (10000*t, 3.0E6) for t < 720
- Q = max (3.e6-13000*(t-720), 5000) for t > 720

- The required InForm is:

```
save13begin
(source of tem1 at fire1 is min(1e4*tim,3.e6) with $
 if(tim.le.720) ! WHOL)
(source of tem1 at fire1 is max(3.e6-1.3e4*(tim-$
 720),5e3) with if(tim.gt.720) ! WHOL)
save13end
```

- TIM is the PHOENICS variable for the current time.
- ! applies another condition; the use of this and of “if” is described later in this lecture.
- This example is described in full in InForm Tutorial 2.

Common Keywords – MAKE1 and STORE1

- The **MAKE1** keyword is used to create a single-value parameter.
  (**MAKE1** of **PARAM** is **VALUE**)
- PARAM is the parameter to be created.
- VALUE (optional) is the initial value of the parameter.

- The **STORE1** keyword is used to set the value of a parameter.
  (**STORE1** of **PARAM** is **VALUE**)
- The parameter PARAM is set to VALUE.
- Before it can be set by STORE1, a parameter must be created by MAKE1.

- The **PRINT** keyword is used to print the parameter...

Common Keywords – PRINT

CHAM

- The **PRINT** keyword is used to print out parameters which have been set using STORE1.
  (**PRINT** of **TITLE** is **VALUE**)
- TITLE is a character string used to label the output (15 characters maximum).
- VALUE is the parameter which is to be printed. The parameter will generally have been created with MAKE1, and set using STORE1.
- The printout appears the file ‘Inforout’, which can be viewed by clicking “File” / “Open file for editing”.
- InForm Tutorial 4 gives a worked example.

Common Keywords – PRINT

- **EXAMPLE –**
  Print the area-averaged z-velocity at object HOLE.

```
save21begin
(make atot is 0.)
(make wtot is 0.)
(store1 of atot at HOLE is sum(ahigh))
(store1 of wtot at HOLE is sum(w1*ahigh))
(print of w1_mean is wtot/atot)
save21end
```

- The SUM function sums the quantity over the location.
- W1 is the z-velocity, AHIGH is the z (“high”) cell-face area.
- For x and y cell-face areas, use AEAST and ANORTH.

Common Keywords – TABLE

CHAM

- The **TABLE** keyword is used to produce a table of values, either with sweep or with time step.

  (**TABLE** in **FILENAME** is **GET**(COL1,COL2,.. COLn) with **HEAD**(H1,H2,...Hn) ! sweep)
- FILENAME is the name of the file to write.
- COL1, COL2,... are formulae for each column of the table.
- H1,H2,... Are the headings for the columns.
- “! sweep” means write a line at every sweep. To write a line every transient time-step, use “! time” instead.
- InForm Tutorial 4 gives a worked example of using the TABLE keyword to create extra monitoring information.

And now …

CHAM

- Having gone through the list of basic keywords, we now move on to discuss other elements of InForm in more detail:
- - Variable
  - Location
  - Formula
  - Operators
  - Functions
  - Condition
  - Property index IMAT

VARIABLE

- VARIABLE defines the variable that the In-Form statement is to operate on. A “variable” is a quantity which has a value for every cell in the domain.
- It can be any of the PHOENICS variables, which you can see listed in “Models” / “Solution control - extra variables”.
- Alternatively it can be any variable which has been created using the InForm “STORED” command.
- **Example -** Create a new variable to store temperature in fluid regions only. TEM1 and TEMF are variables.

```
save7begin
(stored of TEMF is TEM1 with imat<100)
save7end
```

LOCATION

CHAM

- LOCATION defines the location **where** the In-Form command will be applied. If it is absent, the command will apply to the whole domain.
- LOCATION will generally be the name of a VR Object.
- If different formulae are required at different locations or at different times, a separate In-Form statement will be needed for each location / time-range.

FORMULA

CHAM

- This is the formula, long or short, simple or complex, which describes what the In-Form coding is intended to implement.
- In-Form formulae, for setting properties, initial values, sources or anything else, are arrangements of **operators**, **functions** and **operands**.
- These conform to rules which are similar to those of algebra and of programming languages.
- No significance attaches to whether upper- or lower-case characters are used.

FORMULA - Operators

CHAM

- The operators which may be used are:

  + - * / ( )

  all of which have their usual significances; and

  ^

  which represents exponentiation.

- (Note that the ****** of Fortran cannot be used for exponentiation.)

FORMULA - Functions

- Available functions, listed in full in POLIS, include:

- **conventional mathematical functions:**
  - *ABS ACOS ASIN ATAN COS MAX MIN SIN SQRT TAN* which have their usual significances;
  - *EXP* is the exponential to base **e**;
  - *LOGE* is the **natural** logarithm, with base **e**;
  - *LOG10* is the **Napierian** logarithm, with base 10.

FORMULA - Functions

- **formula-name functions** (a selection)**:**
  - *POL2 POL3 POL4 POL5 POL6*, polynomials of the appropriate order;
  - *PWL3*, a piece-wise-linear function, with three parts;
  - *SPL5*, a cubic-interpolation spline function passing through five points; and
  - *PWLF*, a piece-wise linear function of which the defining points are specified in a file. This is useful e.g. for interpolating inputs from discrete data.
- **neighbour-location functions:**
  - *EAST, WEST, NORTH, SOUTH, HIGH, LOW* and *OLD* have meanings which are conventional in PHOENICS.

FORMULA - Functions

- Each FORMULA contains implied loops, with limits set by the LOCATION keyword – there is no need for a DO construct.
- To access the values of a variable at a particular location, use
  - [i,j,k] for the value at cell i,j,k
    - e.g. P1[2,5,12]
  - [+/-i, +/-j, +/-k] for an offset from the current cell (blank for current cell)
    - e.g. U1[+1,,-1]
  - {x,y,z} for the value at coordinates x,y,z (metres)
    - e.g. TEM1{1.2,3.7,5.2}

CONDITION

CHAM

- !, like “with”, specifies a condition.
- “with” may only be used for the first condition. Subsequent conditions must be specified by ! rather than “with”.
- There are numerous post-formula options - many apply only to particular keywords (see POLIS) and will not be described here.
- There are two important ones which apply to many keywords - IMAT and IF.

CONDITION - Limitation to a particular material

CHAM

- It is possible to limit the action of a formula to those locations at which the material-property index, IMAT (*alias* PRPS), has a specific value.
- The complete set of 'with IMAT' conditions allowable in In-Form statements is:

  'with IMAT>value' means 'for PRPS greater than value'

  'with IMAT<value' means 'for PRPS less than value'

  'with IMAT>=value' means 'for PRPS greater than or equal to value'

  'with IMAT<=value' means 'for PRPS less than or equal to value'

  'with IMAT=value' means 'for PRPS equal to value'

  'with IMAT!=value' means 'for PRPS not equal to value'
- Material numbers 0 – 99 denote fluids; 100 and greater denote solids; 198 denotes a non-participating solid with friction.

CONDITION - Limitation to a particular set of locations

- Variables other than PRPS can be used in a similar way.
- Thus, for example, it would be possible to:
  - define a new whole-field stored variable called, say, "MARK";
  - use In-Form to ascribe values for it (either 0 or 1) over the whole field;
  - specify a source term, which will apply only in cells where MARK=1.

CONDITION

CHAM

- Another generally-applicable post-formula option is the

  "IF (*condition*)"

  construct, as we saw in one of the SOURCE examples above.

- But note that you cannot write

  “if … then … else …” .

  A second “IF” with the reverse condition must be used instead.

Further Examples

- Some further examples of In-Form…

- **EXAMPLE -** Make density a quadratic function of temperature (TEM1) for material 123:

```
save9begin
(property RHO1 is 1.0+0.1*TEM1+1.E-6*TEM1^2 with imat=123)
save9end
```

- The ‘with imat=123’ condition limits the formula to those cells where the property marker PRPS is 123.
- Everywhere else the density takes its usual value.

Further Examples

- **EXAMPLE -** set the velocity at the object FAN to 20 m/s when the temperature at coordinate (23.4, 12.8, 2.2) exceeds 60°:

```
save13begin
(source of V1 at FAN is 20 with if(tem1{23.4, 12.8, 2.2}.gt.60)
    ! fixval)
save13end
```

- The "! fixval" condition specifies that the source is a fixed value (i.e. a velocity in this case). Without it, the source would be a fixed flux (i.e. for V1, a force).
- The {} brackets indicate (x,y,z) coordinates in metres.

In-Form and Objects

CHAM

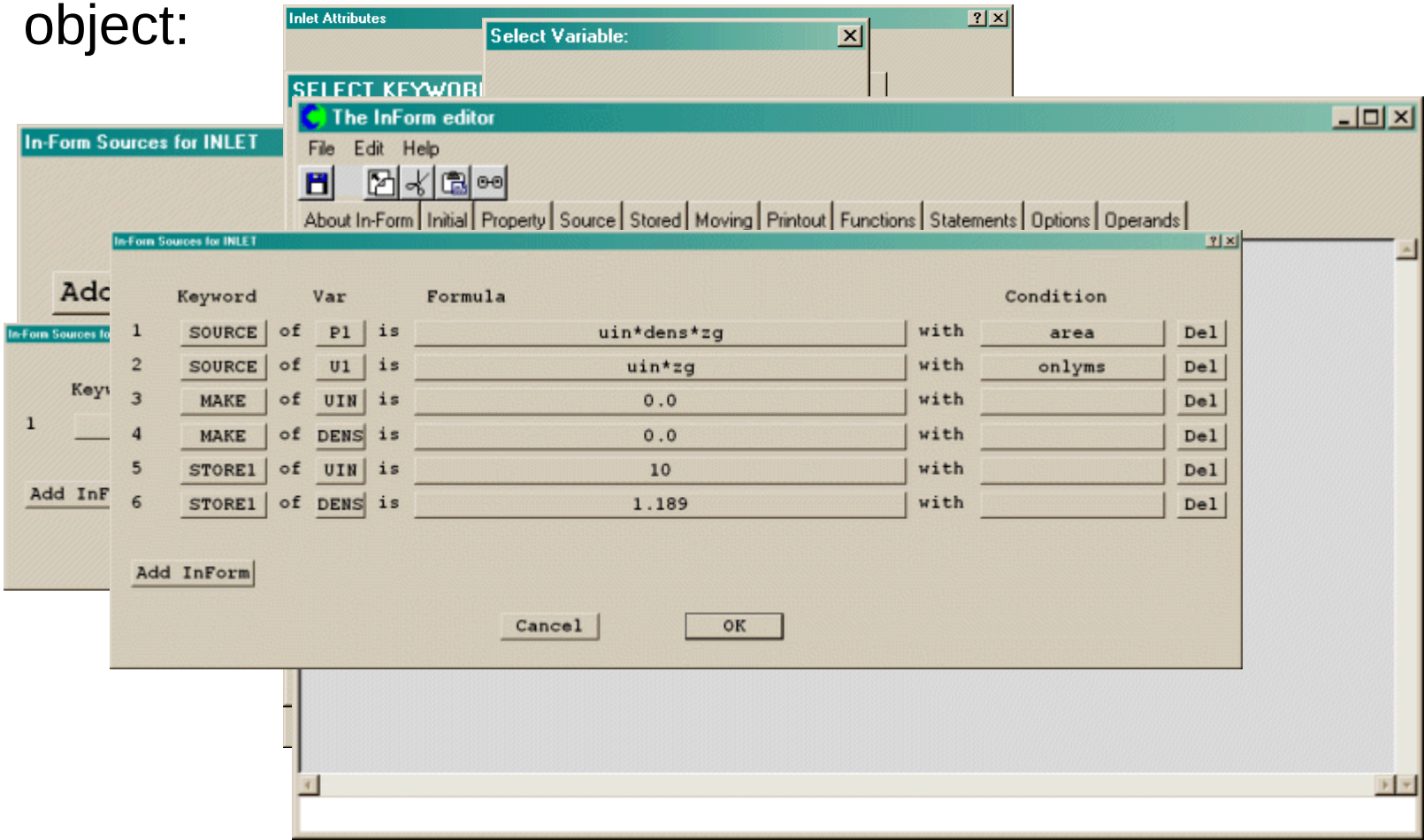
- **In-Form and OBJECTS**

- The In-Form described above is the ‘classic’ form, which has to be “hand-edited” into the Q1.
- It is very powerful, especially when combined with the programmability of PIL.
- It can be accessed via the In-Form buttons on the Main Menu panels.
- However - there is a newer form of In-Form, more directly linked to the objects.
- It is accessed from the In-Form buttons on the object dialogs.

In-Form and Objects

CHAM

- The following dialogs allow a limited range of common In-Form commands to be attached to an object:



In-Form and Objects

CHAM

- These settings are held in the Q1 as part of the object definition.

```
> OBJ,INFSRC_TEM1, min(1e4*tim,3.e6) with if(tim.le.720)!whol
> OBJ,INFSRC_TEM1, max(3.e6-1.3e4*(tim-720),5e3) with if(tim.gt$
> OBJ,INFSRC_TEM1, .720)!whol
```

- This is equivalent to

```
(source of tem1 at fire1 is min(1e4*tim,3.e6) with $
 if(tim.le.720) ! WHOL)
(source of tem1 at fire1 is max(3.e6-1.3e4*(tim-$
 720),5e3) with if(tim.gt.720) ! WHOL)
```

- If the object is copied, the In-Form commands will be copied as well.
- In-Form Tutorial 3 gives an example of using Object-related Inform.

To conclude…

CHAM

- This presentation is a basic introduction to In-Form.
- For full particulars, see the In-Form entry in POLIS.
- The best way to learn how to use In-Form is to practise using it. The Tutorials are recommended.