2. PROBLEM SOLVING IN EXAMPLES

According to the three kinds of mathematical models the three kinds of users has been evolved and each of them considers FEMINA as a different program even if the program control and presentation of results is similar:

- ❑ The simplest job concerns the users considering FEMINA only as a simple regression program. They only need to know how to prepare data (observation matrix with independent and dependent variables, e.g. time and measured response) and they must learn two commands: how to read data file (command ROM or READ) and how to run the regression program (command NELREG) – all other actions are automatic, including selection of the best regression model (models of axial dispersion, Gaussian distribution, response of ideally mixed tanks in series and many others are implemented).

- ❑ The second group form users dealing with stimulus-response experiments, peoples, making analysis of residence time distributions and looking for a suitable model of a continuous system (usually lumped parameter model) describing their data-measured responses. They need not know anything about finite element method, they only need to learn approximately 20 basic commands and first of all they have to know which basic RTD models are available, how to modify existing models and how to prepare new ones for specific applications. Typical analysis consists of the following steps:

  - Reading experimental time courses of stimulus and response functions from files to the so called observation matrix (command ROM).
  - The „raw“ data processing, covering corrections of different kinds, for example extrapolation or approximation of tail (command TAIL), half-life correction (TCF), raising background correction (TCBGR) and eventually „manual“ corrections, e.g. local smoothing, noise filtering (FFT) and normalisation of time courses (NORM). Further on integral characteristics of responses, i.e. moments (MOM) are usually calculated.
  - In the case, that the stimulus as well as the response functions are known, the impulse response of a system can be calculated using fast Fourier transform (FFT) method with regularised deconvolution. FFT can be also used for calculation of correlation functions and power spectral densities of processed responses.
  - Selection of suitable lumped parameter model represented by a system of ordinary differential equations for time courses of concentrations described in a script file (read by command RMODEL). A part of the script file are also specifications of model parameters, their bounds and default values – all these data can be modified (commands PARSET,PARLIM) and selected parameters (PARFIT) can be optimised by comparison with experimental responses using one of two available optimisation algorithms (OPTIMA or SOMA).

- ❑ The third group of users considers FEMINA as a more or less standard finite element program, which is little bit similar to the much bigger program COSMOS/M from point of view of range of applications and user interface. This part of FEMINA is the most demanding, because it is necessary to learn approximately 50 basic commands, names of finite elements (PIPE2D, PLANE2D, FLOW2D,…), names of calculated degrees of freedom (TEMP, PRES, UX, UY, …), names of important element parameters (ETau, ESxx - stresses, ERe - Reynolds, EQ - flowrates, …) and also names of system variables used for interactive definition of interpreted functions (XX, YY, TIME, ...). Finite element solution of a problem consists of the following steps:

  - Building geometrical model. Definition of region using entities POINT (PT), CURVE (CR), SURFACE (SF), VOLUME (VL). A basis of a model are points, as all other entities are defined in terms of points and therefore a change of a point position is automatically projected into changes of curves, surfaces and volumes.

- Building finite element model. Specified region is meshed by finite elements (EL) and nodal points (ND) connecting elements (remark: point PT is an entity of the geometrical model, while the node ND is the finite element model entity, these two notions should be distinguished). Any created element is defined by a list of nodes defining its geometry and further on by assigned group of parameters EGROUP (specifying algorithms), a group MPROP (material properties) and RCONST (real constants). In addition to the coordinates *x,y,z* several nodal parameters DOF (Degree Of Freedom), which are primary targets of solution, e.g. temperatures, pressures, velocities,…, are assigned to every node.
- As soon as nodes are formed, the boundary conditions, initial conditions and loading can be specified. It is possible to fix any DOF in any node (this is so called strong boundary condition) or just only to specify its initial value, which will be changing during calculations (initial conditions). In the same way it is possible to assign loads or more generally concentrated sources to any DOF, e.g. a point heat source to temperature or a force component to a displacement. Specified values of nodal parameters need not be constants, they can be functions of time, temperature, coordinates and these functions can be defined either as an algebraic expression or as a table of values.
- Calculation of selected nodal parameters by steady or transient solution of partial differential equations describing given problem. Transformation of differential equations to a system of algebraic equations is based upon weighted residual method, assembly and solution of this system is performed by frontal method.
- Postprocessing is usually performed automatically after solution and consists in calculation of quantities derived from DOFs (e.g. element stresses calculated from nodal displacement or velocities calculated from stream function).
- Graphical and numerical outputs.
- Data import and export. Data describing geometry and finite element model as well as results of calculations can be saved to text files, all database can be saved or loaded to binary files (database is formed by all important variables describing not only current status of finite element model, but also active residence time distribution models, observation matrix, i.e. all time responses, and others). Journal of processed commands and also of the control commands like loops, jumps,…(session file) is automatically created and can be modified by any text editor. The session file can be later on read and interpreted line by line. The session file represents parametrical description of a problem which can be solved for different values of parameters and thus it is possible to look for an optimal solution minimising deviation between model prediction and experimental data.

- ❑ We expect, that the fourth group of users will be formed, because FEMINA code is completed by modules for import results from FLUENT and COSMOS codes, by algorithms of collimated detectors γ-radiation and by effective methods for simulation stimulus-response experiment with suppressed distortion by numerical diffusion. Therefore FEMINA can be used as a CFD postprocessor, a tool for verification of CFD results.

2.1. Program control

Program FEMINA is controlled first of all by commands and not by using menu (only few commands are accessible also from menu and dialog windows). Commands can be prepared in advance as a text file and interpreted using command FILE (batch mode), or you can write them directly into the window CONSOLE as a line of text (interactive mode). You can mix the both modes freely during a session, some part is better to process as a batch another parts interactively. Structure of a command line is simple:

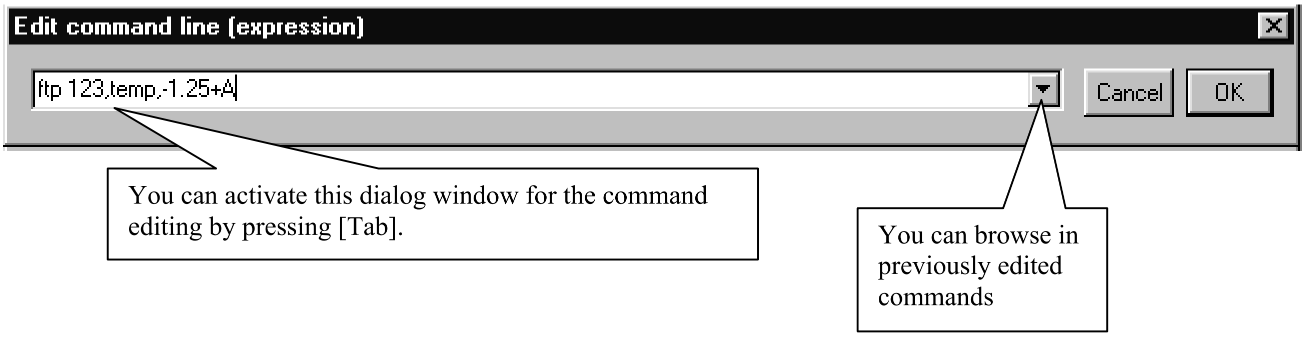
key-word list of parameters separated by spaces or commas, terminated eventually by semicolon;

Key-word is 6 characters long (more precisely, only first 6 characters are distinguished), and synonyms are frequently defined, e.g. MPL is the same as MPLIST (list of material parameters). Command parameters are processed as algebraic expressions, with constants, variables, arithmetic operators, parentheses, functions and so on. Parameters identifying calculated DOFs (Degree Of Freedom), element parameters or element names can be specified by name:

TEMP /temperature/, UX,UY,UZ /displacements/, RX,RY,RZ /rotations/, VOLT /voltage/, VX,VY,VZ /velocities/, PRES /pressure/, OMG /vorticity/, PS,PSX,PSY,PSXX,PSYY,PSXY /stream function and its derivatives/, CA,CN,CD /concentration/, EQ /flowrate/, PIPE2D, PLANE2D,FLOW2D /the most common finite elements for modelling pipes, plates, and two dimensional fluid flow/

Example:

FPT 123,TEMP,-1,25+A; prescribes temperature 25+A ^{0}C in the node nearest to the point 123.



ftp 123,temp,-1.25+A

You can activate this dialog window for the command editing by pressing [Tab].

You can browse in previously edited commands

Processing of each line begins by identification of the key-word (e.g. FPT in the previous example). If the word is not found in the list of standard commands, FEMINA tries to interpret the line as a simple script command according to the syntax which will be described later (the same script language is used for the RTD model definition). The simplest and the most frequent command is the assignment *Variable=expression*, where *Variable* is one from 250 system variables (full list of variables will be presented in chapter 3)[1]. For example A,B,…Z, are general purpose simple variables, NPT, ND, NE, NTC specify number of points, nodes, elements and time curves (responses), DT is a time step, NT number of observation points, vectors KX(i), CP(i), DEN(i), VISC(i) are material parameters of the *i*-th material group (thermal conductivity, specific heat capacity, density, viscosity,…). In the *expression* you can use any arithmetic operator (+,-,*,/,**), parentheses, standard functions (SIN(x),EXP(x),ABS(x),MIN(x,y…),…) and of course any user defined variables.

Example:

KX(1)=0.6 sets the value of thermal conductivity of the material parameter in the group number 1.

[1] You can also define your own variables or vectors, e.g. REAL GEORGE,JOHN(20) INTEGER DORIS(5), and use them in the same way as system variables. However, system variables cannot be overrided and e.g. REAL DT has no effect upon the system variable DT. The commands REAL,INTEGER are examples of script commands of interpreter.

It is not necessary to specify all parameters in the command line, because it is sufficient (and highly recommended for interactive work) to write only the keyword, the name of command, and all parameters will be in this case specified in a controlled dialog[2]. If a parameter ends with semicolon (even during dialog), the parameter processing ends and default values are assigned to remaining parameters.

Little bit different is processing commands which make use the mouse either for picking already defined entities (points PID, nodes NID, curves CID, elements EID, responses TID …) or for new points or nodes definition. For example the command for the definition of point can be written with all four parameters (PT index,X,Y,Z) and in this case the mouse is not needed. However, if we write only one parameter (PT index), the mouse is activated and the located point position is confirmed by left-click. In this way it is possible to create arbitrary number of points with successive indices using one command, because each left-click generates one point – the sequence ends with right-click.

A sequence of commands can be prepared in a file and executed as a program. The commands written into a batch file cannot comprise graphics, lists or optimisation commands (commands like GRAPH or OPTIMA will be skipped). The reason why the graphics is suppressed is obvious, because it would have no sense to project graphs or lists on screen in a fast sequence during run. The optimisation (commands OPTIMA and SOMA) has been excluded from batches because possible optimisation of optimisation algorithm itself would cause recursions and is therefore forbidden. On the other hand control commands, first of all #LOOP and #IF commands, can be used as a part of batch and applied for example for processing results in the NT-time steps:

```
T=0
#LOOP lab,NT
        processing results in time T
T=T+DT
#LABEL lab
```

Batch files can be created using any text processor, however it is much easier to create them in the interactive mode: the sequence RECORD ... ENDREC records all processed commands into a batch file automatically. A batch file can be processed either by the command FILE (this is a typical batch processed without an user intervention[3]) or issuing the command MACRO: in this case the commands are processed in fact in the interactive mode, including graphics and dialogs but without possibility to use e.g. LOOPs. Instead of MACRO you can write simply the name of batch file as a keyword and in this way it is possible to customise FEMINA by your own commands.

Program FEMINA works under operating systems Windows 95,98,2000. The following files should be placed into active directory:

FEMINA.EXE,	pre/postprocessor, optimisation
RUNFEM.EXE,	FEM solver
RUNFEMD.EXE,	FEM solver in double precision
RUNMOD.EXE,	RTD models solver
RUNBOX.EXE,	Cartesian boxing
RUNDET.EXE,	Collimated detector processing
RUNDET1.EXE,	Collimated detector processing
RUNMRG.EXE,	Mesh automatic a region
RUNMCT.EXE	Mesh interior of contours

[2] Please, take in mind, that for example W=3*L/U is *not* a basic command with the keyword W= and parameter 3*L/U. It is a script command of interpreter which cannot be obviously processed part by part in a controlled dialog.

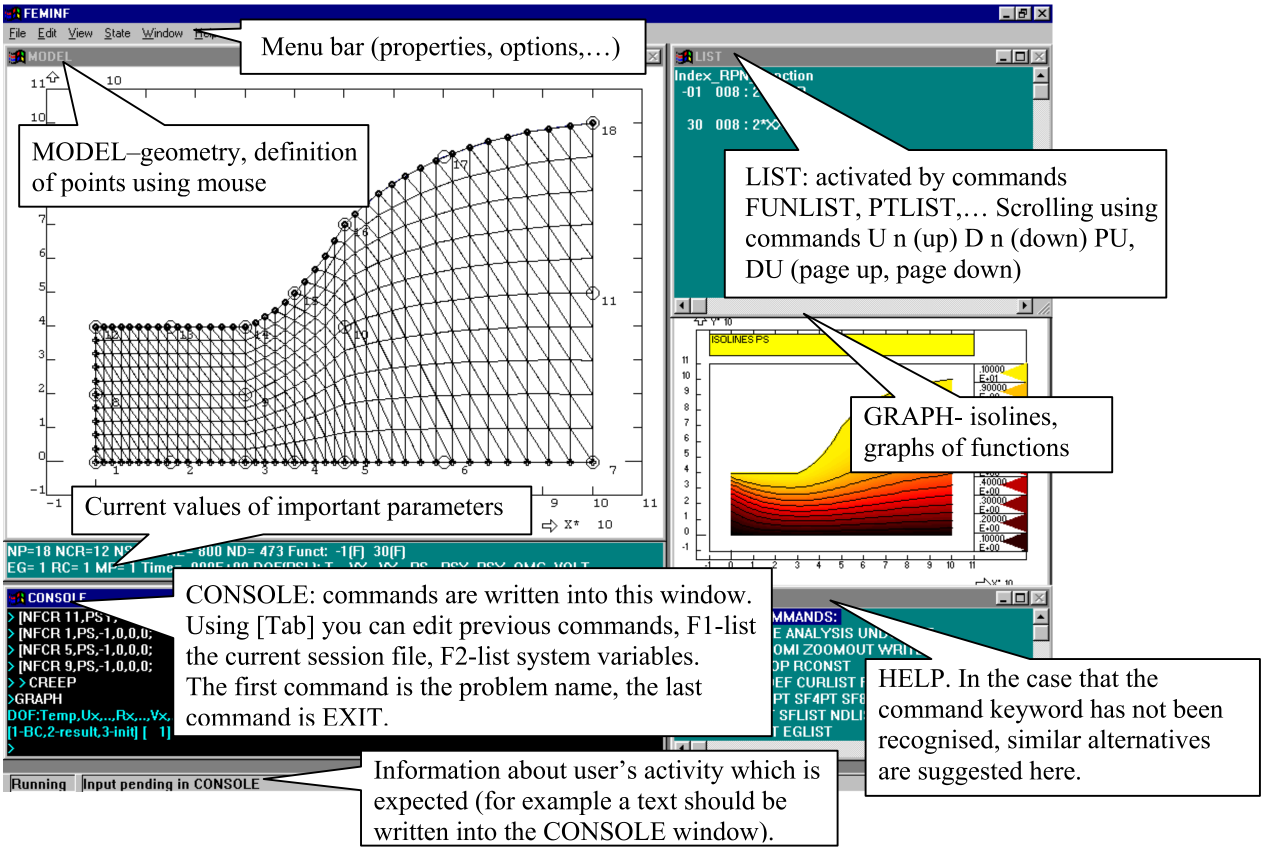
[3] The only important and frequently used exception is the command VALUE *variable* which always expects numerical value from keyboard which is assigned to the *variable*.

RUNMND.EXE	Create mesh of elements from defined set of nodal points
RUNMPT.EXE	Create mesh of elements from a set of key-pints
RANLREG.EXE,	Linear regression
RANNREG.EXE,	Nonlinear regression
RANNREG2.EXE,	Multiple nonlinear regression

Auxiliary files (model definition files, pictures) are located in the subdirectory /FILES/

$FEMINA.CMD,$FEMWIN.CFG,$FEMINA.TXT, $COMMAND.TXT, $VARLIST, $PUMPS.TXT, RTD model definition files *.mdt and also bitmap icons *.bmp. The file $FEMINA.CMD is a setup file (lists of RTD models, external programs) and $FEMWIN.CFG sets initial position and size of child windows; both the files can be modified by any text processor.

After start up FEMINA.EXE the following windows will be displayed:



Menu bar (properties, options,…)

MODEL–geometry, definition of points using mouse

LIST: activated by commands FUNLIST, PTLIST,… Scrolling using commands U n (up) D n (down) PU, DU (page up, page down)

GRAPH- isolines, graphs of functions

Current values of important parameters

```
NP=18 NCR=12 NS      800 ND= 473 Funct:  -1(F)  30(F)
EG= 1 RC= 1 MF= 1 Time
```

CONSOLE: commands are written into this window. Using [Tab] you can edit previous commands, F1-list the current session file, F2-list system variables. The first command is the problem name, the last command is EXIT.

```
> [NFCR 11,PS
> [NFCR 1,PS,-1,0,0,0;
> [NFCR 5,PS,-1,0,0,0;
> [NFCR 9,PS,-1,0,0,0;
> > CREEP
>GRAPH
DOF:Temp,Ux,..,Rx,..,Vx,
[1-BC,2-result,3-init] [   1]
>
```

HELP. In the case that the command keyword has not been recognised, similar alternatives are suggested here.

Information about user's activity which is expected (for example a text should be written into the CONSOLE window).

Program always begins with a problem name specification (e.g. TEST), which becomes the name of created files, specifically TEST.SES (session file, copy of commands), TEST.DBG (copy of listings being sent do display using commands PTLIST, ELIST,…), TEST.OUT and TEST.TEP (results).

The commands with parameters presented in the following examples are in fact copies of session files, and this text could have been literally retyped into the console window. However it is much better to write only key-words and to specify parameters in subsequent dialog – only then you can understand the parameter meaning.

The examples are presented in three nearly independent sections: Finite element models (2D temperature, 2D flow and 1D pipelines), RTD models and response processing (deconvolution), and the last section is devoted to model identification (optimisation).

2.2. Finite element models for transport processes

The command sequence is nearly the same for all finite-element problems: first reference points are defined (Points), then curves CR, surfaces SF and volumes VL. Afterwards, a mesh of finite elements and node points is generated on the curves, surfaces and volumes (commands MCR, MSF, or MVL). Boundary conditions are prescribed at selected nodes (commands NFCR, FPT, …). It is always necessary to specify definitions of material properties and other element characteristics (commands MPROP, RCONST, EGROUP), it does not matter whether it is done at the beginning or at the end of the problem specification. These parameters are different for different kinds of finite elements, the most important elements are PLANE2D (temperature, concentration and electrical potential in a plate), FLOW2D (fluid element, where velocities, pressures, stream functions, temperature, concentrations and electrical potential can be calculated), PIPE2D (pipes), PUMP and CSTR (Continuously Stirred Tank Reactor). Problem solution is always started by the SOLVE command with parameters which determine number of time steps and maximum number of iterations for flow, electrical field, temperature, concentrations (transport equations) and stresses calculation.

Equations for heat and mass transport are of this general form

$$\Lambda(\frac{\partial \phi}{\partial t} + \vec{u} \cdot \nabla \phi) = \nabla \cdot \lambda \nabla \phi + Q - S\phi \tag{2.2-1}$$

where ϕ is temperature T or concentrations c_N, c_D, c_A, while λ is the heat conductivity or diffusion coefficients D_N, D_D, D_A, respectively. Q is a source term and S is a sink; these terms can be defined (or modified) interactively by user as functions of time, coordinates, temperature, concentrations and other variables. Let as explain this concept in more details for the heat (enthalpy) transfer: the heat source $Q=Q_e+Q_u$ (units W/m^3) consists of two parts, the first one, Q_e, electric ohmic heating is predefined in FEMINA, while the second part Q_u can be supplied by user in form of a function (FUNDEF) or a table (CURDEF). Thus a dissipation of mechanical energy by friction, internal microwave heating, reaction enthalpies and similar production terms describing reactions kinetics for mass transport equations can be defined. Indices of functions used for definition of Q_u and S terms are specified as EGROUP parameters (there are 4 pairs of indices for T, c_N, c_D, c_A equations and one additional function defining hydraulic characteristics, together nine user defined functions for any kind of finite element). This is not only the source or the sink term, but also transport coefficients λ and Λ which can be defined by user as functions and the same holds for boundary and initial conditions thus ensuring maximum flexibility of problem specification.

The most frequently used commands in the following examples are:

SCALE *xmin,xmax,ymin,ymax*	ranges of axes X,Y
ZI, ZO	zoom in, zoom out (zooming is defined by mouse)

PT *i,x,y,z* reference point definition (i=index of point) of geometrical model (using a mouse)

CIRCLE *i,central point, point on perimeter, number of segments* circle described by two points (and created from at least 4 segments – parabolic curves)

CR2PT *i,i1,i2* curve number *i* described by 2 points *i1,i2* (straight line). A similar command CR3PT creates a quadratic curve from three points. It is possible to define a sequence of points and linear or quadratic curves simultaneously using the command CRSPOLY.

PMERGE this command merges and renumbers key-points having the same coordinates (or nearly the same, within the tolerance TOL). Curves and surfaces are merged too during this operation. It is recommended to perform this command after CRSPOLY and before surfaces will be created from curves. Anyway, PMERGE makes geometrical model more clear and transparent.

SF8FP *i,i1,i2,…,i8* surface described by 8 points (the first four points are corners of a curved quadrilateral). Surface (entity SF) is in this case a quadrilateral with four curved sides (CR), and these curves will be created automatically. Similar commands SF3PT, SF4PT form surfaces from 3 or 4 points respectively.

Note: Created entities (points, curves, surfaces, nodes, elements,…) can be displayed by commands with LIST at the end, e.g. PTLIST, CRLIST, SFLIST for points, curves and surfaces respectively. Entities can be plotted (PTPLOT, CRPLOT,...), or deleted (PTDEL, CRDEL,...) in a similar way.

MSF *index of surface, Nx,Ny, Lx,Ly, number of nodes, number of internal nodes*

generation of finite elements on a specified surface. All created elements are of the same type, they have the same index of the material properties group MPROP, real constants group RCONST and the group EGROUP specifying computational algorithm of element matrices. The number of nodes determines the element topology: for example 3 and 6 represent triangles with three and six nodes, 4 and 8 are quadrilateral elements. The parameters *Nx,Ny* determine the number of segments, dividing two adjacent sides of the quadrilateral surface, and *Lx,Ly* determines a non-uniformity of this division.[4] A positive value gives the ratio of the last and the first segment length (i.e. *Lx*=1 uniform division, *Lx*<1 element size decreases towards the side end). A side orientation, that is the identification of the beginning and the end, is depicted as a diamond symbol near beginning of side. A negative value of *Lx* is an instruction for generation of elements diminishing symmetrically towards the ends (*Lx*<-1) or towards the centre of side (-1<*Lx*<0). Similar operations are MCR and MVL, meshing curves or volumes respectively.

MRG *index of contour, number of nodes, number of internal nodes, default number of nodes on curve, smoothing iterations, subdivision of triangles, attractivity of boundary nodes, size, near*

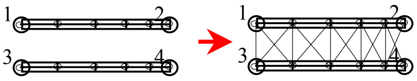
automatic generation of a triangular mesh on a 2D region defined by a contours (contour is a list of curves, specified by the command CT interactively). Size of generated elements is determined by predefined position of nodes along the contour and can be specified using command CRN for each curve of the contour individually (parameters of CRN are similar as in the following command MCRC). In the case that the number of nodes is not specified by the CRN command, the parameter *default number of nodes* is applied to each boundary curve. The parameter *Smoothing* concerns an iterative procedure which makes distances between nodes similar, with the exception of boundary points. These boundary nodes could attract the nearest internal nodes (*attractivity* >1) giving finer mesh along boundary curves. The parameter *subdivision of triangles* increases number of generated elements (=1 four-times, =2 sixteen-times and so on, because each triangle is decomposed to 4 triangles in one step). The last parameters *size, near* are not so important (default values 1 are usually the best), they have only a slight effect upon the shape and density of generated triangles.

Similar commands are MCT (mesh multiple contours) , MPT (mesh points – generates triangular elements connecting a set of key-points PT) and MND (mesh nodes). These commands are not aimed primarily for a finite element analysis, rather for situation, when you have a list of points (x_i,y_i) and corresponding values in a data file and you need to diplay contours, calculate surface integrals etc.

MCRC *CR-first, Nx, Last/First, Number of Nodes, RC-group, CR-last, CR-step*

generates *Nx* two-node elements (pipes, pumps or mixed tanks) on curves starting from index *CR-first*, up to the *CR-last* with step *CR-step*. The elements will be defined by parameters in the *RC-group* (these parameters determine the equivalent pipe diameter, cross section area, heat transfer coefficient at the outer side, ambient temperature T_e, and electric field intensity).

[4] If the division has been prescribed at an adjacent surface, these parameters are submitted as default.



MCR4 *CR pipe 1, CR pipe 2, RC-HEXC*

generates special 4-node elements *HEXC* (heat exchangers) which connects formerly defined *PIPE2D* elements on curves *CR pipe 1* and *CR pipe 2* (number of these elements must be the same on both curves otherwise the operation *MCR4* fails). See also *MCR2*, which creates *PIPE2D* and *HEXC* simultaneously on a given pair of curves.

NMERGE

To ensure connectivity of elements the operation which merges nodes with identical (or very close) coordinates *x,y,z* has to be performed. These nodes usually arise at the interface of two surfaces or curves (after MSF or MCR operations). The nodes with the distance shorter than TOL (this value can be assigned by a command, e.g. TOL=0.001) will be merged and renumbered.

NFCR *curve index, parameter name, parameter type, P1,P2,P3*

specifies boundary conditions or loads at nodes situated on, or near to a specified curve. *Parameter name* is for example TEMP-temperature, PRES-pressure, etc. *Parameter type (status)* is an integer value in the range from –10 to 50: a negative value means strong boundary condition (fixed value of parameter), zero denotes a free and therefore computed parameter, while a positive value indicates a load or a source term located at the corresponding node. At the same time, this value (parameter *status*) is an index of a user defined function of time, coordinates or temperature, which enables specification of variable boundary conditions or variable loads – this function multiplies the parameter value specified by the NFCR command, which is exactly the same procedure used by the COSMOS/M[5] program. If the corresponding function is not defined, identity is assumed and the computation proceeds with the directly entered constant values *P1,P2,P3* in reference points of the curve (two ending points and the middle one). These three values define a quadratic interpolating polynomial (a constant in case of identical values *P1,P2,P3*) for the selected parameter at nodes which lie or are very close to the specified curve. Similar commands are NFPT and NFSF for setting boundary conditions in the nearest node to a point PT or to the nodes on a surface SF.

INITIAL *parameter name, index of function*

specifies initial condition for a selected variable (TEMP, PRESS, CN,…) using a function. If the *index of function* is zero, zero initial condition applies.

FUNDEF *function index, f(XX,YY,ZZ,TIME,TEMP,DP,RE,DE,HE,...)*

This command defines a function as an algebraic expression (with common operators + - * / **, parenthesis ( ) and functions, sin,cos,abs,exp,log,…). The following names are predefined as independent variables: *XX,YY,ZZ* – coordinates, *TIME* – time, *TEMP* – temperature, and other not so important quantities. The function index is within the same range as the status of node parameters, i.e. from –10 to 50. These functions can be used for example with commands NFCR (see above) or MPROP (see below). Sometimes the desired behaviour cannot be properly described by a single algebraic expression, then a table of values and interpolation can be used, see operation CURDEF.

CURDEF *function index*,'*0-time,1-x,2-y,3-z,4-temp,5-ux,6-uy,7-uz,8-II'*, *No of points* (*max* 8),
x1,y1,x2,y2,x3,y3,x4,y4,x5,y5,x6,y6,x7,y7,x8,y8

[5] If -10<=*status*<=10, a transforming function $f_{status}(x,y,t,T)$ multiplies entered values DOF. These values are interpreted either as strong boundary conditions (*status*<0), or as a load (*status*>0). If *status*>10 the value of DOF meaning (if entered) depends on the context. For example, the value *status*>20 determines a node where the boundary condition of the third kind is prescribed for a specific DOF (and input value DOF is interpreted as the heat transfer coefficient).

MPROP *group index of material properties, element type, p1, f1, p2, f2, …*

Material properties of an element group. FEMINA considers 20 parameters, *p1* to *p20*: thermal conductivity, specific heat capacity, density, electrical conductivity, modulus of elasticity, Poisson constant, viscosity, thermal volumetric expansion, diffusion coefficient, enthalpy of formation, frequency factor, latent heat, melting point and others. The parameter *element type* selects material properties which are relevant for the specified element type (e.g. for the element PLANE2D – plate, the viscosity is not relevant). An index of a function *f1,f2*,… is assigned to each material parameter and if the index is non-zero and the function has been defined, the specified parameter value is multiplied by this function. In this way for example a temperature dependency of material properties can be defined.

RCONST *group index of real constants, element type, p1, p2,..*

Real parameters of elements which cannot be derived just only from the nodal coordinates, e.g. plate width, pipe diameter, heat transfer coefficient, wall roughness and so on.

EGROUP *index of element group, element type, i1,i2,…*

Integer element parameters, which usually specify a computational algorithm of element matrices. For example, they specify Cartesian or cylindrical coordinate system, number of points for Gauss numerical integration[6] or the solution method of Navier Stokes equations for fluid elements (e.g. FLOW2D). For PIPE elements, it is possible to specify an index of function describing a hydraulic characteristic and in this way for example a pump with a prescribed flow rate dependency on pressure drop can be modelled. Using EGROUP it is also possible to determine indices of functions which describe an internal (volumetric) source of heat and sources or sinks of species in a mixture, for most element types.

Remark: Indices of EGROUP, RCONST and MPROP are assigned to an element in the moment of the element creation (ACTSET). However, this assignment can be changed later, using commands ERMOD, ERMSF, ERMEL (for elements inside a region, on a surface, or picked by mouse).

SOLVE *t0,number of time steps,Δt,append,iter-electric,iter-flow,iter-thermal,iter-conc,iter-stress*

Start-up of the external program RUNFEM.EXE which reads the actual database and calculates electrical, flow, thermal, concentration and deformation fields (in this sequence) in every time step. Maximum number of iterations in single time step is specified for each of the mentioned analysis types – in case of zero number of iterations the corresponding analysis is omitted. Number of iterations should be greater than 1 in case of nonlinear and unsteady problems when it is necessary to iterate on each time level. If only a steady state solution is searched and accuracy in each time step is not important, then 1 can be specified as the number of iterations. Each time steps then represents one iteration step and the time step size plays the role of an underrelaxation factor – this can affects a convergence of the iteration process (the underrelaxation factor can be specified explicitly RELFAKT=…, the default value is 1)[7]. Results of each time step are stored in files with the suffix OUT (nodal parameters) and TEP (element parameters), and these files can be viewed by any text processor. After the RUNFEM program terminates, the control is return back to FEMINA, the database is updated and results of solution can be displayed as graphs or tables.

DSOLVE is the same as SOLVE, however instead of RUNFEM, the RUNFEMD program solving equations in double precision is invoked.

[6] You can specify 1,2,3,4 Gauss points for quadrilateral or brick elements, and 1,3,4,6,7,12,or 16 for triangular elements

[7] There are other parameters specifying methods and operational parameters of solver, use the command OPTION.

GRAPH *parameter name, zone of parameters* Displays contours of specified parameter on the basis of actual values in the database. Parameter zones go from 1 to 4: actual solution results are in zone 2, initial conditions are in zone 3, values of previous iteration are in zone 4. Zone 1 reflects only specified boundary conditions which is not of great importance when displaying.

GRATIM *parameter name* displays time history of a parameter in nodes selected by a mouse pointer. The file *problem.OUT* is used because results from previous time steps are not present in memory. Nodal values from all calculated time steps are transferred to the matrix of observation points for further processing or just for export into a file using command WRITE.

LTN *time*, and LTE *time* These commands read results corresponding to *time* from files *problem.OUT* (nodal parameters) and *problem.TEP* (element parameters), respectively. The parameter *time* need not match discrete calculated times because linear interpolation of both nodal and element parameters from two time levels will be applied.

NID Node identification by mouse, nodal parameters are reported in the status window. Similar commands are used for identification of elements, points, curves, graphs (EID, PID, CID, …).

IC *number of curve, function number* integral of a selected function along a curve (integration is numerical with NINT points, result is in the variable INTEGRAL). Integration is numerical and accuracy can be changed by system variable NINTG, that defines number of integration points (default value is NINTG=200).

IS *number of surface, function number* integral of a selected function on a surface (similar operation as IC).

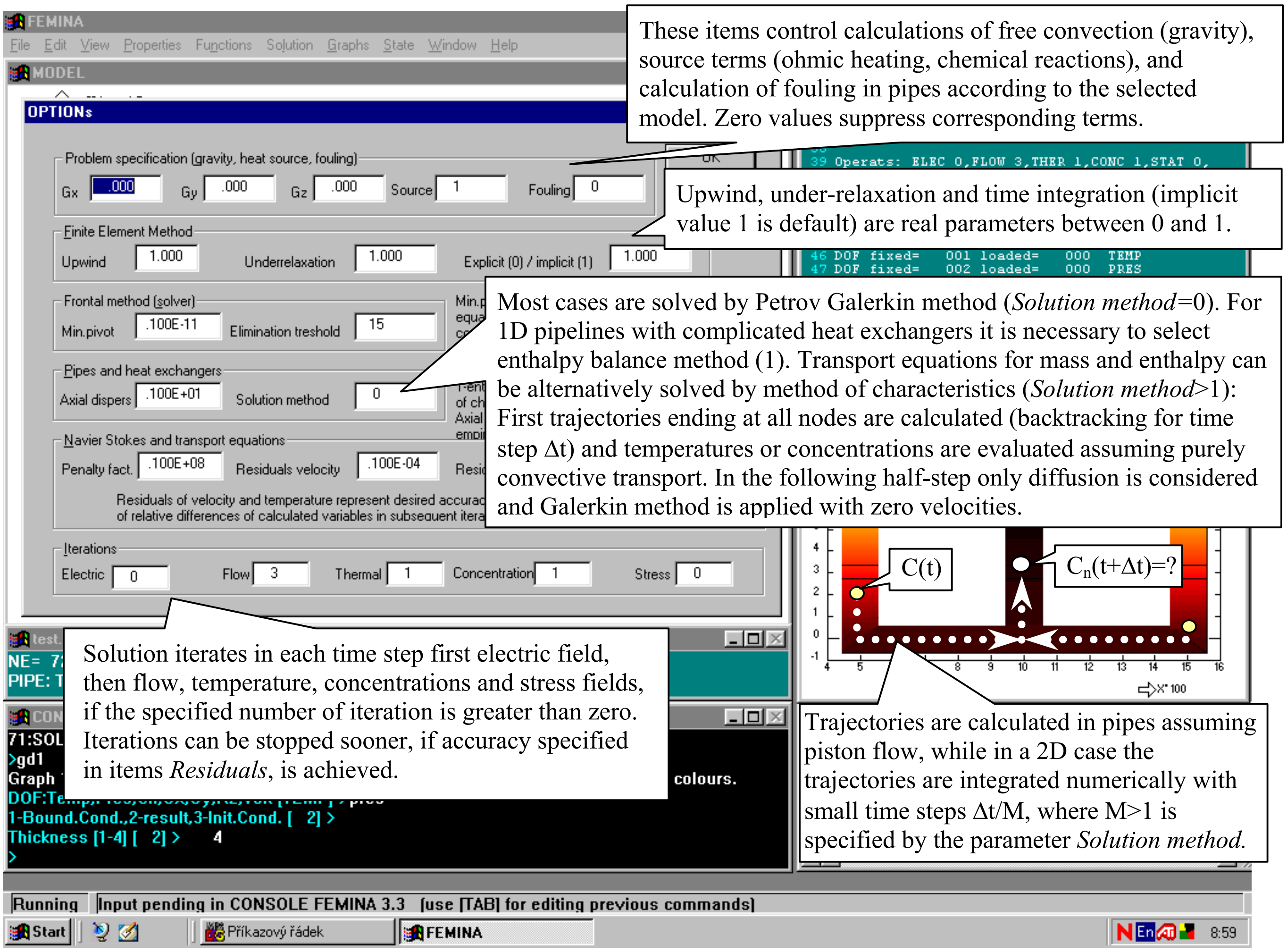
IE *dof, function number,e-first,e-last,e-step* integral of selected nodal parameter multiplied by a user function over specified range of elements. The elements can be 1D lines, 2D triangles or quadrilaterals or 3D solids, hexa- or tetragons. Result is in the system variable INTEGRAL.

EXIT Terminates FEMINA program.

Solution performed by RUNFEM.EXE and started by the command SOLVE is controlled by many parameters. Their default values can be changed either by using the command OPTION, or by system variables, e.g.

GY=-9.81	sets acceleration
OHMI=1	internal heat source terms will be respected (or suppressed if OHMI=0)
FOUL=1	selects the model of fouling
RUPW=1	sets upwind coefficient (=0 upwind suppressed)
TAUD=0	sets time constant in the model of axial dispersion (0-suppresses axial dispersion)
BL=0.3	reduces calculated thermal boundary layer in pipes to one third
HEPI=1	instead of Galerkin method, the method of enthalpy balances will be used (pipes)
BUOY=1	temperature dependent density is used for modelling buoyancy (=2 Boussinesque)
RELFAKT=.3	reduce underrelaxation factor when convergency problems are encountered

The same parameters ca be adjusted in the following dialog panel (menu bar *Solution*, *Option*):



OPTIONs

Problem specification (gravity, heat source, fouling)
Gx .000 Gy .000 Gz .000 Source 1 Fouling 0

Finite Element Method
Upwind 1.000 Underrelaxation 1.000 Explicit (0) / implicit (1) 1.000

Frontal method (solver)
Min.pivot .100E-11 Elimination treshold 15

Pipes and heat exchangers
Axial dispers .100E+01 Solution method 0

Navier Stokes and transport equations
Penalty fact. .100E+08 Residuals velocity .100E-04

Residuals of velocity and temperature represent desired accurac
of relative differences of calculated variables in subsequent itera

Iterations
Electric 0 Flow 3 Thermal 1 Concentration 1 Stress 0

```
39 Operats: ELEC 0,FLOW 3,THER 1,CONC 1,STAT 0,
46 DOF fixed=   001 loaded=   000  TEMP
47 DOF fixed=   002 loaded=   000  PRES
```

```
71:SOL
>gd1
Graph                                    colours.
DOF:Te
1-Bound.Cond.,2-result,3-Init.Cond. [  2] >
Thickness [1-4] [  2] >    4
>
```

```
Running   Input pending in CONSOLE FEMINA 3.3   [use [TAB] for editing previous commands]
```

These items control calculations of free convection (gravity), source terms (ohmic heating, chemical reactions), and calculation of fouling in pipes according to the selected model. Zero values suppress corresponding terms.

Upwind, under-relaxation and time integration (implicit value 1 is default) are real parameters between 0 and 1.

Most cases are solved by Petrov Galerkin method (*Solution method*=0). For 1D pipelines with complicated heat exchangers it is necessary to select enthalpy balance method (1). Transport equations for mass and enthalpy can be alternatively solved by method of characteristics (*Solution method*>1): First trajectories ending at all nodes are calculated (backtracking for time step Δt) and temperatures or concentrations are evaluated assuming purely convective transport. In the following half-step only diffusion is considered and Galerkin method is applied with zero velocities.

Solution iterates in each time step first electric field, then flow, temperature, concentrations and stress fields, if the specified number of iteration is greater than zero. Iterations can be stopped sooner, if accuracy specified in items *Residuals*, is achieved.

Trajectories are calculated in pipes assuming piston flow, while in a 2D case the trajectories are integrated numerically with small time steps Δt/M, where M>1 is specified by the parameter *Solution method*.

For more details look at the chapter 4 (theory).

Besides these basic commands it is useful to become familiar with some system variables frequently used in the finite element analysis

A,B,C,…,Z user variables (FEMINA does not interfere them, the variable I, which is used as a counter, is an exception)

ND, NE number of nodes and number of elements. For example you can delete a mesh by writing ND=0 and NE=0.

NPT,NCR,NSF,NVL number of points, curves, surfaces and volumes.

TIME, DT these variables are usually used in definition of time varying boundary conditions, e.g. the command FUNDEF –1,A+B*SIN(TIME*F) defines an oscillating function number -1.

XX,YY,ZZ coordinates used for example for definition of initial conditions.

TEMP, CN temperature and concentration used for material properties, production and sink terms definition.

It is true that material and element parameters can be specified using commands MPROP, EGROUP, RCONST or from menu *Properties*, however, sometimes it is easier to use the system variables:

TRANS(i),GAUSS(i),AXIS(i), HYDR(i),HEAT(i),HSINK(i),CNSRC(i),CNSINK(i)
transient(1)/steady(0) state, no. of Gauss points, coordinate system (0-Cartesian, 1-cylindrical), indices of functions defining hydraulic characteristics, heat source, heat sink, component c_N sources and sinks.

HH(i),DD(i),ALPHA(i),AREA(i) thickness, diameter of pipe, heat transf.coef., cross-section,

KX(i),CP(i),DENS(i),KAPPA(i),VISC(i),BETA(i),DN(i) thermal conductivity, heat capacity, density, electric conductivity, dynamic viscosity, thermal expansion, and diffusion coefficient in the i-th group EGROUP, RCONST or MPROP respectively.

Vectors of nodal parameters and interpolating functions of nodal parameters are also important
V1(LPU(*parameter name*)+I) value of selected parameter in the I-th node (boundary and loads)
V2(LPU(*parameter name*)+I) value of selected parameter in the I-th node (actual results)
V3(LPU(*parameter name*)+I) value of selected parameter in the I-th node (initial conditions)
The most import *parameter name* are TEMP (temperature), PRES (pressure), VX,VY,VZ (velocities), VOLT (voltage), CN,CD,CA (concentrations)

Examples: DISP V2(LPU(CN)+10) displays actual concentration c_N in 10th node.
V1(LPU(TEMP)+5)=A point heat source in the 5th node is set to the value A

Vector of element parameters makes use the function IEP(*parameter name*,*element index*) for calculation of a specified parameter index. Frequently used *parameter name* are ERE (Reynolds), EQ (volumetric flowrate), ETAU (shear stress, usually at wall), EPOWER (for example duty of a heat exchanger), EII (second invariant of rate of strain tensor), EFOUL (thickness of fouling layer), EDISS (dissipated power), ETX,ETY,ETZ (gradient of temperature), ETMEAN (mean temperature in an element), EMISE (von Mises stress), ELAV (Length or Area or Volume of element).

Example: DISP E1(IEP(EFOUL,5)) displays thickness of fouling in element 5

While the vectors V1(i),V2(i),V3(i) and E1(i) ensure a direct access to nodal and element parameters, the following functions DOF and EPA transfer interpolated values of nodal and element parameters at any point *x,y,z* (and not only at a node).

DOF(*parameter name*, *zone*, *x,y,z*) this function interpolates selected nodal parameter in specified zone (zone =1 boundary conditions and loads, zone=2 actual results, zone=3 initial conditions) at point x,y,z.

EPA(*parameter name*, *x,y,z*) this function interpolates parameter of elements, gradients, stresses and so on.

Remark: The functions DOF, EPA and also other operations which need to calculate values in a general point *x,y,z* can be effectively realised only if it is known inside which element the point *x,y,z* occurs. This relationship between a point and the corresponding element is hidden in the file $RUNBOX.BIN which is created by the operation BOX *h*. This operation encloses a finite element mesh into a fine rectangular grid with uniform spacing Δx, Δy, Δz and this spacing is determined by the parameter *h* (*h* is ratio of the grid size Δx, Δy, Δz to the smallest element size h_x, h_y, h_z , default value is 0.3). Then for each grid point an enclosing element is found (if such an element exists, of course) and its index is saved to the file $RUNBOX.BIN.

2.2.1. Temperature distribution in a plate

This example demonstrates a typical procedure of the finite element modelling, using simple geometry (rectangular plate with a hole), simple equation (heat transfer) and simple triangular elements (PLANE2D - only one degree of freedom TEMP is used at every node).

Heat transfer modelled by the element PLANE2D can be rather complex, and generally the following transport equation

$$\rho c_p (\frac{\partial T}{\partial t} + u_x \frac{\partial T}{\partial x} + u_y \frac{\partial T}{\partial y}) = \frac{\partial}{\partial x}(\lambda \frac{\partial T}{\partial x}) + \frac{\partial}{\partial y}(\lambda \frac{\partial T}{\partial y}) + \kappa[(\frac{\partial U}{\partial x})^2 + (\frac{\partial U}{\partial y})^2] + Q - ST \, , \quad (2.2.1\text{-}1)$$

or similar equation for cylindrical coordinate system, can be solved.

2.2.1.1. Steady state temperature distribution, Dirichlet boundary conditions

In the case that velocities are zero, that there are no production or sink terms (Q=S=κ=0), that the heat conductivity λ is constant and that we are looking only for a steady temperature field, the equation (1) reduces to the Laplace equation for temperature

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = 0 \, . \qquad (2.2.1\text{-}2)$$

And this is just this equation we shall try to solve first:

```
C* Opening test date:27.11.02 at 11:55
ANALYS 2;
EGROUP 1,PLANE2D,0,4,0,0;
```

EGROUP Element group 1 consists of PLANE2D elements (a plate). The following values determine the calculation procedure of local matrices. 0 – steady state solution, 4 - four-points Gauss integration (permitted values are 1,3,4,6,7,12,16 for triangular elements, and 1,2,3,4 for quadrilaterals), 0 – Cartesian coordinate system.

```
MPROP 1,PLANE2D,.6,0,4200,0,998,0,0,0,0,0,0,0,0,0;
```

MPROP Material property in group 1, necessary for elements PLANE2D: Kx=0.6 – thermal conductivity, transformation of Kx (0 - none), heat capacity cp=4200, transformation of cp, density ρ=998, transformation of ρ and so on (see dialog). In this case all functions have index zero, therefore all material properties are constant, independent of T,t,x,y,…

```
RCONST 1,PLANE2D,1,0,0,0;
```

RCONST sets real constants of group number 1 (next parameter 1 = means the plate thickness)

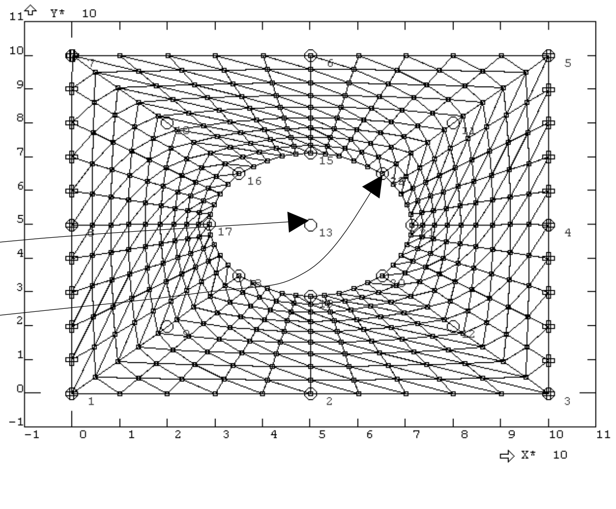
```
SCALE -0.1,1.1,-0.1,1.1;
PT 1,0,0;
PT 2,.5,0;
PT 3,1,0;
PT 4,1,.5;
PT 5,1,1;
PT 6,.5,1;
PT 7,0,1;
PT 8,0,.5;
PT 9,.2,.2;
PT 10,.2,.8;
PT 11,.8,.8;
PT 12,.8,.2;
PT 13,.5,.5;
PT 14,.65,.65;

CIRCLE 1,13,14;
```

Scale X,Y for graphs (xmin,xmax,ymin,ymax)

Reference points. Using a mouse, L-click confirms the selection, R-click finishes the entering sequence.

Circle center

Point on the circle, divided to 4 quadratic curves.



```
SF8PT 1,1,3,20,18,2,12,19,9;
SF8PT 2,3,5,14,20,4,11,21,12;
SF8PT 3,5,7,16,14,6,10,15,11;
SF8PT 4,7,1,18,16,8,9,17,10;
```

Surface from 8 points

```
A=15
B=14
```

Sometimes, it is useful to specify variables or expressions instead of numerical values of command parameters and the whole problem can be parametrised (we could use it in all previous commands describing geometry). In this case we will use parameters A,B to determine number of segments, on which surface edges will be divided when generating elements using MSF commands. Some identifiers are pre-declared, e.g. single-character general purpose variables A,B,C,…,Z.
Instead the assignments A=15 and B=15 the commands VALUE A and VALUE B could be used. In this case the variables would have been specified interactively even in the case, that the problem is processed as a batch.

```
MSF 1,A,B,1,1,6;
MSF 2,A,B,1,1,6;
MSF 3,A,B,1,1,6;
MSF 4,A,B,1,1,6;
```

Grid generation on surfaces 1 to 4, A x B, regular (1,1), triangles (6 nodes, permitted values 3,4,6,8)

```
NMERGE ;
```

Nodes with the same coordinates *x,y* is to be merged. Coincident nodes arise on surface boundaries because each MSF does not take into account that previous MSF command, applied on the adjacent surface, has already generated nodes on the coincident edge. Coincident nodes will be merged and renumbered using operation NMERGE.

At this moment, a grid of triangular elements *PLANE2D* is created and nodes with their parameters (*DOF*) are defined – we will use only one parameter, the temperature *TEMP*. Each *DOF* has an attribute, so called status of node parameter (integer number). So far, this status is zero at all nodes, which means that all *DOF* are to be calculated. Strong boundary conditions (prescribed temperatures on the plate edge in our case) are characterised by negative status (the range from –10 to –1 corresponds up to 10 different transformation functions, defining variable strong boundary conditions).

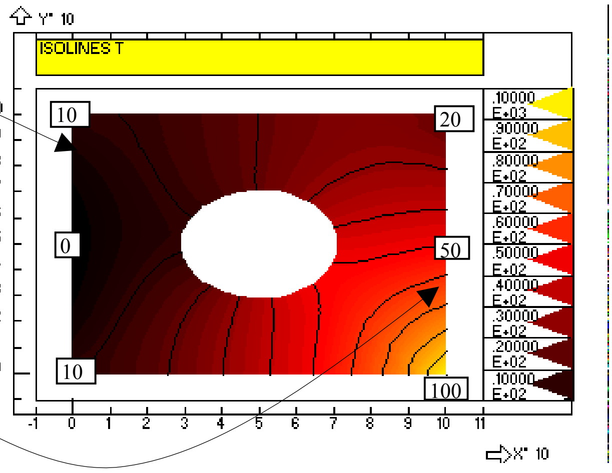
NFCR fixes values of the selected DOF on the curve 12

DOF

Entered values represents boundary conditions (minus sign) with transformation number –1. The transformation -1 is not defined, so the temperature on the boundary is fixed. The value 10 is specified at end-points, while at the mid-point it is 0 ^{0}C. This defines quadratic temperature profile on the curve 12.

```
NFCR 12,TEMP,-1,10,10,0;
```

```
NFCR 8,TEMP,-1,100,20,50;
```



```
SOLVE 0,1,10000,0,0,0,1,0,0;
```

The command SOLVE starts temperature analysis. 0 is initial time, 1=number of time steps, 10000=time step. Beacuse only steady-state analysis is demanded (see EGROUP), the value of time step is not important and results are independent of it. The following parameters specify maximum number of iterations in each time step for analysis of electrical field, flow field, temperatures, concentrations and stresses. Because our problem is linear, one iteration of temperature field calculation is sufficient. Calculated temperatures are written into a file xxxxx.OUT in alphanumeric form and they are transferred into a zone of "initial conditions" at the end of each time step.

```
GRAPH TEMP,2
```

The first parameter of the GRAPH command specifies a variable (here it is temperature TEMP) to be displayed as isolines. The second parameter (2) determines if the results (2) or initial conditions (3) will be presented. The GRAPH (or synonymum GD2) can be used only in 2D problems, for 1D graphs use its equivalent GD1. If you want to draw a temperature profile in a cross section use commands GCR or GFCR, see chapter 3.

The calculated results correspond to the Dirichlet boundary conditions of the first kind or to the Neumann boundary conditions of the second kind on a perfectly insulated boundary. In other words, the parts of boundary with no temperature specification are considered as insulated (note that the isotherms are perpendicular to the boundary in this case). In the following part we shall show how to include also boundary conditions of the third kind into the problem definition.

2.2.1.2.Steady state temperature distribution, Newton's boundary condition

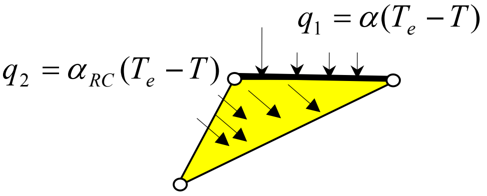
Newton's boundary conditions of the third kind (finite thermal resistance at boundary) are more complicated:

$$\lambda \frac{\partial T}{\partial n} = \alpha (T_e - T) \quad . \qquad (2.2.1\text{-}3)$$

First it is necessary to specify the part of boundary, where the heat transfer coefficient α and ambient temperature T_e are to be prescribed: The nodes lying in this part of boundary are assigned with status from 21 to 50 (remind that the values of status from –10 to –1 denote strong boundary conditions, status =0 has a calculated parameter without any source or load, and status 1 to 10 characterise calculated parameters with a point source and the values of status higher than 20 indicate boundary conditions of the third kind). Thus the prescribed value of nodal parameter, having status higher than 20, does not represent neither temperature nor a point source, but the heat transfer coefficient value. Ambient temperature is an element parameter (see RCONST) and can be specified either by calling RCONST, or more simply by assignment the system vector TE(*i*), where *i* is index of group RCONST:

```
TE(1)=200
```

Ambient temperature is assigned to elements having one or more sides on boundary, T_e=200 ^{0}C.

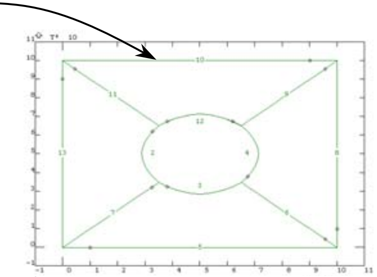


Remark: It seems to be strange, that not only the ambient temperate T_e but also some heat transfer coefficient α_{RC} appear as an RCONST parameter. However, this coefficient concerns the heat transfer at the element surface and not through the element boundary (see the heat flux q_2 in the figure) – what is interesting now is the heat flux q_1.

We shall specify the boundary condition of the third kind at the top of plate, which is the curve number 10.

```
NFCR 10,TEMP,21,100,100,100
```

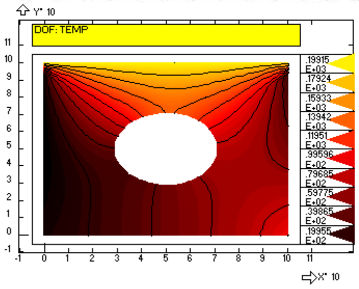
Set status 21 to TEMP in nodes on curve number 10. Heat transfer coefficient is constant, 100 W/m^{2}K (this value is assigned to all three definition points of curve, different values would define a parabolic function).



```
NID
```

This command NID (Node IDentification) has nothing to do with computation – it is only very useful tool for quick information about calculated values in nodes picked by mouse (similar command is EID for information about elements)

```
Node 0477 XY: .2000 1.0000 Zone 1: TEMP status[ 21] value= .100E+03
```



```
SOLVE 0,1,10000,0,0,0,1,0,0;
```

Let us summarise:

At the part of boundary with no boundary conditions, perfect insulation is assumed, the part with prescribed status -10,-9,...,-1 has prescribed temperatures (can be a function of *x,y*), the part with status 1,2,...,20 is insulated but with point sources of heat, and finally the part with status of nodes 21,22,... is characterised by heat transfer across the boundary with the heat transfer coefficient specified as the nodal parameter TEMP at input. Looking forward: The heat transfer coefficient need not be a constant as soon as a function with the index 21,22,..., corresponding to the status of node, has been defined by FUNDEF or CURDEF (therefore the heat transfer coefficient can be a function of calculated temperatures). Very specific is treatment of the part of boundary with status 21,22,... at direct ohmic heating: In this case besides the heat transfer resistance, the part of boundary is considered also as an electric current resistance and therefore a line source of heat (this option is important when calculating effect of fouling layer to distribution of temperature as well as electrical field).

2.2.1.3.Transient temperatures, initial conditions defined by function

Transient temperature field is a solution of partial differential equation

$$\rho c_p \frac{\partial T}{\partial t} = \lambda\left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2}\right). \qquad (2.2.1\text{-}4)$$

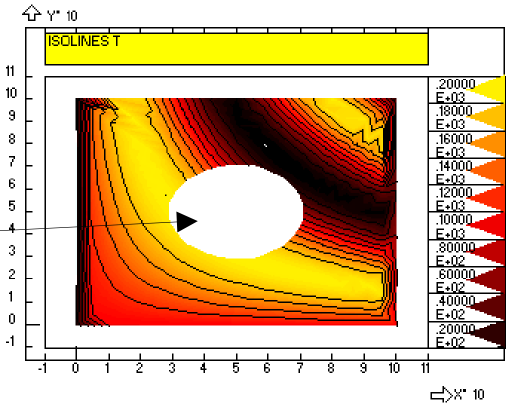
There exists a parameter EGROUP S/T (steady/transient), determining algorithm applied in individual elements. Some elements can be therefore set to transient according to Eq.(4) and other to a steady solution according to Eq.(2)[8]. However, there is only one group of elements in this case and therefore we switch to transient using the command EGROUP. Initial condition must be defined as a function of coordinates XX,YY.

Redefined attribute TRANSIENT (1)

```
EGROUP 1,PLANE2D,1,4,0,0;
```

Define transform No.1 (in this case as an expression). Variables TIME,TEMP,XX,YY,UX,UY can be used.

```
FUNDEF 1,100*(1+SIN(10*XX*YY));
```



Initial condition for temperature is defined as function number 1.

```
INITIA TEMP,1;
```

Plot contours of initial temperature field (initial conditions are in the zone 3).

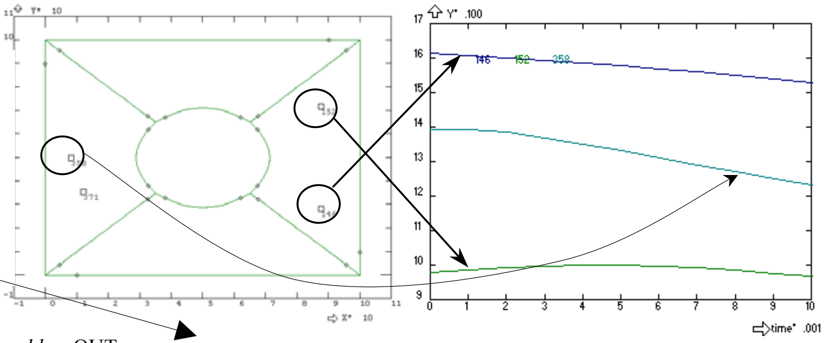
```
GRAPH TEMP,3
```

```
SOLVE 0,10,10000,0,0,0,1,0,0;
```

10 steps with a constant time step 10000 s.

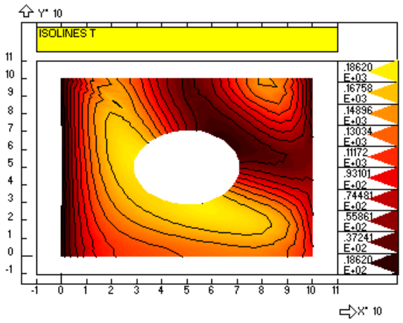
```
GRATIM TEMP
```

GRATIM - Using mouse it is possible to pick several nodes (L-click pick a node, R-click ends selection). Afterwards the time courses of specified parameter (in this case temperature) in the selected nodes will be plotted. Data are retrieved from the file *problem*.OUT and saved in memory as a table – you can list the table (TCL) and process the curves (TCF,…).



Results have been written into the text file *problem*.OUT (see example, 440 nodes and 10 time steps). It is possible to read data from this file into the zone of initial conditions using command LOADT.

```
THERMAL ANALYSIS test      ND=    440
         .0          1000.000 (time, dtime) INITIAL
     1    10.000
     2   100.000
         ....
   440   194.081
    1000.0           1000.000 (time, dtime)
     1    10.000
     2   101.050
         ....
   440      .000
     ....
   10000.0           1000.000 (time, dtime)
     1    10.000
         ....
   440      .000
```



[8] This can be important only in elements, which „react“ much faster than others, an example is combination of materials with very different thermal capacities and conductivities (this option enables to overcome the “stiff problem“).

2.2.1.4.Transient temperatures, boundary conditions defined by function

Boundary conditions as well as loads can be also functions of time. For example we shall show how to solve the problem when temperature at one quadrant of circle (curve number 3) oscillates with a constant frequency and amplitude:

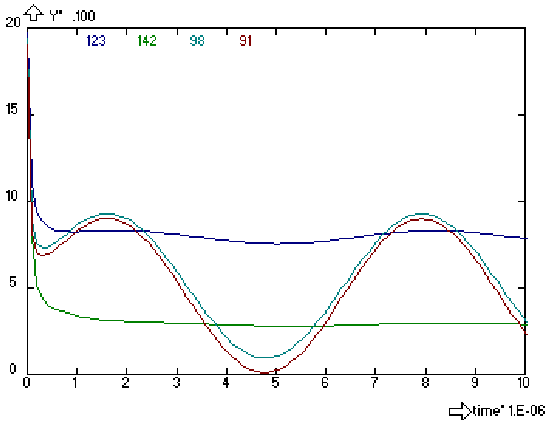
```
FUNDEF -2,50*(1+SIN(1E-6*TIME))
FUNLIST;
```

```
Index_RPN_Function
 -02  015 : 50*(1+SIN(1E-6*TIME))
 -01  005 :    1
  01  018 : 100*(1+SIN(10*XX*YY))
```

```
NFCR 3,TEMP,-2,1,1,1
```

Temperature (TEMP) is prescribed on curve 3 (one quadrant of circular hole) as a strong boundary condition number -2 (-2 is the status of temperature in nodes on curve 3). The last three values generally define quadratic course of temperature along the curve, which is reduced to a constant in our case (triplet of constant values 1,1,1). This basic course is multiplied by the function having the index of status, if such a function exists. And it exists in our case – thus the boundary values are product of the constant 1 times the function number -2.

```
INITIA TEMP,1
SOLVE 0,100,100000,0,0,0,1,0,0;

GRATIM TEMP
```



Let us summarise:

Simple initial conditions are defined by using FUNDEF (simple expression of variables XX,YY), while the time dependent boundary conditions by FUNDEF (expression with variable TIME) or also by CURDEF (small table of points t_i,y_i). Looking forward: More complicated function, defined by a program or by large tables of experimental points, can be defined too, see the end of this chapter.

2.2.1.5.Non-linear problems, temperature dependent conductivity

The problem has been linear so far, and iterations were not necessary. Non-linearity can be caused by temperature dependence of boundary conditions or material properties, e.g. thermal conductivity λ.

$$0=\frac{\partial}{\partial x}(\lambda\frac{\partial T}{\partial x})+\frac{\partial}{\partial y}(\lambda\frac{\partial T}{\partial y}) \qquad (2.2.1\text{-}5)$$

There is an index of a user defined function assigned to any material property and if such a function exists the material parameter (a constant) is multiplied by this function. Thus the material property can be defined as a function of temperature (TEMP), time (TIME), pressure (PRES), velocities (VX,VY,VZ), rate of strain (II) and others. In this example we shall substitute formerly constant thermal conductivity (λ=0.6 W/m/K) by the linear function of temperature λ=0.6 + 0.01 T.

```
EGROUP 1,PLANE2D,0,4,0
```

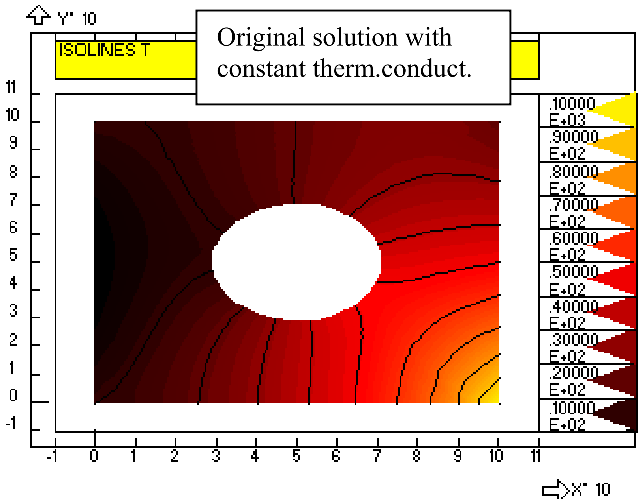
Reset attribute Static/Transient to static (0)

```
FUNDEF 2,0.6+0.01*temp
```

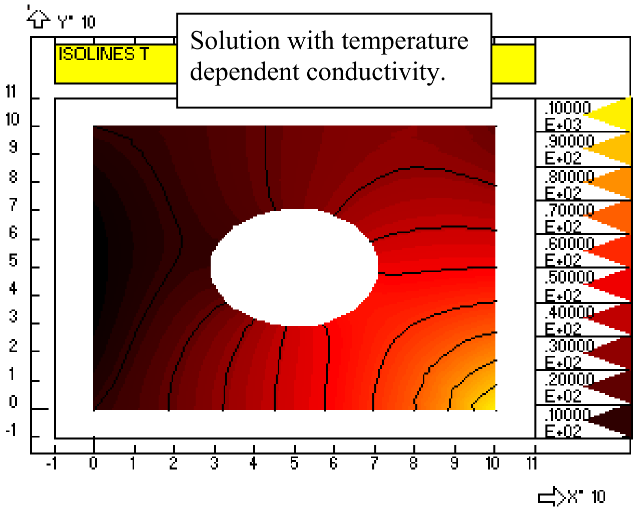
Material parameter *Kx* (=1.0) will be multiplied by function 2

```
MPROP 1,PLANE2D,1.0,2,4200,0,998,0,0,0,0,0,0,0,0,0;

SOLVE 0,1,1,0,0,0,5,0,0;
```

5 iteration steps are sufficient (and result will be independent of specified time step DT=1).
You can see (GRAPH) changes of contours due to nonlinearity.



Original solution with constant therm.conduct.



Solution with temperature dependent conductivity.

2.2.1.6.Internal heat sources

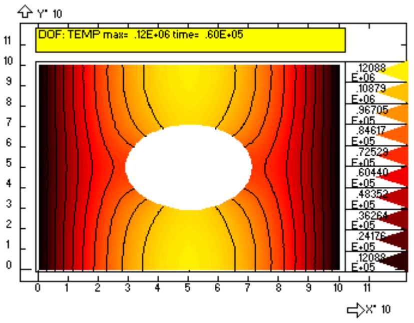
There are three possible ways how to define internal sources of heat: Volumetric heat source is calculated automatically if a voltage is applied and electrical conductivity is non-zero. Point sources of heat can be defined as loads in selected nodes (FND): if you define for example non-zero nodal parameter TEMP (e.g. 1.23) and positive status (within the range 1-10), the value 1.23 will not be a temperature, but intensity of heat source in the node (1.23 Watts). The third and the most frequent method is to prescribe the volumetric heat source Q (W/m^3) as a function f(TIME,TEMP,XX,….) or by a table of values.

$$0 = \lambda(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2}) + Q \qquad (2.2.1\text{-}6)$$

Index of this function Q must be specified as an EGROUP parameter (heat source), see example

```
FUNDEF 10,1E6*(1-XX);

EGROUP 1,PLANE2D,0,4,0,0,10;
```

$$Q = 10^6(1-x)$$



```
OHMI=1
```

Calculation of heat sources is suppressed as default. It must be switched on for example by using assignment OHMI=1, or using the command OPTION.

```
SOLVE .200E+05,1,.100E+05,0,0,0,1,0,0;
```

2.2.1.7.Ohmic heating and fouling

A special case of internal heat source is ohmic heating, where volumetric source depends upon intensity of electric field (intensity E is gradient of electric potential-voltage):

$$Q = \kappa[(\frac{\partial U}{\partial x})^2 + (\frac{\partial U}{\partial y})^2] \qquad (2.2.1\text{-}7)$$

where κ [S/m] is specific electrical conductivity. Computation of temperature field has to be preceded by solution of differential equation describing electric field. This equation is a perfect analogy of Fourier equation with electric conductivity replacing thermal conductivity

$$\frac{\partial}{\partial x}(\kappa\frac{\partial U}{\partial x})+\frac{\partial}{\partial y}(\kappa\frac{\partial U}{\partial y})=0. \qquad (2.2.1\text{-}8)$$

However aim of this paragraph is something different: to explain the way how to describe the effect of a thin layer of deposits at surface to heat transfer and ohmic heating. This is a typical problem of fouling: gradually forming layer of deposits, for example minerals or denaturated proteins, represents additional thermal and electrical resistance at wall. Thickness of this layer *h* is a key parameter and its time growth is described in FEMINA by a rate equation d*h*/d*t*=*f* , where *f* is a user defined function of temperature, shear stress, concentrations,... Properties of this layer are characterised by 3 parameters: thickness *h*, temperature conductivity λ_f (system variable KXF(i)) and electrical conductivity κ_f (KAPPAF(i)) and these parameters enable to calculate additional thermal resistance of the fouling layer ($r=h/\ \lambda_f$). More complicated is the case of ohmic heating, because then the fouling layer not only increases electric resistance but becomes additional line source of heat! FEMINA accounts for these effects by modification of Newton's boundary conditions for temperature and electric field. What is modified (automatically and only temporarily): external temperature (system variable TE(i)) and external voltage (UE(i)) as well as heat transfer coefficient and analogous electric current transfer coefficient.

It is obvious, that a lot of additional parameters have to be specified; it is possible to use commands MPROP, RCONST, EGROUP, but if only some parameters are to be changed the system variables assignment is more straightforward. The last approach will be used for the solution of case when outer perimeter is hold at zero temperature and voltage, while a part of the hole (curve 3) is an electrode with a layer of fouling (external voltage 220V, external heat transfer coefficient 100 W/m^2/K, fouling layer of the thickness 0.01 m is generated by an auxiliary rate equation assuming a constant fouling rate):

```
C* Rate of fouling defined by rate equation No.3
FUNDEF 3,A;
A=1E-7
RT=3
OHMI=1
KAPPA(1)=1
KAPPAF(1)=0.01
TE(1)=0
UE(1)=220
KX(1)=0.6
KXF(1)=0.2
NFCR 5,TEMP,-1,0,0,0;
NFCR 8,TEMP,-1,0,0,0;
NFCR 10,TEMP,-1,0,0,0;
NFCR 12,TEMP,-1,0,0,0;
NFCR 5,VOLT,-1,0,0,0;
NFCR 8,VOLT,-1,0,0,0;
NFCR 10,VOLT,-1,0,0,0;
NFCR 12,VOLT,-1,0,0,0;
NFCR 3,VOLT,21,1E10,1E10,1E10;
NFCR 3,TEMP,21,100,100,100;
INITIA TEMP,0;
TRANS(1)=1
SOLVE 0,10,.100E+05,0,1,0,1,0,0,0,0;
```

List of variables, which is to be defined for ohmic heating with a layer of fouling:
RT – model of fouling defined by function 3 (dh/dt=A)
OHMI – flag activating inner sources calculation
KAPPA(1) – electric conductivity of media (in this case material of plate, group 1)
KAPPAF(1) – electric conductivity [S/m] of fouling layer
TE(1) – external temperature [^{0}C] for elements in group RCONST = 1
UE(1) – external voltage [V] for elements in group RCONST=1
KX(1) – thermal conductivity [W/m/K] for elements in group MPROP=1
KXF(1)- thermal conductivity [W/m/K] of fouling layer

Status of voltage and temperature is changed to 21 on curve 3 – it means that at this part of boundary Newton's boundary conditions are prescribed. Value of nodal parameter VOLT is reciprocal electric resistence (10^{10} is a very high conductivity of metalic electrode) and heat transfer coefficient of parameter TEMP (100 – corresponds for example water cooled electrode)

2.2.1.8.How to calculate integral quantities (mean temperatures, heat flux,...)

An arbitrary function defined either by FUNDEF as an expression or by CURDEF as a table can be integrated along a curve (CR) or on a surface (SF) using commands IC (Integrate on Curve) or IS (Integrate on Surface)[9]. You can use a system function *DOF*(*parameter_name, zone, x,y,z*), giving value of any selected nodal parameter DOF in an arbitrary point *x,y,z* as a part of the integrand definition. The command BOX should be issued before the function DOF calling because the BOX command creates the file $RUNBOX.BIN defining correspondence between a general point *x,y,z*, and an element enclosing the point (this information is necessary for efficient interpolation of nodal parameters). We shall illustrate the procedure on example of calculation of the mean temperature on a selected surface.

```
BOX 0.3;
FUNDEF 11,DOF(TEMP,2,XX,YY,ZZ);
FUNDEF 12,1;
IS 1,11;
S=INTEGRAL
IS 1,12;
DISP S/INTEGRAL
```

Function number 11 is defined as a temperate in a general point x,y,z, and function 12 as identity (we shall use it for calculation of area). The command IS 1,11 calculates integral of temperature across a surface number 1 and result is in variable INTEGRAL. We save the value into variable S and calculate area by repeated integration IS 1,12. Mean temperature is then S/INTEGRAL.

```
EXIT
```

End of FEMINA

2.2.1.9.How to define more complicated functions

FUNDEF and CURDEF allow to define only simple functions. For example CURDEF enables to define a table with maximum of only 8 points. This restriction can be overcome easily by calling system function *CVT*(*i,x*), interpolating data in the i-th column of sufficiently large matrix of observation points, where the first column is independent variable *x* (e.g. time, temperature, coordinate,…). This approach will be explained in more details later in the RTD analysis.

In the case that a function cannot be defined as a single algebraic expression by FUNDEF, it is possible to define this function as a program in a script language used by FEMINA for interpreting commands or models. While the standard functions are referred by indices -10 up to 60 the extended functions are referred by indices -105, -104,-103,-102,-101 and 101,102,103,104,105 (negative indices for definition of Dirichlet boundary conditions, positive indices for Newton's boundary conditions, for variable material properties, or for more complicated initial conditions). The extended model definition is in fact identical with the description of models usually used for RTD models (see later) and the whole procedure will be explained on example: Functions, which we are going to use, must be written as a text into a file (for example with the name *func1.mdt*)

```
\\ini
if temp>100 then yv(1)=c+d*temp+e*temp**2 else yv(1)=a+b*temp
yv(2)=f*time*exp(-g*time)
\\
```

This definition file has to be announced to FEMINA in the initialisation file $FEMINA.CMD, where keywords (e.g. *F1*) are assigned to the model definition files (e.g. *func1.mdt*).

```
…
\\model
f1 func1.mdt
…
```

The selected model must be activated using the command *RM keyword*, and for example *RM F1* reads and compiles the file *func1.mdt*. Since then two new functions having indices 101 and 102 (and at the

[9] It is also possible to calculate integrals over selected finite elements using INTGEL DOF,function,... see example 2.2.1.12

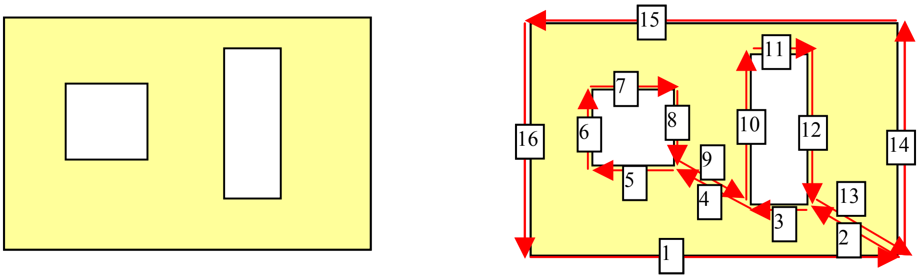
same time -101, -102 for strong boundary conditions) are available and their values are elements YV(1) and YV(2) in the model definition file above.

The extended functions 101,...,105 or -101,...,-105, defined in terms of the *.mdt script file, can be used whenever it is necessary to perform a more specific actions: for example if we need to prescribe boundary condition or a source at a node, and intensity of the source is to be calculated by integration of flowrate (element parameter). Example of a more complicated function, describing attenuation of radiation in a mixed vessel with internal baffles, will be presented later, in chapter 2.5. The extended functions can be also used as „hooks“, called at the beginning of each time step of the FE solution – index of this function must be specified in the menu OPTION (in this case the script file makes use the variable I which informs us of the type of operation, I=1-voltage, I=2-flow, I=3-thermal, I=4,5,6-concentrations, I=7-stress analysis).

2.2.1.10. Advanced meshing

Meshing is a formation of elements and nodes (finite element entities) using information from a geometrical model. In the previous paragraphs and also in most of the following chapters the so called parametric meshing is used, based upon decomposition of complex geometrical model to simple forms, like surfaces with three or four sides (curves) or in the case of 3D into hexahedral “bricks”. These simple elements are covered by a structured mesh of M x N elements. Parametric meshing results in a mesh which is quite under your control and you can draw a mesh approximately by hand or you can calculate the number of generated elements in advance. However, sometimes it is rather laborious to decompose a model into many small surfaces and volumes, and still, especially if you are working with parametrically defined geometry (for example if the radius and the position of the hole in rectangle is defined by variables), an optimum meshing need not be achived. In this case it is better to use automatic generators of triangular mesh, which fills a region defined by a contour. The contour is nothing else than the list of succesive curves, which is closed. The list of curves is created by the command CT *index of contour* and the counter-clockwise oriented list of curves is specified by mouse (left-click, right-click as usually).

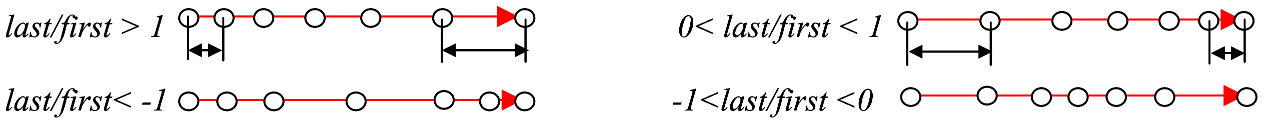
The command MRG (Mesh ReGion) is capable to fill interior of only one region and therefore in the case of multiply connected regions you have to include inner boundaries into a one closed contour for example in the way shown in the following figure



Further on it is desirable to specify location of nodes along the contour, it means at curves defining this contour. This is done by the command

CRN *index of curve, number of nodes, last/first*

with parameters having the same meaning as parameters of the command MSF used previously. The parameter *last/first* determines uniformity of distribution of nodes along the curve[10]



[10] Values *last/first*=1 or –1 correspond to uniform distribution of nodes (equidistant nodes).

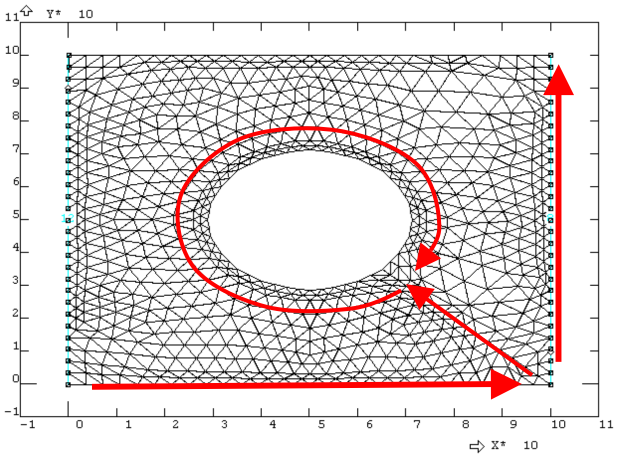
The command CRN does not create nodes, it only assigns the specified parameters to selected curve and this information is used later by the command MRG

MRG *index of contour, number of nodes, number of internal nodes, default number of nodes on curve, smoothing iterations, subdivision of triangles, attractivity of boundary nodes, size, near*

The MRG command creates triagular elements with 3 or 6 nodes (this is a restriction with respect to the MSF command which enables generation of a more complicated elements). Parameters of MRG seems to be complicated, however in most cases default values can be accepted.

We shall demonstrate the whole procedure using previous geometry – rectangle with a hole.

```
NE=0
ND=0
CTR 1,10,5,6,3,2,1,4,6,8,10,12;
CRN 5,15,1;
CRN 8,15,1;
CRN 10,15,1;
CRN 12,15,1;
CRN 6,9,1;
CRN 1,5,1;
CRN 2,5,1;
CRN 3,5,1;
CRN 4,5,1;
MRG 1,3,0,2,50,0,2,1,1;
```

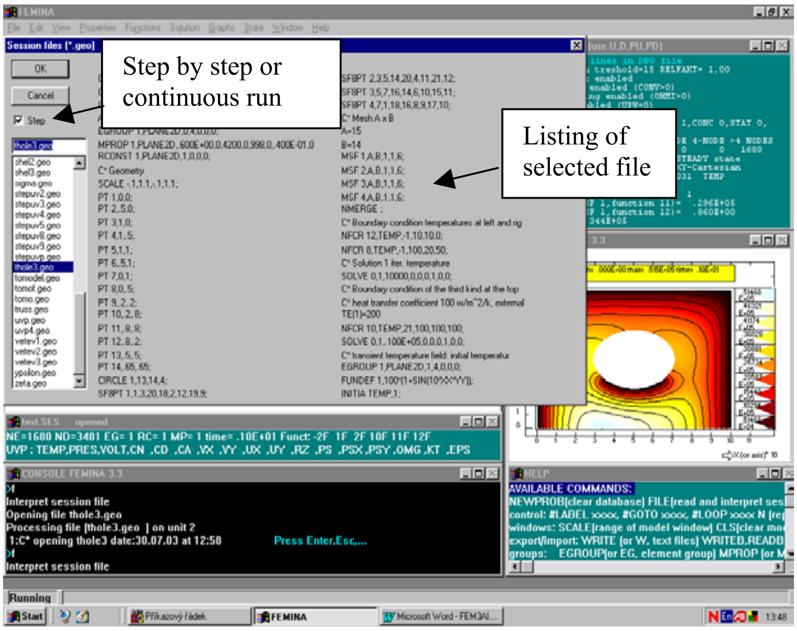


2.2.1.11. Processing problem as a batch or by using macro

The whole rather long example is available also as the session file THOLE3.GEO which is a part of the installation package. You can run the session either as a batch (command F) or by using the command M (macro) step by step or continuously. Batch processing is necessary for solution of problems requiring repetition of commands in a loop, macro is better for interactive processing:

M

The command M activates selection of files with postfix .GEO for interpretation.



Step by step or continuous run

Listing of selected file

```
EGROUP 1,PLANE2D,0,4,0,0,0;            SF8PT 2,3,5,14,20,4,11,21,12;
MPROP 1,PLANE2D,.600E+00,0,4200,0,998,0,.400E-01,0   SF8PT 3,5,7,16,14,6,10,15,11;
RCONST 1,PLANE2D,1,0,0,0;              SF8PT 4,7,1,18,16,8,9,17,10;
C* Geometry                            C* Mesh A x B
SCALE -,1,1,1,-,1,1,1;                 A=15
PT 1,0,0;                              B=14
PT 2,.5,0;                             MSF 1,A,B,1,1,6;
PT 3,1,0;                              MSF 2,A,B,1,1,6;
PT 4,1,.5;                             MSF 3,A,B,1,1,6;
PT 5,1,1;                              MSF 4,A,B,1,1,6;
PT 6,.5,1;                             NMERGE ;
PT 7,0,1;                              C* Boundary condition temperatures at left and rig
PT 8,0,.5;                             NFCR 12,TEMP,-1,10,10,0;
PT 9,.2,.2;                            NFCR 8,TEMP,-1,100,20,50;
PT 10,.2,.8;                           C* Solution 1 iter. temperature
PT 11,.8,.8;                           SOLVE 0,1,10000,0,0,0,1,0,0;
PT 12,.8,.2;                           C* Boundary condition of the third kind at the top
PT 13,.5,.5;                           C* heat transfer coefficient 100 w/m^2/k, external
PT 14,.65,.65;                         TE(1)=200
CIRCLE 1,13,14,4;                      NFCR 10,TEMP,21,100,100,100;
SF8PT 1,1,3,20,18,2,12,19,9;           SOLVE 0,1,.100E+05,0,0,0,1,0,0;
                                       C* transient temperature field: initial temperatur
                                       EGROUP 1,PLANE2D,1,4,0,0,0;
                                       FUNDEF 1,100*(1+SIN(10*XX*YY));
                                       INITIA TEMP,1;
```

```
test.SES    opened
NE=1680 ND=3481 EG= 1 RC= 1 MP= 1 time= .10E+01 Funct: -2F  1F  2F 10F 11F 12F
UVP : TEMP,PRES,VOLT,CN  ,CD  ,CA  ,VX  ,VY  ,UX  ,UY  ,RZ  ,PS  ,PSX ,PSY ,OMG ,KT  ,EPS
```

```
CONSOLE FEMINA 3.3
>f
Interpret session file
Opening file thole3.geo
Processing file [thole3.geo  ] on unit 2
 1:C* opening thole3 date:30.07.03 at 12:58            Press Enter,Esc,...
>f
Interpret session file
```

```
HELP
AVAILABLE COMMANDS:
NEWPROB[clear database] FILE[read and interpret ses
control: #LABEL xxxx, #GOTO xxxx, #LOOP xxxx N [re
windows: SCALE[range of model window] CLS[clear mo
export/import: WRITE [or W, text files] WRITEB,READB
groups:   EGROUP[or EG, element group] MPROP [or M
```

2.2.1.12. Processing a batch with loops

This example demonstrates how to prepare a parametric model, how to repeat solution for different parameters using a loop, and how to store results. Suitable problem is evaluation of hydraulic characteristics of a duct with different cross-sections. Steady and fully developed flow of Newtonian liquid in such a duct is described by the following Poisson equation for axial velocity u(x,y)

$$\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}=-P \qquad (2.2.1\text{-}9)$$

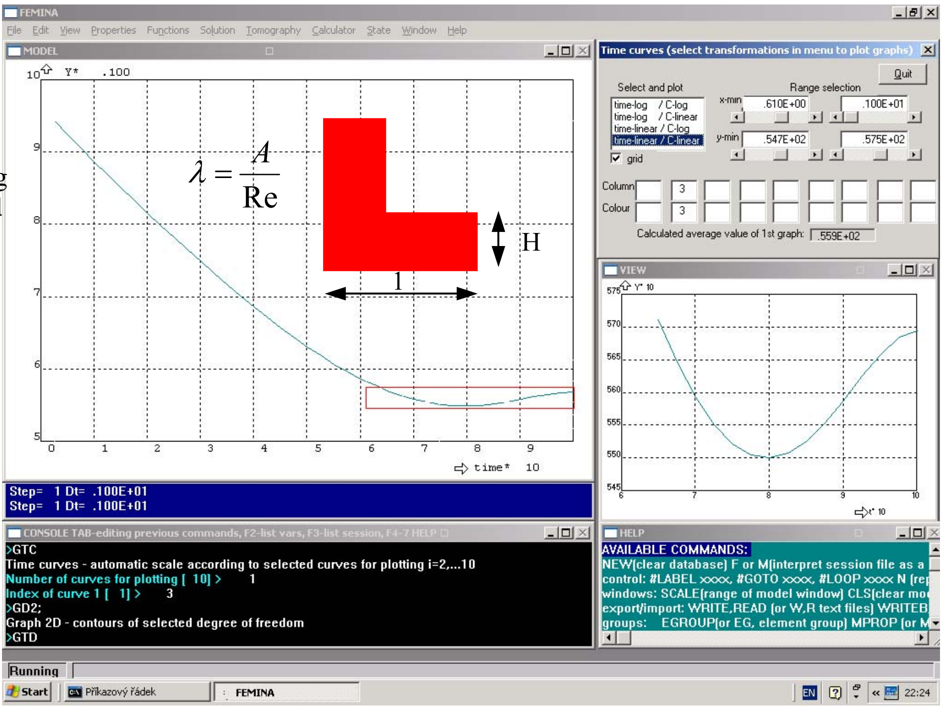
where P is an axial gradient of pressure along the duct, divided by viscosity. As soon as we are able to solve the distribution of velocity u(x,y) for an arbitrary P, we can calculate hydraulic characteristics, relationship between pressure gradient and mean velocity. This characteristic is usually expressed in terms of friction factor λ or f (Fanning friction factor), which is indirectly proportional to Reynolds number for laminar flows

$$\lambda=\frac{A}{\mathrm{Re}} \qquad (2.2.1\text{-}10)$$

where A is a constant, that depends only upon the shape of cross section (A=64 for circular cross-section of pipe, A=96 for very narrow slit). The following batch calculates A for L-profiles with different relative thickness of arms H/L=0.025,0.05,…,0.975 solving temperature field which is also described by Eq.(9). This approach is an application of principle of analogy between temperature, electric of velocity fields. Value P (=1) corresponds to a constant heat source (function =1) and unit material properties (unit heat capacity, density and thermal conductivity).

```
H=.025
OHMI=1
K=0
#LOOP LAB,39;
K=K+1
NPT=0
NCR=0
NSF=0
NE=0
ND=0
PT 1,0,0;
PT 2,1,0;
PT 3,1,H;
PT 4,H,H;
PT 5,H,1;
PT 6,0,1;
SF4PT 1,1,2,3,4;
SF4PT 2,1,4,5,6;
MSF 1,50,40,-1.5,-1.5,4,
MSF 2,40,50,-.150E+01,-1
NMERGE ;
NFCR 1,TEMP,-1,0,0;
NFCR 2,TEMP,-1,0,0;
NFCR 3,TEMP,-1,0,0;
NFCR 5,TEMP,-1,0,0;
NFCR 6,TEMP,-1,0,0;
NFCR 7,TEMP,-1,0,0;
FUNDEF 1,1;
EGROUP 1,PLANE2D,0,3,0,0,1,0,0,0,0,0,0,0,0;
RCONST 1,PLANE2D,1,0,0,0,0,0;
```

LOOP will be repeated 39-times, with increasing H (thickness of arm) and K (counter)



Time curves (select transformations in menu to plot graphs)

Select and plot

- time-log / C-log
- time-log / C-linear
- time-linear / C-log
- time-linear / C-linear

☑ grid

Range selection

x-min .610E+00 .100E+01

y-min .547E+02 .575E+02

Column 3

Colour 3

Calculated average value of 1st graph: .559E+02

```
Step=  1 Dt=  .100E+01
Step=  1 Dt=  .100E+01
```

CONSOLE TAB-editing previous commands, F2-list vars, F3-list session, F4-7 HELP

```
>GTC
Time curves - automatic scale according to selected curves for plotting i=2,...10
Number of curves for plotting [  10] >      1
Index of curve 1 [   1] >      3
>GD2;
Graph 2D - contours of selected degree of freedom
>GTD
```

HELP

AVAILABLE COMMANDS:
NEW(clear database) F or M(interpret session file as a
control: #LABEL xxxx, #GOTO xxxx, #LOOP xxxx N (rep
windows: SCALE(range of model window) CLS(clear mo
export/import: WRITE,READ (or W,R text files) WRITEB
groups: EGROUP(or EG, element group) MPROP (or M

Function number 1 defines unit heat source

```
MPROP 1,PLANE2D,1,0,1,0,1,0,.400E-01,0,.210E+12,0,.280E+00,0,.120E-04,
0,0,0,0,0,.100E-08,0,.100E-08,0,.100E-08,0,0,0,0,0,0,0,0,0,0,0;
SOLVE 0,1,1,0,0,0,1,0,0,0,0;
INTGEL TEMP,1,1,NE,1;
C1(K)=H
C2(K)=INTEGRAL
C3(K)=2*(2*C1(K)-C1(K)**2)**3/C2(K)
H=H+.025
#LABEL LAB;
NT=39
GTC 1,3;
```

Operation INTGEL integrates temperatures multiplied by function number 1 (in this case it is identity) over all NE finite elements. Result is in the system variable INTEGRAL.

Value A in Eq.(10) is placed into 3rd column of matrix of observation points with NT=39 rows. This final result can be listed by TCL or plotted using GTC (see the Figure above, parameters GTC 1,3 denote only one graph, and 3rd column).

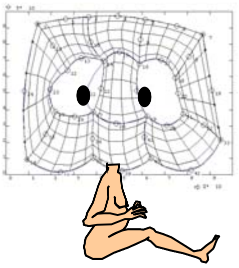
Remark: In this example the integration over geometrical entities (over surfaces 1 and 2) was substituted by numerical integration over elements (command INTGEL instead of INTGSF). In this case it is not necessary to use the box operation (BOX) which speeds up computation substantially.

2.2.1.13. Debugging

A short notice at the end of this introductory example: How to debug a problem? What to do when your results are obviously wrong or even none? You should realise that the solution is described only by finite elements, nodes, EGROUP, RCONST, MPROP groups, by defined functions and by parameters of solver. Geometrical entities like points PT, curves CR and so on, are not important and need not be checked. Recommended procedure is as follows:

1. Check number of elements, nodes and groups looking at the status window.
2. Look at the LIST window (or to the *.DBG file), where information about setting of switches (coordinate system, steady/transient, upwind, …) and elements used in the problem are displayed.
3. Use commands ELIST, NDLIST, NFLIST to check elements and nodes. You should identify missing boundary conditions, coincident nodes or wrong number of nodes of specified kinds of elements (for example PIPE2D should have 2 nodes). Sometimes it is easier to use NID for checking boundary conditions (parameters in zone 1) and status of nodal parameters. In the same way EID can be used for checking elements and associated groups EGROUP, RCONST, MPROP.
4. Check functions using FUNLIS (RPN parameter, length of translated function, should be greater than zero for each function) or simply by GF (graph of function).
5. Problems with solution can be caused by improper setting of time step, maximum number of iterations, relaxation factor, number of Gauss points for numerical integration, wrong or missing material parameters (zero viscosity, thermal conductivity, diffusivity), parameters of elements (e.g. zero thickness of plate, zero diameter of tube). Use tables in menu *Properties* and *Solution* for checking.
6. Execute RUNFEM.EXE manually after exiting FEMINA program. In this way you can locate errors like division by zero, overflow,…

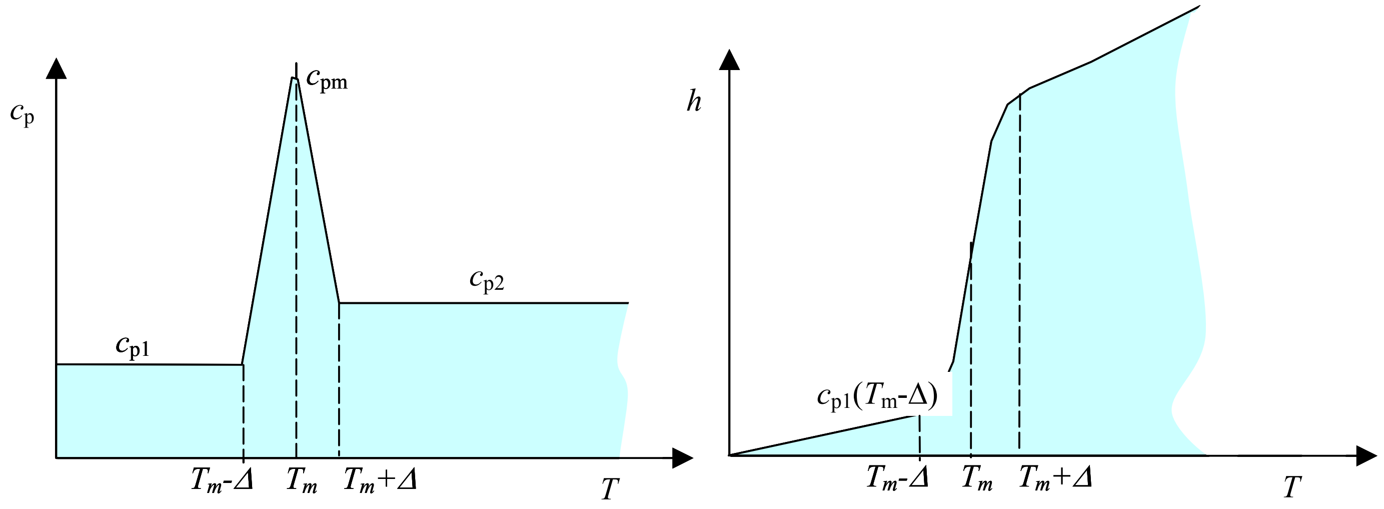
Anyway, FEMINA is not free from errors. In doubts do not hesitate to send a message to the address Rudolf.Zitny@fs.cvut.cz .



2.2.2. Phase changes (melting/solidification)

Previous example analysed a steady or transient temperature field in a body (it was a solid block, however it could have been a flowing fluid as well) without phase changes. Analysis of freezing or melting is more complicated and several different methods can be used:

The simplest approach is based upon specification of temperature dependent specific heat capacity $c_p(T)$ in the following way (the function is defined as a table using CURDEF command)

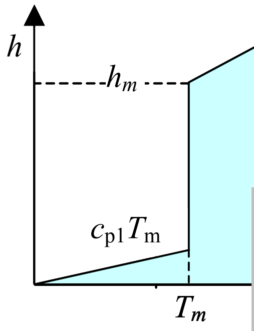


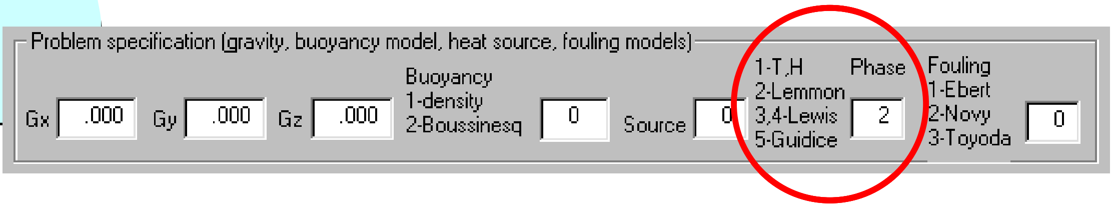
where peek of specific heat capacity

$$c_{pm} = \frac{r}{\Delta} + \frac{1}{2}(c_{p1} + c_{p2})$$

depends upon the latent heat r , selected width of the phase change temperature range Δ and also upon specific heats of solid and liquid phases. This method cannot accurately describe an isothermal phase change, because the melting temperature range Δ cannot be zero and in fact must be rather high or the time step must be extremely small.

FEMINA offers also methods based upon enthalpy balances, where melting point T_m [^{0}C] and melting enthalpy h_m [J/kg] are explicitly specified as material parameters. These methods are selected by the parameter *Phase Change* in the OPTION command or in the OPTION menu (selected from the toolbar *Solution*):





Problem specification (gravity, buoyancy model, heat source, fouling models)

Gx .000 Gy .000 Gz .000

Buoyancy 1-density 2-Boussinesq: 0

Source: 0

Phase 1-T,H 2-Lemmon 3,4-Lewis 5-Guidice: 2

Fouling 1-Ebert 2-Novy 3-Toyoda: 0

0 - this is the previously described method based upon temperature dependent c_p

1 - two nodal parameters, temperature T and specific enthalpy h, are computed simultaneously.

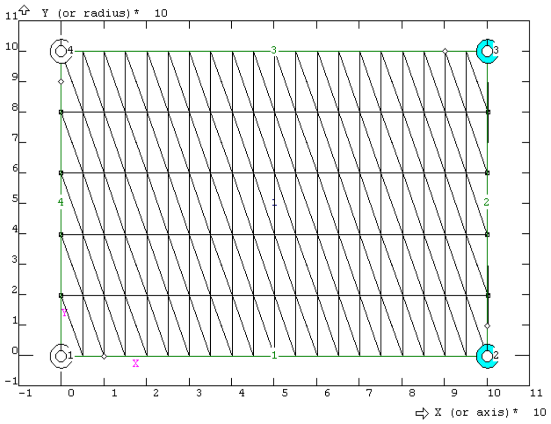
The following methods take into account only temperature as a computed nodal parameter, enthalpy is calculated internally at an element level and the methods differs by the way how the term dh/dT is approximated using the nodal temperatures and associated enthalpies

2 - Lemmon's method

3,4 - Lewis, Morgan

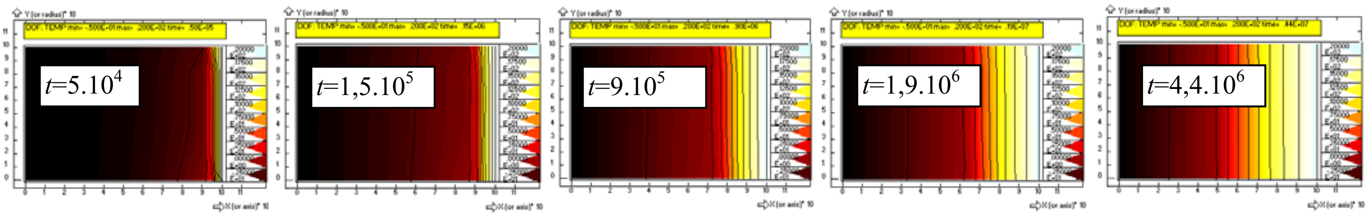
5 - Del Guidice.

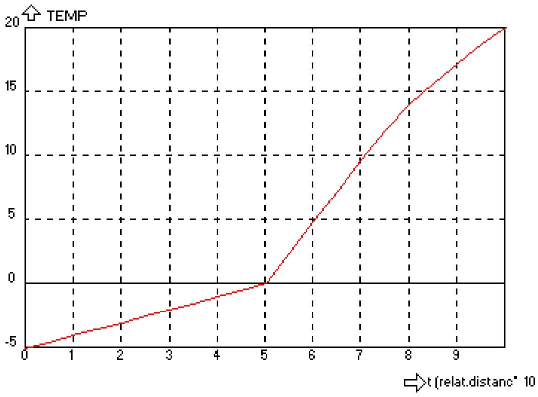
As an example we shall solve a simple problem of thawing rectangular box of ice, having initial temperature T_0=-5^{0}C, left side is hold at temperature -5^{0}C and right side at temperature 20^{0}C. Temperature of thawing is 0^{0}C, latent heat of fusion r is 340 kJ/kg, specific heat of ice c_{p1}=2 kJ/(kg.K), specific heat of water c_{p2}=4.2 kJ/(kg.K), thermal conductivity of ice is λ_1=2.2 W/(m.K) and thermal conductivity of water is λ=0.6 W/(m.K). Density of ice is 917 kg/m^3 and density of water 1000 kg/m^3. All these material parameters are defined as step-wise functions



```
C* TMELT.GEO melting a frozen box
ANALYS 2;
PT 1,.000E+00,.000E+00;
PT 2,.100E+01,.000E+00;
PT 3,.100E+01,.100E+01;
PT 4,.000E+00,.100E+01;
SF4PT 1,1,2,3,4;
EGROUP 1,PLANE2D,1,3,0,0,0;
RCONST 1,PLANE2D,1,0,0,0;
MPROP 1,PLANE2D,.600E+00,1,4200,2,1000,3,
.400E-01,0,.210E+12,0,.280E+00,0,.120E-04,
0,0,0,340000,0;
CURDEF 1,4,4,-
10,2.2/0.6,0,2.2/0.6,0.001,1,10,1;
CURDEF 2,4,4,-10,2/4.2,0,2/4.2,.100E-02,1,10,1;
CURDEF 3,4,4,-10,917/1000,0,917/1000,.100E-02,1,10,1;
MSF 1,20,5,1,1,3;
NFCR 4,TEMP,-1,-5,-5;
NFCR 2,TEMP,-1,20,20;
FUNDEF 4,-5;
INITIA TEMP,4;
OPTION 0,0,0,0,1,0,0,15,.1E-11,1,.1E+08,1,.1E+01,1,0,2,.1E-04,.1E-
04,0,0,0,0,0,0,0,
SOLVE 0,50,1000,0,0,0,3,0,0,0,0;
```

Resulting temperature contours up to the time 4.4 millions of seconds (it means approximately 51 days) for method 2 (Lemmon) are shown in the following figure



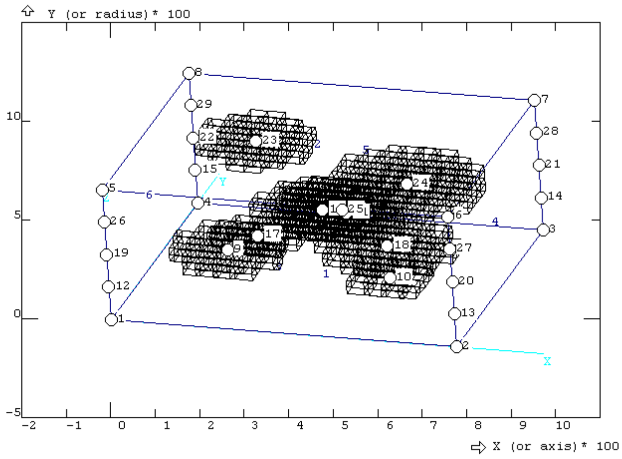


Temperature profile along the bottom of the block at the last time shows, that the temperature profile at the thawed part is still nonlinear (and it seems to be suspicious because at this time the temperature profile should be steady and λ is a constant?)

Hint: For plotting temperature profiles in a cross section you can use a curve passing through nodes and plot selected degree of freedom by command GCR. If you want to plot it along an arbitrary curve (not passing through nodes), you should perform BOX operation first and plot an arbitrary function (e.g. FUNDEF 1,DOF(TEMP,2,XX,YY,ZZ)) by command GFCR.

2.2.3. Ohmic heating 3D

The following example is a 3D model of temperature and electrical field in an electrically heated block of solid material with dimensions 8 x 8 cm (bottom) and height 10 cm. It is assumed that the block is filled by material having thermodynamic properties of water and that several rather big spherical particles with different material properties are located inside the block, see figure



The two opposite sides of block (left and right) represent cooled electrodes maintained at a constant temperature 10^0 C. The remaining sides are electrically insulated, however heat transfer to the ambient is considered, assuming constant ambient temperature and heat transfer coefficient α=50 W/(m^{2}K).

Problem is described by partial differential equations for steady state balances of heat and conservation of electrical charge

$$0 = \frac{\partial}{\partial x}(\lambda\frac{\partial T}{\partial x}) + \frac{\partial}{\partial y}(\lambda\frac{\partial T}{\partial y}) + \frac{\partial}{\partial z}(\lambda\frac{\partial T}{\partial z}) + \kappa(T)[(\frac{\partial U}{\partial x})^2 + (\frac{\partial U}{\partial y})^2 + (\frac{\partial U}{\partial z})^2] \qquad (2.2.3\text{-}1)$$

$$0 = \frac{\partial}{\partial x}(\kappa\frac{\partial U}{\partial x}) + \frac{\partial}{\partial y}(\kappa\frac{\partial U}{\partial y}) + \frac{\partial}{\partial z}(\kappa\frac{\partial U}{\partial z}), \qquad (2.2.3\text{-}2)$$

where λ and κ are thermal and electrical conductivities, T is temperature and U electric potential. While temperature as well as electrical potential are fixed at electrodes (x=0 and x=0.08 m), zero normal derivative of potential U and Robin's boundary conditions for temperature

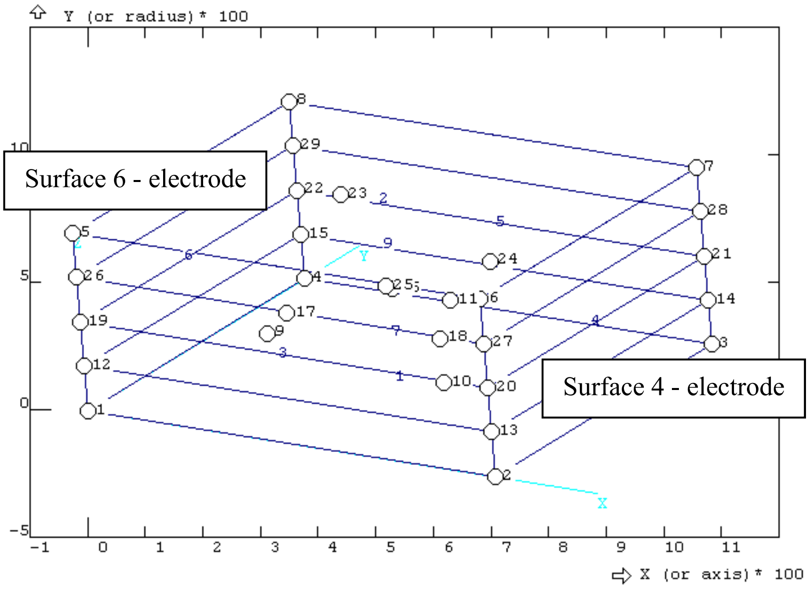
$$\lambda\frac{\partial T}{\partial n} = -\alpha(T - T_e) \qquad (2.2.3\text{-}3)$$

are to be applied at lateral faces of box.

Geometrical model is represented in this case by a hexahedron with 8 vertices which can be created directly from vertices (command VL8PT) or, as will be done in this case, by extrusion of a surface in a certain direction (command SFEXTR). To define elements with different properties of particles, several auxiliary points PT, defining centres of spheres (of particles) are located inside the box. These points and auxiliary surfaces (number 7,8,9) will be located in planes z=0.25, 0.5 and 0.75 (the surfaces have nothing to do with computation, they will be used for graphic purposes only).

```
PT 1,.000E+00,.000E+00;
PT 2,.800E-01,.000E+00;
PT 3,.800E-01,.800E-01;
PT 4,.000E+00,.800E-01;
SF4PT 1,1,2,3,4;
SFEXTR 1,1,0,0,.1;
```

```
ZDEF .025;
PT 9,.200E-01,.300E-01;
PT 10,.600E-01,.200E-01;
PT 11,.400E-01,.600E-01;
PT 12,.000E+00,.000E+00;
PT 13,.800E-01,.000E+00;
PT 14,.800E-01,.800E-01;
PT 15,.000E+00,.800E-01;
SF4PT 7,12,13,14,15;
ZDEF .05;
PT 16,.400E-01,.400E-01;
PT 17,.300E-01,.200E-01;
PT 18,.600E-01,.200E-01;
PT 19,.000E+00,.000E+00;
PT 20,.800E-01,.000E+00;
PT 21,.800E-01,.800E-01;
PT 22,.000E+00,.800E-01;
SF4PT 8,19,20,21,22;
ZDEF 0.075;
PT 23,.200E-01,.600E-01;
PT 24,.600E-01,.400E-01;
PT 25,.500E-01,.200E-01;
PT 26,.000E+00,.000E+00;
PT 27,.800E-01,.000E+00;
PT 28,.800E-01,.800E-01;
PT 29,.000E+00,.800E-01;
SF4PT 9,26,27,28,29;
```

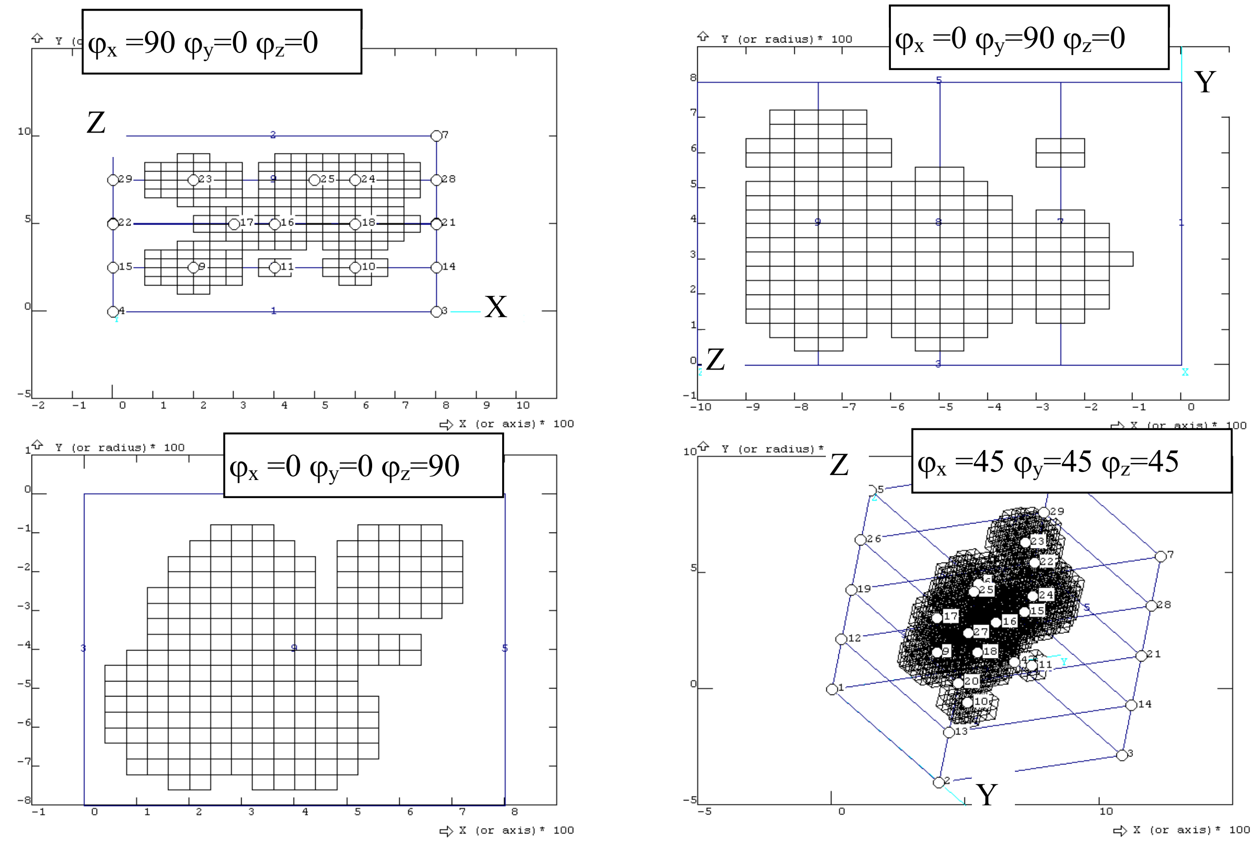


Having defined geometry, a mesh of 20 x 20 x 20 elements SOLID with 8 nodes can be created using MVL command. All the generated elements will have the attribute EGROUP=RCONST=MPROP=1 by default, however the index of material parameters should have been changed to MPROP=2 for elements having gravity centre inside the spheres with centres PT=9,10,… (MPROP=1 describes properties of liquid and MPROP=2 properties of spherical particles inside the box). This change can be accomplished by repeating use the command ERMOD (Egroup,Rconst,Mprop mODification).

```
MVL 1,20,20,20,1,1,1,8;
EGROUP 1,SOLID,0,2,0,1,0;
MPROP 1,SOLID,.600E+00,0,4200,0,998,0,.400E-01,0,.100E-02,0,.500E-03,0,.100E-
08,0,0,0,0,0;
MPROP 2,SOLID,1.,0,2500,0,1500,0,0.15,0,1,0,0,0,0,0,0,0,0,0;
ERMOD 0,9,.013,1,1,2;
ERMOD 0,10,.009,1,1,2;
ERMOD 0,11,.006,1,1,2;
ERMOD 0,16,.015,1,1,2;
ERMOD 0,17,.009,1,1,2;
ERMOD 0,18,.015,1,1,2;
ERMOD 0,23,.013,1,1,2;
ERMOD 0,24,.017,1,1,2;
ERMOD 0,25,.015,1,1,2;
```

EGROUP specifies type of elements (SOLID), steady state solution (0) and number of Gauss points (2 x 2 x 2). Note: If only one Gauss point of integration is used, results exhibit slight oscillations.
The command MPROP 1,… defines „liquid“ while the command MPROP 2,… spherical „particles“. Each particle is described by the command ERMOD Sphere(0)/Cube(1),center,radius,EGROUP,RCONST,MPROP.

3D graphic allows to rotate coordinate axis (x,y,z) by specified angles using the command VIEW 0, φ_x, φ_y, φ_z . It is also possible to select axonometric projection by specifying position of observer's eye using VIEW 1, x_{eye}, y_{eye}, z_{eye}. The command EPLOT for element plotting can be specified in such a way, that only the elements having index MPROP=2, e.g. elements inside „particles", will be plotted:



Boundary conditions are to be applied on the surface of cube. This is especially easy for electric potential, where strong boundary conditions are specified only on surfaces 4 and 6 (electrodes), while natural boundary conditions hold on the remaining sides (insulation). Boundary conditions of the third kind are applied on sides 1,2,3,5 for heat transfer, by setting status of nodes to 21. Then the specified nodal parameters are interpreted as the heat transfer coefficient α. Ambient temperature is parameter of RCONST and can be set by assignment TE(*i*)=value, where *i* is index of the RCONST group.

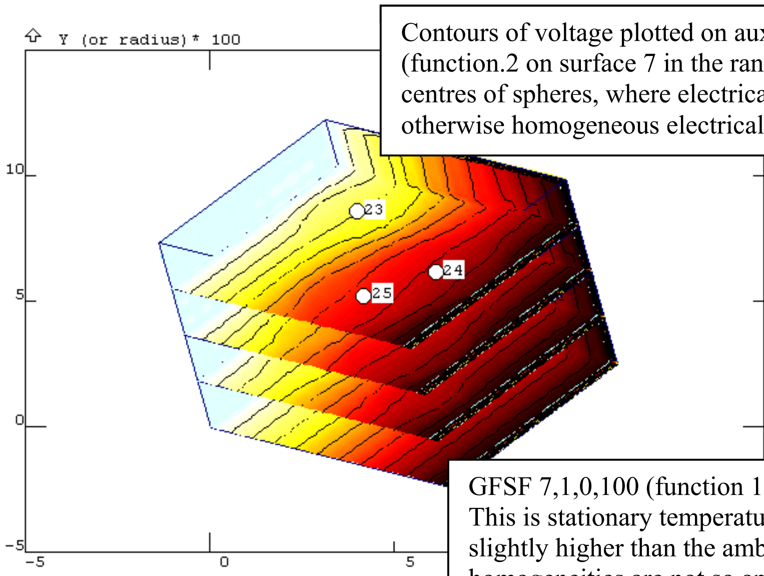
```
TE(1)=100
NFSF 4,VOLT,-1,0,0,0,0;
NFSF 6,VOLT,-1,100,100,100,100;
NFSF 6,TEMP,-1,10,10,10,10;
NFSF 4,TEMP,-1,10,10,10,10;
NFSF 1,TEMP,21,50,50,50,50;
NFSF 2,TEMP,21,50,50,50,50;
NFSF 3,TEMP,21,50,50,50,50;
NFSF 5,TEMP,21,50,50,50,50;
```

Calculation of internal heat source can be set either by command OPTION or by using system variable OHMI (volumetric heating by electric current). Because steady state and constant properties are considered, only one time step and one iteration is sufficient:

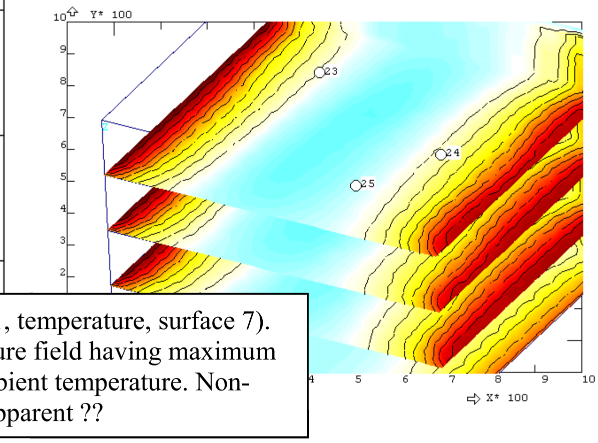
```
OHMI=1
SOLVE 100,1,100,0,1,0,1,0,0;
```

Command GFSF will be used for drawing contours of an interactively defined function on a selected surface. Because we need to plot nodal parameters TEMP and VOLT, we shall define functions number 1 and 2 as standard interpolating function DOF, calculating values of any nodal parameter in an arbitrary point *x,y,z*. This function DOF requires an auxiliary file, assigning finite elements to grid points, voxels, of a fine and uniform Cartesian grid (this file helps in fast search for element enclosing specified point *x,y,z*). This file is created by the command BOX.

```
BOX .5;
FUNDEF 1,DOF(TEMP,2,XX,YY,ZZ);
FUNDEF 2,DOF(VOLT,2,XX,YY,ZZ);
```



Contours of voltage plotted on auxiliary surfaces 7,8,9 by command GFSF 7,2,0,100 (function.2 on surface 7 in the range of voltage 0 - 100V. Points PT 23,24,25 are centres of spheres, where electrical conductivity is about 4 x higher and disturbs otherwise homogeneous electrical field.



GFSF 7,1,0,100 (function 1, temperature, surface 7). This is stationary temperature field having maximum slightly higher than the ambient temperature. Non-homogeneities are not so apparent ??

2.2.4. Concentration of tracer in a pipe (stimulus response experiment)

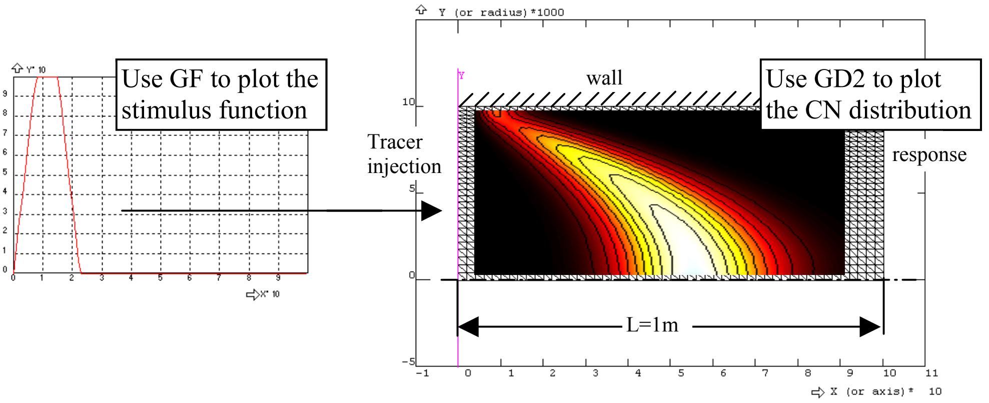
In the same way as a temperature field, a concentration field $c_N(t,x,y)$ can be calculated, because transport equations for temperature and concentration are in principle the same.

$$\frac{\partial c_N}{\partial t}+u\frac{\partial c_N}{\partial x}+v\frac{\partial c_N}{\partial r}=\frac{1}{r}\frac{\partial}{\partial r}(rD\frac{\partial c_N}{\partial r})+\frac{\partial}{\partial x}(D\frac{\partial c_N}{\partial x}) \qquad (2.2.4\text{-}1)$$

If velocities *u,v* are known, it will be possible to predict responses to a tracer injection by solving transient concentration field $c_N(t,x,r)$. In the following example we shall consider seemingly trivial case: fully developed laminar flow in a straight pipe with velocity profile

$$u(r)=2\bar{u}(1-(\frac{2r}{d})^2) \qquad (2.2.4\text{-}2)$$

where *r* is radial coordinate, *d* is diameter of tube, and $\bar{u}$ is mean velocity. If a tracer will be injected at inlet uniformly, it's concentration will be a function of only two spatial coordinates *x,r* and the problem can be formulated as a 2D case, using the same elements PLANE2D as previously.



2.2.4.1. Two dimensional solution of tracer distribution by Galerkin method

The problem can be described and solved by the following program

```
INTEGER NX,NY
H=1
R=0.01
U=0.1
W=3*H/U
NX=50
NY=30
L=400
```

User variables are used for parametric definition of geometry, mesh and boundary conditions: H-length of pipe, R-radius, U-mean velocity, W-maximum time, L-number of time steps, NX x NY mesh.

```
PT 1,0,0;
PT 2,H,0;
PT 3,H,R;
PT 4,0,R;
SF4PT 1,1,2,3,4;
MSF 1,NX,NY,1,.5,3;
BOX 0.3;
```

Operation BOX is something new. This operation is necessary if we want interpolate calculated nodal values (concentrations CN) to any point *x,y,z*. This operation creates direct access file assigning corresponding element to any grid point of a rectangular box enclosing a geometric model.

```
FUNDEF 1,2*U*(1-(YY/R)**2);
FUNDEF 2,YY*U*(1-(YY/R)**2)*DOF(CN,3,H,YY,0)
FUNDEF 3,YY*DOF(CN,3,H,YY,0)
```

Three functions are defined: function no.1 is radial velocity profile, function no.2 is used for calculation of mean mass concentration and function 3 will be used for calculations of area mean concentration. Because the command BOX has been performed, it is possible to use the system function DOF(*param,zone,x,y,z*) interpolating nodal values of *param* to a point *x,y,z*. Zone=3 indicates the zone of initial conditions.

```
INITIA VX,1;
INITIA VY,0;
AXIS(1)=1
UPW=1
EGROUP 1,PLANE2D,1,3,1,0,0;
```

EGROUP sets transient solution, 3 Gauss points, and cylindrical coordinate system for PLANE2D.

```
CURDEF -1,0,4,0,0,W/L,1,2*W/L,1,3*W/L,0,W,0;
NFCR 4,CN,-1,1,1;
INITIA CN,0;
SOLVE 0,L,W/L,0,0,0,0,1,0;
```

Time course of tracer concentration at inlet is defined in form of a table as the function -1. NFCR uses this function at inlet (inlet is the curve no 4).

2.2.4.2. How to evaluate response (mean concentration of tracer at outlet)

The calculated time history of concentration is saved in the file *problem*.OUT. Responses should have been in fact integrals of concentrations across the cross-section of pipe and these integrals can be calculated using the command IC (integral on CR) repeatedly at selected time steps. These time steps need not be necessary the same as those used in solution of transient problem, because the operation LTN *time* interpolates reloaded values from the file OUT to an arbitrary specified time. Result of integration must be transferred to the so called matrix of observation points (MOP), so that it would be possible to normalise and plot these time responses, to calculate moments and so on. Ten columns of MOP are accessible using system vectors C1(i),C2(i),…,C10(i) and we shall use the first column C1(i) for time, the second and the third columns for responses c_2 and c_3

$$\bar{c}_2(t) = \int_0^{d/2} ru(r)c(L,r,t)dr \qquad \bar{c}_3(t) = \int_0^{d/2} rc(L,r,t)dr\,. \qquad (2.2.4\text{-}3,4)$$

You can see, that the integrands of c_2 and c_3 have been yet defined as functions 2 and 3, respectively. Because the integration must be carried out at several tenths of time step, it would be very inefficient to repeat the operation manually. This is just the case, when it is better to use a batch file with #LOOP command:

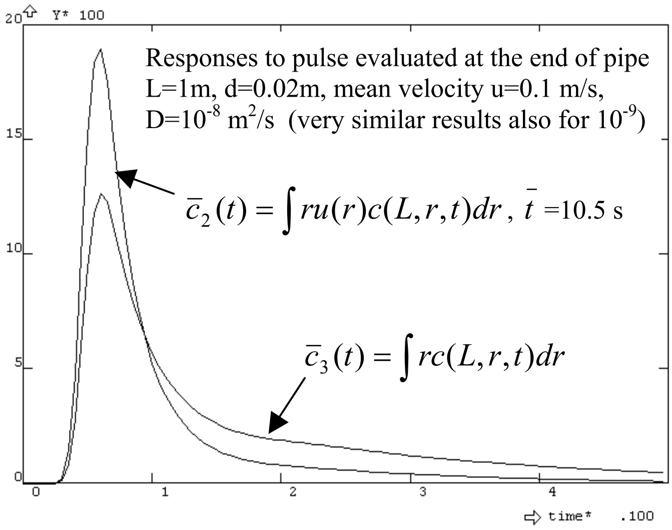
```
RECORD GTIME
       I=0
       T=0
       NT=50
       #LOOP LAB,NT;
       I=I+1
       LTN T;
       C1(I)=T
       IC 2,2
       C2(I)=INTEGRAL
       IC 2,3
       C3(I)=INTEGRAL
       T=T+0.5
       #LABEL LAB;
ENDREC
```

This batch calculates always 50 points at times 0,0.5,1,1.5,…,24.5 no matter which time step has been used in FEM solution.

I-row of the matrix of observation points, T-temporary time, NT-number of evaluated time steps (number of loops). For example command IC 2,3 calculates integral on curve 2 of function number 3. Result of operation IC is always in the system variable INTEGRAL.

This is where the loop begins and ends

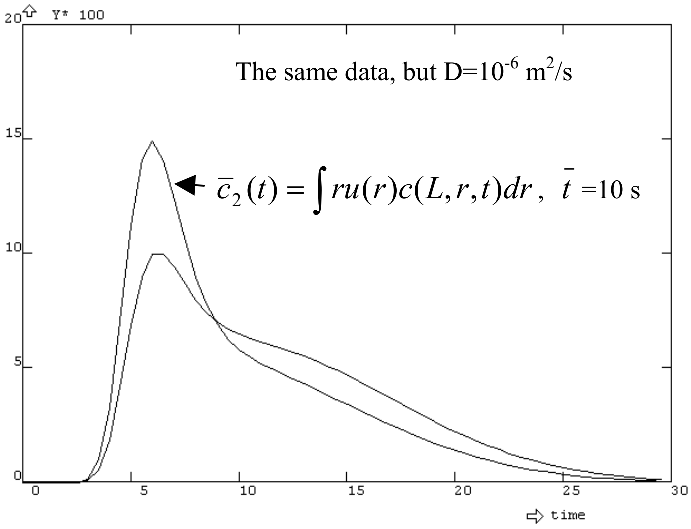
After the batch has been processed (F GTIME) it is possible to normalise responses (NORM), plot them (GTC), and calculate moments (MOM):



Responses to pulse evaluated at the end of pipe L=1m, d=0.02m, mean velocity u=0.1 m/s, D=10^{-8} m^2/s (very similar results also for 10^{-9})

$\bar{c}_2(t) = \int ru(r)c(L,r,t)dr$, $\bar{t}$ =10.5 s

$\bar{c}_3(t) = \int rc(L,r,t)dr$



The same data, but D=10^{-6} m^2/s

$\bar{c}_2(t) = \int ru(r)c(L,r,t)dr$, $\bar{t}$ =10 s

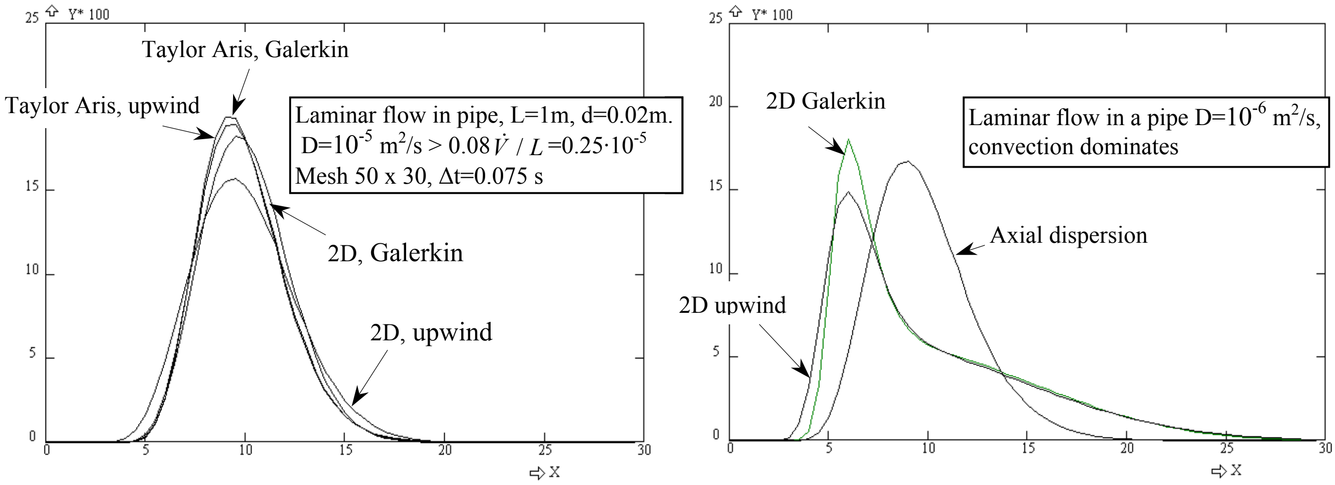
Results show, that the responses depend strongly upon the diffusion coefficient. If it is small (of the order 10^{-8} or less, typical values for liquids), problem with numerical dispersion appears, because Peclet number of element is extremely high (of the order 10^5 !) and solution exhibits slight deficiencies at axis, no matter that the full upwind has been applied. Even at $D=10^{-6}$ the Peclet number is still rather high, however the solution is smoother and quite different (radial diffusion starts to be significant), which is apparent from the quite different time responses (it is remarkable that the evaluated mean residence times more or less agree with the theoretical value 10.1 s).

2.2.4.3. Comparison with 1D solution – axial dispersion model

In the following chapters we shall demonstrate applications of simple 1D elements PIPE2D based upon the model of axial dispersion, with the dispersion coefficient D_e calculated according to the Taylor (1953), Aris (1958) theory

$$\frac{\partial \bar{c}_N}{\partial t} + \bar{u}\frac{\partial \bar{c}_N}{\partial x} = \frac{\partial}{\partial x}(D_e \frac{\partial \bar{c}_N}{\partial x}), \qquad D_e = \frac{R^2\bar{u}^2}{48D} \quad (R\text{-radius}) \qquad (2.2.4\text{-}5,6)$$

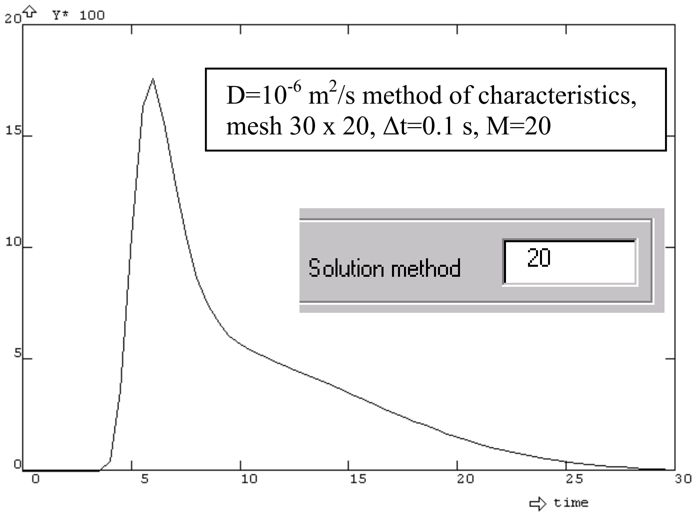
derived for laminar and Newtonian flow (it means for the same parabolic velocity profile as in the 2D case). This model is quite good if the diffusion coefficient D is at least ten times larger than the ratio of volumetric flow-rate and the length of pipe. This restriction holds e.g. for D=10^{-5} m^2/s and corresponding results are shown in the figure left together with responses calculated from 2D models. It seems that in this case the simple 1D model of axial dispersion using only 30 elements is more accurate and insensitive to numerical dispersion than the 2D model with 3000 elements. The figure in right looks different: Diffusion coefficient is ten times lower (but still about 3 orders greater than a typical value for liquids) and the model of axial dispersion is not able to describe the impulse response correctly, because convection dominates in the tracer transport.

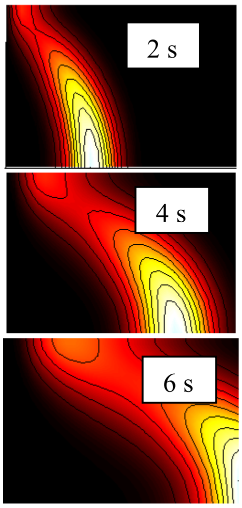


What is apparent: Responses calculated from the 2D model are distorted by numerical diffusion, because results obtained using the same mesh and the same time step, but once with and once without upwind, differ considerably and it is obvious that the upwind increases the response variances .

2.2.4.4. Two dimensional solution using method of characteristics

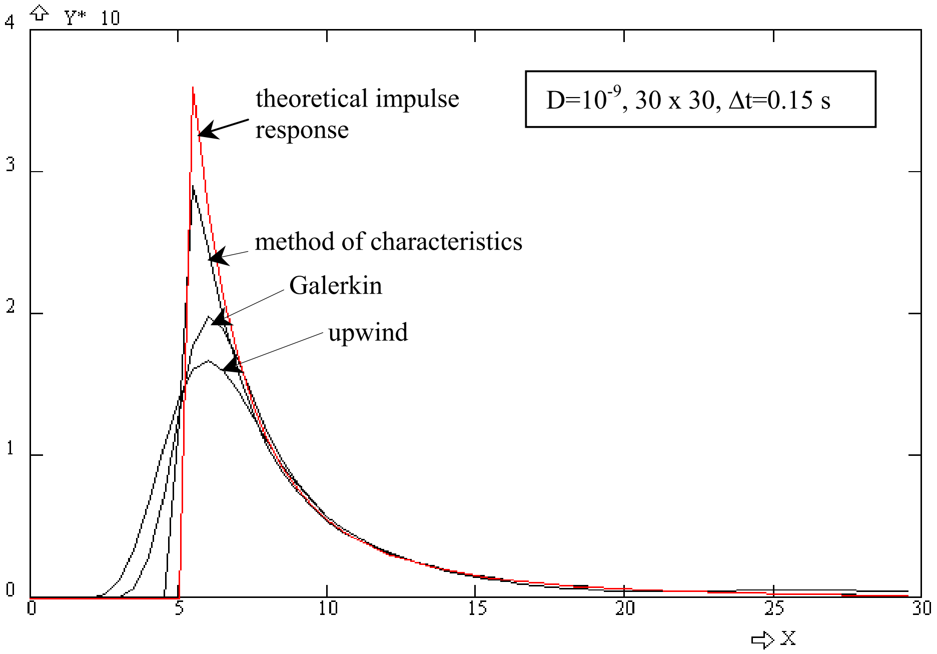
An attempt how to decrease the numerical dispersion is the method of characteristics. This method decomposes the time step Δt into two phases. The first purely convective phase calculates trajectories starting from nodes at time t+ Δt and ending at time t somewhere (these trajectories are characteristics of transport equation, along which the concentration remains constant). The trajectories are calculated from velocity field by numerical integration with a finer time step Δt/M. If the trajectory ends inside an element, the concentration at time *t* is obtained by interpolating nodal values, otherwise it had to cross the boundary somewhere at inlet. Then the time of crossing is estimated and initial concentration is calculated from specified boundary condition at the nearest boundary node. The second phase of time step is pure diffusion, solved by a classical Galerkin method with zero velocities.





This result obtained by the method of characteristics seems promising, because responses calculated with a coarser mesh and a higher time step compare well with the previous result.

Comparison between method of characteristics, Galerkin and Galerkin Petrov methods is shown in the following figure, calculated with the same mesh (30 x 30) and the same time step. Diffusion coefficient corresponds to liquids, and therefore is much lower, D=10^{-9} m^2/s.



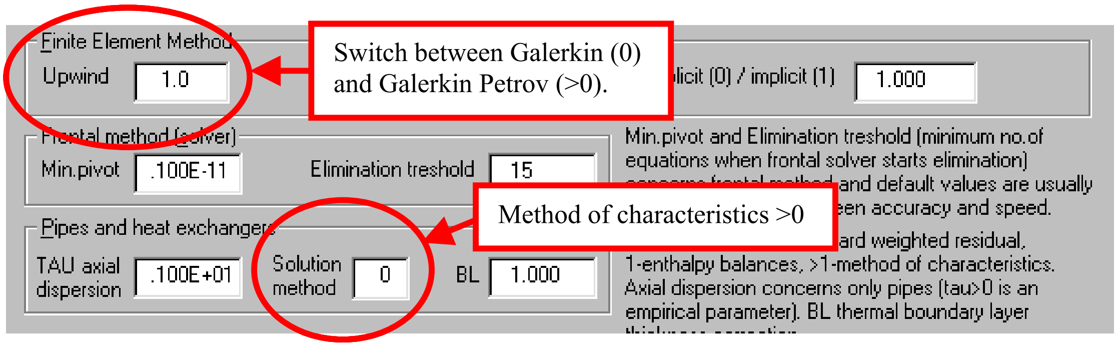
The theoretical impulse response, given by the expression, see Thýn (2000),

$$E(t) = \frac{1}{2(t/\bar{t})^3} \qquad \text{for} \qquad t > \bar{t}/2 \qquad (2.2.4\text{-}7)$$

describes pure convection, and can be used as a reference response. It is obvious, that in this case the method of characteristics is superior, however it is not fool proof, sometimes fails and therefore the method is still under development.

Note: If you want to plot several responses in one graph (GTC), use the command TCF for copying columns in matrix of observation points, and for creation responses defined by an analytical formula, e.g. the command TCFR 10,60,c3=1/c1**3 defines values in the third column and in the rows 10,11,…,60 as a function 1/t^3.

Note: You can use system variables for selection of solution methods, e.g. HEPI=3 selects method of characteristics (solution method=3), RUPW=0.5 sets Galerkin Petrov method with optimum upwind reduced by coefficient 0.5. The same effect can be accomplished by using the OPTION command or its dialog equivalent selected from the toolbar *Solution* and item *Option* :



Finite Element Method

Upwind: 1.0

licit (0) / implicit (1): 1.000

Frontal method (solver)

Min.pivot: .100E-11

Elimination treshold: 15

Pipes and heat exchangers

TAU axial dispersion: .100E+01

Solution method: 0

BL: 1.000

Min.pivot and Elimination treshold (minimum no.of equations when frontal solver starts elimination) concerns frontal method and default values are usually …een accuracy and speed. …ard weighted residual, 1-enthalpy balances, >1-method of characteristics. Axial dispersion concerns only pipes (tau>0 is an empirical parameter). BL thermal boundary layer thickness correction

Switch between Galerkin (0) and Galerkin Petrov (>0).

Method of characteristics >0

2.2.5. Laminar flow (stream function)

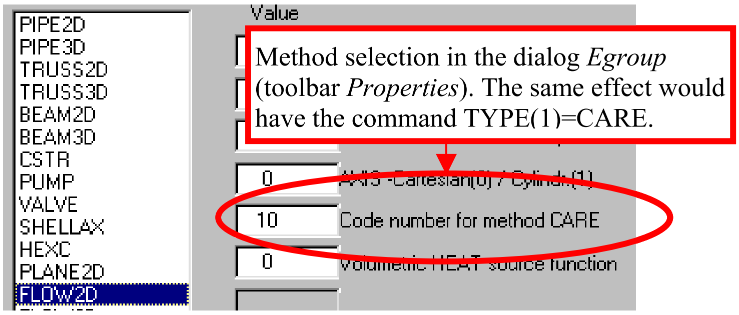
FEMINA offers several methods for solution of two-dimensional flow field, i.e. for finding velocities and pressures of an incompressible liquid in laminar flow regime. One group of the methods (CREE, PSIN, CARE) is based upon the flow description by the stream function ψ and this approach has the advantage that the continuity equation is automatically satisfied. This is a direct consequence of the stream function definition which is tied together with velocity components by the following relations:

Plane flow

$$u_x = \frac{\partial \psi}{\partial y} \qquad u_y = -\frac{\partial \psi}{\partial x} \qquad (2.2.5\text{-}1,2)$$

Rotationally-symmetrical flow (axis z)

$$u_r = \frac{1}{r}\frac{\partial \psi}{\partial z} \qquad u_z = -\frac{1}{r}\frac{\partial \psi}{\partial r} \qquad (2.2.5\text{-}3,4)$$



0 AXIS -Cartesian(0) / Cylindr.(1)

10 Code number for method CARE

0 Volumetric HEAT source function

Method selection in the dialog *Egroup* (toolbar *Properties*). The same effect would have the command TYPE(1)=CARE.

CREE method calculates steady-state creeping flow of Newtonian or even non-Newtonian liquids by minimizing dissipation of energy *F*, expressed in terms of stream function ψ, (δ*F*(ψ)=0),

$$\text{CREE} \qquad F(\psi) = \iint_{x,y} 2\mu(\vec{\vec{\Delta}} : \vec{\vec{\Delta}})dxdy = \iint_{x,y} \mu[4(\frac{\partial^2 \psi}{\partial x \partial y})^2 + (\frac{\partial^2 \psi}{\partial x^2} - \frac{\partial^2 \psi}{\partial y^2})]dxdy\,. \qquad (2.2.5\text{-}5)$$

Only the degrees of freedom PS, PSX, PSY - values of the stream function and its first derivatives, are used in the solution, while velocities VX, VY are derived from ψ afterwards. If viscosity μ is constant no iteration is necessary because the creeping flow is described by *linear* Stokes equation and therefore the CREE method is suitable as an initial approximation for higher Reynolds number flows solved by the method PSIN. The method PSIN is derived from Navier Stokes equations with eliminated pressure terms (shortly: momentum equation in the direction *x* is differentiated with respect *y* and subtracted from the *y*-equation differentiated with respect *x*; thus the pressure are cancelled). Remaining velocity terms in resulting equation are expressed in terms of stream function ψ:

$$\text{PSIN} \qquad \rho \frac{D}{Dt}\nabla^2 \psi = \mu \nabla^4 \psi\,. \qquad (2.2.5\text{-}6)$$

The method PSIN makes use the same degrees of freedom and the same boundary conditions as CREE, but calculates unsteady solution and the influence of inertial terms is considered (it is suitable for Re only slightly greater than 0, because "upwind" is not implemented, see later). CREE and PSIN methods are implemented for triangular elements with three nodes only[11] – this restriction does not apply to the CARE method which solves the stream function ψ and the vorticity ω simultaneously, and makes use of simpler base functions (therefore quadrilateral elements can be also used). The CARE method follows from the formulation (2.2.5-6) by introducing vorticity ω

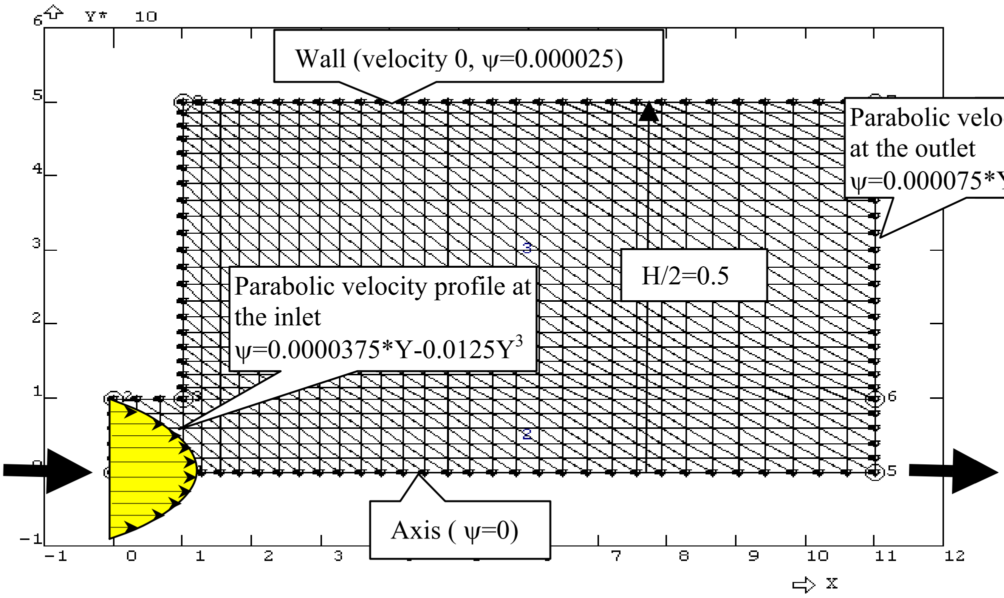
$$\text{CARE} \qquad \omega = \frac{\partial u_x}{\partial y} - \frac{\partial u_y}{\partial x} = \nabla^2 \psi \qquad \rho \frac{D\omega}{Dt} = \mu \nabla^2 \omega \qquad . \qquad (2.2.5\text{-}7)$$

The ψ-ω method is suitable even for higher values of Reynolds numbers due to implementation of Galerkin-Petrov upwind method.

[11] At least cubic polynomials having continuous first derivatives at nodes must be used for approximation of ψ(x,y) because also second derivatives (and not only first derivatives) appear in integrands of both the CREE and PSIN method.

2.2.5.1. Steady laminar flow in a planar channel (CREE and PSIN)

As an example of the described methods the analysis of laminar flow in a planar channel with geometry shown in the following figure will be presented (Cartesian coordinate system, the channel is symmetrical with respect to *x* – axis and dimensions are designed for Reynolds numbers of around 100)


Y* 10
Wall (velocity 0, ψ=0.000025)
Parabolic velocity profile at the outlet
ψ=0.000075*Y-0.0001Y^3
H/2=0.5
Parabolic velocity profile at the inlet
ψ=0.0000375*Y-0.0125Y^3
Axis (ψ=0)
6
5
4
3
2
1
-1
-1
0
1
2
3
7
8
9
10
11
12
X

When using the stream function, values of ψ and their first derivations at the inlet section must be specified. We will assume a fully developed parabolic velocity profile at the inlet

$$u_x(y)=\frac{3}{2}\bar{u}_x[1-(\frac{2y}{H})^2], \qquad \psi(y)=\frac{3}{2}\bar{u}[y-\frac{y^3}{3}(\frac{2}{H})^2]. \tag{2.2.5-8,9}$$

The mean velocity at the inlet section is taken as 0.00025 m/s, and the corresponding mean velocity in the expanded outlet section is 0.00005 m/s (H_{inlet}=0.2 m, H_{outlet}=1 m). Hydraulic diameter D_e measured at the outlet is double of the channel width, i.e. 2m. If water is used as a model liquid (viscosity 1 mPa.s), the Reynolds number at the outlet section will be Re_{outlet}=100 which is the same as Re_{inlet}=100 at the inlet section (product of the mean velocity and diameter is constant in a planar channel). We estimate the channel length so that the model region would contain the velocity profile evolution behind the expansion according to relation

$$\frac{L}{D_e}=0.05\,\mathrm{Re}. \tag{2.2.5-10}$$

NEW! You can define inlet velocity profiles (parabolic, logarithmic, uniform) also using the command INLET

```
C* Opening test date:22.04.03 at 08:53
ANALYS 2;
SCALE -.500E-02,12,-.500E-02,.6;
PT 1,.000E+00,.000E+00;
PT 2,.000E+00,.100E+00;
PT 3,.100E+01,.100E+00;
PT 4,.100E+01,.000E+00;
PT 5,1.100E+01,.000E+00;
PT 6,1.100E+01,.100E+00;
PT 7,1.100E+01,.500E+00;
PT 8,.100E+01,.500E+00;
SF4PT 1,1,4,3,2;
SF4PT 2,4,5,6,3;
SF4PT 3,3,6,7,8;
EGROUP 1,FLOW2D,0,7,0,CREE,0;
MSF 1,3,5,1,1,3;
```

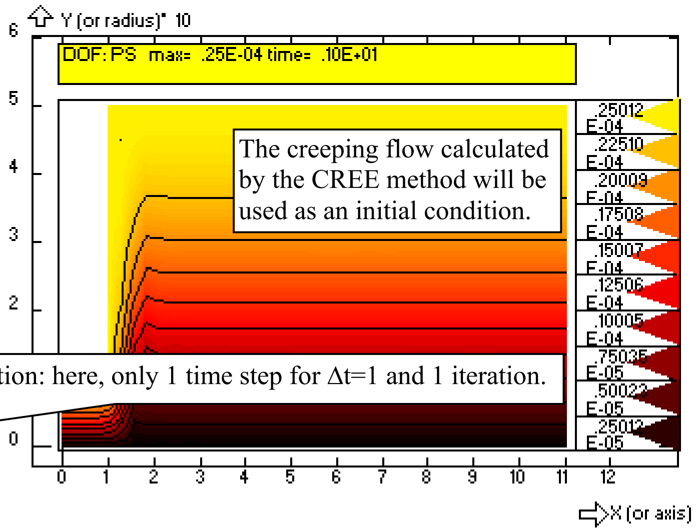
Elements FLOW2D. The CREE method is selected (creeping flow) and used for calculationof initial conditions: only one iteration is needed in this case (if the viscosity and density are constant).

```
MSF 2,30,5,1.5,1,3;
MSF 3,30,20,1.5,-1.5,3;
NMERGE ;
FUNDEF -2,0.000375*YY-.0125*YY**3;
FUNDEF -3,0.000075*YY-0.0001*YY**3;
FUNDEF -4,0.000375*(1-100*YY**2);
```

Functions –2, -3, -4 describe strong boundary conditions: stream function profiles at the inlet and outlet sections, and velocity profile at the inlet $(\psi_{,y})$.

Velocity profiles at the inlet and outlet of the channel. In fact, the profile of the ψ(y) function calculated from the velocity profiles is specified.

```
NFCR 4,PS,-2,1,1;
NFCR 4,PSX,-1,0,0;
NFCR 4,PSY,-4,1,1;
NFCR 1,PS,-1,0,0;
NFCR 1,PSX,-1,0,0;
NFCR 5,PSX,-1,0,0;
NFCR 5,PS,-1,0,0;
NFCR 3,PSX,-1,0,0;
NFCR 3,PSY,-1,0,0;
NFCR 10,PSY,-1,0,0;
NFCR 10,PSX,-1,0,0;
NFCR 9,PSX,-1,0,0;
NFCR 9,PSY,-1,0,0;
NFCR 3,PS,-1,0.000025,0.000025;
NFCR 10,PS,-1,0.000025,0.000025;
NFCR 9,PS,-1,0.000025,0.000025;
NFCR 6,PSX,-1,0,0;
NFCR 8,PSX,-1,0,0;
NFCR 6,PS,-3,1,1;
NFCR 8,PS,-3,1,1;
SOLVE 0,1,1,0,0,1,0,0,0;
```



The creeping flow calculated by the CREE method will be used as an initial condition.

Start the computation: here, only 1 time step for Δt=1 and 1 iteration.

```
EGROUP 1,FLOW2D,0,7,0,PSIN,0;
RELFAKT=.3
OPTION 0,0,0,1,0,15,.100E-11,.1,.100E+08,1,.100E-04,.100E-04,0,20,0,0,0;
```

The method PSIN solves the unsteady Navier-Stokes equation (the parameter *Steady/Transient* of the EGROUP command is ignored, the type of solution is always *Transient* if the method PSIN is selected). It is necessary to specify the time step and number of iterations even if the aim is a steady-state solution. A rough estimate about the time scale can be obtained from the ratio of the channel length (L=10 m) and the mean velocity (0.00005 m/s), i.e. 200000 s. The time step is chosen 20 times lower, i.e. 10000 s.

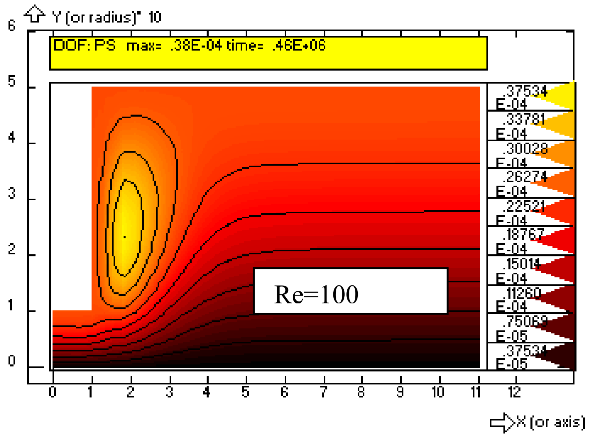
```
SOLVE 10,3,10000,0,0,20,0,0,0;
```

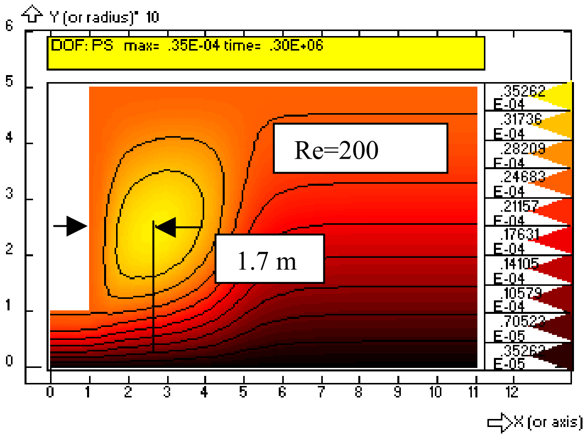
Note: The main problem in solution of the Navier-Stokes equation is a convergence which can be influenced primarily by the time step and eventually by the underrelaxation factor (see command RELFAKT=0.3). The final solution is usually obtained through a repeated sequence of computations with different time steps. The essential thing is that the procedure SOLVE updates initial conditions in every time step, so the repetition of the SOLVE command continues the results of the previous time step. If we like to repeat the whole computation from the beginning, we would have to overwrite the initial conditions explicitly (e.g. by the INI command or, in this case, by the solution of steady-state problem using the method CREE for creeping flow).

```
SOLVE .300E+05,3,.100E+05,2,0,20,0,0,0;
VISC(1)=VISC(1)/2
```

Method PSIN for Re = 100. Streamlines ψ>0.000025 mark off recirculation zones.

```
SOLVE .296E+06,1,3000,2,0,20,0,0,0;
```

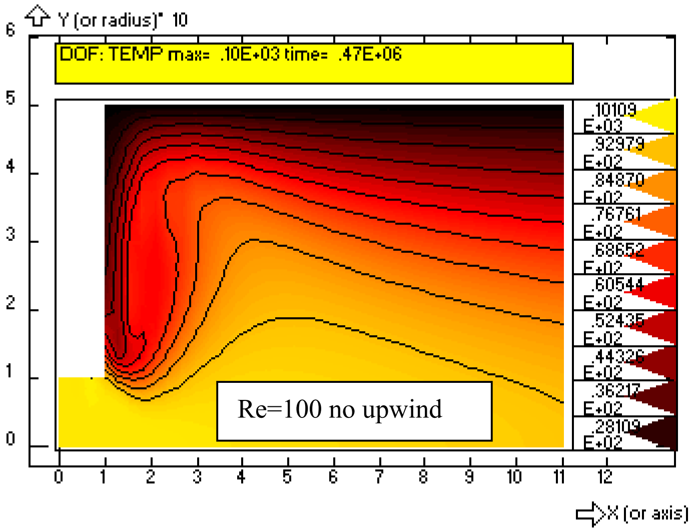




2.2.5.2. Heat and mass transfer in planar channel (upwind)

Because the velocity field is already known, it is possible to solve the temperature field in the same way as in the previous paragraph (i.e. arbitrary boundary conditions and time and temperature dependencies of parameters can be specified, etc.). We will consider a simple case when the temperature of the inlet liquid is 100 ^{0}C, and each wall except the upper extended part of the channel is thermally insulated (temperature of the upper wall is 20 ^{0}C). We change the parameter EGROUP to the steady-state solution (thus the steady-state solution of a temperature field can be obtained in only one "time" step) and we shall try to compare two variants of the solution, one without the upwind modification ("pure" Galerkin method) and the other with upwind (Galerkin-Petrov). The upwind method is to be used when solution stability problems arise at high values of Péclet numbers in the element, i.e. at high flow velocities and a coarse grid of finite elements. The disadvantage is of course a numerical diffusion (in our case it is rather a numerical thermal diffusivity) and consequently inaccurate results. This modification together with other operational parameters can be set by the OPTION command (it is also sufficient to write UPW=1, the variable UPW means "upwind"):

```
NFCR 4,TEMP,-1,100,100;
NFCR 9,TEMP,-1,20,20;
EGROUP 1,FLOW2D,0,7,0,6,0;
SOLVE .463E+06,1,2000,2,0,0,2,0,0;
OPTION 0,0,0,0,0,15,.100E-11,.100E+00,.100E+08,1,.100E-04,.100E-04,0,0,2,0,0;
SOLVE .465E+06,1,2000,2,0,0,2,0,0;
```



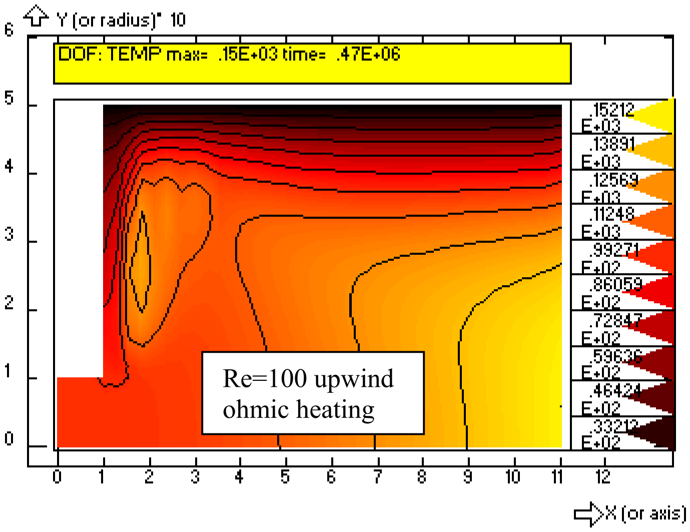
It is evident that the temperature profile is not fully developed and the effect of the recirculation zone is self-evident. The results also show that the upwind modification is not required at all. It is surprising because the Péclet number of the element is very high (more than 100 near the channel axis).

When the direct ohmic heating is to be solved, the electric potential field must be calculated first. We shall assume that electrodes are formed by the opposite walls of the channel and the potential difference is 220 V. The potential anti-symmetry will be used to specify boundary conditions: potential is set to zero in the centre (y=0), and 110 V at electrode (at the upper wall):

```
NFCR 9,VOLT,-1,110,110;
NFCR 1,VOLT,-1,0,0;
NFCR 5,VOLT,-1,0,0;
SOLVE .467E+06,1,2000,2,1,0,1,0,0;
```

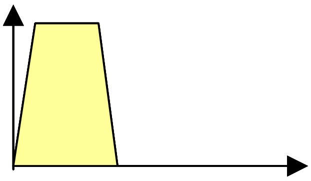
Using the calculated electric potential field, the temperature field corresponding to direct ohmic heating of transported liquid will be computed. First, a parameter which ensures calculation of volumetric heat sources must be set using the OPTION command (or by assignment OHMI=1)

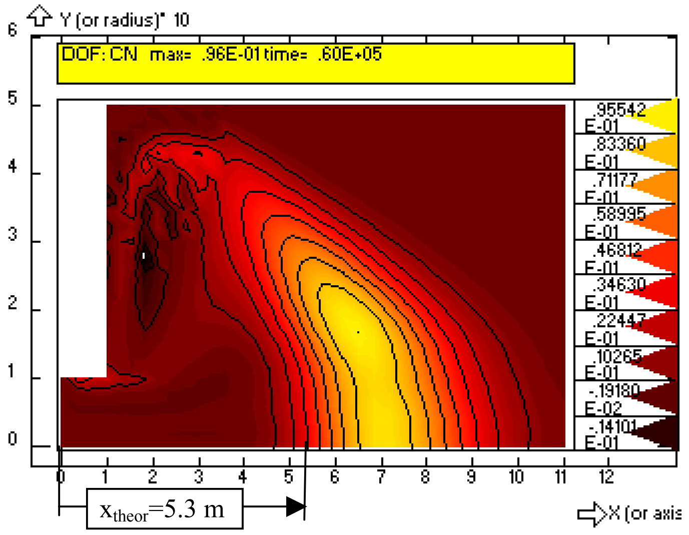
```
OPTION 0,0,1,1,0,15,.100E-11,.100E+00,.100E+08,1,.100E-04,.100E-04,1,0,1,0,0;
SOLVE .469E+06,1,2000,2,1,0,1,0,0;
```

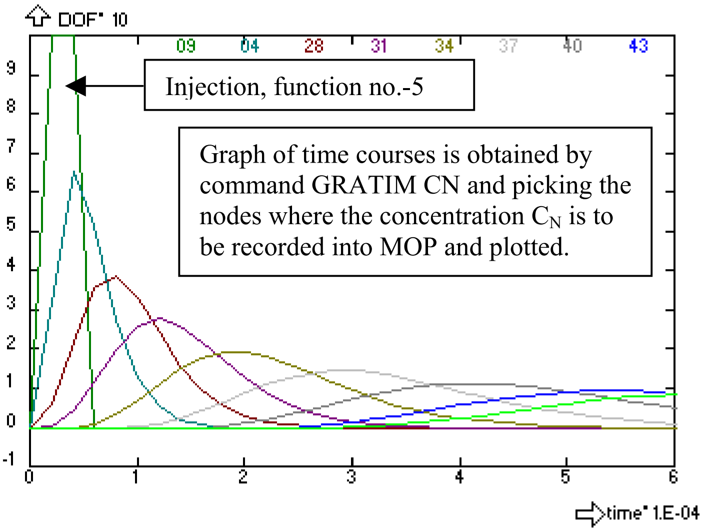


Knowing the flow field, mass transport problems can be solved, for example concentration responses to a tracer, injected instantaneously into the inlet section. The injection is modelled by prescribing a time dependency of the tracer concentration c_N at each node of the inlet section with the following table (this table stands for the function number –5 applied on the curve number 4 – it is just the inlet section)

```
CURDEF -5,0,5,0,0,1000,1,5000,1,6000,0,1000000,0;
NFCR 4,CN,-5,1,1;


EGROUP 1,FLOW2D,1,7,0,6,0;
SOLVE 0,30,2000,0,0,0,0,1,0;
```





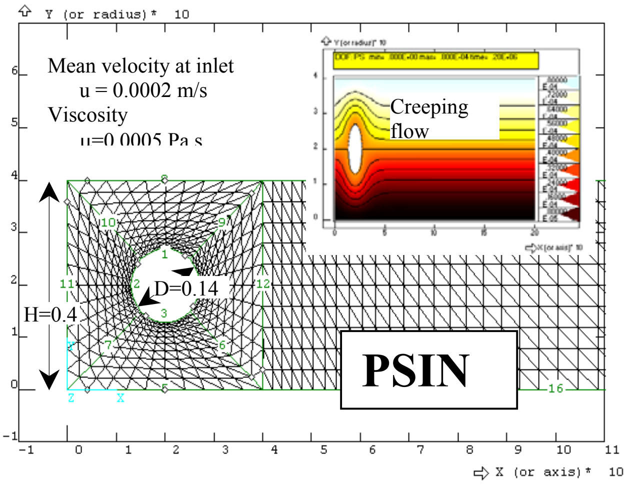


Graph of time courses is obtained by command GRATIM CN and picking the nodes where the concentration C_N is to be recorded into MOP and plotted.

The figure on the left shows the tracer concentration field at time 60000 s. If the liquid was moving at constant velocity along the whole channel axis (0.000075 m/s in the extended part[12]) the tracer should reach the distance 5.3 m. However it is clear that the mass centre of the tracer goes further – approximately 7 m. It can be caused (and with no doubt it is) by the cross-section contraction behind the step, but uncertainties are evoked by the concentration field which is evidently distorted by the numerical diffusion (diffusion coefficient D_N=10^{-9} m^2/s and the corresponding penetration depth for time 60000 s is approximately only 1.5 cm!).

2.2.5.3. Transient laminar flow over obstacle in a channel

Precceeding example has been aimed to the steady state flow analysis even if the solution was obtained by a transient method (the method PSIN can not solve the steady state directly). However, sometimes a steady state does not exist at all, for example in the so called Karman vortex street formed by periodically detached vortices behind an obstacle in flow. We shall try to use the method PSIN for evaluation of transversal flow around a cylinder in a planar channel as an example. We shall assume fully developed parabolic velocity profile at the channel inlet

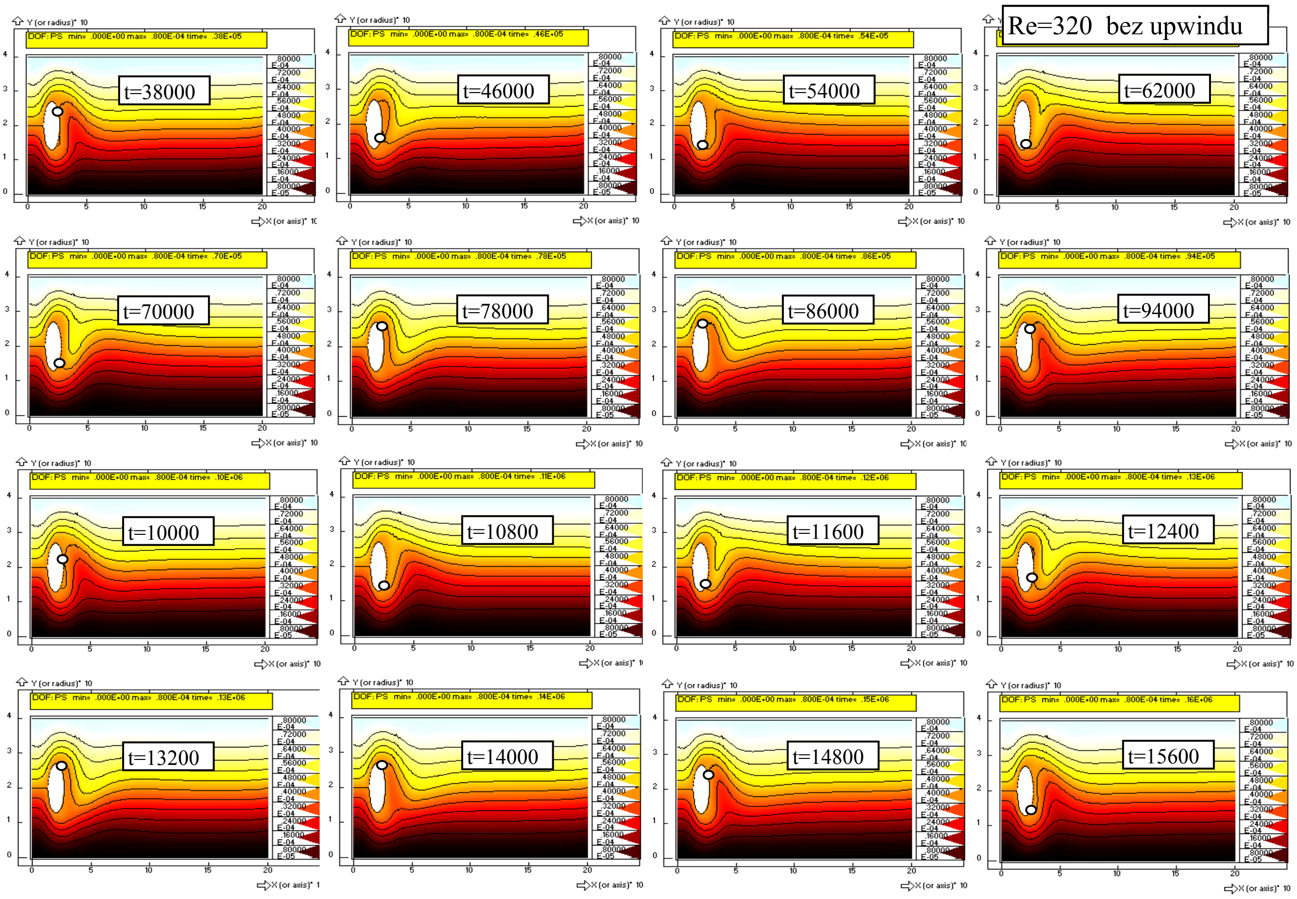


$$u(y) = \frac{6\bar{u}y}{H}(1-\frac{y}{H}) \qquad \psi(y) = \frac{\bar{u}y^2}{H}(3-2\frac{y}{H}). \qquad (2.2.5\text{-}11,12)$$

At the cylinder surface a constant value of stream function, corresponding to symmetric distribution of flow, is assumed. This assumption is not quite correct and generally speaking the methods, operating with stream functions, are not very suitable for modelling of flow around obstacles because values of stream function at the obstacle surface are not known in advance. The value ψ at the surface should not

[12] Results correspond to Re=100, mean velocity at inlet 0.00025 m/s and mean velocity in expanded part of channel 0.00005 m/s. Assuming fully developed parabolic velocity profile, the maximum velocity at axis is 1.5 times the mean velocity (it is a slit and not a pipe!), that is the stabilised velocity at axis equals 0.000075 m/s.

be fixed by a strong boundary condition and its constant value should follow from the condition that the both first derivatives of stream function are zero. However tests which have been performed by FEMINA shown that it does not work – maybe due to unsuitable formulation of problem (see theoretical part) and improper natural boundary conditions for stream function.



These results (see the session file *fkarman.geo*) correponds to Re=320 (related to the hydraulic diameter of channel) and do not describe generation and detachment of isolated vortices, only a periodical oscilation of flow behind cylinder at frequency around 50000 s. Corresponding value of Strouhal number is

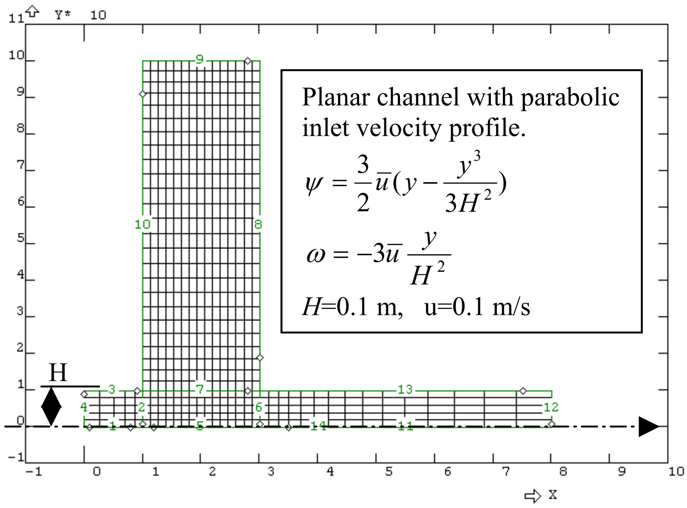
$$S = \frac{D}{\bar{u}T} = \frac{0.14}{0.0002 \cdot 50000} = 0.014 \quad . \qquad (2.2.5\text{-}13)$$

2.2.5.4. Campion Renson method (stream function and vorticity)

In this paragraph we shall illustrate the method, suggested by Campion-Renson and this is why it is called CARE in FEMINA. This method solves a pair of partial differential equations of the second order for stream function ψ, and vorticity ω, and these equations substitute the biharmonic equation of the fourth order for the stream function. Therefore much simpler base functions may be used in the CARE on contrary to the PSIN method, and any kind of elements, i.e. not only 3-nodal triangles, can be used. The only difference from point of view of user are boundary conditions, which have to be prescribed for stream function at inlet, axis and wall, while for vorticity only at inlet and at axis - not at the wall! This is because the boundary conditions for normal derivative of the stream function at wall are replaced by *natural* boundary conditions for vorticity. Boundary conditions at inlet and axis must be derived from the vorticity definition

$$\omega = \frac{\partial u_x}{\partial y} - \frac{\partial u_y}{\partial x} = \frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} \qquad (2.2.5\text{-}14)$$

```
ANALYS 2;
PT 1,.000E+00,.000E+00;
PT 2,.100E+01,.000E+00;
PT 3,.100E+01,.100E+00;
PT 4,.000E+00,.100E+00;
PT 5,.300E+01,.000E+00;
PT 6,.300E+01,.100E+00;
PT 7,.300E+01,.100E+01;
PT 8,.100E+01,.100E+01;
PT 9,.800E+01,.000E+00;
PT 10,.800E+01,.100E+00;
SF4PT 1,1,2,3,4;
SF4PT 2,2,5,6,3;
SF4PT 3,3,6,7,8;
SF4PT 4,5,9,10,6;
EGROUP 1,FLOW2D,1,3,0,CARE,0;
MSF 1,5,5,.5,1,4;
MSF 2,15,5,1,1,4;
MSF 3,15,25,1,-.150E+01,4;
MSF 4,15,5,4,1,4;
NMERGE ;
CR2PT 14,1,9;
NFCR 14,PS,-1,0,0;
NFCR 14,OMG,-1,0,0;
FUNDEF -2,0.15*(YY-100*YY**3/3);
FUNDEF -3,-30*yy;
NFCR 4,PS,-2,1,1;
NFCR 4,OMG,-3,1,1;
NFCR 12,OMG,-3,1,1;
NFCR 12,PS,-2,1,1;
NFCR 3,PS,-1,0.01,0.01;
NFCR 10,PS,-1,0.01,0.01;
NFCR 9,PS,-1,0.01,0.01;
NFCR 8,PS,-1,0.01,0.01;
NFCR 13,PS,-1,0.01,0.01;
SOLVE 0,10,100,0,0,10,0,0,0,0,0;
VISC(1)=10
SOLVE 1000,10,100,0,0,10,0,0,0,0,0;
```

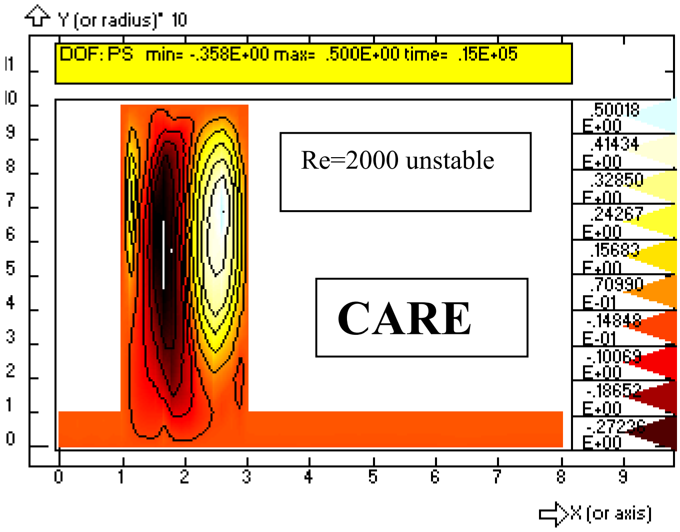


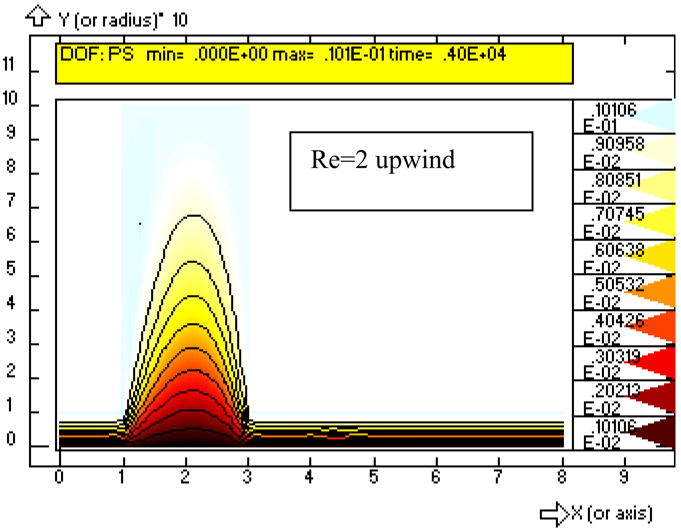
Planar channel with parabolic inlet velocity profile.

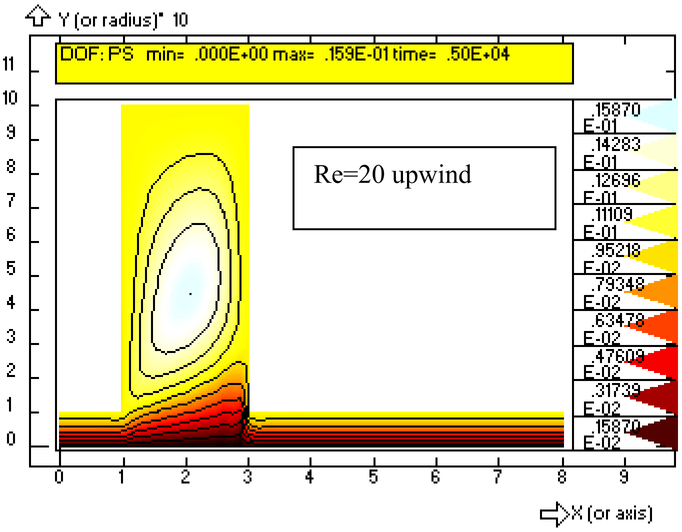
$$\psi = \frac{3}{2}\bar{u}\left(y - \frac{y^3}{3H^2}\right)$$

$$\omega = -3\bar{u}\frac{y}{H^2}$$

H=0.1 m, u=0.1 m/s







2.2.6. Navier Stokes equations in primitive variables, velocities and pressure

The methods UVP,UVPP,PENS calculating velocities and pressure (so called primitive variables) directly, are more frequently used than the stream function methods, because they can be easily extended to 3D. FEMINA implements the methods in Cartesian or cylindrical 2D coordinate system.

UVP: $$\rho\frac{D\vec{u}}{Dt} = -\nabla p + \nabla\cdot\mu\nabla\vec{u} + \rho\vec{g}(1-\beta T), \qquad \nabla\cdot\vec{u} = 0 \qquad (2.2.6\text{-}1)$$

UVPP: $$\rho\frac{D\vec{u}}{Dt} = -\nabla p + \nabla\cdot\mu\nabla\vec{u} + \rho\vec{g}(1-\beta T), \qquad \nabla\cdot\vec{u} = -\frac{1}{\lambda}\frac{\partial p}{\partial t} \qquad (2.2.6\text{-}2)$$

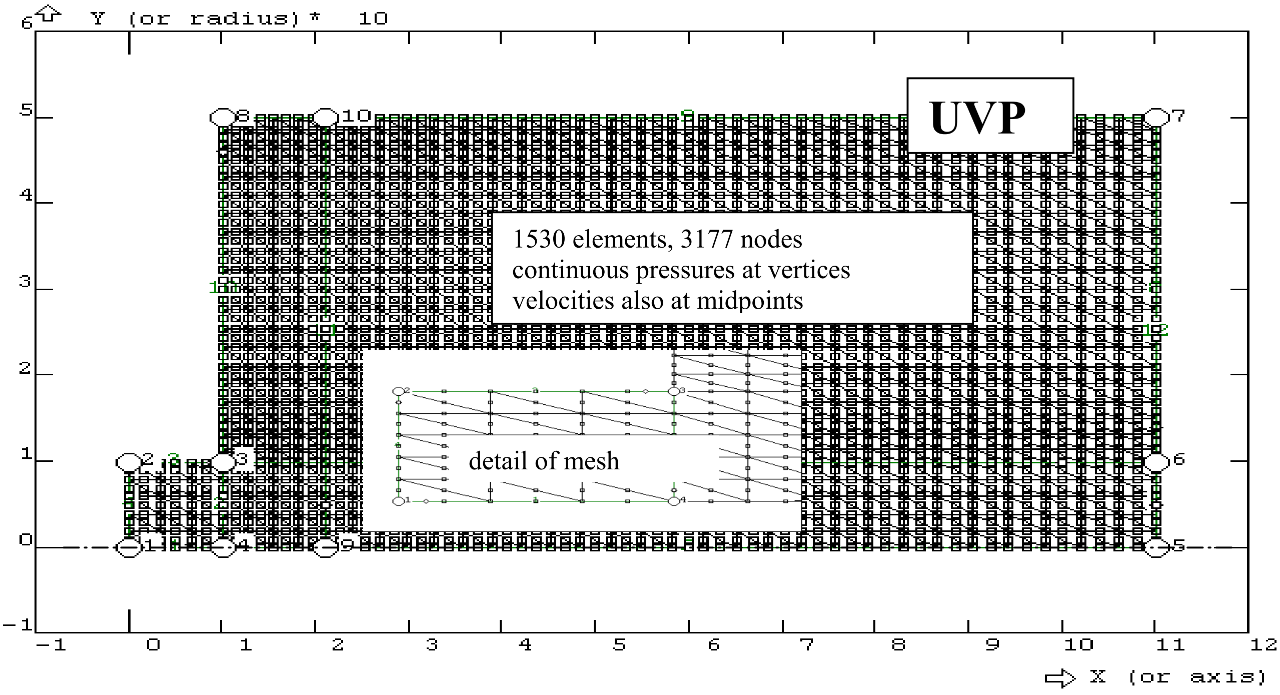
PENS: $$\rho\frac{D\vec{u}}{Dt} = \lambda\nabla(\nabla\cdot\vec{u}) + \nabla\cdot\mu\nabla\vec{u} + \rho\vec{g}(1-\beta T) \qquad (2.2.6\text{-}3)$$

Remarks: All the methods consider buoyancy term, with β - coefficient of thermal volumetric expansion. UVPP is pseudocompressibility method approximating continuity equation $\nabla\cdot\vec{u} = 0$ by Eq. (2) with very high value of user defined parameter λ. In principle the same parameter is used in the penalty method PENS for elimination of pressure from Navier Stokes equations.

All these methods are implemented for triangular elements with 6 nodes (velocities, approximated by quadratic functions, are identified by 6 nodes, vertices and midpoints of sides, while pressures, approximated only by linear polynomials are defined by 3 nodes in the triangular element vertices). Another option represents triangular elements with 3 nodes for linear approximation of velocities and one internal node for discontinuous pressure. The 3 nodes, allocated for velocities, can be either vertices (seems to be natural, but in fact not a very good idea) or midpoints of sides (much better elements). In a similar way quadrilateral finite elements with 5, 8 or 9 nodes are designed. All the methods make use upwind for improvement of convergence at higher Reynolds number flows.

2.2.6.1.Planar channel – UVP and UVPP method

The same problem, which has been solved in the preceding chapter, i.e. flow in symmetric planar channel with sudden expansion, will be used as an example of solution. Geometry and boundary conditions (parabolic velocity profile at inlet, Re=100) as well as number of triangular elements will be preserved.


Y (or radius) * 10
UVP
1530 elements, 3177 nodes
continuous pressures at vertices
velocities also at midpoints
detail of mesh
-1
0
1
2
3
4
5
6
7
8
9
10
11
12
X (or axis)

```
ANALYS 1;
SCALE -.500E-02,12,-.500E-02,.6;
PT 1,.000E+00,.000E+00;
PT 2,.000E+00,.100E+00;
PT 3,.100E+01,.100E+00;
PT 4,.100E+01,.000E+00;
PT 5,1.100E+01,.000E+00;
PT 6,1.100E+01,.100E+00;
PT 7,1.100E+01,.500E+00;
PT 8,.100E+01,.500E+00;
SF4PT 1,1,4,3,2;
SF4PT 2,4,5,6,3;
SF4PT 3,3,6,7,8;
EGROUP 1,FLOW2D,0,7,0,UVP,0;
MSF 1,3,5,1,1,6;
MSF 2,30,5,1.5,1,6;
MSF 3,30,20,1.5,-1.5,6;
NMERGE ;
```

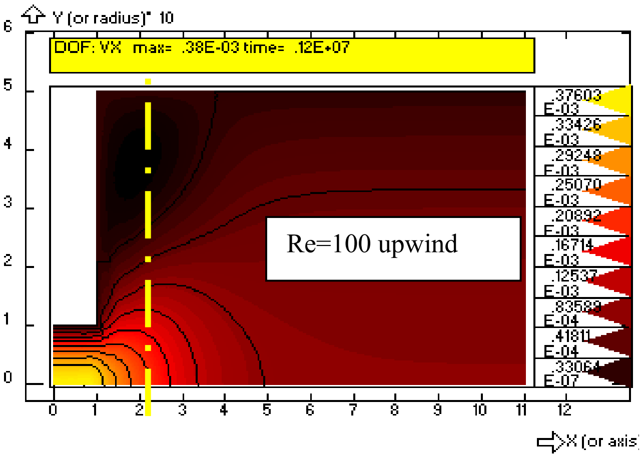
```
FUNDEF -4,0.000375*(1-100*YY**2);
FUNDEF -3,0.75E-4*(1-(YY/.5)**2);
NFCR 4,VX,-4,1,1;
NFCR 4,VY,-1,0,0;
NFCR 1,VY,-1,0,0;
NFCR 5,VY,-1,0,0;
NFCR 3,VY,-1,0,0;
NFCR 10,VY,-1,0,0;
NFCR 9,VY,-1,0,0;
NFCR 3,VX,-1,0,0;
NFCR 10,VX,-1,0,0;
NFCR 9,VX,-1,0,0;
NFCR 8,VX,-3,1,1;
tol=tol*5
NFCR 6,VX,-3,1,1;
NFCR 6,VX,-3,1,1;
NFCR 6,VY,-1,0,0;
```

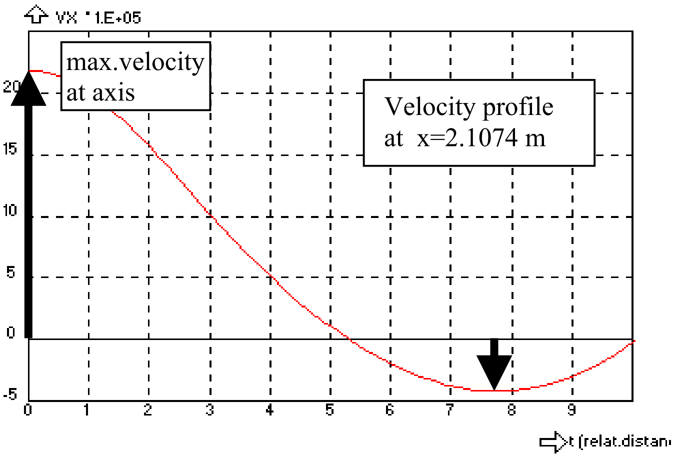
Note: There is no strong boundary condition for pressures!

Note: Boundary conditions for velocities VX and VY were prescribed by commands NFCR. The command looks for nodes having distance from a selected curve less than TOL, where boundary conditions are applied (there is no direct connectivity between nodes and curves). Some nodes on the curve could be sometimes skipped due to round off errors and then the tolerance TOL should be enlarged (see the command TOL=5*TOL). However, the fact, that there is no connectivity between nodes and curves, can be utilised for definition of auxiliary curves and for plotting graphs based upon nodes being near the curves (graphs in arbitrary cross sections) using GCR command.

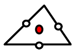
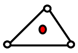
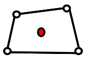
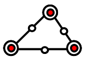
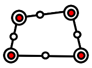
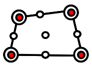
```
OPTION 0,0,0,1,0,15,.100E-11,1,.100E+08,1,.100E-04,.100E-04,0,3,0,0,0;
SOLVE .604E+06,20,.300E+05,0,0,3,0,0,0;
PT 9,2.1074,0;
PT 10,2.1074,.5;
CR2PT 11,9,10;
GCR 11
```

Graph of axial velocity at a selected cross section using newly defined curve number 11 (see next figure)



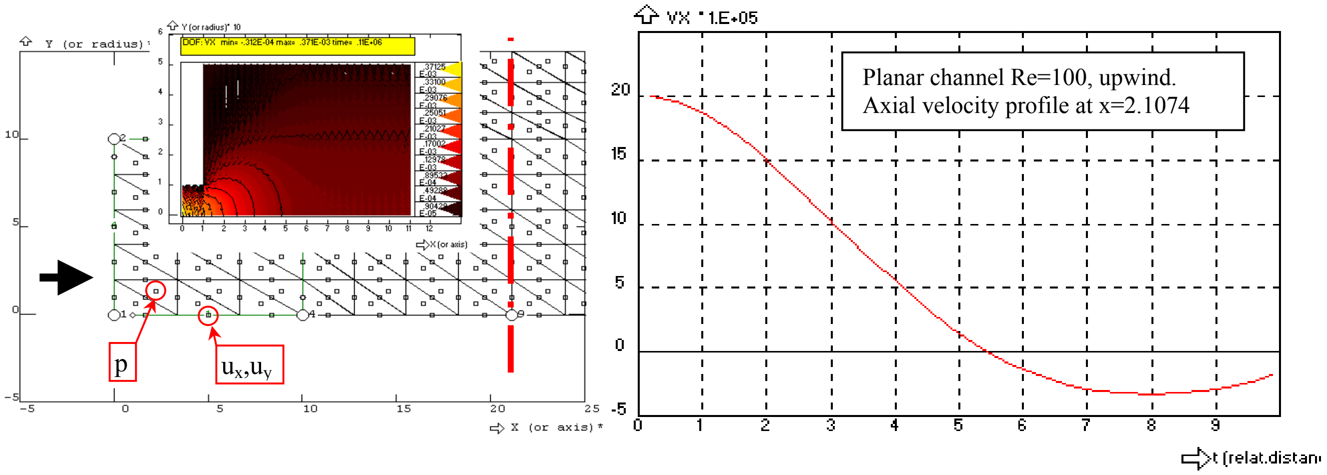


When using UVP and UVPP methods it is possible to select several different elements, and it is sufficient to change only the parameter in the MSF command. Available options are

`MSF 1,3,5,1,1,-4,1;`		T1P0
`MSF 1,3,5,1,1,4,1;`		P1P0 (frequently locks, LBB not fulfilled)
`MSF 1,3,5,1,1,5,1;`		Q1P0 (LBB not fulfilled)
`MSF 1,3,5,1,1,6;`		P2P1 (LBB stable, very good)
`MSF 1,3,5,1,1,8;`		$Q2^8$ Q1 (serendipity, Hood 1973, LBB stable not very good)
`MSF 1,3,5,1,1,9,1;`		$Q2^9$ Q1 (LBB stable, better element)

●	pressure
◉	veloc+pressure
○	velocities

The following figure corresponds to results obtained by using triangular elements with mid side nodes (3 velocities and constant pressure inside element):

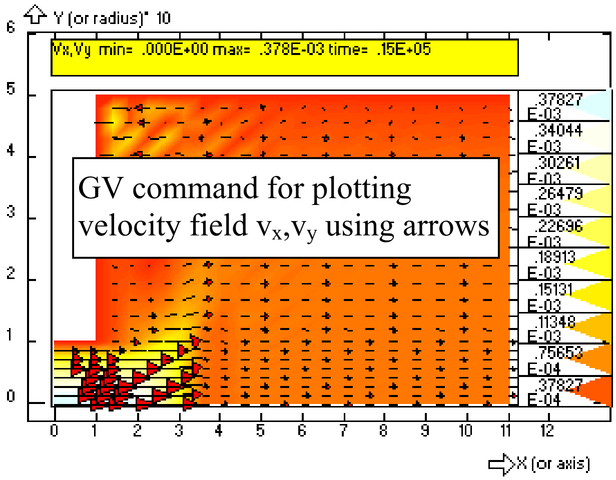


Planar channel Re=100, upwind.
Axial velocity profile at x=2.1074

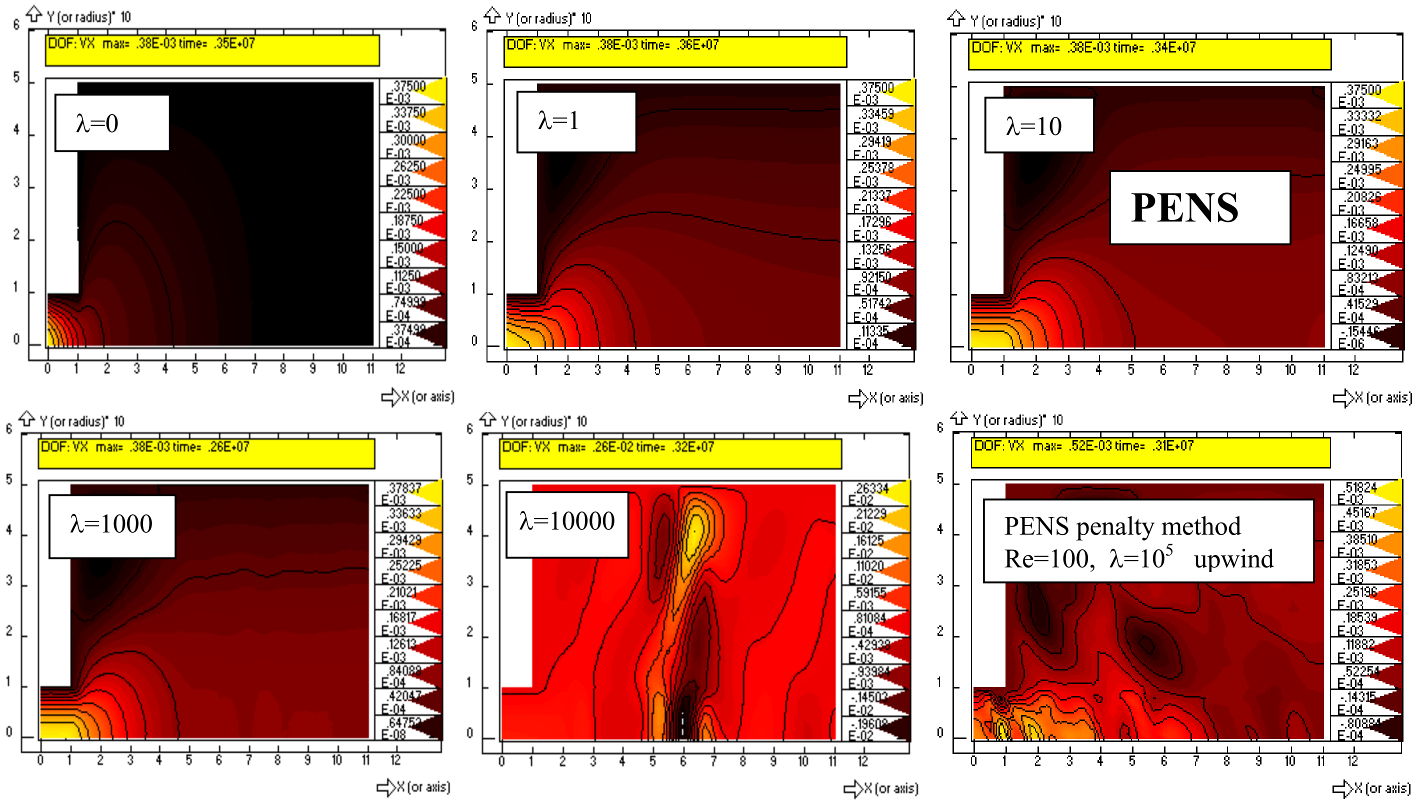
2.2.6.2.Planar channel – penalty method PENS

The penalty method PENS differs substantially from the previous methods UVP and UVPP. First of all by the fact that pressure is not a calculated nodal parameter and therefore any elements with arbitrary number of nodes can be used. The principle of the penalty method is as follows: The pressure is assumed to be proportional to the divergence of velocities (residuum of continuity equation), and this divergence, multiplied by a penalty factor λ, is substituted into the Navier Stokes equations instead of pressure. Thus the pressure is eliminated from the equations solved. A cost of this simplification is considerable – results depend upon value of penalty factor λ - if it is too small the continuity equation will be violated and if it is too large the Navier Stokes equations reduce in fact to the continuity equation having no unique solution. This is illustrated by results calculated for different values of λ from zero up to 10^8 (in all cases 6-node elements, upwind and natural boundary conditions at outlet have been used,)

```
EGROUP 1,FLOW2D,0,7,0,PENS,0;
SOLVE .249E+07,2,.300E+05,0,0,3,0,0,0;
LAMBDA=1E8
SOLVE .261E+07,5,.300E+05,0,0,3,0,0,0;
```

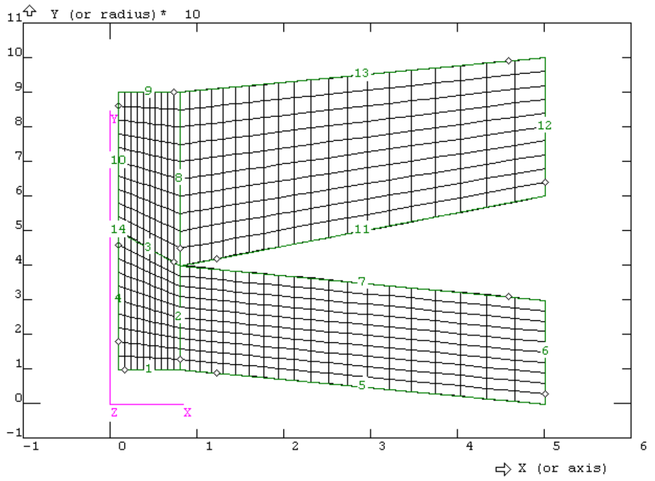


GV command for plotting velocity field v_x,v_y using arrows



2.2.6.3. Y channel – comparison of elements

In this example we compare behaviour of different trinagular and quadrilateral elements when using UVP method. The problem is formulated as laminar flow in a branched duct (planar channels are considered). A uniform velocity profile is prescribed at inlet v_x=1.25 m/s, channel width is 0.8 m (therefore the theoretical flowrate is 1 m^3/s, for unit depth of channel). For viscosity 10 Pa.s and density 1000 kg/m^3, the Reynolds number at inlet is 200.

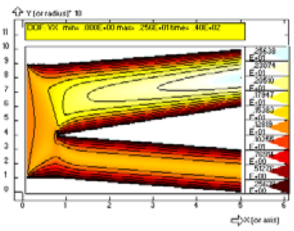
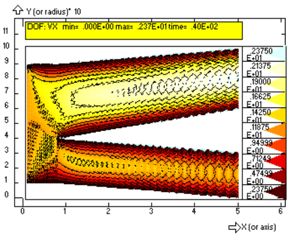
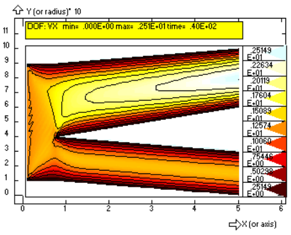
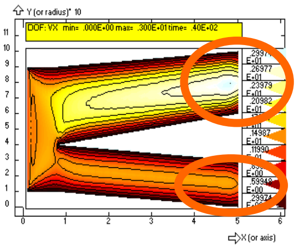


Solution based upon UVP method is described by the following session file, which allows interactive specification of mesh, elements (in terms of number of nodes) and number of Gauss integration points.

```
C* FYPSIL.GEO
ANALYS 2;
FUNDEF 1,DOF(VX,2,XX,YY,ZZ)
VALUE M
VALUE N
VALUE K
PT 1,.100E+00,.100E+00;
PT 2,.800E+00,.100E+00;
PT 3,.500E+01,.000E+00;
PT 4,.100E+00,.500E+00;
PT 5,.800E+00,.400E+00;
PT 6,.500E+01,.300E+00;
PT 7,.500E+01,.600E+00;
PT 8,.100E+00,.900E+00;
PT 9,.800E+00,.900E+00;
PT 10,.500E+01,.100E+01;
SF4PT 1,1,2,5,4;
SF4PT 2,2,3,6,5;
SF4PT 3,4,5,9,8;
SF4PT 4,5,7,10,9;
C* Gauss points
VALUE G
EGROUP 1,FLOW2D,0,G,0,UVP,0;
C* Kind of elements J (outer), I (inner)
VALUE J
VALUE I
```

```
MSF 1,M,N,1,1,J,I;
MSF 2,K,N,3,1,J,I;
MSF 3,M,N,1,1,J,I;
MSF 4,K,N,3,1,J,I;
CR2PT 14,1,8;
NMERGE ;
NFCR 14,VX,-1,1.25,1.25,1.25;
NFCR 14,VY,-1,0,0,0;
NFCR 1,VX,-1,0,0,0;
NFCR 1,VY,-1,0,0,0;
NFCR 5,VY,-1,0,0,0;
NFCR 5,VX,-1,0,0,0;
NFCR 7,VX,-1,0,0,0;
NFCR 7,VY,-1,0,0,0;
NFCR 11,VY,-1,0,0,0;
NFCR 11,VX,-1,0,0,0;
NFCR 13,VX,-1,0,0,0;
NFCR 13,VY,-1,0,0,0;
NFCR 9,VY,-1,0,0,0;
NFCR 9,VX,-1,0,0,0;
VISC(1)=10
BOX 0.3;
SOLVE 0,2,20,0,0,10,0,0,0,0,0;
INTGCR 12,1;
INTGCR 6,1;
INTGCR 14,1;
```

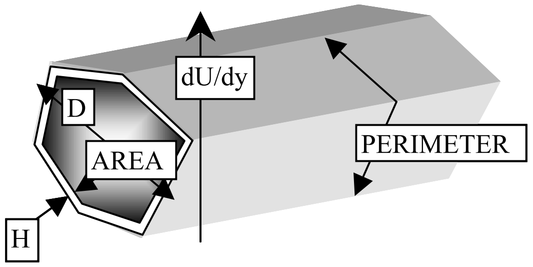
Results shown in the following table look similar, flowrates (V1,V2 at outlet, Vin at inlet) indicate that the continuity equation is satisfied, however the calculated powers P differ substantially.

UVP	T6+T3 3 Gauss p.	T3+1 3 Gauss p.	Q4+1 4x4 Gauss.p.	Q9+Q4 4x4 Gauss p.
Grid 15, 15, 30				
V1	0.693	0.621	0.668	0.736
V2	0.293	0.267	0.279	0.290
Vin	0.982 (2.365 : 1)	0.963 (2.326 : 1)	0.949 (2.394 : 1)	1.020 (2.538 : 1)
P [W]	4756	4186	4422	6183

2.2.7. Pipeline networks

The FEMINA program calculates pressure and flow rate distribution in a pipeline network using two-node elements PIPE2D, PUMP, CSTR (continuously stirred tank reactor) and four node elements HEXC (heat exchangers). The laminar and turbulent flow regimes together with buoyancy effects are considered, and the solution can include even rheologically more complicated liquids (only the dynamics of compressible fluids – hydraulic shocks cannot be solved). Besides the hydraulics, the distribution of temperatures and tracer concentration in liquid can be calculated at steady and unsteady state. Not only the straight pipe sections but also heat exchangers, ideal mixers or devices with user-defined hydraulic characteristics can be used (e.g. pumps, elbows, valves). On the basis of the computed pressures and temperatures, structural analysis and stresses evaluation can be carried out.

The basic element used for modelling pipelines is PIPE2D. The pipe need not have a circular cross section and only a part of its surface transfers a heat generally. And this is the role of RCONST parameters to complete description of geometry as well as some boundary conditions

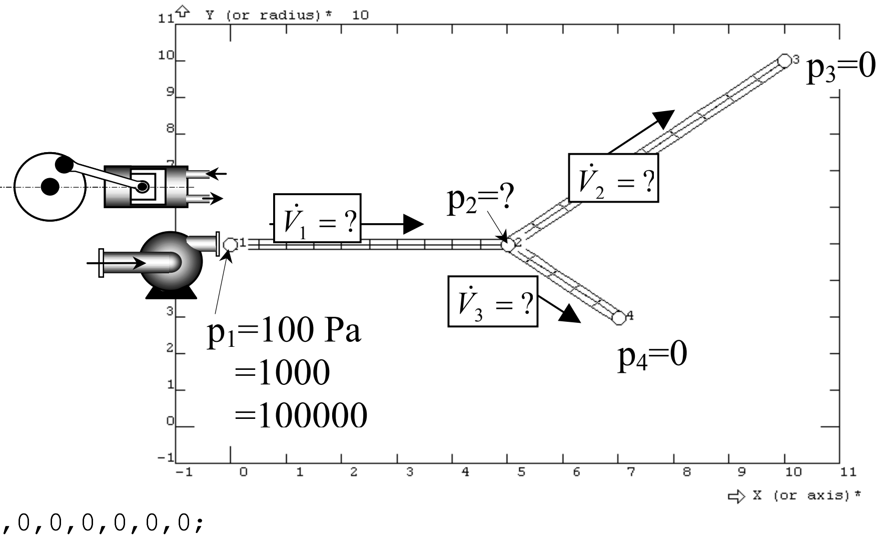


AREA – cross section (mean velocity is calculated as flowrate/AREA)
D - equivalent diameter used in Re and Nu numbers definition
H - thickness of wall is used only in stress analysis.
PERIMETER - is in fact a part of the whole perimeter where heat transfer (defined by heat transfer coefficient ALPHA and external temperature TE), takes place. If specified as zero, substituted by πD.
dU/dy – transversal component of electric field.
Axial profiles of pressure, temperature, concentration, elect. potential and thermal boundary layer thickness as well as fouling layer are calculated.

2.2.7.1. Pressure and flowrates in laminar/turbulent flows

The introductory example will demonstrate a computation of pressure losses and flowrates in a simple piping system which consists of three sections with flow splitting. Inner diameters of all pipes are identical (d=5 mm, roughness k=0) and water is used as a working liquid (μ=0.001 Pa.s). Pressures are specified at each boundary node of the system (zero at the outlet, the inlet pressure is going to be changed).

```
PT 1,.000E+00,.500E+00;
PT 2,.500E+00,.500E+00;
PT 3,.100E+01,.100E+01;
PT 4,.700E+00,.300E+00;
CR2PT 1,1,2;
CR2PT 2,2,3;
CR2PT 3,2,4;
```



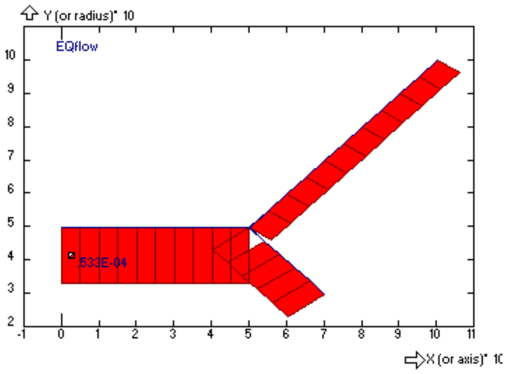
```
EGROUP 1,PIPE2D,0,0,0;
RCONST 1,PIPE2D,0.001,0.005,0,0,0,0,0,0;
MPROP 1,PIPE2D,.600E+00,0,4200,0,998,0,.400E-01,0,.210E+12,0,.280E+00,0,.100E-
02,0,.500E-03,0,.100E-08,0,0
MCR 1,10,1,2;
MCR 2,10,1,2;
MCR 3,5,1,2;
NMERGE ;
NFPT 1,PRES,-1,100000;
NFPT 3,PRES,-1,0;
NFPT 4,PRES,-1,0;
SOLVE 0,1,0,0,0,20,0,0,0;
```

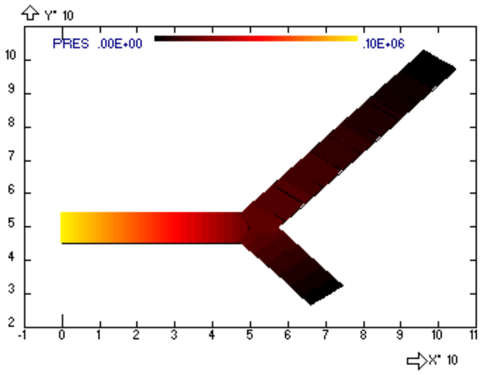
EGROUP enables to specify nonstandard hydraulic characteristics of PIPE as a function (not used here), RCONST defines not only dimensions of pipe (diameter, thickness of wall) but also wall roughness and coefficient of local pressure losses (not used here).

Grid generation on curves 1 through 3. The second parameter is number of elements, the third parameter (1) determines uniform meshing, and the last parameter (2) specifies number of nodes.

```
NFPT 1,PRES,-1,1000;
SOLVE 0,1,0,0,0,10,0,0,0;
```

Display results with commands GE1 (parameters EQ-flow rate, ERE-Reynolds number, ETAU-shear stress on wall) and GD1 (nodal parameter PRES – pressure distribution).





Results of computation are pressures in nodes and flow rates stored as parameters of individual elements (for a quick inspection of results, NID and EID can be used – parameter identification by a mouse pointer). Three different pressures 10^2, 10^3, 10^5 Pa at the inlet correspond to different flow regimes: laminar, mixed and turbulent – the following table presents the results which agree at all valid decimal positions with the accurate analytical solution.

p_1 [Pa]	p_2 [Pa]	$\dot{V}_1$ [ml/s]	$\dot{V}_2$ [ml/s]	$\dot{V}_3$ [ml/s]
100	28.78	2.185 (Re=555)	0.6243 (Re=159)	1.561 (Re=397)
1000	161.4	12.25 (Re=3110)	3.50 (Re=890)	8.75 (Re=3170)
100000	20000	169.7(Re=43100)	63.1 (Re=16000)	106.6 (Re=2700)

If it is not a pressure but a flow rate which is to be specified at the inlet (for example if a positive displacement pump instead of a radial-flow pump or a pressure accumulator is used), the intensity of source, i.e. the mass flow rate (kg/s), must be given as the node parameter,

```
NFPT 1,PRES,2,0.05;
SOLVE 0,1,0,0,0,10,0,0,0;
```

Status of the node parameter PRESS is changed to 2 (a strong boundary condition is replaced by a load), and the mass flow rate value is specified $\dot{V}_1 = 50\, ml/s$ corresponds to the mass flowrate $\dot{m}_1 = 0.05 kg/s$

p_1 [Pa]	p_2 [Pa]	$\dot{V}_1$ [ml/s]	$\dot{V}_2$ [ml/s]	$\dot{V}_3$ [ml/s]
11850	2413	50	18.4	31.6

Remark: You can specify inlet or outlet flowrate at any, even internal, node (in this way for example a leakage can be described). The point source/sink is indicated by the status of nodal parameter PRES within the range 1,2,...,10. Prescribed positive value of parameter PRES represents in this case the inflow in units kg/s or if you set the system variable VOLFLOW=1 in units m^3/s! [13] It is possible to define also time variable flowrate by user defined function with an index 1,2,...,10 (time is represented by system variable TIME). If a function with the index of status PRES exists, prescribed value of nodal parameter (nominal flowrate) is multiplied by this function. In the same way a time dependent pressure can be prescribed using a function with an index -10,...,-1 (strong boundary condition). The rule is simple: Any specified nodal parameter is multiplied by a user defined function with the index of specified status. If such a function has not be defined, identity is assumed, it means that the prescribed value is multiplied by one.

[13] This is not only a problem of units: VOLFLOW=0 means, that method of mass balancing will be applied, while at VOLFLOW=1 distribution of pressures will be solved from equations representing volumetric flowrate balances.

2.2.7.2. Non-Newtonian Power-law liquids

Computation with non-Newtonian liquids is more complicated because the Rabinowitsch, Mooney Weissenberg equation, see Steffe 1996, must be integrated numerically in each finite element connecting nodes *i,j* (radius R_{ij}, length L_{ij}, friction pressure drop Δp_{ij}, shear stress τ):

$$\dot{m}_{ij} = 8\pi\rho(\frac{L_{ij}}{\Delta p_{ij}})^3 \int_0^{R_{ij}} \frac{\tau^3}{\mu_{apparent}(\tau)} d\tau \,. \tag{2.2.7-1}$$

Rheological model characterising properties of a non-Newtonian liquid are usually specified as a function γ(τ), where γ is the deformation rate (1/s) and τ is the corresponding shear stress (Pa). The constitutive equation is specified in FEMINA in form of apparent viscosity μ_{apparent} as a function of shear stress TAU and temperature TEMP. Example: for probably the most often used power-law model (sometimes called Ostwald de Vaele model) the following constitutive equation holds

$$\gamma = (\frac{\tau}{K})^{1/n} \qquad \mu_{apparent}(\tau) = (\frac{K}{\tau^{1-n}})^{1/n} \tag{2.2.7-2}$$

where *K* is the consistency coefficient and *n* is the flow behaviour index. The model parameters can be constants, but it is usually more convenient to use variables (for example, we will use the variable *A* as the consistency coefficient and the variable *B* as the flow behaviour index):

```
     FUNDEF 1,(A/MAX(1E-10,TAU)**(1-B))**(1/B);

MPROP  1,PIPE2D,.600E+00,0,4200,0,998,0,.400E-01,0,.210E+12,0,.280E+00,0,.100E-02,
1, .500E-03,0,0,0,0,0,0,0;
A=1
B=.8
SOLVE 0,1,0,0,0,20,0,0,0;
```

This definition of the function number 1 should replace the Newtonian liquid rheological model, which is done by specifying a non-zero function number of the viscosity material parameter: Computations can be repeated for different values of flow indices *n* (*K*=1 Pa.s^n, p_1=1000 Pa) and the following table shows perfect agreement with the analytical solution (in laminar regime)

$$\dot{V} = \pi R^3 \frac{n}{3n+1}(\frac{\tau_w}{K})^{1/n} \qquad \text{where} \quad \tau_w = \frac{R\Delta p}{2L} \tag{2.2.7-3}$$

n (flow index)	p_2 [Pa]	$\dot{V}_1$ [ml/s]	$\dot{V}_2$ [ml/s]	$\dot{V}_3$ [ml/s]
0.8	312	0.0227	0.00549	0.0173
0.6	335	0.0246	0.00438	0.0202
0.4	352	0.0298	0.00273	0.0270
0.2	361	0.0639	0.000648	0.0633

It should be noticed, that the strong nonlinearity of the rheological model at low values of the flow behaviour index *n* causes problems. For the values of the flow index *n*<0.5 the iterations do not converge and it is necessary to reduce the relaxation factor (system variable RELFAKT, e.g. RELFAKT=0.2).[14]

[14] If the sum of flow rates in branching points of pipes is not zero, something is wrong. In our case the flow rate in the first pipe (it should be same in all elements from 1 to 10) should equal to the sum of flow rates in pipes 2 and 3. If this does not hold true the number iterations must be increased or eventually the RELFAKT must be decreased.

2.2.7.3.Non-Newtonian Bingham, Herschel Bulkley and Casson's liquids

Serious convergence problems have been encountered when calculating Bingham liquids (or generally liquids with significant yield stress) at turbulent flows. The Bingham liquid can be described

$$\mu_{apparent}(\tau)=\frac{\tau\mu_p}{\tau-\tau_0}=\lim_{t\to\infty}\frac{\tau\mu_p}{\tau-\tau_0\frac{t}{N+t}} \tag{2.2.7-4}$$

where μ_p [Pa.s] is plastic viscosity and τ_0 [Pa] is yield stress. In this case improvement has been achieved by selective under-relaxation of the yield stress, more specifically by using the constitutive equation in the following form

```
FUNDEF 1,A*TAU/MAX(1E-10,TAU-B*TIME/(N+TIME))
```

where the user variable A is plastic viscosity μ_p and the variable B is the yield stress τ_0. TIME and TAU are system variables (simulation time and wall shear stress). If the variable N is much greater than TIME, that is at the beginning of calculation, the model reduces to the Newtonian liquid with apparent viscosity μ_p, and only with the increasing time (if TIME>>N) the computational yield stress is approaching to the correct value τ_0.

Resulting pressure and flowrate for p_1=100 Pa at inlet and μ_p=1 Pa.s, calculated for different values τ_0, are summarised in the following table

τ_0 (yield stress)	p_2 [Pa]	$\dot{V}_1$ [ml/s]	$\dot{V}_2$ [ml/s]	$\dot{V}_3$ [ml/s]
0	17.51	0.09979	0.003866	0.06113
0.01	30.31	0.001975	0.000494	0.00148
0.03	33.21	0.001561	0.000246	0.001313

Accuracy of results can be ascertained by comparison with flowrates, calculated from the Buckingham equation (expressed in terms of τ_w - wall shear stress)

$$\dot{m}=\frac{\rho\Delta pR^2}{8\mu_pL}[1-\frac{4\tau_0}{3\tau_w}+\frac{1}{3}(\frac{\tau_0}{\tau_w})^4].\qquad \tau_w=\frac{R\Delta p}{2L} \tag{2.2.7-5}$$

A more general Herschel Bulkley model is a combination of the power law and the Bingham models (the HB model has three parameters, yield stress τ_0, flow behaviour index n and consistency K)

$$\mu_{apparent}(\tau)=\frac{\tau K^{1/n}}{(\tau-\tau_0)^{1/n}}=\lim_{t\to\infty}\frac{\tau K^{1/n}}{(\tau-\tau_0\frac{t}{N+t})^{1/n}}. \tag{2.2.7-6}$$

The HB model can be specified as the function (variables A-consistency, B-yield stress, C-flow index)

```
FUNDEF 1,TAU*(A/MAX(1E-10,TAU-B*TIME/(N+TIME)))**(1/C)
```

Results obtained by Herschel Bulkley model can be verified in laminar regime by comparing with the analytical solution

$$\dot{m}=\pi R^3\rho(\frac{\tau_w}{K})^{1/n}(1-\frac{\tau_0}{\tau_w})^{(n+1)/n}\frac{[2n^2+3n+1+2n(n+1)\frac{\tau_0}{\tau_w}+2(n\frac{\tau_0}{\tau_w})^2]n}{(3n+1)(2n+1)(n+1)}\qquad \tau_w=\frac{R\Delta p}{2L} \tag{2.2.7-7}$$

It can be easily verified, that this equation reduces to the Buckingham equation for n=1 and K=μ_p.

Casson's model is another example of a generalised Newtonian liquid, applied first of all for modelling of blood flow. This model has only two parameters: yield stress τ_0 and viscosity coefficient K_e. In the case that the yield stress vanish, the model reduces to the Newtonian liquid, just like the Bingham model, however unlike the Bingham model the shear stress is not a linear function of shear rate. Apparent viscosity can be expressed as follows

$$\mu_{apparent}(\tau)=\frac{\tau K_e^2}{(\sqrt{\tau}-\sqrt{\tau_0})^2} \qquad (2.2.7\text{-}8)$$

Using variables A for K_e and B for the yield stress, the Casson's model (8) can be defined as

FUNDEF 1,TAU*A**2/MAX(1E-10,SQR(TAU)-SQR(B))**2

Results can be tested again using analytical solution for mass flowrate of Casson's liquid

$$\dot{m}=\pi R^3\rho\frac{\tau_w}{K_e^2}[\frac{1}{4}(1-(\frac{\tau_0}{\tau_w})^4)+\frac{4}{7}\sqrt{\frac{\tau_0}{\tau_w}}(1-(\frac{\tau_0}{\tau_w})^{7/2})+\frac{1}{3}(\frac{\tau_0}{\tau_w})(1-(\frac{\tau_0}{\tau_w})^3)] \qquad (2.2.7\text{-}9)$$

2.2.7.4. Non-Newtonian thixotropic liquids

Thixotropic liquids are characterised by viscosity, that depends not only upon the actual rate of deformation, but also upon the whole deformation history of a material element. Memory of liquid is hidden in a structural parameter λ characterising state of structure: λ=0 – structure is completely destroyed and viscosity is low, λ=1 – structure is perfectly build and viscosity is high. It is assumed that the structural parameter is a transport property. Probably the simplest and still realistic constitutive equation can be written in the following form, which is a generalisation of the power-law liquid

$$\tau=(K+\Delta K\lambda)\dot{\gamma}^n, \qquad \frac{D\lambda}{Dt}=a(1-\lambda)-b\lambda\dot{\gamma}^m, \qquad (2.2.7\text{-}10,11)$$

where $K+\Delta K\lambda$ is effective consistency coefficient, n is power-law index, a is a measure of structure build-up, while b is a coefficient of structure decay. The equation (11) is in fact a rate equation This equation can be rewritten to the form with apparent viscosity expressed as a function of shear stress

$$\mu_{apparent}=\frac{\tau}{\dot{\gamma}}=((K+\Delta K\lambda)\tau^{n-1})^{1/n} \qquad \frac{D\lambda}{Dt}=a\boxed{-}\lambda[a\boxed{+b(\frac{\tau}{K+\Delta K\lambda})^{m/n}].} \qquad (2.2.7\text{-}12,13)$$

We shall use concentration c_N in the role of structural parameter λ described by Eq.(8) with source term (red circle) defined by function 10 with user variable A as the build up parameter *a*

```
FUNDEF 10,A;
```

Sink term (blue circle) is the function 11 with user variables A=*a*, B=*b*, C=*K*, D=Δ*K*, E=*m*, F=*n*

```
FUNDEF 11,A+B*(TAU/MAX(1E-10,C+D*CN))**(E/F);
```

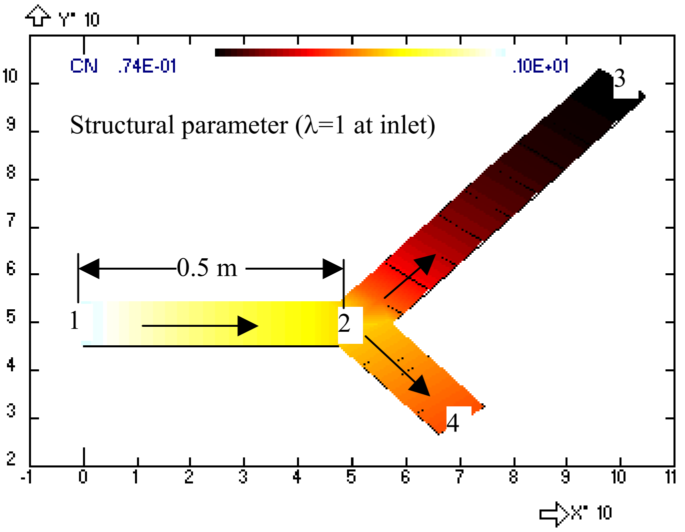
Apparent viscosity is defined by function 1

```
      FUNDEF 1,((C+D*CN)/MAX(1E-10,TAU)**(1-F))**(1/F);
```

Sources and sink term functions are referenced in EGROUP, while the viscosity function 1 is referenced in MPROP.

We shall use the model of thixotropy for description of flow in the same Y-system of pipes as previously for the following values of rheological parameters characterising plaster NEOPONIT, see Šesták et al (1990): K=115 [Pa.s^n], ΔK=94 [Pa.s^n], n=0.42, m=0.4, a=0.0001 [1/s], b=0.003955 [s^{m-1}]. The Eq.(8) assumes only convective transport, however some diffusion coefficient must be specified when using weighted residual method, for example by accepting very small default value ($D_N=10^{-8}$ m^2/s) and suppressing axial dispersion (TAUD=0). The solution is than described by the following batch

```
PT 1,.000E+00,.500E+00;
PT 2,.500E+00,.500E+00;
PT 3,.100E+01,.100E+01;
PT 4,.700E+00,.300E+00;
CR2PT 1,1,2;
CR2PT 2,2,3;
CR2PT 3,2,4;
FUNDEF 10,A;
FUNDEF 11,A+B*(TAU/MAX(1E-10,C+D*CN))**(E/F);
FUNDEF 1,((C+D*CN)/MAX(1E-10,TAU)**(1-F))**(1/F);
FUNDEF 2,1;
MPROP 1,PIPE2D,.600E+00,0,4200,0,998,0,.400E-01,0,.210E+12,0,.280E+00,0,1,1,.500E-
03,0,.100E-08,0,.100E-08,0,.100E-08,0,0,0,0,0,0,0,0,.120E-04,0;
EGROUP 1,PIPE2D,0,0,0,0,0,0,10,11,0,0,0,0;
MCR 1,10,1,2,0,3,1;
NMERGE ;
A=0.0001
B=0.003955
C=115
D=94
E=0.4
F=0.42
NFPT 1,PRES,3,0.01;
NFPT 3,PRES,-1,0;
NFPT 4,PRES,-1,0;
NFPT 1,CN,-1,1;
INITIA CN,2;
RELFAKT=0.5
TAUD=0
OHMI=1
SOLVE 0,3,1,1,0,20,0,10,0,0,0;
```



It follows from results in table, that at higher flowrates the structure is not so much destroyed!

$\dot{m}_{inlet}$ [kg/s]	λ_2	λ_3	λ_4	p_1 [Pa]	p_2 [Pa]
0.001	0.118	0.064	0.0575	3458	1048
0.01	0.62	0.074	0.461	11540	3648
0.05	0.845	0.3551	0.7611	24780	8300

2.2.7.5. Heat transfer in pipes

It is possible to evaluate not only pressures and flowrates, but also concentration and temperature changes in pipes due to either internal sources (for example ohmic heating) or heat transfer between the pipe surface and surrounding. Heat transfer is determined by specified ambient temperature (as the RCONST parameter T_e) and by the overall heat transfer coefficient enclosing thermal resistance of wall and outer heat transfer (this is specified as the RCONST parameter ALPHA), thermal resistance of fouling (this is calculated by selected model of fouling) and first of all by inner thermal resistance of liquid flowing inside the pipe. This last item is calculated separately at laminar and turbulent flow using correlations for thermal boundary layer thickness δ

$$\alpha = \frac{\lambda}{\delta}. \qquad (2.2.7\text{-}14)$$

Methods calculating axial profile of the thickness δ are described in the theoretical part of this manual, and from user's point of view it is important only to know, that the thermal boundary layer thickness calculation works automatically for Newtonian and nonnewtonian liquids as well as for temperature dependent viscosity[15]. Only in non-circular tubes a manual correction is needed (parameter OPTION, BL-correction) and experience shows that this correction (usually BL<1, which means reduction of the thickness δ) is necessary if there exist factors increasing heat transfer, for example by buoyancy induced macroinstabilities, vortex generators,…

2.2.7.6. Heat transfer coefficient – effect of axial dispersion

Temperature profiles along pipes are obtained as a solution of enthalpy balance – Fourier Kirchhoff equation integrated across the cross section of pipe

$$A\rho c_p\left(\frac{\partial T}{\partial t} + \bar{u}\frac{\partial T}{\partial x}\right) = \frac{\partial}{\partial x}\left(A\rho c_p a_e \frac{\partial T}{\partial x}\right) + kO(T_e - T) + A(Q - ST) \qquad (2.2.7\text{-}15)$$

where $T(t,x)$ is a mean calorimetric temperature, a_e is axial dispersion, k is overall heat transfer coefficient, O is perimeter of pipe, A cross section, and T_e is ambient temperature. The coefficient a_e describes axial thermal dispersion – a mechanism of smoothing temperature nonuniformities along a pipe. This dispersion is caused not only by the molecular diffusion in the axial direction (characterised by thermal diffusivity a) but also by a combined effect of radial diffusion and radial velocity profile. The axial dispersion coefficient a_e has been theoretically derived for slow laminar flows by Taylor (1953) and for turbulent flow also by Taylor one year later (1954)[16]. In both cases the predicted axial dispersion a_e is much higher than the molecular diffusivity a, typically hundred times or more and cannot be ignored. However, the axial dispersion model in the form implemented in FEMINA (according to Taylor's results) is correct only if the pipe is insulated[17]. This restriction can be sometimes accepted, for example when modelling dispersion of temperature disturbances in an almost insulated pipe, nevertheless probably the most important problem – stationary heat transfer with a constant wall temperature is distorted significantly at very high values of axial dispersion coefficient a_e. And this is subject of the following example:

We shall consider a straight pipe (d=0.02 m, L=1 m, hydraulically smooth) with nearly negligible external thermal resistance (α_e=10^6 W.m^{-2}.K^{-1}), it means with approximately constant temperature of wall T_e=T_w=100 °C. Inlet temperature is constant T_0=0 °C and we shall calculate outlet

[15] This is accomplished by numerical integration of RMW equation in individual elements, which is activated if the viscosity function is defined as a function of shear stress or if the index of viscosity function is greater than 10 (this is necessary if we need to take into account the influence of variable viscosity of Newtonian liquids to heat transfer).

[16] See also the model of axial dispersion of mass discussed in the paragraph 2.2.4.3.

[17] Not only insulated: Taylor's model of axial dispersion assumes either slow laminar flow or on the other hand turbulent flow. There is a rather wide gap of flowrates (fast, but still laminar flows) when applicability of the axial dispersion model is questionable.

temperatures (mean calorimetric temperatures at distance L=1 m from inlet) for different mass flowrates of water starting with the slow laminar flow at 0.0005 kg/s (Re=32) up to turbulent flow at 0.146 kg/s (Re=9300). There are several parameters in the OPTION dialog panel and corresponding system variables affecting the solution method: weighted residual/enthalpy balance method (variable HEPI), internal heating (OHMI), buoyancy (BUOY), effect of fouling (RF), correction of thermal boundary layer thickness (BL) and modification of axial dispersion model (system variable TAUD). This variable is the time constant τ in the emergency model

$$a_e = a + \frac{u^2\tau}{\pi} \qquad (2.2.7\text{-}16)$$

and FEMINA applies the smallest value of dispersion coefficients calculated according to Eq.(11) or according to Taylor's model for laminar/turbulent flows (with the aim to avoid excessively large dispersion a_e). It means that if we select TAUD=0 axial dispersion will be suppressed (reduced to thermal diffusivity), while very large value, e.g. TAUD=10^6, ensures, that the dispersion will be evaluated according to Taylor. We shall calculate temperature profile for TAUD=0 and 10^6 respectively, and save results (outlet temperatures for different mass flowrates) into second and third column of matrix of observation points. The whole procedure is described by the following command file, where the control command #LOOP is used for repeating solutions for increasing flowrates

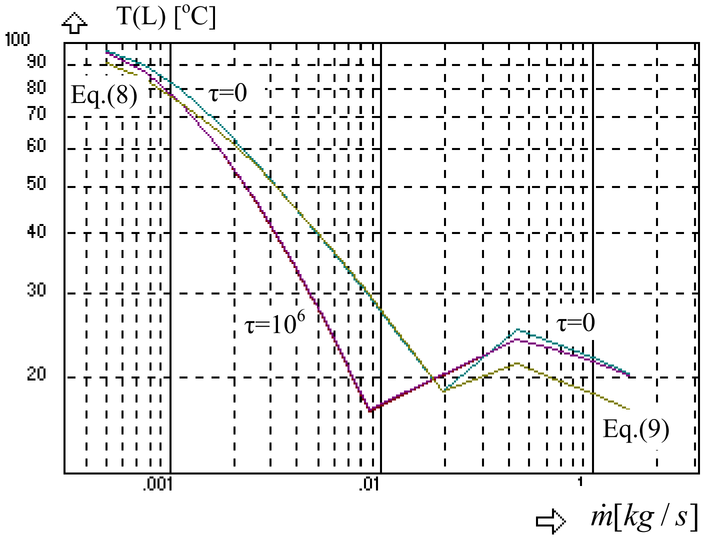
```
PT 1,.000E+00,.000E+00;
PT 2,.100E+01,.000E+00;
CR2PT 1,1,2;
RCONST 1,PIPE2D,.200E-01,0,0,.100E-02,1E8,100;
EGROUP 1,PIPE2D,0,0,0,0,0,0,0,0,0,0,0;
MPROP 1,PIPE2D,.6,0,4200,0,998,0,.04,0,.21E+12,0,.28,0,.001;
VALUE N
MCR 1,N,1,2,0,0,1;
NF ND,PRES,-1,0;
NF 1,TEMP,-1,0;
A=0.0005
J=0
NT=15
#LOOP L,NT
J=J+1
C1(J)=A
NF 1,PRES,1,A
TAUD=0
SOLVE 0,1,1,0,0,10,1,0,0,0,0;
C2(J)=V2(LPU(TEMP)+ND)
TAUD=1E6
SOLVE 0,1,1,0,0,10,1,0,0,0,0;
C3(J)=V2(LPU(TEMP)+ND)
A=A*1.5
#LABEL L
```

Results in MOP (first column C1 is mass flowrates, second column C2 are outlet temperatures with reduced dispersion and third column C3 temperatures according to axial dispersion model) can be completed by column of results calculated according to the most common correlations for Nusselt number at laminar (Leveque) and turbulent (Colburn) flow regime

Laminar: $$\frac{T(x)-T_e}{T_0-T_e} = \exp(-1.615\pi\frac{\lambda}{\dot{m}c_p}\sqrt[3]{\frac{ud^2x^2}{a}}) \qquad (2.2.7\text{-}17)$$

Turbulent $$\frac{T(x)-T_e}{T_0-T_e} = \exp(-\pi\frac{\lambda x}{\dot{m}c_p}Nu), \qquad Nu = 0.023\,\mathrm{Re}^{0.8}\,\mathrm{Pr}^{0.333}. \qquad (2.2.7\text{-}18)$$

Use for example Excel to evaluate Eqs.(17-18), the command TCI to transfer results into matrix of observation points and the command GTD to plot the calculated temperatures as a function of mass flowrate in the log-log graph:



$\dot{m}[kg/s]$	Re	a_e [m^2/s]	T – τ=10^6	T – τ=0	T – Eqs.(8-9)
0.0005	32	0.000037	95.4	96.7	90.8
0.00253	161	0.00095	46.8	56.0	55.5
0.02883	1836	0.0087	12.3	14.7	23.05
0.146	9294	0.0053	20.3	20.6	17.3

Discussion: Axial dispersion predicts always lower temperatures at outlet and apparently lower heat transfer coefficient. This is caused by the increased axial conductivity (thermal diffusivity a=1.4.10^{-7} is more than ten thousand times lower than dispersion a_e) and by fixed temperature at inlet – heat flux is therefore against the flow direction. Therefore it is better to suppress the axial dispersion (e.g. by TAUD=0) if a steady state heat transfer is to be calculated.

2.2.7.7.Heat transfer coefficient – temperature dependent viscosity

Temperature dependent viscosity influences not only the pressure drop, but also the heat transfer coefficient as soon as temperature depends upon radius and deforms the radial velocity profile. A problem is that the 1D solution contains only one piece of information about the radial temperature profile – nodal temperature (this is mean calorimetric temperature). Nevertheless, another piece of information is in the estimate of thermal boundary layer thickness δ. Assuming linear temperature profile within this layer and a uniform temperature in core the wall temperature T_w and the wall shear rate Γ can be estimated

$$\Gamma = \frac{\tau_w}{\mu_w(\tau_w, T_w)}. \qquad (2.2.7\text{-}19)$$

In terms of Eq.(20) the boundary layer thickness and therefore the heat transfer coefficient is corrected

$$\delta^3 = \delta_0^3 + \varphi \frac{\pi a L}{\Gamma}. \qquad (2.2.7\text{-}20)$$

where δ$_0$ is boundary layer thickness in previous element (previous from point of view of flow direction), L is length of element and φ is coefficient determined from experiments (φ=1.845 is used in FEMINA).

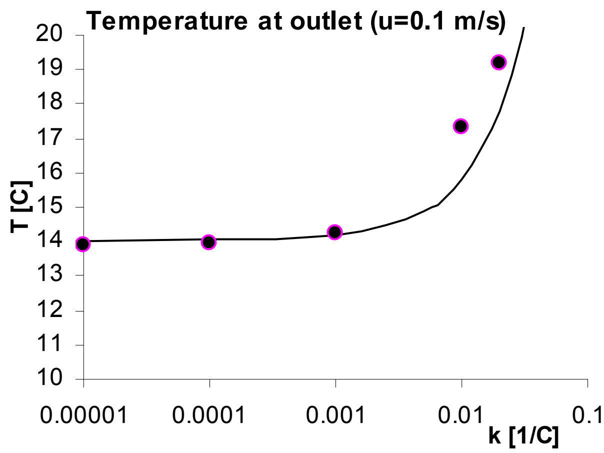
Example: Heating of liquid characterized by viscosity $\mu(T) = \mu_0 e^{-k(T-T_0)}$, in a pipe with the constant wall temperature T_w=100 ^{0}C, when the inlet temperature is T_0=0 ^{0}C. Dimension of pipe D=0.02 m, L=1 m (the same formulation as in the previous example). Mean velocity u=0.1 m/s. Thermodynamic parameters of liquid λ=0.6 W/(m.K), c_p=4200 J/(kg.K), ρ=1000 kg/m^3. The problem can be solved by the following simple program

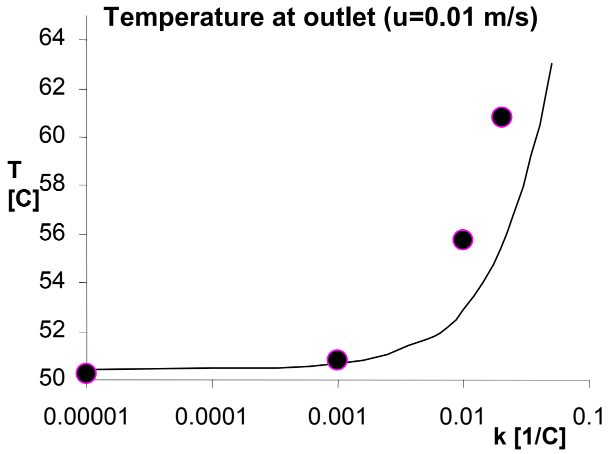
```
PT 1,.000E+00,.000E+00;
PT 2,.100E+01,.000E+00;
CR2PT 1,1,2;
RCONST 1,PIPE2D,0.02,0,0,0.001,1E5,100,0,0,0,0,0,0,0,0,0,0,0;
MPROP 1,PIPE2D,.600E+00,0,4200,0,998,0,.400E-01,0,.210E+12,0,.280E+00,0,.100E-
02,11,.500E-03,0,.100E-08,0,0,0,0,0,.120E-04,0;
MCR 1,50,1,2,0,0,1;
NFPT 1,TEMP,-1,0;
NFPT 1,PRES,1,0.0314;
NFPT 2,PRES,-1,0;
OPTION 0,0,0,0,1,0,0,15,.1E-11,1,.1E+08,1,.1E-8,1,0,0,0,.1E-04,.1E-
04,0,3,1,0,0,0,0,
FUNDEF 11,EXP(-0.001*TEMP)
SOLVE 0,10,1,0,0,3,1,0,0,0,0;
```

Results, temperatures at outlet, corresponding to different values of *k* (in previous program *k*=0.001 in the function number 11) can be compared with the analytical solution, based upon Leveque correlation and Sieder Tate correction

$$Nu_{\ln} = 1.618\sqrt[3]{Gz}(\frac{\mu}{\mu_w})^{0.14}. \tag{2.2.7-21}$$

The graphs demonstrate relative temperature increase with the increasing *k* for the case of heating due to decrease of viscosity at hot wall and therefore increased velocity gradient and decreased thickness of thermal boundary layer. For the case of cooling, when the temperature of wall is less than the inlet temperature, the trends are opposite.





2.2.7.8. Heat transfer and fouling in pipes

Gradually increasing thermal resistance *r* [m^2.K/W] of fouling layer, caused by degradation of oil, proteins or crystallisation, depends first of all upon temperature *T*, but also upon wall shear stress τ and composition. Time changes of thermal resistance *r* are described by a model with the rate function $f(T, \tau, c_N, c_D, c_A)$

$$\frac{dr}{dt} = f. \tag{2.2.7-22}$$

The function *f* (its index) is specified as a parameter OPTION, usually using menu SOLVE-OPTION as a model of fouling (1- is reserved for the predefined Ebert Panchal model, while 2,3,…. are assessed for user defined fouling models).

Let us consider as an example the model for milk fouling described by de Jong, Toyoda and others. The model assumes that the rate of fouling as proportional to the concentration of agglomerated proteins c_A. Let us define this model as the function 6, with the proportionality coefficient A (predefined user variable)

FUNDEF 6,A*CA

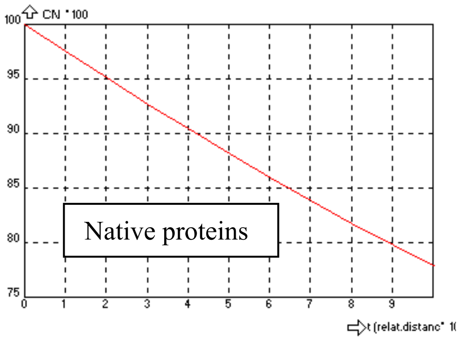
The concentration of agglomerated proteins CA is determined by biochemical reactions between native proteins c_N, denatured proteins c_D and agglomerates c_A (see chapter 4 – theory) and these concentrations have to be solved from appropriate transport equations with user defined source and sink terms, see the following table:

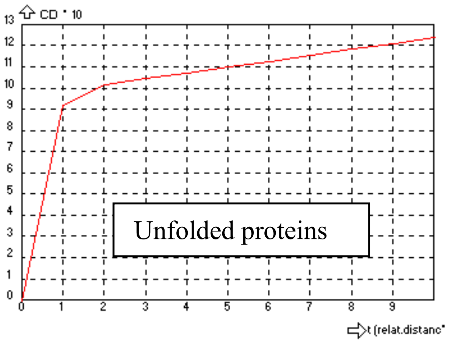
Component	Source term	Sink term
c_N native protein	0	KN*exp(-EN/(R*TEMP))
c_D denatured protein	KN*exp(-EN/(R*TEMP))*CN	CD*KD*exp(-ED/(R*TEMP))
c_A aggregated protein	CD**2*KD*exp(-ED/(R*TEMP))	0

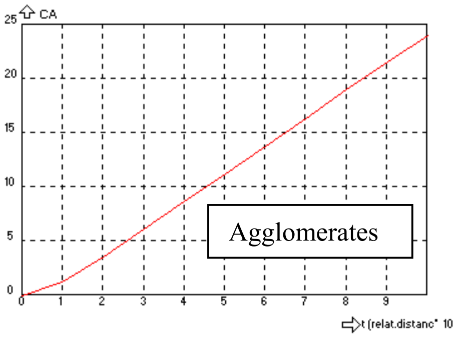
All these source and sink terms must be defined as user functions and their indices should be specified as EGROUP parameters. Let us show this approach on a simple case: straight pipe, laminar flow, ohmic heating (insulated tube)

```
C* Toyoda mode - fouling in a pipe
PT 1,.000E+00,.500E+00;
PT 2,.100E+01,.500E+00;
CR2PT 1,1,2;
EGROUP 1,PIPE2D,1,0,1,0,0,2,3,4,0,0;
REAL KN,EN,KD,ED
C* The following parameters (activation energies) are not realistics
KN=1
EN=300
KD=1
ED=200
EGROUP 1,PIPE2D,1,1,0,0,0,0,2,3,4,5,0,
RCONST 1,PIPE2D,.008,.1E-03,.0,.1E+01,.0,.0,.0,.0,.0,.0,.0,.0,.0,.0,.0,.0,.0,
FUNDEF 2,KN*EXP(-EN/(TEMP+273));
FUNDEF 3,KN*EXP(-EN/(TEMP+273))*CN;
FUNDEF 4,CD*KD*EXP(-ED/(TEMP+273));
FUNDEF 5,CD**2*KD*EXP(-ED/(TEMP+273));
FUNDEF 6,A*CA;
FUNDEF 7,1;
MCR 1,10,1,2,0,0,1;
NF 1,PRES,1,.100E-02;
NF 11,PRES,-1,0;
INITIA CN,7;
INITIA CD,0;
INITIA CA,0;
NF 1,CA,-1,0;
NF 1,CD,-1,0;
NF 1,CN,-1,1;
NF 1,TEMP,-1,20;
A=0.0001
OPTION 0,0,0,1,1,6,0,15,.1E-11,1,.1E+08,1,.1E+01,1,0,0,0,.1E-04,.1E-
04,0,5,1,2,2,2,0,
SOLVE 0,5,100,0,0,5,1,2,2,2,0;
```

The following graphs show results – concentrations of native, denaturated and agglomerated proteins along the pipe.



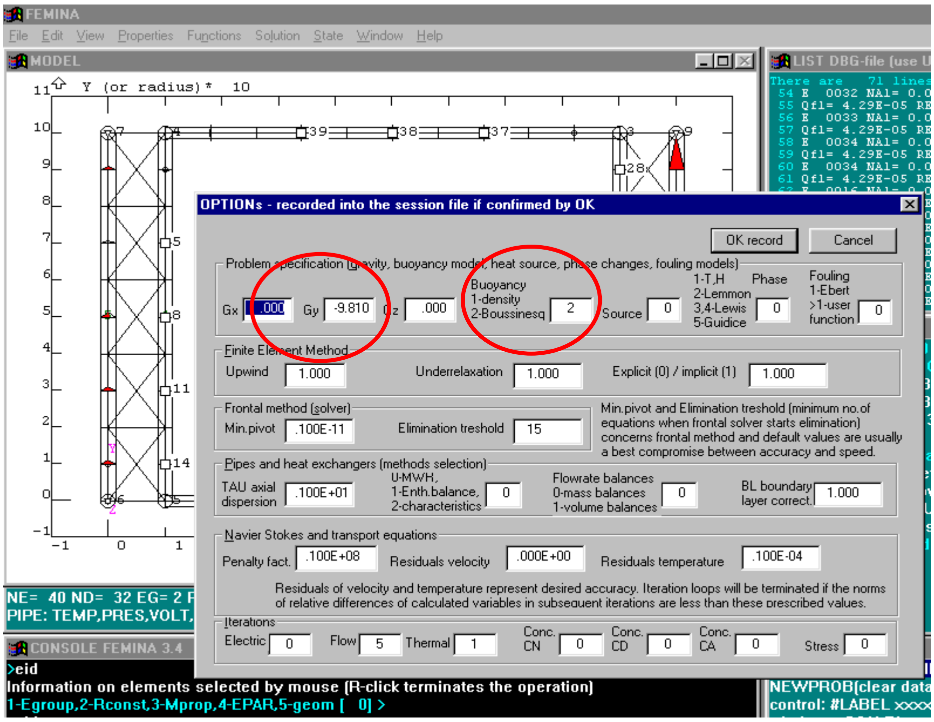




Graphical representation of calculated thermal resistance of the fouled layer can be obtained by using GE1 command.

2.2.7.9. Buoyancy and heat transfer between pipes

In the case that the flow is induced by buoyancy, it is necessary to specify the gravity acceleration (g_y=-9.81 m/s^2) and temperature dependent density of liquid– there are two possibilities how to do it: The first possibility is to prescribe density as a function of temperature (e.g. for water within the temperature range 20-80 ^{0}C correct the reference density ρ_0=1000 kg/m^3 by the function FUNDEF 1,1-0.000121*TEMP-3.2e-6*TEMP**2 where temperature TEMP is in degree of Celsius). The second possibility is to set a constant density (for example ρ_0=1000 kg/m^3), specify nonzero thermal expansion coefficient in MPROP (for example β=0.0005 [1/^{0}C]) and select Boussinesque method in the menu OPTION.



```
There are    71 lines
 54 E   0032 NA1= 0.0
 55 Qfl= 4.29E-05 RE
 56 E   0033 NA1= 0.0
 57 Qfl= 4.29E-05 RE
 58 E   0034 NA1= 0.0
 59 Qfl= 4.29E-05 RE
 60 E   0034 NA1= 0.0
 61 Qfl= 4.29E-05 RE
```

OPTIONs - recorded into the session file if confirmed by OK

Problem specification (gravity, buoyancy model, heat source, phase changes, fouling models)
Gx .000 Gy -9.810 Gz .000 Buoyancy 1-density 2-Boussinesq 2 Source 0 Phase 1-T,H 2-Lemmon 3,4-Lewis 5-Guidice 0 Fouling 1-Ebert >1-user function 0

Finite Element Method
Upwind 1.000 Underrelaxation 1.000 Explicit (0) / implicit (1) 1.000

Frontal method (solver)
Min.pivot .100E-11 Elimination treshold 15
Min.pivot and Elimination treshold (minimum no.of equations when frontal solver starts elimination) concerns frontal method and default values are usually a best compromise between accuracy and speed.

Pipes and heat exchangers (methods selection)
TAU axial dispersion .100E+01 U-MWH, 1-Enth.balance, 2-characteristics 0 Flowrate balances 0-mass balances 1-volume balances 0 BL boundary layer correct. 1.000

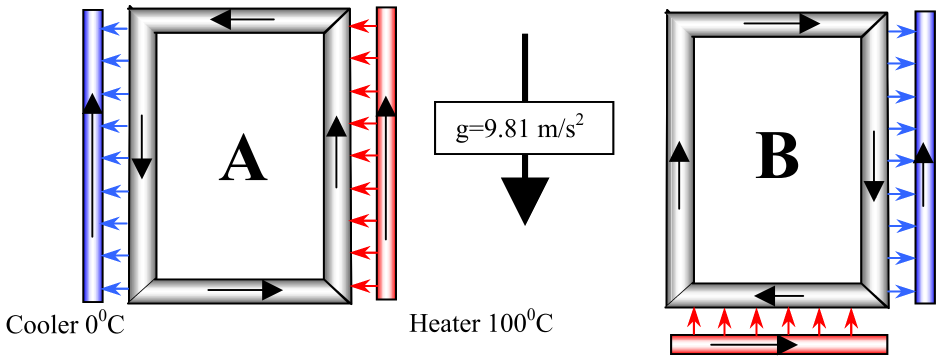
Navier Stokes and transport equations
Penalty fact. .100E+08 Residuals velocity .000E+00 Residuals temperature .100E-04
Residuals of velocity and temperature represent desired accuracy. Iteration loops will be terminated if the norms of relative differences of calculated variables in subsequent iterations are less than these prescribed values.

Iterations
Electric 0 Flow 5 Thermal 1 Conc. CN 0 Conc. CD 0 Conc. CA 0 Stress 0

```
NE=  40 ND=  32 EG= 2 F
PIPE: TEMP,PRES,VOLT,
CONSOLE FEMINA 3.4
>eid
Information on elements selected by mouse (R-click terminates the operation)
1-Egroup,2-Rconst,3-Mprop,4-EPAR,5-geom [   0] >
```

```
NEWPROB(clear data
control: #LABEL xxxx
```

The problems with buoyancy driven flows are usually rather difficult and results must be interpreted carefully. It will be demonstrated in the following example: Buoyancy driven circulation of water in a closed loop with some pipes heated and other cooled (similar arrangement is for example a system for domestic heating).



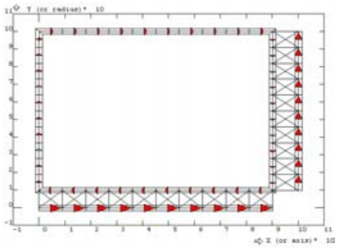
Mutual heat transfer between two PIPE2D elements can be realised by 4-node HEXC finite element (this connection represents the simplest form of a tubular or plate heat exchangers in parallel or counter-parallel flow arrangement). The arrangement when one vertical pipe is heated and the other vertical section is cooled (configuration A) seems to be similar to the arrangement when the loop is heated from below (B). This is not so, the arrangement shown left is nearly impossible to solve, because in this case the driving force depends only upon small differences in temperature profiles in vertical sections. It should be realised that the heat, transferred in both vertical sections, is the same, the flowrates are constant, and therefore the mean calorimetric temperature differences (temperature approaches) are identical. It is true that the mean temperature is not exactly the same as the mean calorimetric temperature, however the difference is small and therefore the mean densities are nearly identical in vertical channels and driving force – buoyancy is very small. It reveals as a strong sensitivity of flowrates upon model parameter (for example coefficient of axial dispersion). The model A is described by the following batch file

A

```
GY=-9.81
BETA(1)=0.00012
BUOY=2
BL=0
PT 1,.100E+00,.000E+00;
PT 2,.900E+00,.000E+00;
CR2PT 1,1,2;
PT 3,.900E+00,.100E+01;
CR2PT 2,2,3;
PT 4,.100E+00,.100E+01;
CR2PT 3,3,4;
PT 5,.100E+00,.000E+00;
CR2PT 4,4,5;
PT 6,.000E+00,.000E+00;
PT 7,.000E+00,.100E+01;
CR2PT 5,6,7;
PT 8,.100E+01,.000E+00;
PT 9,.100E+01,.100E+01;
CR2PT 6,8,9;
MCR2 5,4,5,1,1,1,1,2,1,2;
MCR2 2,6,5,1,1,1,1,2,1,2;
MCR 1,5,1,2,1,3,2;
NMERGE ;
NFPT 1,PRES,-1,0;
NFPT 6,PRES,-1,10;
NFPT 7,PRES,-1,0;
NFPT 8,PRES,-1,10;
NFPT 9,PRES,-1,0;
NFPT 6,TEMP,-1,0;
NFPT 8,TEMP,-1,100;
RCHEX 2,0.1,200;
EGROUP 2,HEXC,0,0;
SOLVE 0,50,10,1,0,35,1,0,0,0,0;
```

Older versions of FEMINA were not able to achieve a convergent solution of this problem. Since October 2004 additional procedure, improving mass conservation, is available and a convergent and at a first glance acceptable result can be obtained. However, when repeating the solution for different values of TAUD (axial dispersion), BL (boundary layer), heat transfer coefficient (see RCHEX), quite different results will be predicted.

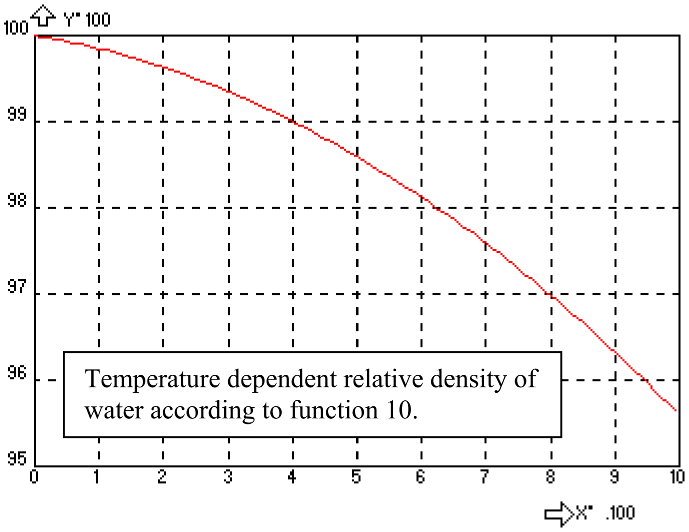
B Much more stable and realistic solution exists for the problem, where the loop is heated from bottom - in this case hot water is in the left ascending pipe, while the descending pipe is cooled – driving force is significantly greater than in the case A and the problem is well determined.



Its formulation is in the following batch file

```
PT 1,.000E+00,.100E+00;
PT 2,.900E+00,.100E+00;
CR2PT 1,1,2;
PT 3,.900E+00,.100E+01;
CR2PT 2,2,3;
PT 4,.000E+00,.100E+01;
CR2PT 3,3,4;
CR2PT 4,4,1;
PT 6,.000E+00,.000E+00;
PT 7,.900E+00,.000E+00;
CR2PT 6,6,7;
PT 8,.100E+01,.100E+00;
PT 9,.100E+01,.100E+01;
CR2PT 8,8,9;
EGROUP 1,PIPE2D;
RCONST 1,PIPE2D,.05,.100E-03,0,0.001;
FUNDEF 10,1-0.000121*TEMP-3.2E-6*TEMP**2
MCR 1,10,1,2,0,4,1;
MCR 6,10,1,2,0,0,1;
MCR 8,10,1,2,0,0,1;
EGROUP 2,HEXC,0,0;
RCONST 2,HEXC,2000,0.1;
MCR4 1,6,2,2;
MCR4 2,8,2,2;
NMERGE ;
NFPT 6,PRES,1,1;
NFPT 6,TEMP,-1,100;
NFPT 8,TEMP,-1,0;
NFPT 8,PRES,1,1;
NFPT 7,PRES,-1,0;
NFPT 9,PRES,-1,0;
BUOY=2
GY=-9.81
TAUD=0
BETA(1)=0.0005
SOLVE 0,10,1,1,0,20,1,0,0,0,0;
```

In this case we compare results obtained with two different methods of solution: The first (BUOY=1) makes use temperature dependent density defined by function number 10, and the second method is a Boussinesque approximation (BUOY=2) with thermal expansion coefficient β=0.0005 [1/K]. This value corresponds to the mean temperature of water 50 ^{0}C, which is mean temperature of water in the circulation loop (hot stream is 100 ^{0}C, cold stream 0 ^{0}C, and heat transfer coefficients equal in both heat exchangers).



Temperature dependent relative density of water according to function 10.

The most important result of calculation is mean velocity of water in the circulation loop, see the following table

Temperature dependence of density BUOY	TAUD [s] (dispersion)	u [m/s]
1 $\rho = 998(1 - 0.000121T - 3.2 \cdot 10^{-6} T^2)$	0	0.0574
1 $\rho = 998(1 - 0.000121T - 3.2 \cdot 10^{-6} T^2)$	1	0.0575
1 $\rho = 998(1 - 0.000121T - 3.2 \cdot 10^{-6} T^2)$	10^6	0.0602
2 β=0.0005	0	0.06
2 β=0.0005	1	0.061
2 β=0.0005	10^6	0.0634

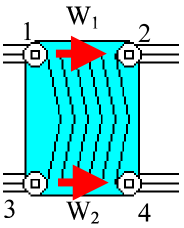
Discussion of results: Velocity (which is rather sensitive parameter) calculated by Boussinesque method agrees acceptably with the method defining temperature dependent density. What is surprising: velocity slightly increases with dispersion-according to common sense it should have been reversaly.

2.2.8. Heat exchangers (enthalpy balance method)

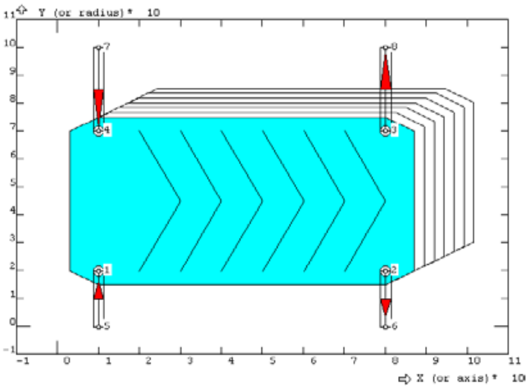
The heat exchange between parallel channels has been modelled as a classical counter-current heat exchanger and using a classical weighted residual method in the previous example. This method cannot be applied in the case that more complicated heat exchangers (e.g. plate, shell and tube or cross-flow heat exchangers) form a part of analysed pipeline network. Then the enthalpy balance method is to be used, selected by the parameter *Solution method* in the OPTION command or menu bar *Solution*. The enthalpy balance method has been implemented also for PIPE2D and CSTR elements, therefore it can be considered as a more general than the weighted residual method from the point of view of range of applications.

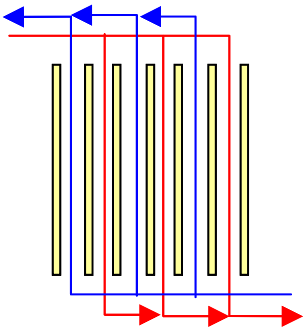
2.2.8.1. Chevron plate heat exchanger

In the following example the enthalpy balance method will be applied for modelling the chevron plate heat exchanger using elements HEXC with connected piping modelled by elements PIPE2D. The elements HEXC have 4 nodes, the nodes 1 and 2 are endpoints of the first stream while the nodes 3 and 4 belong to the second stream. The nodes arrangement should correspond to the schematic figure right, however orientation of flows is arbitrary (it is in fact result of calculations, depending upon pressures at nodes 1,2,3,4).



The heat exchanger considered in this example has 7 plates dividing two incoming streams into triplets of parallel streams, heat transfer surface of a stainless steel plate is 1 m^2, and equivalent hydraulic diameter d_e=0.005 m. Water is used as working fluid in both streams, with inlet temperatures 100^{0}C and 20^{0}C and pressure drops 1 kPa, and 2kPa in the first, and the second stream respectively.





```
PT 1,.100E+00,.200E+00;
PT 2,.800E+00,.200E+00;
PT 3,.800E+00,.700E+00;
PT 4,.100E+00,.700E+00;
PT 5,.1,0;
PT 6,.8,0;
PT 7,.8,1;
PT 8,.1,1;
CR2PT 1,1,5;
CR2PT 2,2,6;
CR2PT 3,4,5;
CR2PT 3,4,8;
CR2PT 4,3,7;
ND .100E+00,.200E+00;
ND .800E+00,.200E+00;
ND .800E+00,.700E+00;
ND .100E+00,.700E+00;
E 4,1,2,4,3;
EGROUP 1,HEXC,0,11;
RCHEV 1,1,1,1,3,0.001,0.005,45,15,1,0.5,1;
EGROUP 2,PIPE2D,0,0,0;
RCONST 2,PIPE2D,0.02,0,0,0,0,0,0,0,0,0;
MCR 1,1,1,2,2,4,1;
NMERGE ;
NF 5,PRES,-1,1000;
NF 6,PRES,-1,0;
NF 7,PRES,-1,2000;
NF 8,PRES,-1,0;
NF 5,TEMP,-1,100;
NF 7,TEMP,-1,20;
NF 6,TEMP,21,0;
NF 8,TEMP,21,0;
```

ND-new node (index is automatically incremented)
E-new element with 4 nodes 1,2,3,4. Associated EGROUP=RCONST=MPROP=1

Element HEXC can be any kind of heat exchangers. The following parameter of EGROUP command *Method*=11 selects chevron type plate heat exchanger and RCHEV defines RCONST parameters for this specific type (3 parallel streams, plate thickness 0.001m, plate distance 0.005m, chevron angle 45^0, thermal conductivity of plate 15 W.m^{-1}.K^{-1}, heat transfer surface of a plate 1 m^{2,}, dimensions of plate (width and height)

Four pipes are connected to the inlet and outlet nozzles of heat exchanger.

Pressures are specified at inlet and outlet, however temperatures at inlets only (100^{0}C and 20^{0}C). Nothing would need to be prescribed at outlet if the weighted residual method is used, because then natural boundary conditions apply. However, when using the enthalpy balance method the outlet points have to be explicitly marked by assigning status of temperature higher than 20 and zero thermal load!

```
HEPI=1
SOLVE 0,1,1,0,0,6,1,0,0;
```

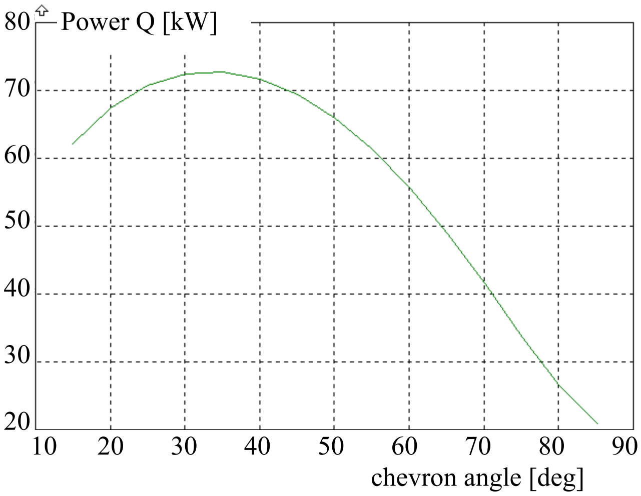
HEPI=1 selects enthalpy balances method (default value 0 is weighted residuals and HEPI=2 would activate method of characteristics).

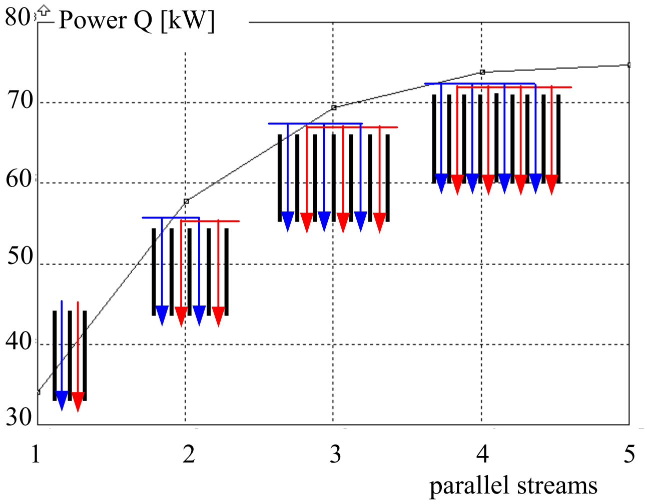
It is important from practical point of view to design an optimal heat exchanger – to find out for example optimum chevron angle, number of plates, or distance of plates, satisfying prescribed heat exchanger duty and minimising pumping power. This is subject of optimisation which will be described later (methods OPTIMA and SOMA). However, it is usually sufficient to observe just only influence of a single key-parameter, for example chevron angle, upon heat exchanger performance. This is demonstrated in the following example, which generates a table (MOP – matrix of observation points) with four columns: chevron angle (1st column), corresponding heating power-duty (2nd column), temperature increase (3rd column) and pressure drop in the 1st stream (4th column). The commands are continuation of the previous case – it means that all design parameters are preserved, with the exception of chevron angle, which is changing from 15 to 85 degrees. Because repetition is ensured by the #LOOP command, it is necessary to process the sequence of commands as a batch (it is necessary to prepare the commands in a file and use F)

```
J=0
A=10
NT=15
#LOOP L,NT
J=J+1
A=A+5
RCHEV 1,1,1,1,3,0.001,0.005,A,15,1,0.5,1;
SOLVE 0,1,1,0,0,10,1;
C1(J)=A
C2(J)=E1(IEP(EPOWER,1))
C3(J)=V2(LPU(TEMP)+3)-V2(LPU(TEMP)+4)
C4(J)=V2(LPU(PRES)+3)-V2(LPU(PRES)+4)
#LABEL L
```

A new feature in this example is the element parameters processing. All the parameters of all finite elements are in the array E1(*ind*) and the index *ind* is calculated by the system function IEP(*name,index of element*), in our case the name is EPOWER (heat exchanger duty) and the heat exchanger is element number 1.

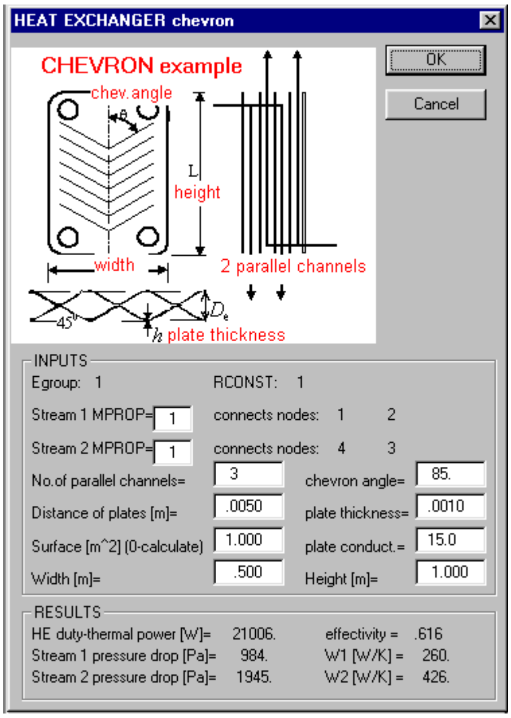
Results located in the matrix of observation points can be listed by using the command TCL, exported to an ASCII file by using command WRITE or shown in form of graphs using GTC, see for example the following figure. The figure left shows relationship between heat exchanger duty and chevron angle at a constant pressure drop.The figure right was created by a modified batch for constant chevron angle 45^0, but for different number of plates (parallel streams):





Results of solution can be inspected also by using mouse and the EID (Element IDentification) command. The EID suggests several options and for example the option 6 enables not only inspection of results but also modification of some parameters, e.g. chevron angle, dimension of plates, see the following dialog panel which appears when the element is picked by mouse (left click as usually). The most important item in the field RESULTS is the heat exchanger duty [W], besides calculated pressure drops and heat capacities of both streams W1,W2 [W/K]. Effectiveness of heat exchanger is the ratio of temperature range in the weak stream (stream having lower heat capacity) and the difference of inlet temperatures.

Remark: Similar information are accesible also for other kinds of heat exchangers, tubular and shell & tube heat exchangers. However, the actions performed in dialog panels are not recorded in the session file!



INPUTS

Egroup: 1 RCONST: 1

Stream 1 MPROP= 1 connects nodes: 1 2

Stream 2 MPROP= 1 connects nodes: 4 3

No.of parallel channels= 3 chevron angle= 85.

Distance of plates [m]= .0050 plate thickness= .0010

Surface [m^2] (0-calculate) 1.000 plate conduct.= 15.0

Width [m]= .500 Height [m]= 1.000

RESULTS

HE duty-thermal power [W]= 21006. effectivity = .616

Stream 1 pressure drop [Pa]= 984. W1 [W/K] = 260.

Stream 2 pressure drop [Pa]= 1945. W2 [W/K] = 426.

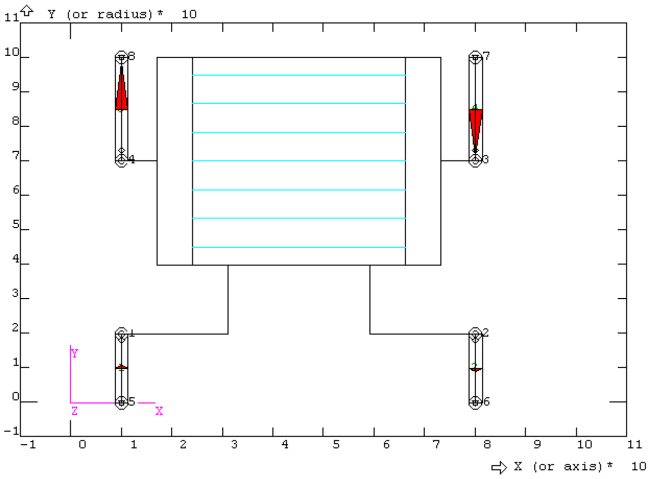
2.2.8.3. Shell and tube heat exchanger

Shell&tube heat exchangers are defined in a similar way as plate heat exchangers. The only difference is the fact that streams 1 and 2 are not exchangable: The first stream connecting nodes 1 and 2 is always shell stream, while the second stream connecting nodes 3 and 4 is the tube stream.

The shell and tube heat exchanger considered in this example has 1 pass in the tube section, 600 tubes, and 5 baffles. Tube outer diameter is 0.05 m, thickness of wall is 0.001 m, tube length is 5 m, tube bundle diameter is 1.9 m, shell inner diameter is 2 m, baffle distance is 1 m, baffle thickness is 0.0025 m, baffle height is 1.75 m, clearance between tube and baffle is 0.001 m, clearance between shell and baffle is 0.002 m, layout angle of tubes is 45°, tube spacing is 0.056, wall roughness is 0.001 m and heat conductivity of tube material (carbon steel) is 50 W.m^{-1}.K^{-1}.

Heat exchanger is designed for oil cooling by water. The first stream is oil with inlet temperature 90^{0}C, and the cooling water enters the exchanger at 5^{0}C. Instead of volumetric flowrate in both streams (which is a more common case) the pressure drop is specified by fixing pressures at inlet and outlet nozzles .

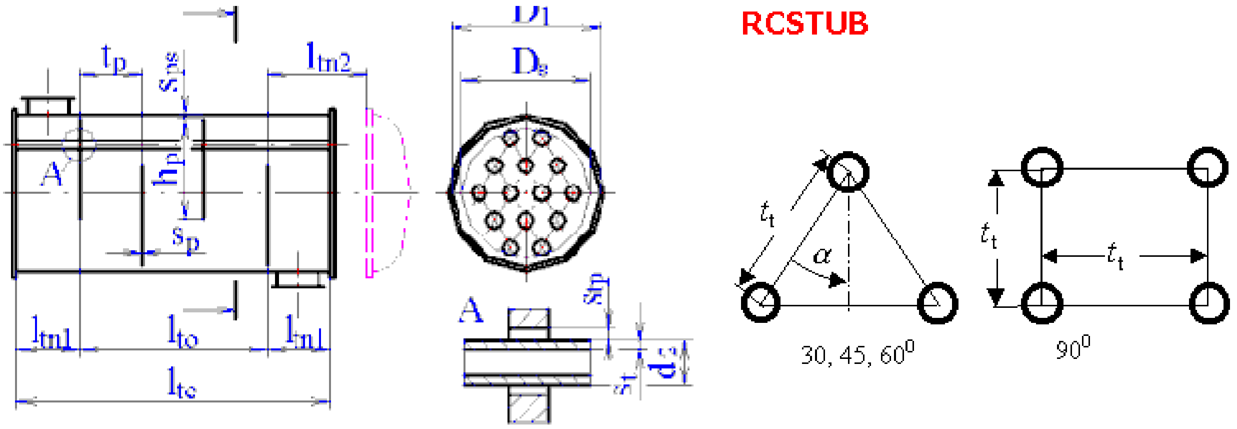
```
PT 1,.100E+00,.200E+00;
PT 2,.800E+00,.200E+00;
PT 3,.800E+00,.700E+00;
PT 4,.100E+00,.700E+00;
ND .100E+00,.200E+00;
ND .800E+00,.200E+00;
ND .800E+00,.700E+00;
ND .100E+00,.700E+00;
PT 5,.1,0;
PT 6,.8,0;
PT 7,.8,1;
PT 8,.1,1;
CR2PT 1,1,5;
CR2PT 2,2,6;
CR2PT 3,4,8;
CR2PT 4,3,7;
E 4,1,2,4,3;
EGROUP 1,HEXC,0,12;
```



The parameter *Method*=12 selects shell and tube heat exchanger

```
RCSTUB
1,1,1,2,1,600,5,0.05,0.001,5,1.9,2,1,0.0025,1.75,0.001,0.002,45,0.055678,0.001,50;
```

RCSTUB defines RCONST parameters for this specific type (1 passes in tube section, 600 tubes, 5 baffles, tube outer diameter 0.05 m, tube thickness 0.001 m, tube length 5 m, tube bundle diameter 1.9 m, shell inner diameter 2 m, baffle distance 1 m, baffle thickness 0.0025 m, baffle height 1.75 m, clearance between tube and baffle 0.001 m, clearance between shell and baffle 0.002 m, layout angle of tubes 45°, tube spacing 0.056, wall roughness 0.001 m and heat conductivity of tube 50 W.m^{-1}.K^{-1}).



```
EGROUP 2,PIPE2D,0,0,0;
RCONST 2,PIPE2D,0.02,0,0,0,0,0,0,0,0,0;
```

Four pipes are connected to the inlet and outlet nozzles of heat exchanger.

```
MCR 1,1,1,2,2,4,1;
MPROP 1,PIPE2D,.598E+00,0,4180,0,998,0,.400E-01,0,.210E+12,0,.280E+00,0,.100E-
02,0,.500E-03,0,.100E-08;
MPROP 2,PIPE2D,.4E+00,0,1890,0,866,0,.400E-01,0,.210E+12,0,.280E+00,0,.4E-
01,0,.500E-03,0,.100E-08;
ERMEL 4,2,2,2;
ERMEL 5,2,2,2;
NMERGE ;
NF 5,PRES,1,0.01;
NF 6,PRES,-1,0;
NF 8,PRES,1,0.1;
NF 7,PRES,-1,0;
NF 5,TEMP,-1,5;
NF 8,TEMP,-1,90;
NF 6,TEMP,21,0;
NF 7,TEMP,21,0;
HEPI=1
OPTION 0,0,0,0,1,0,1,15,.1E-11,1,.1E+08,1,.1E+01,1,0,0,1,.1E-04,.1E-04,
0,10,10,0,0,0,0,
SOLVE 0,1,1,0,0,10,10,0,0,0,0;
```

Pressures are specified at inlet and outlet, however temperatures at inlets only (90^{0}C and 5^{0}C). Nothing would need to be prescribed at outlet if the weighted residual method is used, because then natural boundary conditions apply. However, when using the enthalpy balance method the outlet points have to be explicitly marked by assigning status of temperature higher than 20 and zero thermal load!

Purpose of rating is to calculate outlet temperatures in both streams and heat exchanger duty. These values can be best observed by using EID (picking element by mouse, use the option 6 if you want to modify design parameters interactively).

Remark: There are several restrictions which must be respected: Only one pass in the shell and maximum four passes in tube section can be applied. Only the most frequently “E” shape of the heat exchanger shell (according to the TEMA classification) can be selected and only single-phase flows in heat exchanger can be considered.

2.2.9. Pumps

When using elements PIPE2D it is possible to define your own hydraulic characteristics of pipe $\varphi = \dot{V}/\Delta p$ as a function (index of the function is the parameter *hydraulic characteristics* of EGROUP); this is useful for example when modelling valves with a prescribed time sequence of closing, and this option could be used in principle also for pumps. However practical experience with this procedure is not very good, first of all due to problems with convergence. Therefore a special element PUMP has been designed replacing hydraulic characteristics $\varphi = \dot{V}/\Delta p$ by the hydraulic characteristics of pump in the form $\dot{V}(\Delta p)$. This approach is suitable for modelling centrifugal pumps with “soft” characteristics, and also for positive displacement pumps (piston or screw pumps) with “hard” characteristics independent of Δp. These options are selected using parameter *Method* of EGROUP:

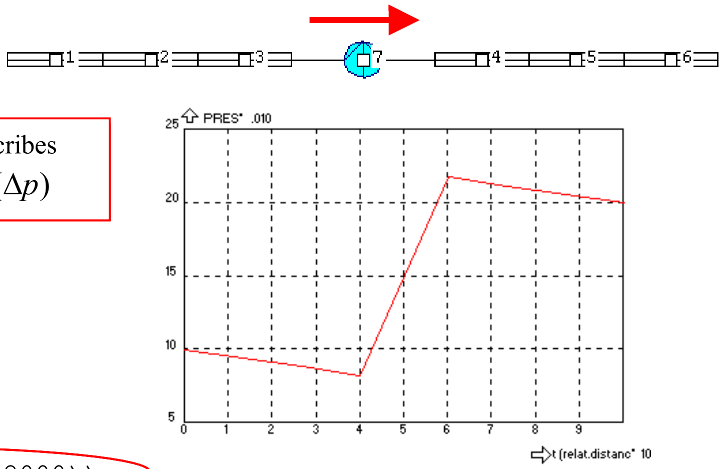
Method <=0 volumetric flow-rate is defined as a function (hydraulic characteristic). Example: Commands EGROUP 1,PUMP,1,-1,5; and FUNDEF 5,F define EGROUP number 1 for positive displacement pumps with flow-rate prescribed by the function number 5 (flow-rate specified by variable F is constant in this case, but could have been for example a function of time).[18]
In a similar way a centrifugal pump with user defined characteristics can be defined, when volumetric flow-rate is a function of discharge pressure.

This approach will be demonstrated in the following example with two straight pipes (suction and discharge pipes) and a pump having linear hydraulic characteristics

$$\dot{V} = 2\cdot 10^{-7}(1-\min(0.9,\frac{|\Delta p|}{2000}))$$

```
C*
PT 1,.000E+00,.500E+00;
PT 2,.400E+00,.500E+00;
PT 3,.600E+00,.500E+00;
PT 4,.100E+01,.500E+00;
CR2PT 1,1,2;
CR2PT 2,3,4;
CR2PT 3,2,3;
MCR 1,3,1,2;
MCR 2,3,1,2;
EGROUP 2,PUMP,1,0,1,0;
MCR 3,1,1,2;
NMERGE ;
RCONST 1,PIPE2D,.005;
NFPT 1,PRES,-1,1000;
NFPT 4,PRES,-1,2000;
VISC(1)=1
FUNDEF 1,.2E-5*(1-MIN(.9,ABS(DP)/2000));
SOLVE 4,3,1,0,0,2,0,0,0;
```

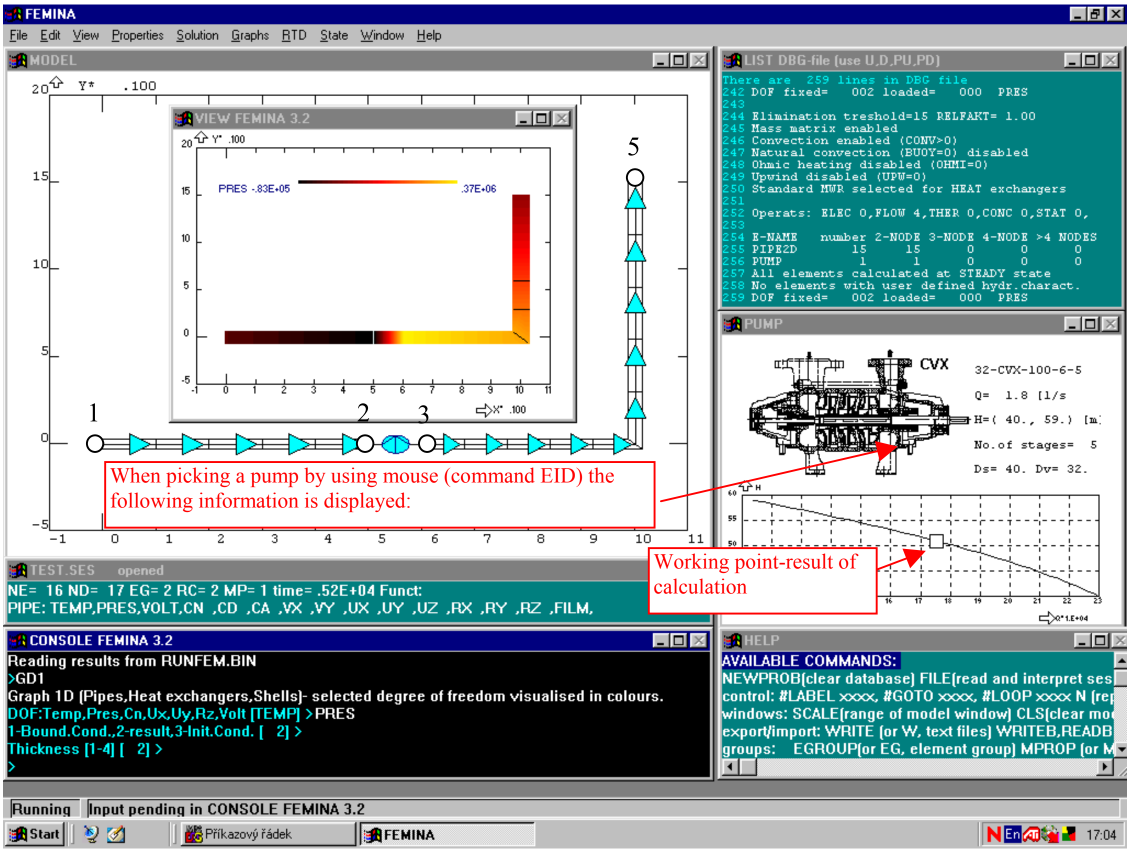
Function number 1 describes charakteristics $\dot{V} = f(\Delta p)$



Method > 0 centrifugal pump selected from a database (file $PUMPS.TXT), where basic characteristics of several tenths of commercial pumps (most of them produced by SIGMA GROUP a.s.) are recorded: pump type, range of applications, bitmap image, performance characteristics, and nozzle dimensions. The parameter *Method* is in fact the order of selected pump in the database. In the case that also the parameter *Hydraulic characteristic* in EGROUP is not zero, it is an index of function and the hydraulic characteristic of pump from database is multiplied by this function (this correction can be a function of time, temperature or Δp).

[18] Volumetric flowrate [m^3/s] and not mass flowrate [kg/s] is to be specified for pumps, which corresponds to custom practice, however flowrates or leakage specified in nodal points in previous chapters are mass flowrates!!

To simplify the pump selection and because the meaning of RCONST parameters differs from other kind of elements, the commands RCPUMP or RCEPUM is to be used instead of the command RCONST. In these commands general specifications of pump are prescribed and the best matching pump is selected. Procedure will be explained in the following example



LIST DBG-file (use U,D,PU,PD)

```
There are  259 lines in DBG file
242 DOF fixed=   002 loaded=   000  PRES
243
244 Elimination treshold=15 RELFAKT= 1.00
245 Mass matrix enabled
246 Convection enabled (CONV>0)
247 Natural convection (BUOY=0) disabled
248 Ohmic heating disabled (OHMI=0)
249 Upwind disabled (UPW=0)
250 Standard MWR selected for HEAT exchangers
251
252 Operats: ELEC 0,FLOW 4,THER 0,CONC 0,STAT 0,
253
254 E-NAME   number 2-NODE 3-NODE 4-NODE >4 NODES
255 PIPE2D       15     15      0      0      0
256 PUMP          1      1      0      0      0
257 All elements calculated at STEADY state
258 No elements with user defined hydr.charact.
259 DOF fixed=   002 loaded=   000  PRES
```

```
32-CVX-100-6-5
Q=  1.8 [l/s
H=( 40., 59.) [m
No.of stages=  5
Ds= 40. Dv= 32.
```

When picking a pump by using mouse (command EID) the following information is displayed:

Working point-result of calculation

TEST.SES opened

```
NE=  16 ND=  17 EG= 2 RC= 2 MP= 1 time= .52E+04 Funct:
PIPE: TEMP,PRES,VOLT,CN  ,CD  ,CA  ,VX  ,VY  ,UX  ,UY  ,UZ  ,RX  ,RY  ,RZ  ,FILM,
```

CONSOLE FEMINA 3.2

```
Reading results from RUNFEM.BIN
>GD1
Graph 1D (Pipes,Heat exchangers,Shells)- selected degree of freedom visualised in colours.
DOF:Temp,Pres,Cn,Ux,Uy,Rz,Volt [TEMP] >PRES
1-Bound.Cond.,2-result,3-Init.Cond. [   2] >
Thickness [1-4] [   2] >
>
```

HELP

```
AVAILABLE COMMANDS:
NEWPROB(clear database) FILE(read and interpret ses
control: #LABEL xxxx, #GOTO xxxx, #LOOP xxxx N (rep
windows: SCALE(range of model window) CLS(clear mod
export/import: WRITE (or W, text files) WRITEB,READB
groups:   EGROUP(or EG, element group) MPROP (or M
```

Running | Input pending in CONSOLE FEMINA 3.2

```
SCALE -.500E-02,100,-.500E-02,200;
PT 1,.000E+00,.000E+00;
PT 2,.500E+02,.000E+00;
PT 3,.600E+02,.000E+00;
PT 4,.100E+03,.000E+00;
PT 5,.100E+03,.150E+03;
CR2PT 1,1,2;
CR2PT 2,3,4;
CR2PT 3,4,5;
CR2PT 4,2,3;
RCONST 1,PIPE2D,0.05;
MCR 1,5,1,2;
MCR 2,5,1,2;
MCR 3,5,1,2;
RCPUMP 2,2,0,0.05,10E-3,30;
MCR 4,1,1,2;
NMERGE ;
NFPT 1,PRES,-1,1000;
NFPT 5,PRES,-1,50000;
BUOY=1
GY=-9.81
SOLVE 0,5,1,0,0,5,0,0,0;
```

Geometry, see previous figure

Parameters in groups EGROUP=2 and RCONST=2 are defined by selection from database $PUMPS.TXT. Parameters 2,2 are indices of groups EGROUP and RCONST, the following parameters specify category of pumps (0-arbitrary), diameter of nozzles (0.05 m), estimated flowrate (10 l/s) and estimated displacement height (30 m).

One element, pump (MPROP=1,EGROUP=RCONST=2), is created on curve number 4.

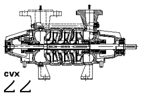
The simplest tool for results inspection is the command EID (Element IDentification by mouse) used with default values of parameters. First of all you should check calculated volumetric flow-rates, because it is the simplest test of convergence and accuracy (flow-rates should be the same in all elements in this example).

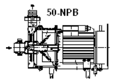
Be careful when defining a pump – solution works only if the required displacement height is within the range of selected pump. For example in the previous example the selected pump is capable do deliver maximum height of 50m of water column, however the vertical pipe is 150 m long! Therefore the solution doesn't converge and it is necessary to reduce the gravity coefficient GY, giving

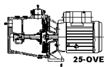
g [m/s^2]	Q [l/s]
0	3.381
-1	2.917
-2	2.231
-3	1.535
-4	0.399
-5	diverge

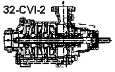
Another type of pumps can be easily added to the file $PUMPS.TXT:

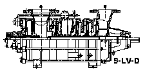
```
2 'che.bmp' '65-cve-200-10' 2 0.065 0.065 5E-3 9E-3 7E-3 22 28 27
…
2 'che.bmp' '80-cve-230-12' 10 0.08 0.08 9E-3 14.4E-3 12E-3 132 170 153
3 'chn.bmp' '100-CHN' 2 0.1 0.1 16E-3 25E-3 20E-3 130 155 150
…
3 'chn.bmp' '100-CHN' 10 0.1 0.1 16E-3 25E-3 20E-3 705 760 750
4 'cvx1.bmp' '32-CVX-100-6-1' 1 0.04 0.032 1.13E-3 2.3E-3 1.75E-3 8 12 10
…
4 'cvx1.bmp' '32-CVX-100-6-12' 12 0.04 0.032 1.13E-3 2.3E-3 1.75E-3 95 142 122
1 '50npb.bmp' '50-NPB-150-12.5' 1 0.05 0.05 2.7E-3 6.5E-3 5E-3 16 26 22.4
…
1 '50npb.bmp' '65-NPB-160-16' 1 0.065 0.065 5E-3 10E-3 8E-3 24 29.5 28
1 'spb_a.BMP' '50-SPB' 1 0.05 0.05 3.2E-3 5.4E-3 4E-3 11.9 29.3 24.2
4 '25ove.bmp' '25-ove-01' 1 0.025 0.025 0.1e-3 0.8e-3 0.5e-3 25 45 33
4 '32cvi2.bmp' '32-cvi' 1 0.032 0.032 1.13e-3 2.3e-3 1.75e-3 8 12 10
…
4 '32cvi2.bmp' '32-cvi' 5 0.032 0.032 1.13e-3 2.3e-3 1.75e-3 40 59 51
4 'slvd.bmp' 'S-LVN-4-D' 3 0.1 0.125 9.1E-3 16.5E-3 13.3E-3 33 45 39
…
4 'slvd.bmp' 'S-LVN-4-D' 7 0.1 0.125 9.1E-3 16.5E-3 13.3E-3 77 105 91
4 '80SVA.BMP' '80-SVA' 1 0.08 0.08 4E-3 13E-3 8E-3 10 40 29
…
4 '80SVA.BMP' '80-SVA' 3 0.08 0.08 4E-3 13E-3 8E-3 29 110 84
4 'HILGE.BMP' 'HILGE'  1 0.02 0.02 0.1E-3  5.55E-3  2.777E-3 12 53 38
```

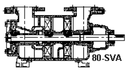


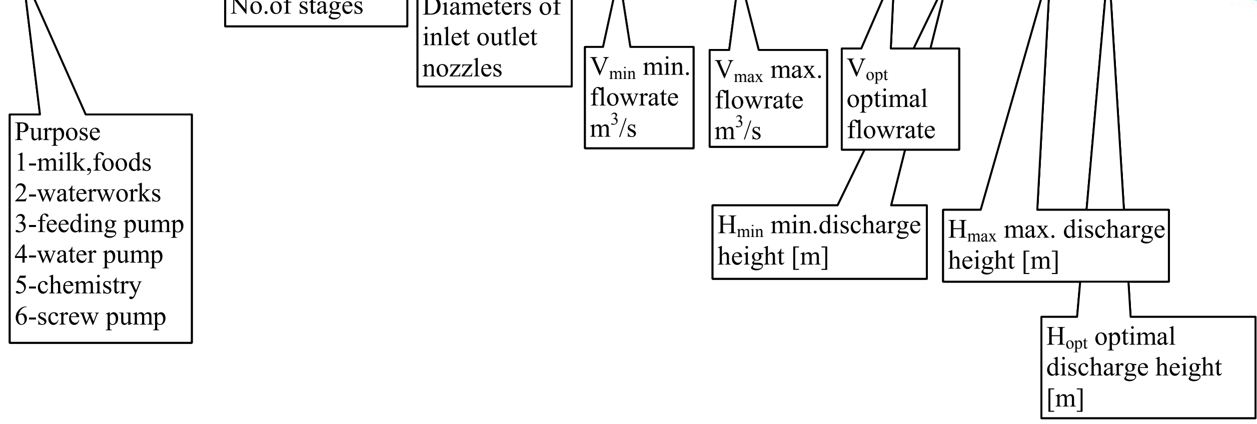




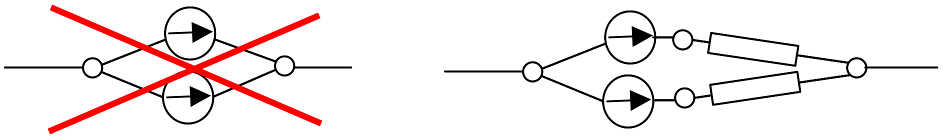






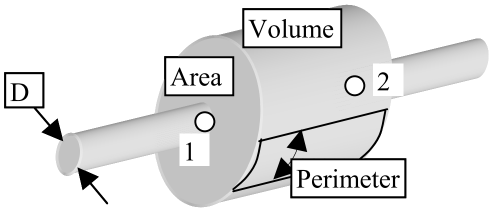

No.of stages
Diameters of inlet outlet nozzles
V_{min} min. flowrate m^3/s
V_{max} max. flowrate m^3/s
V_{opt} optimal flowrate
Purpose
1-milk,foods
2-waterworks
3-feeding pump
4-water pump
5-chemistry
6-screw pump
H_{min} min.discharge height [m]
H_{max} max. discharge height [m]
H_{opt} optimal discharge height [m]

So far the pumps has been used only for calculation of flowrates and pressures, however, elements PUMP are also implemented for heat and mass transfer in a similar way like ideally mixed tanks (see next paragraph). It means that a pump is assumed to be a finite element characterised by a finite volume and inside the volume the enthalpy (temperature) and concentration of species is constant (may be dependent on time, nevertheless the same inside the pump and therefore also at outlet). As a consequence of this assumption is that the nodal parameter (temperature or concentration) corresponding to the outlet node depends only upon the state of inlet node and upon the time accumulation of enthalpy of matter inside the pump. This nodal parameter characterises only the state of pump and is independent of streams connected to this node. Therefore it is not possible to join e.g. outlets of two pumps in the same node; what has to be done is to insert a pipe (maybe very short pipe) behind the outlets of pumps and even then the end points can be merged:



2.2.10. Ideally mixed tanks

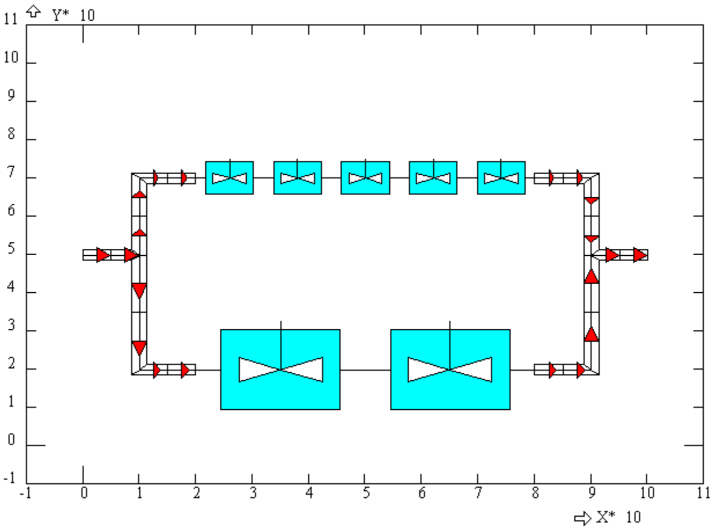
Mixed tanks can be modelled using two node element CSTR (Continuous Stirred Tank Reactor) characterised by a uniform temperature and concentration inside the vessel and these values correspond to the outlet node (which of the two nodes represents outlet is not known in advance of course).



It is assumed, that the vessel has a form of a big tube with cross section and inner volume given by the RCONST parameters (AREA and VOLUME). The cross section and flowrate determine mean axial velocity used for calculation of friction losses. However this part of pressure losses is usually small compared with the losses of kinetic energy at inlet nozzle, and this more significant part of pressure loss is calculated according to specified coefficient of local losses ζ (zeta) and diameter of nozzle *d* in the RCONST group (please do not confuse the meaning of *d* which is *not* the diameter of vessel). If the parameter AREA is not specified (=0) it will be calculated from the volume and the length of vessel, which equals the distance of nodes. The CSTR element can be used also for calculation of a chemical reaction, i.e. concentration of a component and also for heat balancing. In the later case the heat transfer coefficient and external temperature are specified in RCONST group, and heat transfer surface is calculated as the product of perimeter (in RCONST) and the length of vessel.

As an example of CSTR elements application we shall analyse a system with two parallel branches shown in the following figures. Geometry is selected in such a way that the role of CSTR dominates: the inner volume of all seven tanks together is 12 m^3, while the volume of pipes is nearly negligible, 0.00314 m^3. We shall calculate flowrates (given pressures at inlet and outlet) and also concentration response to a short injection of a tracer to the inlet.

Remark: Problems with convergence can be encountered if elements having very different flow resistance are mixed together, and for example if we neglect local pressure losses at inlets to vessels their resistance will be many orders lower than the resistance of pipes. Even thought we specified ζ=1 an underrelaxation was necessary in this case.



```
PT 1,.000E+00,.500E+00;
PT 2,.100E+00,.500E+00;
PT 3,.100E+00,.200E+00;
PT 4,.200E+00,.200E+00;
PT 5,.800E+00,.200E+00;
PT 6,.900E+00,.200E+00;
PT 7,.900E+00,.500E+00;
PT 8,.100E+01,.500E+00;
PT 9,.900E+00,.700E+00;
PT 10,.800E+00,.700E+00;
PT 11,.200E+00,.700E+00;
PT 12,.100E+00,.700E+00;
CR2PT 1,1,2;
CR2PT 2,2,3;
CR2PT 3,3,4;
CR2PT 4,5,6;
CR2PT 5,6,7;
CR2PT 6,7,8;
CR2PT 7,7,9;
CR2PT 8,2,12;
CR2PT 9,12,11;
CR2PT 10,10,9;
CR2PT 7,9,7;
CR2PT 11,4,5;
CR2PT 12,11,10;
EGROUP 1,PIPE2D,1,0,0;
RCONST 1,PIPE2D,.500E-01,.100E-03,0,1,0,0,0,0,0,0;
```

```
MCR 1,2,1,2,0,10,1;
EGROUP 2,CSTR,1,0,0;
RCONST 2,CSTR,0.05,1,0,0,1,0,0,0;
MCR 11,2,1,2,0,0,1;
RCONST 3,CSTR,0.05,1,0,0,2,0,0,0;
MCR 12,5,1,2,0,0,1;
NMERGE ;
NFPT 1,PRES,-1,5000;
NFPT 8,PRES,-1,0;
```

Two CSTR will be created on curve 11 with associated EGROUP=RCONST=2 (parameters RCONST define inner volume of each tank 1 m^3 , diameter of nozzles the same as the diameter of pipes 0.05m and ζ=1).
In a similar way five CSTR are created on curve number 12 (inner volume 2m^3 as defined in associated RCONST=3).
Pressure drop 5 kPa is specified using NFPT commands.

```
CURDEF -2,0,5,0,0,1,1,10,1,11,0,100,0;
NFPT 1,CN,-2,1;
INITIA CN,0;
```

Time course of tracer concentration at inlet is defined by table number –2 and prescribed as a strong boundary condition

```
RELFAKT=0.5
SOLVE 0,1,1,0,0,70,0,0,0;
INITIA CN,0;
RELFAKT=1
SOLVE 0,600,10,1,0,0,0,1,0;
```

Solving pressure distribution and flowrates with underrelaxation (RELFAKT=0.5) and 70 iterations in single time step (the value of time step can be arbitrary because Δt has no effect upon pressure calculation).

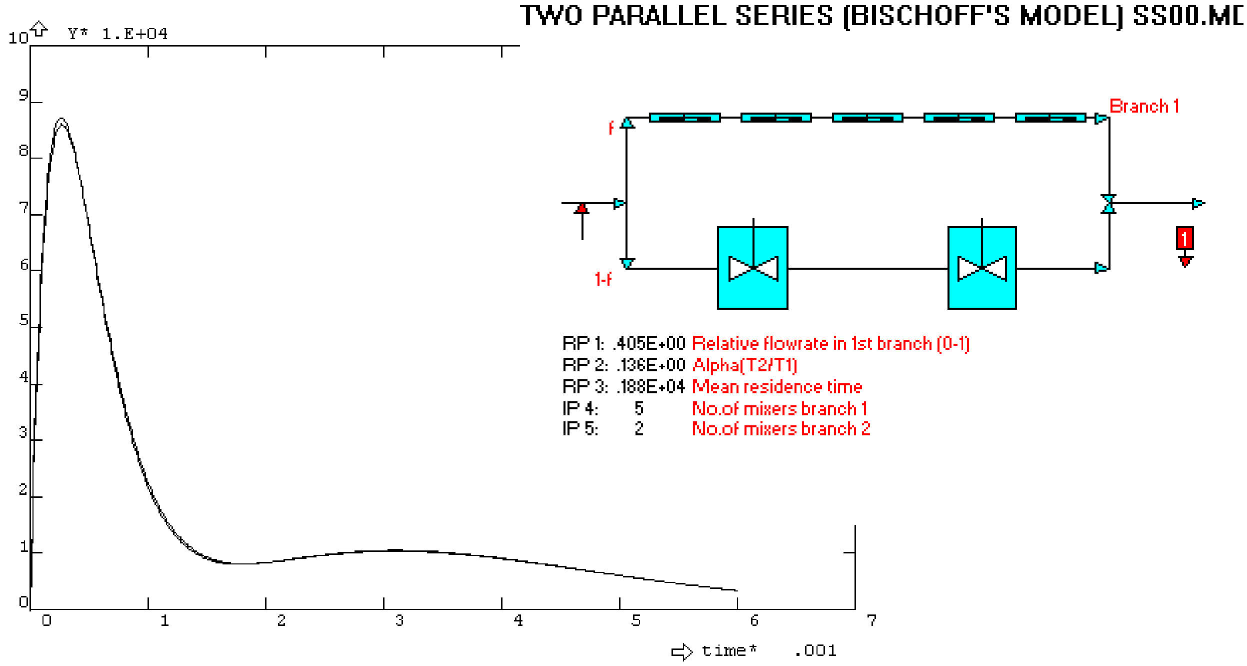
Because flowrates are constants and saved in the zone EPAR, it is sufficient to solve only mass transfer (600 time steps, Δt=10s). The following parameter 1 in the command SOLVE ensures, that the zone EPAR will not be erased.

Results can be inspected using e.g. EID (flowrates, pressures) and GTIME (response of concentration CN at outlet): Overall flowrate 6.371 l/s, relative flowrate in the branch with 5 tanks f=0.4046, mean residence time 1884 s and residence times 3879 s and 527.3 s in the first and in the second branch respectively. Because the response calculated using GTIME has been recorded into the second column of MOP (matrix of observation points), it is possible to normalise it to unit area

```
NORM 2,2,0;
```

and plot it using GTC command.

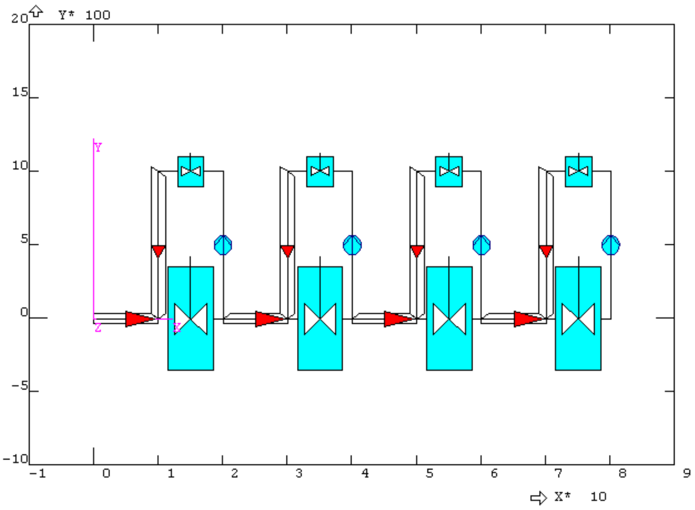
It will be shown in the following chapter, that nearly the same system can be modelled in a different way, as a system of ordinary differential equations (the only difference is that the pipes are infinitely small and the stimulus function need not be approximated by a pulse of a finite duration). Prediction of this model (the Bishoff's model of two parallel series SS00) can be compared with the prediction obtained by the finite element model and you can see that the match is excellent.

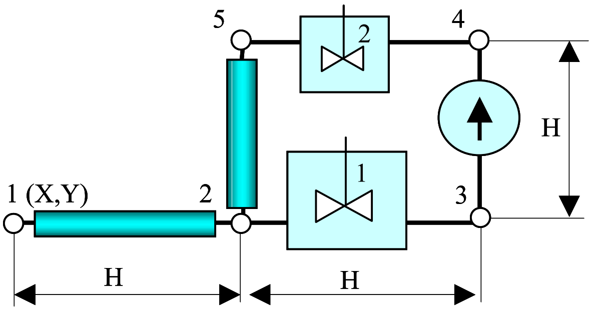


TWO PARALLEL SERIES [BISCHOFF'S MODEL] SS00.MC

```
RP 1: .405E+00 Relative flowrate in 1st branch (0-1)
RP 2: .136E+00 Alpha(T2/T1)
RP 3: .188E+04 Mean residence time
IP 4:     5    No.of mixers branch 1
IP 5:     2    No.of mixers branch 2
```

Combination of CSTR and PUMP elements enables modelling of systems with recycles, stagnant regions or backmixing between CSTR. In the following example we want to show, how useful are macros for definition of basic building blocks. Such a building block is for example mutually connected pair of CSTR with a low level of material exchange forming a basis for modelling stagnant regions. Lets us assume that we want to define a system consisting of four such a units connected in series, see figure





Each building block consists of two CSTR, one positive displacement PUMP (the pump determines intensity of material exchange) and two PIPE2D element with axial dispersion. The elements are localised in space as shown in the figure above (position of point 1 is X,Y, total length is 2H). Let us assume that properties of pipes are described by EGROUP=RCONST=1, properties of CSTR (e.g. volumes) by EGROUP=2 and RCONST=2, 3 for the first and the second CSTR respectively, and PUMP is characterised by EGROUP=3 and RCONST=4. Then the procedure, generating one building block at position X,Y and which automatically increments coordinate X, can be described by the following macro with the name e.g. PSTAS (the macro file can be created interactively)

```
RECORD PSTAS
K=NPT
PT NPT+1,X,Y;
PT NPT+1,X+H,Y;
PT NPT+1,X+2*H,Y;
PT NPT+1,X+2*H,Y+H;
PT NPT+1,X+H,Y+H;
J=NCR
C2P NCR+1,K+1,K+2
C2P NCR+1,K+2,K+3
C2P NCR+1,K+3,K+4
C2P NCR+1,K+4,K+5
C2P NCR+1,K+5,K+2
C* EG=RC=1 PIPES
ACTSET 1,1
ACTSET 3,1
MCR J+1,1,1,2;
MCR J+5,1,1,2;
C* EG=RC=2 CSTR1
ACTSET 1,2
ACTSET 3,2
MCR J+2,1,1,2;
ACTSET 3,3
MCR J+4,1,1,2;
C* EG=3 RC=4 PUMP
ACTSET 1,3
ACTSET 3,4
MCR J+3,1,1,2;
X=X+2*H
ENDREC
```

Using the macro PSTAS it is easy to describe systems with arbitrary number of stagnant region units, for example

```
RCONST 1,PIPE2D,.500E-01,.100E-03,0,1,0,0,0,0,0,0;
EGROUP 1,PIPE2D,1,0,0;
EGROUP 2,CSTR,1,0,0;
EGROUP 3,PUMP,1,-1,1,0;
RCONST 2,CSTR,0.05,1,0,0,1,0,0,0;
RCONST 3,CSTR,0.05,1,0,0,1,0,0,0;
RCONST 4,PUMP,0,0,0,0.01,0,0;
FUNDEF 1,F;
```

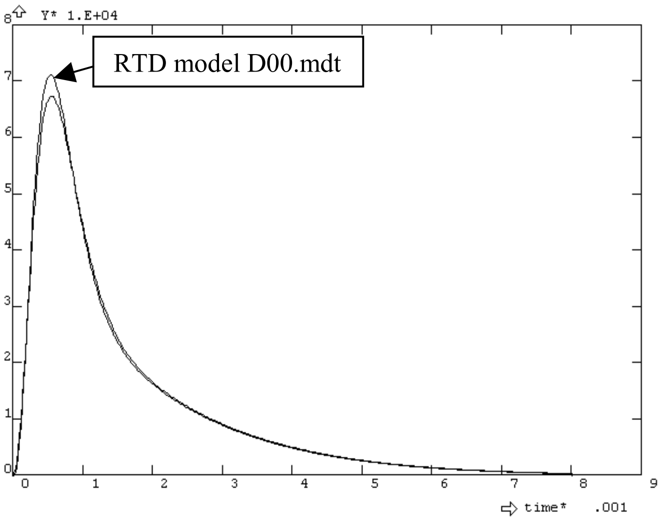
EGROUP for pump selects positive displacement pump with flow-rate defined by function number 1.
RCONST 2 and 3 define CSTR, both the same with diameter of nozzle 0.05 m and inner volume 1 m^3.

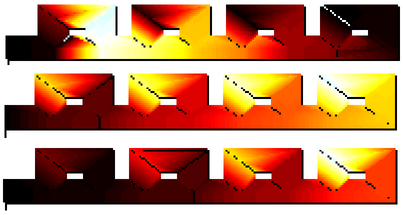
```
F=0.001
CURDEF -2,0,5,0,0,1,1,30,1,31,0,100,0;
X=0
Y=0
H=0.1
PSTAS
PSTAS
PSTAS
PSTAS
NMERGE ;
NFPT 1,PRES,2,5E-3;
NFPT 18,PRES,-1,0;
NFPT 1,CN,-2,1;
SOLVE 0,1,1,0,0,40,0,0,0;
INITIA CN,0;
SOLVE 0,400,20,1,0,0,0,1,0;
```

Stimulus function (tracer injection) defined by table

Just write PSTAS and enter – new building block will be immediately attached at the end of series. This action has to be done interactively, because you cannot call a macro from a batch file.

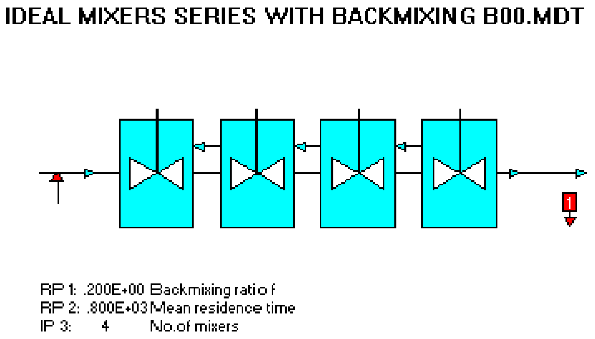
Calculated response can be displayed as a graph using GT command. During execution of this command you can observe evolution of the injected tracer distribution


RTD model D00.mdt

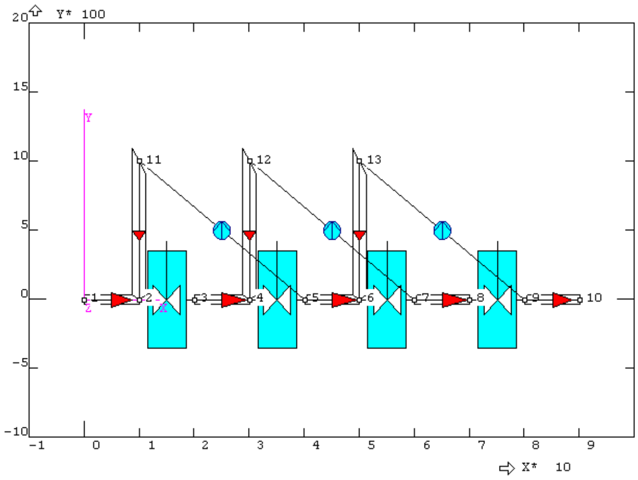


FEMINA automatically calculates inner volume of apparatuses as the sum of volumes of pipes, pumps and tanks, and using this overall volume and the flow-rate at inlet the theoretical mean residence time is calculated and displayed in the LIST window. You can compare this value with the first moment calculated from response (NORM or MOM commands). Even if the tail is very long (which is typical for systems with stagnant regions) the calculated mean residence time is accurate within 1% tolerance, however, comparison with the „exact“ solution using the RTD model D00.mdt reveals the effect of numerical dispersion (the response peek is lower than it should be).

Next example is similar and very frequent: series of ideally mixed vessels with backmixing,


IDEAL MIXERS SERIES WITH BACKMIXING B00.MDT

RP 1: .200E+00 Backmixing ratio f
RP 2: .800E+03 Mean residence time
IP 3: 4 No.of mixers

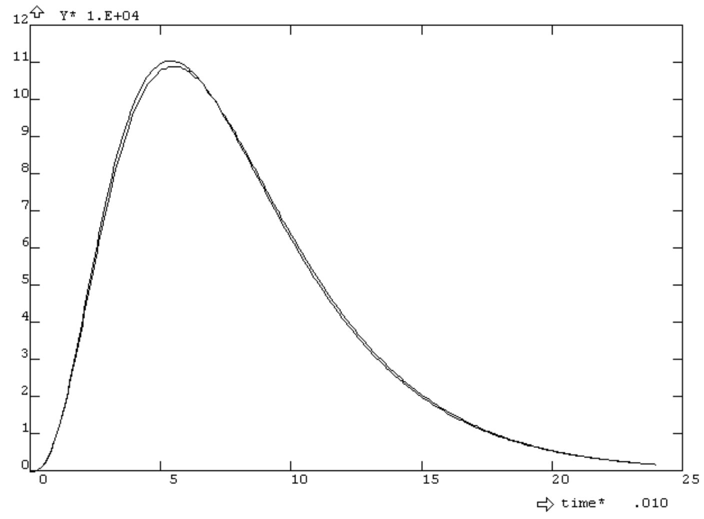


This model seems to be simple, however, when modelling the backmixing stream using a positive displacement pump it is necessary to include PIPE2D elements behind the pump and also behind each CSTR. Not doing this, pressure distribution and flowrates would be calculated correctly, however the impulse response would be quite wrong (even not decreasing to zero, but approaching a positive value). The following program shows also the way how convergence problems can be avoided: calculate flowrates first by adjusting zero backmixing *f* (this always works) and even then continue iterations with desired level of backmixing, or gradually increasing *f*.

```
H=0.1
N=4
G=0.005
V=1
F=0.001
C* backmixing using FEM
C* F-flowrate backmixing,
C* V-volume of 1 vessel
C* G-flowrate (inlet)
C* N-number of units (vessels)
T=N*V/G*3
L=300
W=T/L
INIFEM
EGROUP 3,PIPE2D,1,0,0;
RCONST 3,PIPE2D,0.01;
ND 0,0
ND H,0
E 2,1,2
EGROUP 2,PUMP,1,-1,1,0;
RCONST 2,PUMP,0,0,0,.001,.0001,0;
EGROUP 1,CSTR,1,0,0;
RCONST 1,CSTR,0.01,V,0,0,1,0,0,0;
X=H
#LOOP LAB,N
K=ND
ND X+H,0;
ACTSET 1,1
ACTSET 3,1
E 2,K,K+1
X=X+2*H
ND X,0
ACTSET 1,3
ACTSET 3,3
E 2,K+1,K+2
#LABEL LAB
```

```
X=H
K=1
#LOOP LABK,N-1
ND X,H
ACTSET 1,3
ACTSET 3,3
E 2,2*N+K+2,2*K
ACTSET 1,2
ACTSET 3,2
E 2,2*K+3,2*N+K+2
K=K+1
X=X+2*H
#LABEL LABK
FUNDEF 1,F
CURDEF -2,0,5,0,1,W,1,2*W,1,3*W,0,T,0
K=2*N+2
FND 1,PRES,2,G
FND K,PRES,-1,0
FND 1,CN,-2,1
C* start calculation with zero F
A=F
F=0
SOLVE 0,1,1,0,0,30,0,0,0
F=A
SOLVE 0,1,1,1,0,30,0,0,0
SOLVE 0,L,W,1,0,0,0,1,0
TSTEP L,W
J=0
#LOOP L1,L
J=J+1
C1(J)=W*(J-1)
LTN C1(J)
C2(J)=V3(LPU(CN)+K)
#LABEL L1
NORM 2,2,0
```

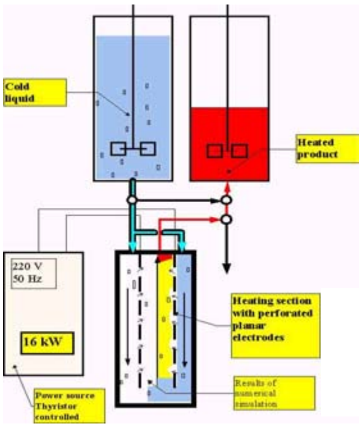
Calculated response of 4 units, f=0.2, mean residence time 800 s, and 300 time steps, shown in the figure right, is compared with the solution of system of ordinary differential equations (model B00.mdt, which will be discussed in the next chapter). As you see agreement is quite good.



2.2.11. Case study – direct ohmic heater

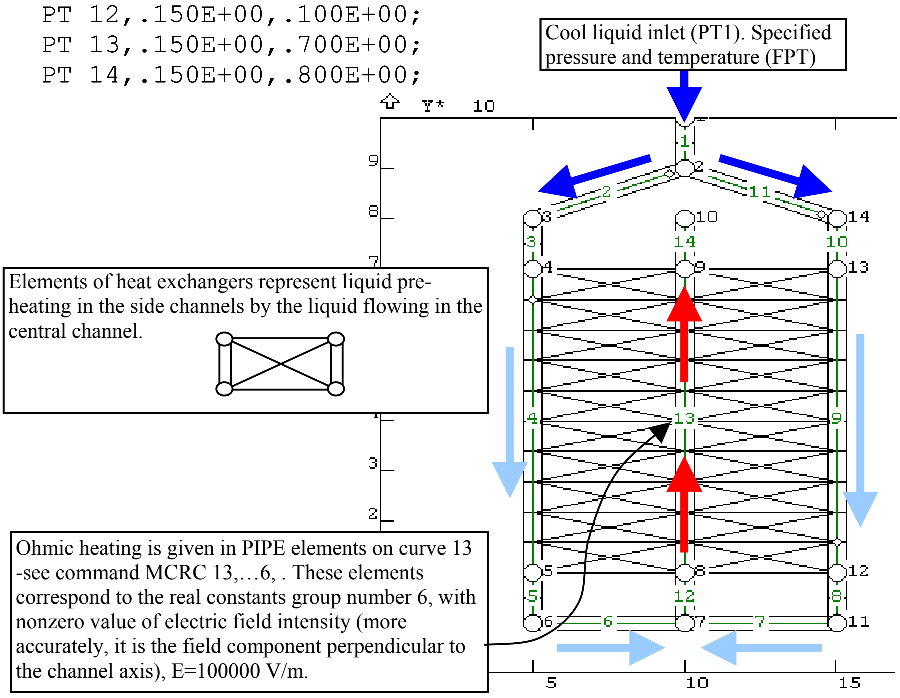
The one dimensional elements (for example PIPE2D) can be used also for modelling of a more complicated systems consisting not only from pipes. In this case the mathematical model is only an approximation of a real system with uncertainties in element characterisation, for example equivalent diameter of pipes, local pressure loss coefficients or even the lengths of pipes. These uncertainties can be reduced by experiments, comparing measured pressures or flow rates with the model prediction. It is also possible to make use of stimulus response experiments for identification of unknown model parameters, because FEMINA allows solution of thermal and concentration fields and therefore to calculate responses to a tracer injection. Both the energy and mass transport analysis is based upon the models of axial dispersion, where the dispersion coefficients are calculated according to the Taylor Aris theory in laminar flows and empirical correlations are used in turbulent flow.

This approach will be demonstrated upon analysis of a direct continuous electric heater (see figure), with the aim to simulate stimulus response experiment. The following model calculates steady state distribution of pressure, flowrates and temperature, when the liquid in the central channel is heated ohmically (it will be demonstrated how to model volumetric heat source and heat transfer between parallel channels). Transient distribution of tracer inside the heater and the response to a simulated trace injection will be calculated as well .


Cold liquid
Heated product
220 V
50 Hz
16 kW
Heating section with perforated planar electrodes
Power source Thyristor controlled
Results of numerical simulation

```
ANALYS 1
SCALE 0,.3,-.1,1;
PT 1,.100E+00,.100E+01;
PT 2,.100E+00,.900E+00;
PT 3,.500E-01,.800E+00;
PT 4,.500E-01,.700E+00;
PT 5,.500E-01,.100E+00;
PT 6,.500E-01,.000E+00;
PT 7,.100E+00,.000E+00;
PT 8,.100E+00,.100E+00;
PT 9,.100E+00,.700E+00;
PT 10,.100E+00,.800E+00;
PT 11,.150E+00,.000E+00;
PT 12,.150E+00,.100E+00;
PT 13,.150E+00,.700E+00;
PT 14,.150E+00,.800E+00;
```

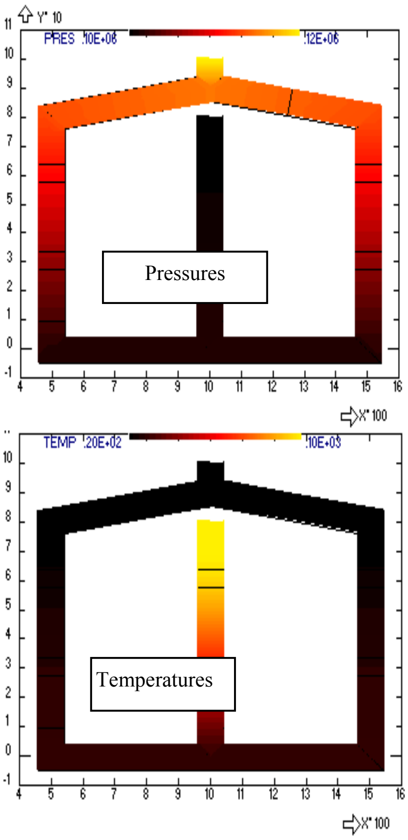
Important points
1. liquid inlet into the head of equipment
2. splitting into side channels
7. stream intermixing at the bottom
8-9. heating section (electric field acts accros the channel)
10. outlet channel



Cool liquid inlet (PT1). Specified pressure and temperature (FPT)

Elements of heat exchangers represent liquid pre-heating in the side channels by the liquid flowing in the central channel.

Ohmic heating is given in PIPE elements on curve 13 -see command MCRC 13,…6, . These elements correspond to the real constants group number 6, with nonzero value of electric field intensity (more accurately, it is the field component perpendicular to the channel axis), E=100000 V/m.



```
CR2PT 1,1,2;
CR2PT 2,2,3;
CR2PT 3,3,4;
CR2PT 4,5,4;
CR2PT 5,5,6;
CR2PT 6,6,7;
CR2PT 7,7,11;
CR2PT 8,11,12;
CR2PT 9,12,13;
CR2PT 10,13,14;
CR2PT 11,14,2;
CR2PT 12,7,8;
CR2PT 13,8,9;
CR2PT 14,9,10;
C* MCRC CR-first,nelements,last/first,no of.nodes,RC-group, CR-last, CR-step
MCRC   1,2,1,2,1,1;
ACTSET 3,2;
MCRC   2,2,1,2,2;
MCRC   3,2,1,2,3;
MCRC   4,10,1,2,3;
MCRC   5,2,1,2,3;
MCRC   6,2,1,2,4,7;
MCRC   8,2,1,2,3;
MCRC   9,10,1,2,3;
MCRC   10,2,1,2,3;
MCRC   11,2,1,2,2;
MCRC   12,2,1,2,5;
MCRC   13,10,1,2,6;
MCRC   14,2,1,2,5;
NMERGE ;
EGROUP 1,PIPE2D,1,0;
EGROUP 2,HEXC,1;
MCR4 4,13,7,2;
MCR4 13,9,7,2;
RCONST 1,PIPE2D,0.02,0,0,0,10,20;
RCONST 2,PIPE2D,0.02,0,0,0,10,20;
RCONST 3,PIPE2D,0.02,0,0,0,10,20;
RCONST 4,PIPE2D,0.03,0,0,0,10,20;
RCONST 5,PIPE2D,0.04;
RCONST 6,PIPE2D,.04,0,0,0,0,0,0,0,1E5;
RCHEX 7,.1,5000;
NFPT 1,PRES,-1,1.2E5;
NFPT 10,12,-1,1E5;
NFPT 1,TEMP,-1,20;
NFPT 1,CN,-2,1;
CURDEF -2,0,4,0,0,1,1,2,0,3,0;
OHMI=1
VISC(1)=1
DN(1)=1E-8
INITIA CN,0;
SOLVE 0,100,0.25,0,0,3,1,1,0;
```

MCRC – generation of elements on a curve (or curves). Only two-node elements PIPE2D are used here, which differs in various groups RC-*Real Constants* (the fifth parameter), i.e. in various channel equivalent diameters, heat transfer coefficients and also transversal component of electric field intensity, determining intensity of volumetric heating in the central channel.

MCR4 – generation of four-node elements HEXC which represent heat exchangers – they model heat transfer between pairs of PIPE2D elements already defined on a pair of curves by command MCRC.

RCONST defines tube diameter (e.g. 0.02 m), heat transfer coefficient (e.g. 10 W.m^{-2}.K^{-1}) and ambient temperature (e.g. 20^{0}C) corresponding to heat losses from lateral channels.

RCONST defines also transversal intensity of electric field in the central channel (E_y=10^5 V/m).

RCHEX defines heat transfer surface (related to the unit length of heat exchanger) and heat transfer coefficient between central and lateral channels.

Pressure is fixed at inlet (120 kPa) and outlet (100 kPa), constant temperature is specified only at inlet (20^{0}C). At inlet a time course of concentration simulates injection of tracer. This time course is defined by table number –2.

OHMI sets calculation of internal sources, in this case volumetric heat source. VISC and DN define viscosity and diffusion coefficient in material group 1. INITIA sets initial concentration zero. SOLVE performs 100 time steps, with Δt=0.25 s, 3 iterations for velocity field and 1 iteration for temperature and concentration in each time step.

Currently only results, corresponding to the time 25 seconds (100 steps x 0.25 seconds), are available in memory. You can pick nodes or elements (NID, EID) to look at the current values of pressures, flow-rates, temperatures and concentration in selected nodes and elements, you can use command GD1 to display contours of pressure, temperature and concentration, but all these values hold for time 25 s. The whole time history is available only in files *.OUT and *.TEP for nodal and element parameters respectively. Command GT reads the OUT-file and plots the time courses of nodal parameters selected by mouse. This procedure was used in the previous example (2D flow in a channel) and could have been used here as well. However, we shall use more complicated, but also a more general approach, based upon the command LTN *time* which reloads all nodal parameters from a specified time, using linear interpolation of data in the OUT-file (see also 2.2.2 where the same method has been applied). Using this, the procedure which transfers nodal values of concentrations for t=0,0.5,1.,1.5,…,25 into 50 rows of the observation matrix can be prepared in the following way

```
RECORD GTIME
      I=0
      T=0
      NT=50
      #LOOP LAB,NT;
      I=I+1
      LTN T;
      C1(I)=T
      C2(I)=V3(LPU(CN)+1)
      C3(I)=V3(LPU(CN)+ND)
      T=T+0.5
      #LABEL LAB;
ENDREC
```

LTN reloads initial values of all nodal parameters by values from time T.

Assign current time T into the 1st column, values of concentration in the node 1 (inlet) to the 2nd and the value of response (the last node ND is outlet) to the 3rd column of MOP. The vector V3 is vector of initial conditions of all nodal parameters (similarly V1 is vector of boundary conditions, V2 is vector of current results). The vector LPU serves for localization of a parameter in vectors V1,V2,V3, e.g. LPU(TEMP) is the beginning of temperatures, LPU(PRES) pressures, and LPU(CN) concentration.

The batch file GTIME is processed using command

F GTIME

and resulting response in the 3rd column of MOP can be processed in a standard way (i.e. moments can be calculated, and time course plotted).

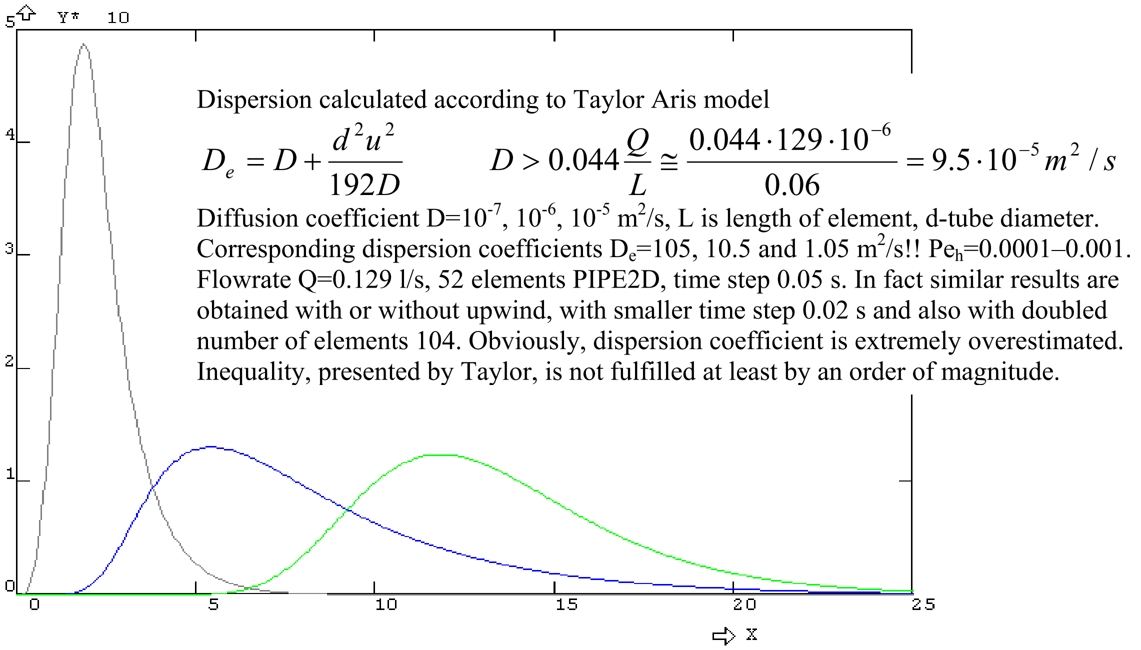
Before presenting and discussing results of stimulus-response experiments, we shall point out some problems related to the axial dispersion of enthalpy and concentration in the 1D flow in a pipe. In the analysed example a narrow triangular pulse of concentration has been defined at inlet as a stimulus function. At turbulent flow this concentration pulse is transferred in the flow direction by mean velocity of flow and at the same time is dispersed first of all by turbulent eddies. This situation can be described quite accurately by the model of axial dispersion, where dispersion coefficient is expressed as a function of Reynolds number. This is not so easy in laminar flows, because then at least two different regions must be analysed separately: region of a very slow flow, when the radial diffusion is sufficient to smooth out radial concentration profiles, and the region when diffusion can be completely neglected and dispersion is caused by convection only (by a non-uniform radial velocity profile). The slow flow region is characterised by inequality

$$D > k\frac{\dot{V}}{L} \qquad (2.2.11\text{-}1)$$

where D is diffusion coefficient, L is a characteristic length of pipe, $\dot{V}$ is volumetric flowrate and the constant k is 0.044 according to Taylor (1953), or 0.08 according to Hunt (1977). Example: Blood flow is almost always laminar and can be modelled by axial dispersion in most cases (in arteries or capillaries), however the flow in an aorta does not fulfil the inequality (1) – convection dominates and in the case of zero hematocrit even the turbulent flow could be expected (Re ≅ 2500).

In fact only the regions of turbulent or a very slow flow can be described using the model of axial dispersion correctly, because a pure convection at medium velocities is qualitatively different: convection disperses an initial short pulse into a broader pulse (but still a discontinuous pulse) and the speed of the pulse dispersion is constant and equals mean velocity of flow assuming parabolic velocity profile. On the other hand the diffusive dispersion is characterised by a continuous concentration profile and first of all by a decreasing rate of the peek expansion with time. In another words: while the initial peek is extended to the distance *u.t* in the convection dominated dispersion, the width is only $\sqrt{(\pi D_e t)}$ if a dispersion D_e is used in the axial dispersion model. Therefore if we want at least to match velocities of broadening a pulse, the dispersion coefficient should be proportional to time ($D_e \approx t$) !

Let us return back to the ohmic heater. The calculated flow is laminar in all elements for specified parameters (volumetric flowrate 0.129 l/s) and theoretical mean residence time calculated by FEMINA is 13.1 s. The following figure presents calculated responses corresponding to the model of Taylor Aris dispersion for three different values of diffusion coefficient:



Dispersion calculated according to Taylor Aris model

$$D_e = D + \frac{d^2u^2}{192D} \qquad D > 0.044\frac{Q}{L} \cong \frac{0.044\cdot 129\cdot 10^{-6}}{0.06} = 9.5\cdot 10^{-5}\, m^2/s$$

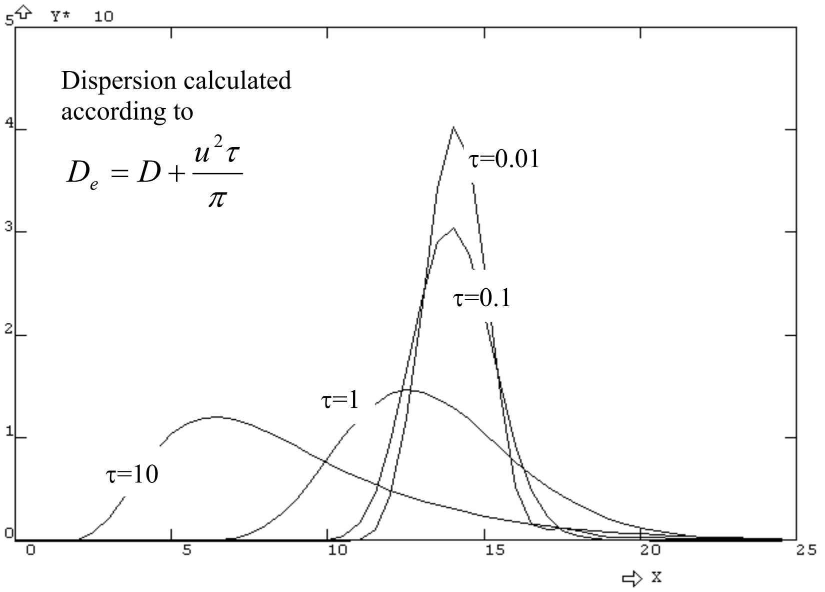
Diffusion coefficient D=10^{-7}, 10^{-6}, 10^{-5} m^2/s, L is length of element, d-tube diameter. Corresponding dispersion coefficients D_e=105, 10.5 and 1.05 m^2/s!! Pe_h=0.0001–0.001. Flowrate Q=0.129 l/s, 52 elements PIPE2D, time step 0.05 s. In fact similar results are obtained with or without upwind, with smaller time step 0.02 s and also with doubled number of elements 104. Obviously, dispersion coefficient is extremely overestimated. Inequality, presented by Taylor, is not fulfilled at least by an order of magnitude.

It is obvious, that the responses calculated for small (realistic) diffusion coefficients are quite wrong, tracer appears too soon at the outlet and the predicted mean residence time is too short.

The reason why the results are wrong is obvious: Taylor Aris model of axial dispersion cannot be applied because the flow is too fast and the radial concentration is not smoothed out by diffusion sufficiently. This is the worst case, having no satisfying solution so far. The following figure presents result corresponding to the axial dispersion model with modified dispersion coefficient D_e

$$u\tau = \sqrt{\pi D_e \tau} \quad \rightarrow \quad D_e = D + \frac{u^2\tau}{\pi} \qquad (2.2.11\text{-}2)$$

which stems from comparison between the width of pulse at pure convection and the characteristic diffusion depth at time τ.

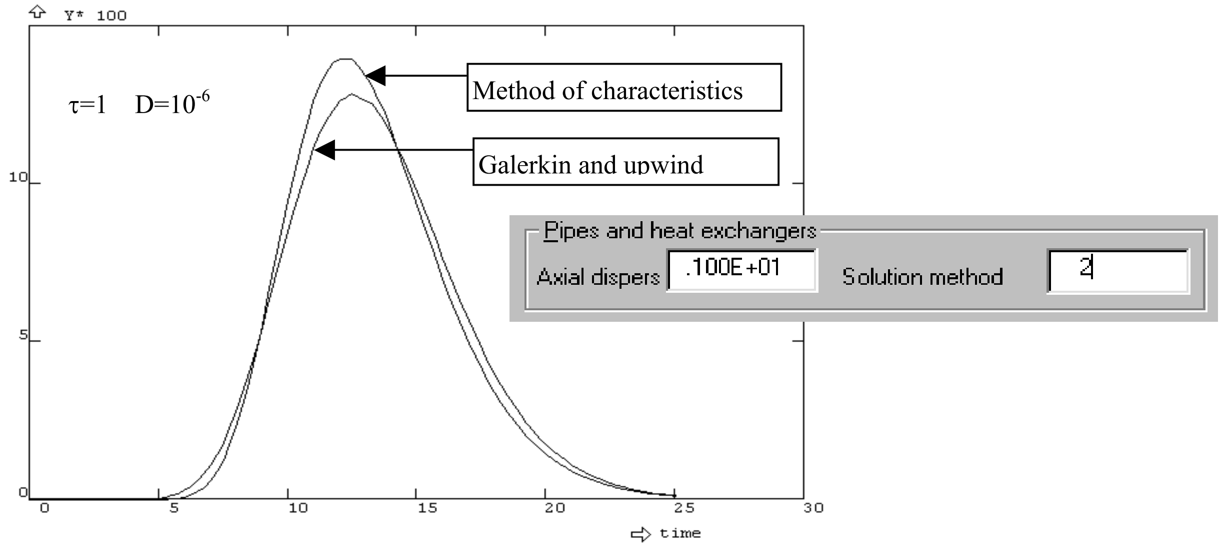


Dispersion calculated according to

$$D_e = D + \frac{u^2\tau}{\pi}$$

This is quite empirical approach, and the time τ is only a model parameter (system variable TAUD) which is to be identified either from real or from numerical experiments, as will be shown in the following chapter “Optimisation”.

In a similar way like in a 2D problem it is possible to solve the transport equations using method of characteristics, i.e. as a pure convection assuming piston flow and dispersion superposed in the following step. This option is activated from menu *Solution-Option*



Pipes and heat exchangers

Axial dispers .100E+01 Solution method 2

It is difficult to come to the conclusion, that the method of characteristics decreases numerical dispersion, from a single numerical experiment, however in this case the method of characteristic predicts response having significantly smaller variance in comparison with the Petrov Galerkin and even pure Galerkin methods.

2.3. Finite elements for stress analysis

Capability of FEMINA for stress analysis is rather restricted. Only static and linear analysis is available for some elements, it means that calculation of dynamics, large deformations or plastic behaviour of materials is not implemented. However, the most frequently encountered problems, for example static analysis of pipelines (where inner pressure and temperature have been calculated by the previous methods) or static analysis of rotationally symmetric pressure vessels can be performed.

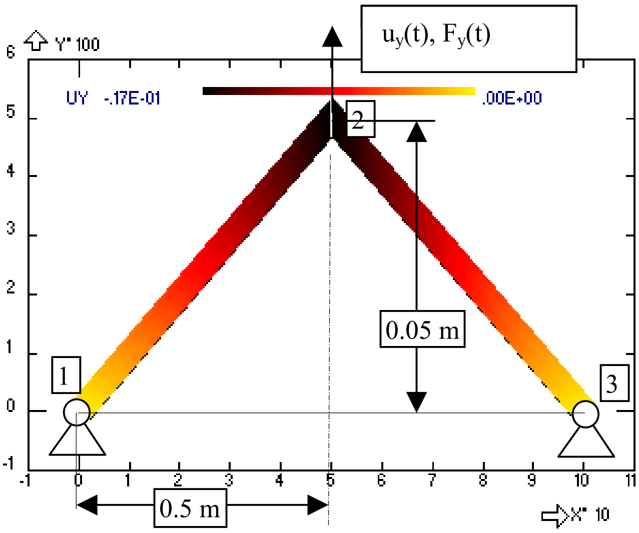
2.3.1. Trusses – nonlinear static analysis

The simplest static element is TRUSS2D – 1D element connected by a joint for example with elements BEAM, PLANE,…. So far this is the only element applicable also for *nonlinear* static, i.e. for large deformations and displacements, as will be demonstrated on the example of a strut frame formed by two trusses with relatively large angle of span. In this case the stiffness changes significantly with load and loss of stability can be expected if the load exceeds a critical limit.

```
PT 1,.000E+00,.000E+00;
PT 2,.500E+00,.0500E+00;
PT 3,.100E+01,.000E+00;
CR2PT 1,1,2;
CR2PT 2,2,3;
EGROUP 1,TRUSS2D,0,0;
RCONST 1,TRUSS2D,0,0,1E-4,0;
MCR 1,1,1,2;
MCR 2,1,1,2;
NMERGE ;
NFPT 1,UX,-1,0;
NFPT 3,UX,-1,0;
NFPT 3,UY,-1,0;
NFPT 1,UY,-1,0;
NFPT 2,UY,1,-1;
SOLVE 0,1,0,0,0,0,0,0,3;
```

Cross section 1 cm^2

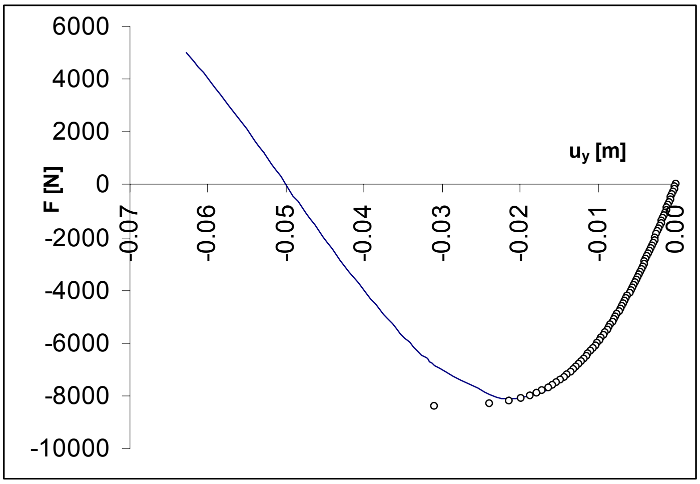
Load in joint 2 by force F= -1 N



Stability can be examined by gradual loading and monitoring the joint 2 displacement. The problem can be considered as a quasi-stationary, when force is a function of time, however the time plays no role from point of view of the system dynamics. We shall consider linear time increase of the force F_y (defined as function number 1) varying from 0 up to 8500 N at 85 increments. Predicted relationship between a load and displacement is compared with the analytical solution

$$F_y = \frac{EA}{L^3} u_y (H - u_y)(2H - u_y)$$

where A is cross-section of truss, L is length and H is an initial elevation of joint. Solution converges only up to the critical load when the trusses click. Up to this load the solution by FEMINA is remarkably accurate.



```
FUNDEF 1,TIME;
NFPT 2,UY,1,-1;
SOLVE 0,100,85,0,0,0,0,0,5;
```

2.3.2. Beams and pipes

Pipeline network, where pressures, flowrates and temperatures have been calculated, can be analysed also from point of view of stresses and it is possible to calculate forces and moments caused by internal pressure, dilatation and external loads. That is because the element PIPE2D is calculated as a beam with second moment of cross section

$$J_z = \frac{\pi D_e^4}{64}[1-(\frac{D_i}{D_e})^4]$$, according to D_i and h parameters in group RCONST

It also possible to calculate beams with arbitrary cross section, and in this case it is necessary to specify both the cross section area A and the moment of inertia J_z explicitly (using the command RCONST). Beam elements (PIPE2D, BEAM2D) can be loaded by four possible ways

1) Forces F_x, F_y (N) or moments M_z (N.m) in arbitrary nodes.
2) Continuous load, specified as a pressure p in parameters RCONST. However, this is not an inner pressure, but a force acting on a unit length of pipe or beam perpendicularly to the element axis.
3) Dilatation forces, caused by different temperature of wall (it is assumed that the wall temperature is the same as the temperature of liquid inside the pipe) and a reference mounting temperature T_e, which is also specified as a RCONST parameter.
4) The element PIPE2D, i.e. a pipe loaded by inner pressure (nodal parameter), exhibits a contraction, having a similar effect as the thermal dilatation.

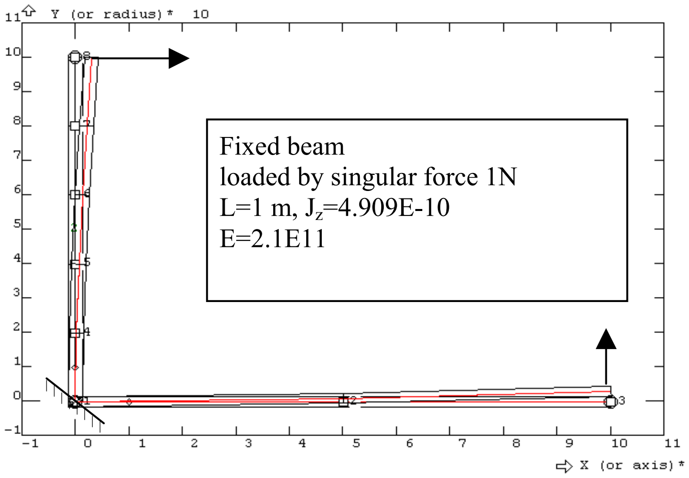
Results of solution are nodal parameters – displacements and rotations at nodes, and inner forces are calculated during postprocessing. These forces are located in the zone of element parameters: axial force (N) under name ENA (N_α), transversal force ENB (N_β), and bending moment EMA (M_α).

The following example is a clamped beam loaded by a singular force at the end. In fact there are two identical beams in the example, demonstrating that results are independent of number of elements used (2 and 5 elements):

```
PT 1,0,0;
PT 2,1,0;
PT 3,0,1;
CR2PT 1,1,2;
CR2PT 2,1,3;
EGROUP 1,BEAM2D,0,0;
RCONST 1,BEAM2D,0,0,0.0000785,0,4.909E-10;
MCR 1,2,1,2;
MCR 2,5,1,2;
NMERGE ;
NFPT 3,UX,1,1;
NFPT 1,UX,-1,0;
NFPT 1,UY,-1,0;
NFPT 1,RZ,-1,0;
NFPT 2,UY,1,1;
SOLVE 0,1,1,0,0,0,0,0,1;
```

Cross section and moment of inertia

Calculated displacements and rotation can be compared with the analytical solution



Fixed beam
loaded by singular force 1N
L=1 m, J_z=4.909E-10
E=2.1E11

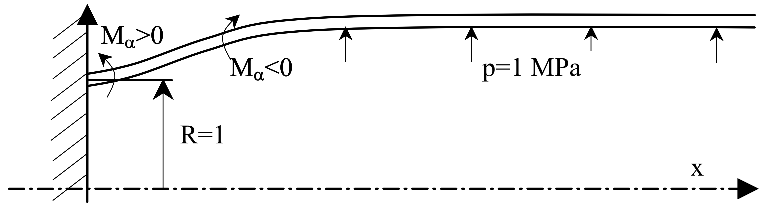
$$u_y = \frac{F_y L^3}{3EJ_z} = 0.003233\, m, \qquad \varphi_z = -\frac{F_y L^2}{2EJ_z} = 0.00485\,.$$

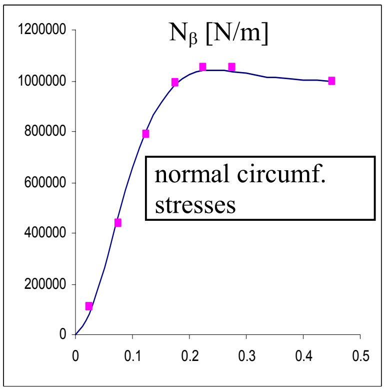
2.3.3. Rotationally symmetric vessels (shells)

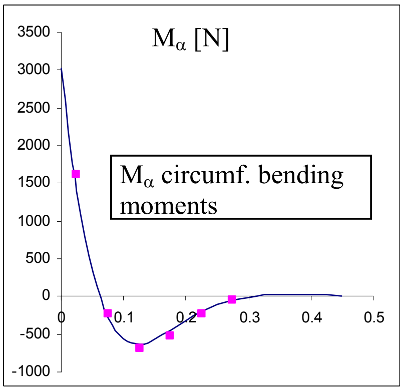
Shell elements SHELLAX calculate deformations and distribution of internal forces and moments in rotationally symmetric and symmetrically loaded shells with the symmetry axis *x* (the coordinate *y* is radius). Shell can be loaded by inner pressure (specified as a parameter of RCONST) or by singular forces and moments. Finite elements having 3 degree of freedom in each node (displacements UX,UY and rotation RZ) are used.

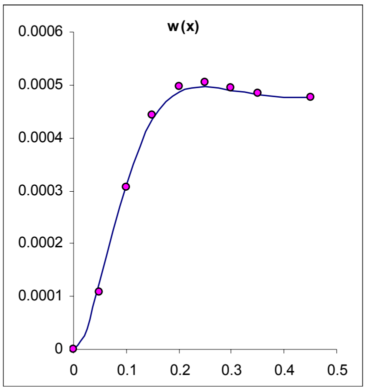
First we introduce examples, where analytical solution exists: We shall analyse a horizontal pipe loaded by inner pressure with one end clamped and the other end free (radius R=1m, thickness of wall h=0.01m, overpressure p=1 MPa, material steel E=2.1E11 Pa, μ=0.3).

The following figures show graphs of circumferential stresses N_β (positive, because radius is increased), bending moments M_α (sign is changing, there is a big positive moment near the clamp, causing inner filament elongation) and also radial displacement.









Continuous graphs correspond to the analytical solution (Křupka 1986), points have been obtained by the following program

```
PT 1,.000E+00,.100E+01;
PT 2,.100E+01,.100E+01;
CR2PT 1,1,2;
EGROUP 1,SHELLAX,0;
RCONST 1,SHELLAX,0.01,1E6;
MPROP 1,SHELLAX,.210E+12,0,.3,0;
MCR 1,20,1,2;
NFPT 1,UX,-1,0;
NFPT 1,UY,-1,0;
NFPT 1,RZ,-1,0;
SOLVE 0,1,0,0,0,1,1,0,1;
```

element SHELLAX, stationary case (0)

wall thickness H and pressure

modulus of elasticity and Poisson const.

20 elements, uniform distr. (1), two nodes

Fix displacements Ux,Uy and rotation Rz in point 1 (boundary condition no..-1, with value 0).

Solution. Only the last parameter is important, because specifies number of iteration in STRESS analysis.

```
NDLIST 1,2,1
NIDENT
EIDENT
WRITE 1
WRITE 2
WRITE 4
```

list nodes

pick nodal parameters (UX,UY,RZ) using mouse

pick element parameters (ENA,…) using mouse

Write information about nodes, elements and element parameters into files (*.NOD,*.ELE,*.EPA)

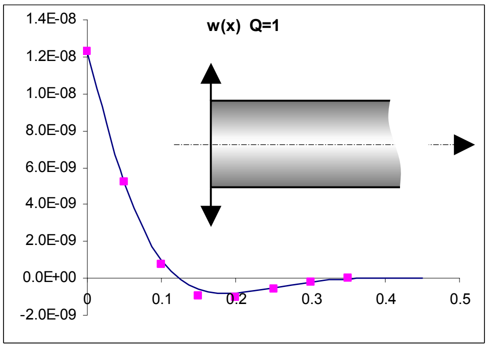
The commands GD1,GE1,GC plot graphs of calculated nodal and element parameters.

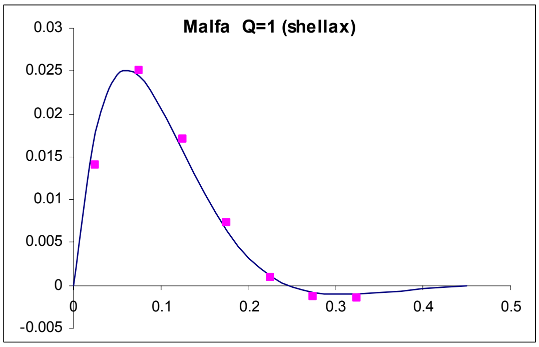
Almost the same program can be used for the pipe loaded by lateral moment and radial force (related to unit circumference).

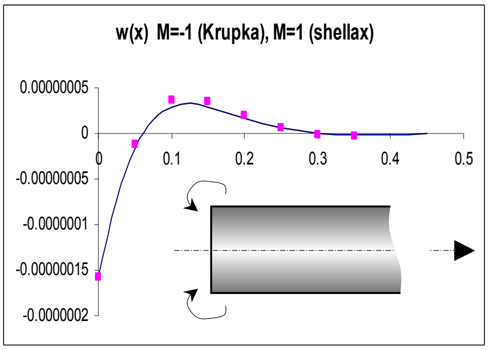
```
RCONST 1,SHELLAX,0.01,0;
NFPT 1,RZ,1,1;
SOLVE 0,1,0,0,0,1,1,0,1;
NFPT 1,UY,1,1;
SOLVE 0,1,0,0,0,1,1,0,1;
```

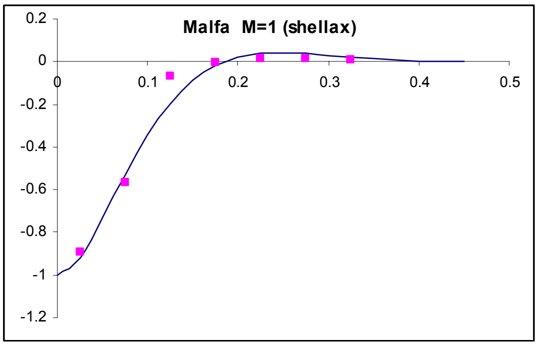
Status of rotation RZ in node 1 (originally clamped) is changed to a positive value (within the range 1 up to 20). Specified value of DOF will be in this case interpreted as a corresponding load (moment M_α). The same procedure will be applied to force (parameter correponding to a force acting in the direction y is UY).

The following figures present results of numerical solution with 20 elements and analytical solution (continuous graphs).





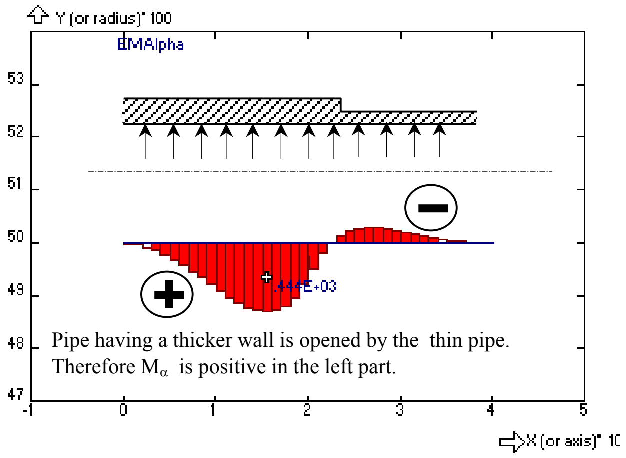




Note: The calculated quantities are displacements UX,UY, rotations RZ (in radians) and unit forces, evaluated from deformation of individual elements (they are parameters of elements, named by ENA-N_α, ENB-N_β, EMA-M_α, EMB-M_β, EQ-Q). All these unit forces are related to the unit length of the shell section: N_α, N_β (unit N/m) are normal forces acting in the meridian or in the circumferential direction respectively, M_α, M_β (unit N) are bending moments invoking bending stresses in the meridian and circumferential direction and Q (N/m) is a force normal to the shell surface. Sign conventions differ when setting and interpreting these forces (because forces are specified in a global coordinate system *x,r*, while results are in the coordinate system of element): Positive *calculated* values N_α, N_β indicate normal *tensile* stresses, and positive moments M_α, M_β correspond to *tensile* stresses at *inner* surface (simply speaking positive calculated moments tend to "open" the shell). On the other hand a positive *specified* value of the bending moment M_z corresponds to the loading moment always oriented counter clockwise (therefore the positive value M_z=1 prescribed at the left edge of pipe would cause the edge shrinking, while the same positive moment at the right edge would cause the pipe opening).

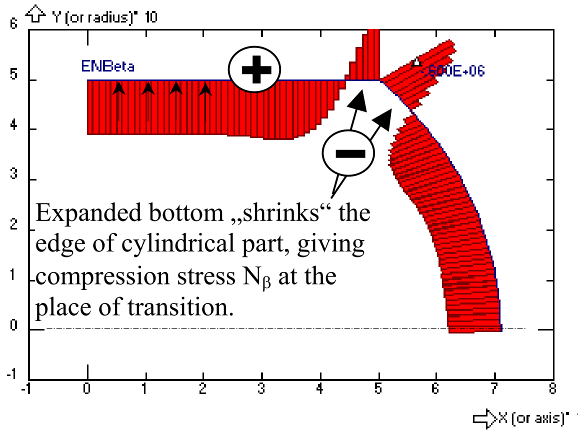
The following example is a model of connection of two pipes with different thickness of wall (radius is the same R=0.5 m, but one pipe has thickness 0.02 m and the second only 0.01m). Membrane stresses are perturbed and waves of bending stresses M_α appear when the pipe is loaded by inner pressure p=1 MPa. The range of disturbance depends upon the thickness of wall ≅ $3.12\sqrt{Rh}$ as you can observe in the following figure

```
PT 1,.000E+00,.500E+00;
PT 2,.200E+00,.500E+00;
PT 3,.400E+00,.500E+00;
CR2PT 1,1,2;
CR2PT 2,2,3;
EGROUP 1,SHELLAX,0;
MPROP 1,SHELLAX,.210E+12,0,0.333,0;
RCONST 1,SHELLAX,.02,1E6;
MCR 1,20,1,2;
RCONST 2,SHELLAX,.01,1E6;
MCR 2,20,1,2;
NMERGE ;
NFPT 1,UX,-1,0;
SOLVE 0,1,0,0,0,1,1,0,1;
```

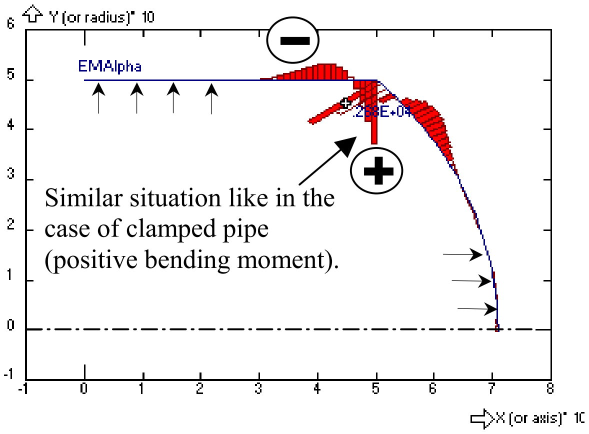


Pipe having a thicker wall is opened by the thin pipe. Therefore M_α is positive in the left part.

The most important applications of elements SHELLAX are calculations of symmetrical and symmetrically loaded pressure vessels, tanks, bellows and so on. As an example a cylindrical vessel with spherical bottom will be calculated (material steel, wall thickness 0.01 m, inner pressure 1 MPa). The transition between the cylindrical and the spherical part of vessel is not smooth, because assumed angle of meridian at the edge of spherical bottom is 45^0. Therefore waves of bending stresses M_α and circumferential stresses N_β appear as shown in the following figure (use the command GE1):



Expanded bottom „shrinks“ the edge of cylindrical part, giving compression stress N_β at the place of transition.



Similar situation like in the case of clamped pipe (positive bending moment).

```
PT 1,.000E+00,.500E+00;
PT 2,.400E+00,.500E+00;
PT 2,.500E+00,.500E+00;
PT 3,.000E+00,.000E+00;
CR2PT 1,1,2;
ARC 2,3,2,-45,1;
EGROUP 1,SHELLAX,0;
RCONST 1,SHELLAX,0.01,1E6;
MPROP 1,SHELLAX,.210E+12,0,.3,0;
MCR 1,50,1,2;
MCR 2,50,1,2;
NMERGE ;
NFPT 1,UX,-1,0;
SOLVE 0,1,0,0,0,1,1,0,1;
```

The only new command ARC – forms approximately circular arch using one or more quadratic curves. What is specified: center (PT 3), initial point (PT 2), angle (-45^0 means clockwise rotation) and numbe r of segments (1). It is an analogy of the command CIRCLE, keypoints can be picked by mouse.

Actual membrane and bending stresses can be calculated from unit forces using expressions

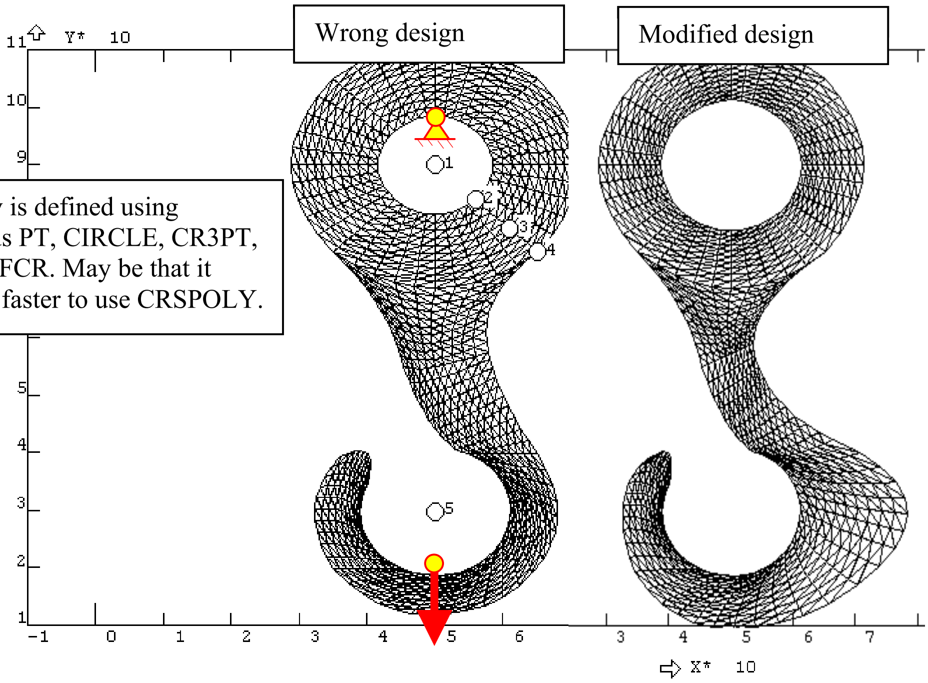
$$\sigma_{membrane} = \frac{N}{h} \qquad \sigma_{bending} = \frac{6M}{h^2}$$

2.3.4. Plane stress, plane strain

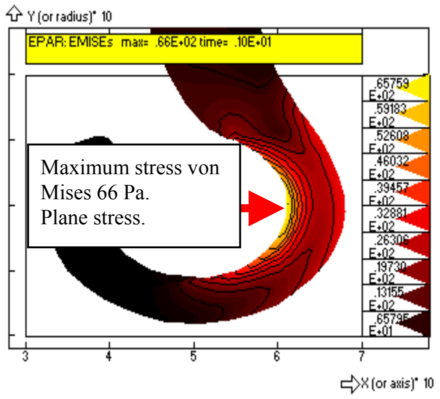
Programs for plane stress and strain analysis are more or less standard, models can be designed using triangular or quadrilateral elements. Results of post-processing are stresses σ_{xx}, σ_{yy}, τ_{xy}, σ_{Mises} (mean element values).

```
PT 1,.500E+00,.900E+00;
PT 2,.550E+00,.850E+00;
PT 3,.600E+00,.800E+00;
PT 4,.650E+00,.750E+00;
PT 5,.500E+00,.300E+00;
CIRCLE 1,1,2,4;
CIRCLE 5,1,3,4;
CIRCLE 9,1,4,4;
CR3PT 13,2,4,3;
CR3PT 14,7,21,14;
CR3PT 15,9,23,16;
CR3PT 16,11,25,18;
PT 27,.400E+00,.650E+00
PT 28,.450E+00,.550E+00
PT 29,.600E+00,.650E+00
PT 30,.579E+00,.577E+00
PT 31,.500E+00,.550E+00
PT 32,.500E+00,.450E+00
PT 33,.550E+00,.400E+00
PT 34,.600E+00,.400E+00
PT 35,.600E+00,.500E+00
PT 36,.650E+00,.400E+00
CIRCLE 17,5,33,4;
CIRCLE 21,5,34,4;
CIRCLE 25,5,36,4;
SFCR 1,11,16;
SFCR 2,12,13;
SFCR 3,9,14;
SFCR 4,10,15;
SF8PT 5,28,30,4,25,31,29,26,27;
CR3PT 32,33,36,34;
CR3PT 33,42,56,49;
CR3PT 34,41,55,48;
SF8PT 6,28,33,36,30,44,34,35,31;
SFCR 7,32,28;
CR3PT 37,40,54,47;
CR3PT 38,38,52,45;
SFCR 8,33,27;
SFCR 9,37,26;
EGROUP 1,PLANE2D,0,3,0,0,0;
MSF 1,10,10,1,1,3;
MSF 2,10,10,1,1,3;
MSF 3,10,10,1,1,3;
MSF 4,10,10,1,1,3;
MSF 5,10,10,1,1,3;
MSF 6,10,10,1,1,3;
MSF 7,10,10,1,1,3;
MSF 8,10,10,1,1,3;
MSF 9,10,10,1,1,3;
NMERGE ;
NF 436,UY,-1,0;
NF 436,UX,-1,0;
NF 847,UY,1,-1;
NF 847,UX,-1,0;
OPTION 0,0,0,1,0,30,.100E-11,1,.100E+08,1,.100E-04,.100E-04,0,1,1,0,0;
SOLVE 0,1,1,0,0,0,0,0,1;
```

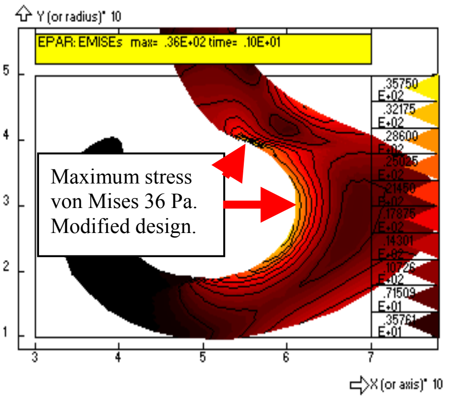
Geometry is defined using commands PT, CIRCLE, CR3PT, SF8PT, SFCR. May be that it would be faster to use CRSPOLY.



Parameter of EGROUP, STRESS=0 – plane stress, STRESS=1 – plane strain. Results are almost the same in this case.



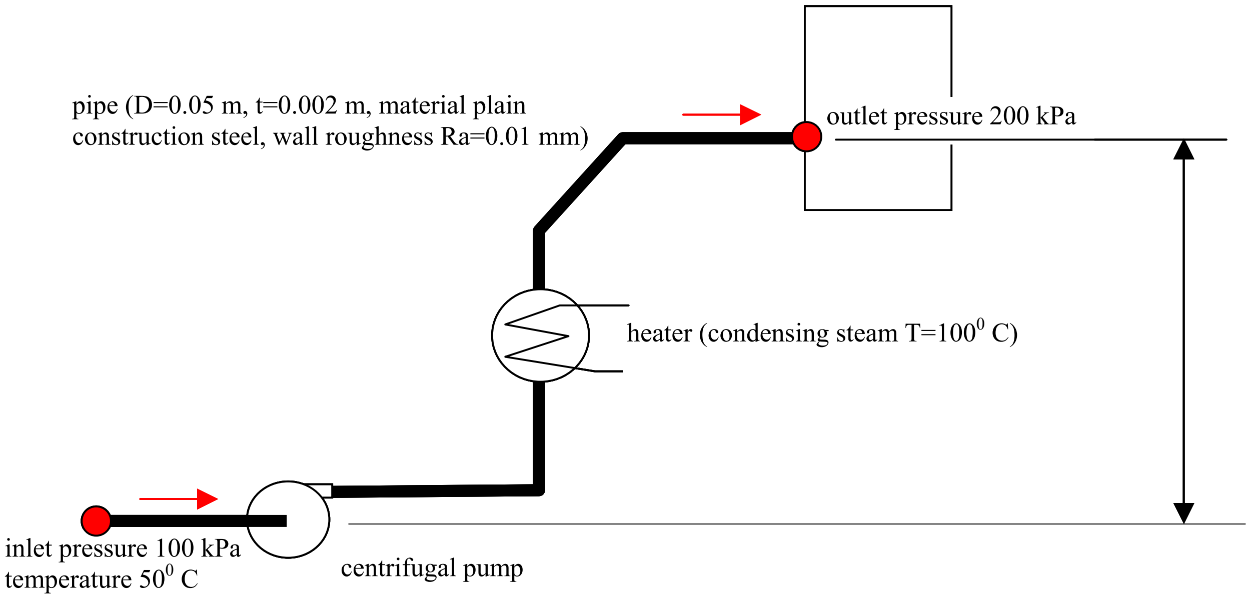
Maximum stress von Mises 66 Pa. Plane stress.



Maximum stress von Mises 36 Pa. Modified design.

2.3.5. Case study: pipeline design

In this example we shall consider basic design of a simple pipeline with a pump and heater, comprising fluid flow analysis (flowrates), heat transfer analysis (temperature profiles at a steady state) and finally stress analysis taking into account dilatations and pressure load of pipes. First step is always definition of problem, shown in the following schematic figure


pipe (D=0.05 m, t=0.002 m, material plain construction steel, wall roughness Ra=0.01 mm)
outlet pressure 200 kPa
heater (condensing steam T=100^0 C)
inlet pressure 100 kPa
temperature 50^0 C
centrifugal pump

Next step is a decision which kinds of finite elements should be used: Obvious choice is selection the PIPE2D elements for pipes and PUMP elements for pump. Heater is a heat exchanger and there are several possibilities, depending first of all upon the type of heat exchanger (plate, shell&tube, hairpin,…). In our case we select instead of the HEXC four nodal elements again a PIPE2D element with appropriate modification of boundary conditions, corresponding to the heat transfer from the condensing steam.

Solution of the problem starts with definition of geometrical model, in fact only a polyline consisting of curves 1 (inlet pipe), 2 (pump), 3 (pipe), 4 (heated section), and 5-7 (insulated pipes). This can be created by a single command CRS using mouse (it is useful to set a grid using GON to simplify positioning of keypoints).

Then the curves are meshed by selected finite elements using commands RC (definition of parameters – diameter, wall thickness and roughness for pipes), EG (definition of the element type, in our case PIPE2D or PUMP) and MCR (mesh curve). This is probably the most important and difficult part of our solution, because some parameters must be estimated, for example the heat transfer coefficient at the outer surface of pipes in the heater section and coefficients of local friction losses (you can use auxiliary tools accessed from menu bar Calculation or in dialog panels for setting parameters). Special attention should be devoted to pump: there are several options depending upon the type of pump (positive displacement pump, pump with a used defined hydraulic characteristics or selection a pump from database). We choose the last option: In this case the optimal pump is selected from the SIGMA catalogue, according to the specified requirements (kind of pump, diameter of nozzles, estimated flowrate and estimated pressure height). Meshing must be completed by the command NMERGE which connects the elements on adjoined curves.

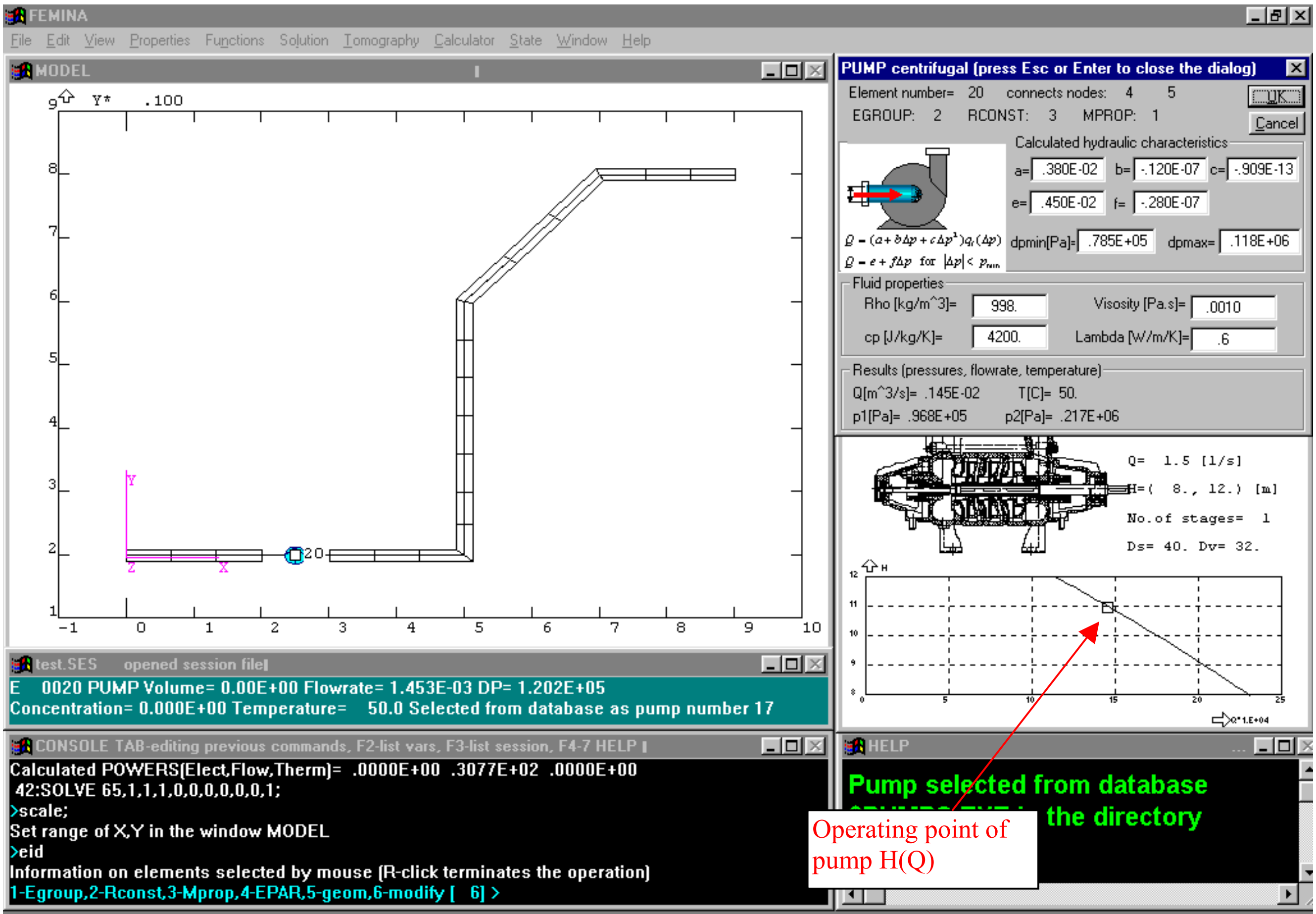
Because we are going to carry out three kinds of analysis (flow, heat and stress), three kind of boundary conditions must be specified for PRESS (pressure), TEMP (temperature), UX,UY,RZ (displacements and rotations). Assuming that there are no external supports, it is sufficient to specify these parameters only at inlet and outlet (red circles) using commands FPT (Fix PoinTs).

From now on it is possible to start finite element analysis by the command SOLVE, where it is sufficient to specify only 1 time step (we consider only steady state analysis – magnitude of time step is unimportant) and number of iterations for flow (due to nonlinear characteristic of pump the number should be rather large, for example 20), heat transfer (1 iteration is sufficient because we consider temperature independent properties) and stress analysis (1 iteration – small deformation and linear elasticity is assumed). The three kinds of analysis can be performed together, using one command SOLVE, however in the following session file the analysis were performed separately – we believe that it is a good practice to start with the flow analysis, check results (pressure profiles and first of all flowrates – in this case flowrates should be the same in all elements!) and even than start the thermal and stress analysis.

```
C* PIPELINE with a centrifugal pump
C* FLOW,THERMAL, and STRESS analysis
C*
PT 1,.000E+00,.200E+02;
PT 2,.200E+02,.200E+02;
CR2PT 1,1,2;
PT 3,.300E+02,.200E+02;
CR2PT 2,2,3;
PT 4,.500E+02,.200E+02;
CR2PT 3,3,4;
PT 5,.500E+02,.300E+02;     GEOMETRY (points PT and curves CR). This is accomplished by single
CR2PT 4,4,5;                command CRS
PT 6,.500E+02,.600E+02;
CR2PT 5,5,6;
PT 7,.700E+02,.800E+02;
CR2PT 6,6,7;
PT 8,.900E+02,.800E+02;
CR2PT 7,7,8;
RCONST 1,PIPE2D,.5E-01,.1E-03,0,.1E+01,0,0,.196E-02,0,0,0,0,0,0,0,0,.12E-04,0,
MCR 1,3,1,2,0,0,1;
MCR 3,3,1,2,0,0,1;
MCR 6,3,1,2,0,7,1;            Meshing pipes (use dialog menu bar Properties, RCONST)
MCR 4,2,1,2,0,0,1;
RCONST 2,PIPE2D,.5E-01,.1E-03,0,0,.1E+05,100,.196E-02,0,0,0,0,0,0,0,0,.120E-04,0;
MCR 5,5,1,2,0,0,1;            Meshing heat exchanger
EGROUP 2,PUMP,0,-2,0,0;
RCONST 3,PUMP,0,0,0,0,0,0,.100E-02;
RCEPUM 20,0,0.05,0.001,10;     Create element PUMP
MCR 2,1,1,2,0,0,1;
NMERGE ;
NFPT 1,PRES,-1,1E5;             Fixed (constant) pressure at inlet and outlet
NFPT 8,PRES,-1,2E5;
RELFAKT=.1
SOLVE 0,1,1,0,0,20,0,0,0,0,0;
NFPT 1,TEMP,-1,50;              Fixed (constant) temperature at inlet (outlet temperature is calculated)
SOLVE 25,20,1,1,0,0,1,0,0,0,0;
NFPT 1,UX,-1,0;
NFPT 1,UY,-1,0;
NFPT 1,RZ,-1,0;                 Fixed support (displacement UX,UY and rotation RZ) at inlet,
NFPT 8,UX,-1,0;                 Fixed displacements at outlet. You can fix also for example the pump
NFPT 8,UY,-1,0;                 or heat exchanger. Remark: PUMP is considered as a rigid body.
SOLVE 65,1,1,1,0,0,0,0,0,0,1;
```

To see results use the commands NID (node identification), EID (element identification) or plot temperature and pressure profiles by commands GD1 (graphs of Degree of freedoms), moments, forces, flowrates by GE1 (graph of Element parameters)[19]. The following screens show dialog panels displayed when picking a pump and a pipe by mouse (command EID):

[19] All results are of course in output files *.OUT, *.TEP as usually, journal of processing in *.SES and *.DBG files.



PUMP centrifugal (press Esc or Enter to close the dialog)

Element number= 20 connects nodes: 4 5

EGROUP: 2 RCONST: 3 MPROP: 1

OK Cancel

Calculated hydraulic characteristics

a= .380E-02 b= -.120E-07 c= -.909E-13

e= .450E-02 f= -.280E-07

$Q=(a+b\Delta p+c\Delta p^2)q_r(\Delta p)$

$Q=e+f\Delta p$ for $|\Delta p|<p_{min}$

dpmin[Pa]= .785E+05 dpmax= .118E+06

Fluid properties

Rho [kg/m^3]= 998. Visosity [Pa.s]= .0010

cp [J/kg/K]= 4200. Lambda [W/m/K]= .6

Results (pressures, flowrate, temperature)

Q[m^3/s]= .145E-02 T[C]= 50.

p1[Pa]= .968E+05 p2[Pa]= .217E+06

test.SES opened session file

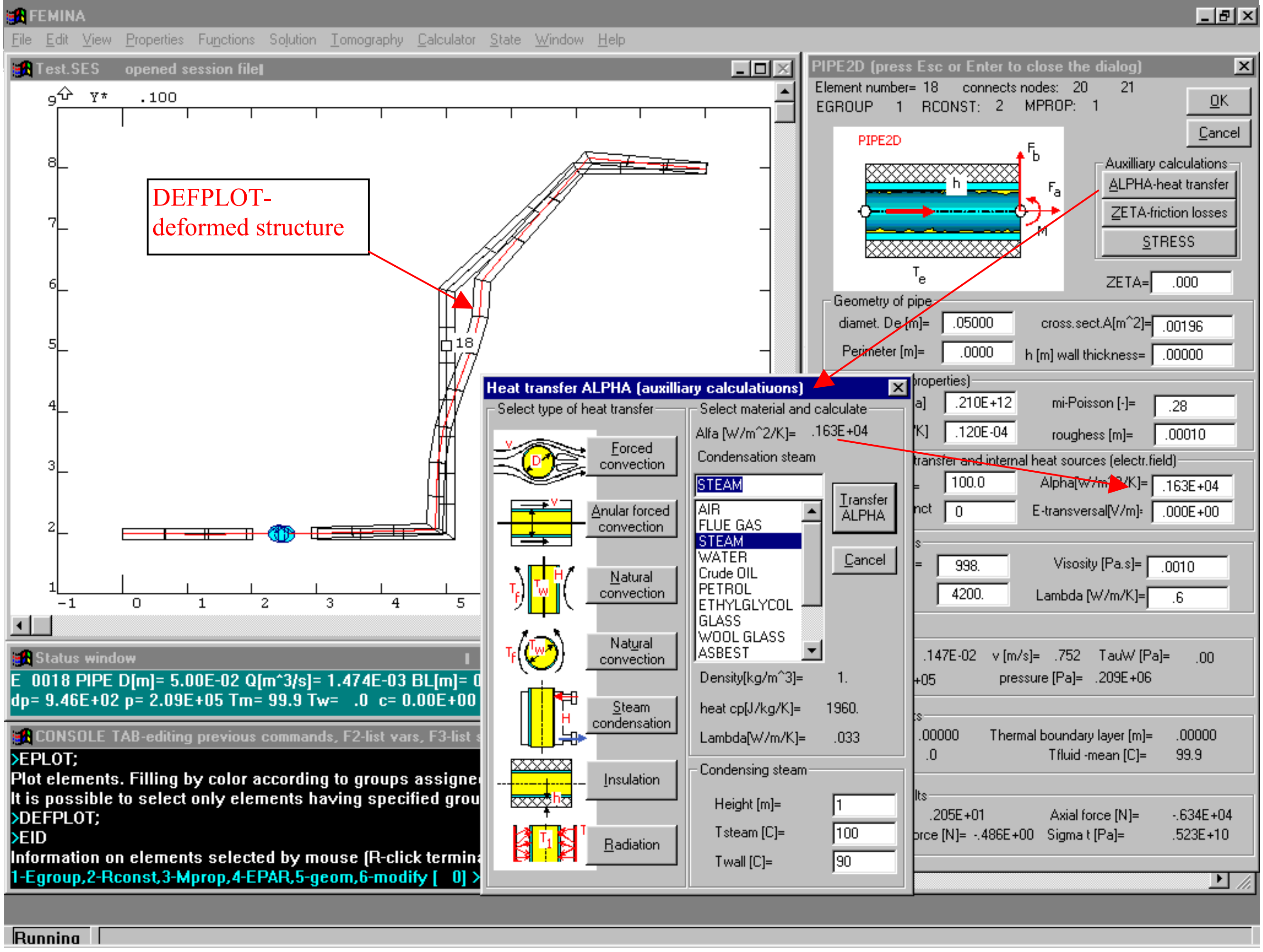
```
E  0020 PUMP Volume= 0.00E+00 Flowrate= 1.453E-03 DP= 1.202E+05
Concentration= 0.000E+00 Temperature=   50.0 Selected from database as pump number 17
```

CONSOLE TAB-editing previous commands, F2-list vars, F3-list session, F4-7 HELP

```
Calculated POWERS(Elect,Flow,Therm)=  .0000E+00  .3077E+02  .0000E+00
 42:SOLVE 65,1,1,1,0,0,0,0,0,0,1;
>scale;
Set range of X,Y in the window MODEL
>eid
Information on elements selected by mouse (R-click terminates the operation)
1-Egroup,2-Rconst,3-Mprop,4-EPAR,5-geom,6-modify [  6] >
```

Pump selected from database the directory

Operating point of pump H(Q)



DEFPLOT- deformed structure

PIPE2D (press Esc or Enter to close the dialog)

Element number= 18 connects nodes: 20 21

EGROUP 1 RCONST: 2 MPROP: 1

OK Cancel

Auxilliary calculations: ALPHA-heat transfer, ZETA-friction losses, STRESS

ZETA= .000

Geometry of pipe

diamet. De [m]= .05000 cross.sect.A[m^2]= .00196

Perimeter [m]= .0000 h [m] wall thickness= .00000

properties)

a] .210E+12 mi-Poisson [-]= .28

K] .120E-04 roughess [m]= .00010

transfer and internal heat sources (electr.field)

= 100.0 Alpha[W/m^2/K]= .163E+04

nct 0 E-transversal[V/m]= .000E+00

= 998. Visosity [Pa.s]= .0010

4200. Lambda [W/m/K]= .6

.147E-02 v [m/s]= .752 TauW [Pa]= .00

+05 pressure [Pa]= .209E+06

.00000 Thermal boundary layer [m]= .00000

.0 Tfluid -mean [C]= 99.9

.205E+01 Axial force [N]= -.634E+04

rce [N]= -.486E+00 Sigma t [Pa]= .523E+10

Heat transfer ALPHA (auxilliary calculatiuons)

Select type of heat transfer

Select material and calculate

Alfa [W/m^2/K]= .163E+04

Condensation steam

STEAM

AIR
FLUE GAS
STEAM
WATER
Crude OIL
PETROL
ETHYLGLYCOL
GLASS
WOOL GLASS
ASBEST

Transfer ALPHA Cancel

Density[kg/m^3]= 1.

heat cp[J/kg/K]= 1960.

Lambda[W/m/K]= .033

Condensing steam

Height [m]= 1

Tsteam [C]= 100

Twall [C]= 90

Status window

```
E  0018 PIPE D[m]= 5.00E-02 Q[m^3/s]= 1.474E-03 BL[m]= 0
dp= 9.46E+02 p= 2.09E+05 Tm= 99.9 Tw=   .0  c= 0.00E+00
```

CONSOLE TAB-editing previous commands, F2-list vars, F3-list s

```
>EPLOT;
Plot elements. Filling by color according to groups assigne
It is possible to select only elements having specified grou
>DEFPLOT;
>EID
Information on elements selected by mouse (R-click termin
1-Egroup,2-Rconst,3-Mprop,4-EPAR,5-geom,6-modify [  0] >
```

Remark: The previous example assumed that the heater is formed by single vertical jacketed pipe (steam condenses in a duplicator). However it is also possible to substitute the pipe with the bundle of pipes just only by changing a proper parameter in EGROUP (N-number of parallel pipes) – in this case any specified and any calculated parameter (diameter, cross section, roughness of wall, as well as wall shear stress, flowrate etc) concerns only one pipe of the bundle.

Remark: Previous figures demonstrate some auxiliary tools for engineering estimates of pressure drop in pipes, for evaluation of stresses in pipes, for calculation of heat transfer coefficient at outer surface of pipes and for estimation of local friction loss coefficient ζ. These tools are accessible from menu bar *Calculator* and also from dialog panels activated during inspection of elements by command EID-modify.

2.4.RTD models and response processing

RTD modelling and response processing has nothing to do with finite element data structures and geometrical models (entities PR,CR,SF,VL,ND,EL), the only important data structure is the matrix of observation points (MOP) consisting of time responses stored as columns of MOP. The first column of MOP matrix is time and the following columns (2,3,…,10) are corresponding functions. Therefore the observation matrix can contain up to 9 different time functions represented by a table of values with a common time base. By the way, the MOP has been used in the previous examples for storing temperature and concentration time courses predicted by FEM and further on it will be used in regression and lumped parameter model analysis for storing inputs and predicted outputs of a system, as well as for the measured responses.

The basic tasks concern the problem how to read data into MOP from an ASCII file, and how to modify, display and plot the data.

Basic commands ensuring these operations are

ROM reads selected columns of MOP from a file with postfix TXT or DAT, which can be created by any text processor, Excel or a data logger. Data can be in a free format with numbers or text items separated by one or more spaces (e.g. –1.2 –1,2 –1.2e+3 are valid formats of numbers), and in the rows describing the observation points only numerical items are expected. The data file is selected from menu and the selection is enhanced by continuous displaying the content of browsed files. After a file is selected it is necessary to specify the row containing column labels (used for headers) and first of all the rows with data which are to be transported into MOP. Further on a correspondence between columns in the data file and columns in MOP must be established (it is possible to select up to 10 columns in data and assign them arbitrary columns of MOP). The operation ROM is sufficient for most cases and only if it is necessary to import time courses having a different time base, the general command READ has to be used (in this case the data are transformed to a different time base using quadratic interpolation[20]).

T *n,Δt* Creation of an empty matrix having *n* rows (*n*-observation points) and a time base (first column of MOP) with a constant time step. NT (number of points) and DT (time step) are system variables.

TCEDIT (TCE)[21] graphic editing of a selected time curve using mouse.
TCINPUT (TCI) manual input of MOP entries from keyboard.
TCLIST (TCL) list of all 10 columns of MOP in the window MODEL (TCS command can be used for naming selected columns)
GTC *n* plot several selected time curves (new window with automatic scale)
TCP *index* plot a selected time curve (add the curve to the existing graph)
ZI,ZO,SCALE,CLS,MA Zoom-in, zoom-out, new scale, clear screen, magnify (the same commands as in FEM)
TID identify a time curve and a point of the curve using mouse (Left-Click picks, Right-Click ends the inspection routine); information on a picked curve and point is immediately displayed (coordinates of point, number of points, mean residence time, variance and even the third moment of the picked curve). This is a similar operation like NID,EID,PID, CID, SID,… in FEM.

[20] This is option TC of the command READ. Quadratic interpolation (in fact quadratic regression splines) is used if there is only a few data points, while in the opposite case, if there is a huge amount of data, much greater than the number of available rows in the matrix MOP, the data are integrated and only average values are placed into MOP.

[21] Any command in FEMINA has several synonyms and for example the keywords TCEDIT, TCE and TED have the same meaning.

TCF *assignment* in this way it is possible to define the whole column as an algebraic expression. Columns are variables C1,C2,…,C10 and the variable I is the row index. For example the command TCF C1=0.1*I defines equidistant times (the first column will be 0.1, 0.2, 0.3,…), and the command TCF C3=C2*EXP(-0.05*C1) represents in fact a half life decay correction of a response in the second column. In the case that only selected range of rows should be calculated use the TCFR command instead of TCF.

SMOOTH i_{source},$i_{smoothed}$,*method* smoothing a part of the time curve i_{source} using regression splines or medians (smoothing method is selected by the parameter *method*). The smoothed part is localised by mouse.

TAIL i_{source},$i_{smoothed}$,*method* tail approximation either by the exponential functions *A+B* exp(-*Ct*) (*method*=1), *A+B* t exp(-*Ct*) (*method*=2), or by the convective model $A+B/t^3$ (*method*=3). The coefficients *A,B,C* are evaluated by nonlinear regression method. The parameter *A* is an asymptotic value of curve for a very long times, interpreted as a value of background, which is automatically subtracted. It is possible to suppress the background calculation and subtraction by specifying negative values of parameter *method*=-1,-2,-3.

TCREG i_{source},$i_{corrected}$,*model,i1,i2* regression of time curve using specified range of points i1,i2.

TCBGR i_{source},$i_{corrected}$ background raise correction is performed automatically (see chapter 4).

NORM i_{source},$i_{normalised}$,*type* normalise a time curve either to unit area or even to a unit mean time

TCINTG i_{source},i_F time curve integration (calculation of integral distributions, e.g. F(t))

TCRND i_{source},$i_{corrected}$,*absolute_relative,sigma* Noise superposition.

CS i_{source},$i_{corrected}$,*shift*[22] vertical shift of a selected curve (for example background subtraction)

TS i_{source},$i_{corrected}$,*shift* time shift of a selected curve (new time origin)

TCS i_{source},$i_{corrected}$,*st,sc* horizontal and vertical shift of a selected curve

TCZ i_{source},$i_{corrected}$ fill zeroes into a part, usually an initial part, of a selected curve (the part is specified by mouse).

TCLIP *i* suppress negative values of a selected curve (replacement by zeroes)

TCEND t_{min},t_{max} set new common time range for all curves (by mouse)

TCDT n_{steps},Δt recalculate all time curves to a constant time step. It is assumed, that the first data point corresponds to the time zero, it means that the original time curve is shifted and only then transformed to the new time basis 0, Δt, 2Δt,…(n_{step}-1) Δt by linear interpolation.

CTO i_{source},i_{copy} copy a time curve

MOMENT *i* calculates moments (area, mean time and variance of the time curve *i*)

There are several operations for the time curve processing making use FFT (Fast Fourier Transform), summarised in a single command FFT. Because the implemented FFT method expects that the number of transformed points is a power of two, the processed curves are automatically extended by zero points to the nearest higher power of two (if you want to reduce the number of the processed points just change the number of observation points in MOP using assignment NT=…). Power spectral densities of resulting functions (PSD) as well as integrals of PSD are always calculated and represented in graphs as a function of frequency. Then it is possible to select a threshold frequency using mouse and Fourier coefficients corresponding to higher frequencies will be suppressed.

- FFT 1,*stimulus function, response, calculated impulse response, freq. cuttoff, reference model, weight of model*
  Deconvolution (given response and stimulus function the impulse response is calculated)
- FFT 2,*stimulus function, impulse response, calculated response, freq. cuttoff*

[22] If a zero *shift* is specied (default value of the parameter) new zero level is located by mouse. The same holds for TS,TCS.

Convolution (response is calculated as a convolution of a stimulus function and impulse response)

- ❑ FFT 3, *function X, function Y, calculated correlation of X,Y, freq. cuttoff*
  Cross-correlation of two time courses (auto correlation is a special case)
- ❑ FFT 4, *noised function, filtered function, method*
  Time course filtering (Wiener filter or simply a higher frequencies cutoff).

Theoretical impulse responses for simple systems can be generated using commands

SERIE $i_{response}, n_{mixers}, t_{mean}$ generalised model of an ideal mixer series; the word generalised means that the number of mixers need not be an integer number, because gamma functions are used instead of factorials in the model definition.

PASERI $i_{response}, n_{mix1}, n_{mix2}, f, \alpha, t_{mean}$ generalised Bishoff's model of n_{mix1} and n_{mix2} units in two parallel series, where f is the relative flowrate in the first stream and α is the relative volume of the vessels in this stream.

The main tool for the RTD modelling are of course lumped parameter models, described by a system of ordinary differential equations of the first order:

$$\frac{dc_i}{dt} = f_i(t, x, c_1, ..., c_N), \qquad \text{i=1,2,…,N}$$

where $c_i(t)$ are concentrations in basic flow units, ideally mixed tanks, t is time and $x(t)$ represents one or more stimulus functions. This system of equations is described by the same simple language, that is used in FEMINA's command interpreter. For example the series of N ideal mixers

$$t_1 \frac{dc_1}{dt} = x(t) - c_1(t),$$

$$t_1 \frac{dc_i}{dt} = c_{i-1}(t) - c_i(t), ...i = 2, ..., N$$

is described by the following commands

```
dcm(1)=(xv(1)-cm(1))/t1
for i=2,n do dcm(i)=(cm(i-1)-cm(i))/t1
yv(1)=cm(n)   yv(2)=yvt(1,time-delay)
```

where *dcm*(*i*),*cm*(*i*) are first derivative and concentration in the *i*-th unit, while *xv*(1) is the first stimulus function, and *yv*(1) is the first response, concentration in the last mixed tank. The last command defines the second output response *yv*(2), calculated as a time delay of the first response, it means that *yv*(2) is a response of the series plus a piston flow unit. While the vectors *cm*(i) and *dcm*(i) are internal, the time courses *xv*(i), *yv*(i) are represented by selected columns of MOP, therefore a model can have maximum 9 stimulus functions and responses.

The description of differential equations is not sufficient – it is also necessary to define initial conditions (important if we want to calculate *impulse* responses) and first of all relationships between model parameters and auxiliary variables, for example t_1, *n*, *delay*. Let us assume that our model has 3 parameters: mean residence time, ratio of piston flow volume and total volume, and of course number of units *n*. The first two parameters are real, while the third is an integer. You can use *any* predefined system variable as a parameter, however for simplicity we shall use two vectors *RP*(i) and *IP*(i) for real and integer parameters of RTD models. Now we can write the initialisation section as

```
real t1,delay   integer i,n
delay=rp(2)*rp(1)     n=ip(3)         t1=(rp(1)-delay)/n
cm(1)=1/t1
```

where the last command describes initial concentration in the first unit, corresponding to the impulse injection of tracer (this command is ignored if the response to a general stimulus function is required).

The text of several typical models is prepared in files with postfix MDT (the same convention as in the previous versions of RTD programs) and these models can be activated by command RMODEL, which enables selection, import and modification of prepared models.

A part of model definition file is also information about model parameters (names, default values, lower and upper bounds, …), specification of solution method (either Euler or Runge Kutta with fixed or variable integration steps) and also directing stimulus and response functions to columns of MOP. All these parameters can be updated either during operation RMODEL in a dialog panel, or later, using commands

INPUT *Number of stimulus functions x, column of x_1, column of x_2,...*
Number of stimulus functions and their position in MOP

OUTPUT *Number of response functions y, column of y_1, column of y_2,...*
Number of response functions and their position in MOP

PARDEF *No.of parameters,p_1-name,p_2-name,...*definition of model parameters. You can use any system variable, for example simple variables (A,B,…), however in predefined models vectors RP(i) and IP(i) are used for real and integer parameters exclusively.

PARSET *p_1, p_2,....* set actual values of parameters

METHOD *M: 0-Euler,>0-RK fixed Δt,<0-variable step, No.of equations, No. of basic steps, Basic time step*
Method of integration: Euler method of the first kind (M=0) is recommended only for algebraic models, while for differential equations the 4-point Runge Kutta (M≠0) is always better. You can select either a fixed time step of integration (M>0) which is M-times less than the basic time step (the time step in MOP), or variable integration time step (M<0), adjusted so that the relative accuracy 10**M will be achieved.

The model calculates impulse response (stimulus functions suppressed, initial cond. respected)
IMPULS *method, no of equations, no of basic steps, basic time step*
or calculates responses to specified stimulus functions (with zero initial conditions)
CONVOL *method, no of equations, no of basic steps, basic time step*
or integrates the model without zeroing initial condition and with stimulus functions
INTEGR *method, no of equations, no of basic steps, basic time step.*

System variables and functions important for RTD analysis and lumped parameter models:

A,B,…,Z	user variables (I,J,K,L,M,N are integers)
C1(i),…,C10(i)	columns of MOP (C1(i) is always interpreted as time)
NT,DT	number of points (rows of MOP) and time step
METH	method of integration (=0 Euler, >0 RK fixed step, <0 RK variable integ.step)
RP(i),IP(i),NP	model parameters (real/integer) and number of parameters

Remark: Vectors RP(i), IP(i) are duplicated by equivalence, therefore you cannot define for example a model with two parameters, the first real and the second integer, as RP(1) and IP(1), but RP(1) and IP(2).

PMIN(i),PMAX(i)	prescribed minimum and maximum values of model parameters
REGR(i)	zero values suppresses identification (see later) of i-th model parameter
CM(i),DCM(i)	values of concentrations and their derivatives in the i-th flow unit at current time
XV(i),YV(i)	actual values of stimulus functions and responses.
NINP,VINP(i)	number of stimulus functions and corresponding columns in MOP
NOUT,VOUT(i)	number of responses and corresponding columns of MOP.
CVT(i,time)	is an interpolating function, passing value of a function in the i-th MOP column.

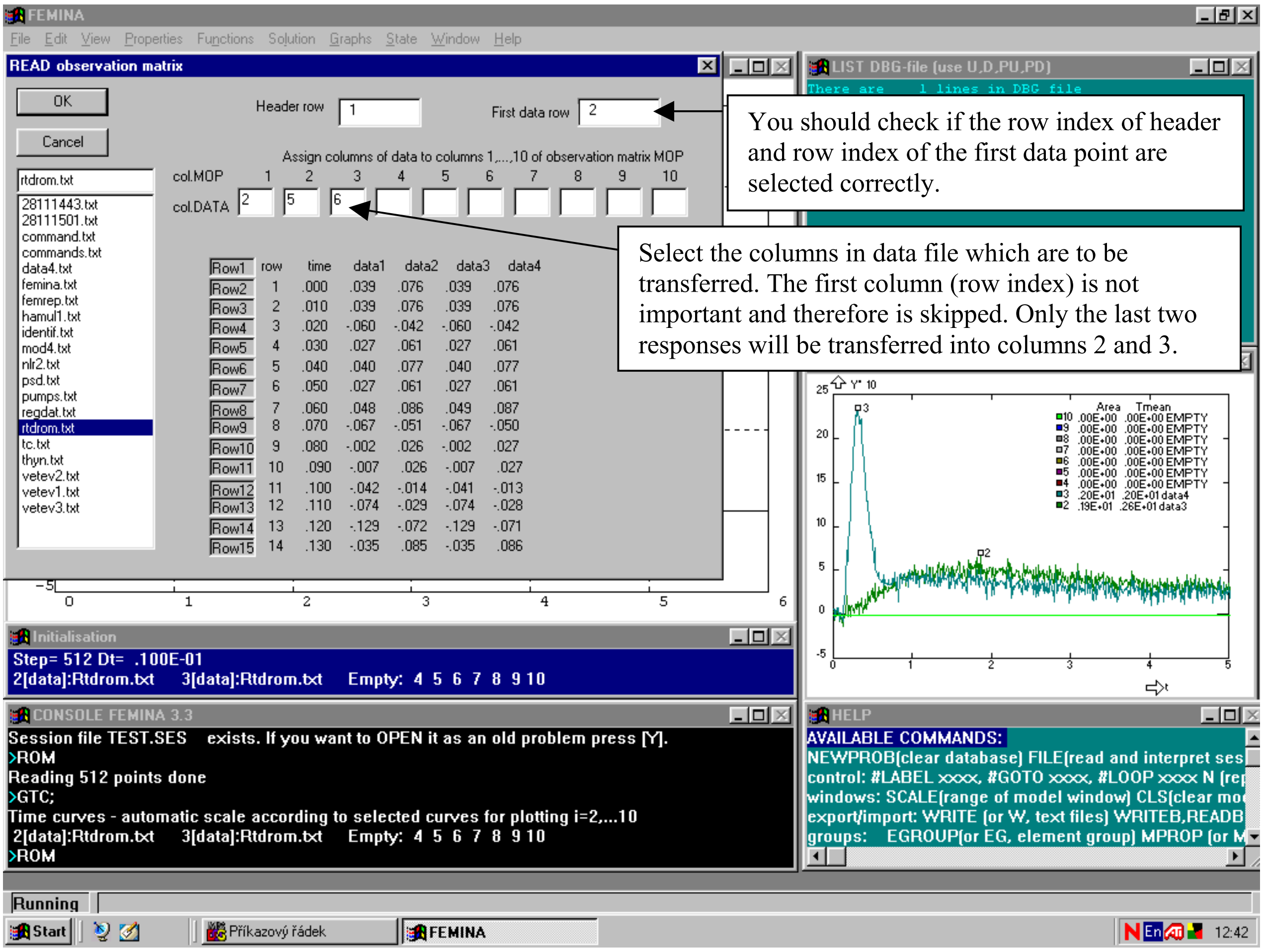
2.4.1. Response processing

Let us assume, that 512 time points of four measured responses are in the following file RTDROM.TXT (the format can be more or less arbitrary, with or without header rows):

```
 row        time      data1     data2     data3     data4
    1       .000      .039      .076      .039      .076
    2       .010      .039      .076      .039      .076
…
   511     5.100      .011      .024      .304      .254
   512     5.110      .026      .074      .122      .135
```

This file can be transferred to the matrix of observation points using command REOMAT (ROM is a synonym which can be used as well)

```
REOMAT rtdrom.txt,1,2,3,2,1,5,2,6,3;
```



Header row 1 First data row 2

Assign columns of data to columns 1,...,10 of observation matrix MOP

col.MOP	1	2	3	4	5	6	7	8	9	10
col.DATA	2	5	6							

Row1	row	time	data1	data2	data3	data4
Row2	1	.000	.039	.076	.039	.076
Row3	2	.010	.039	.076	.039	.076
Row4	3	.020	-.060	-.042	-.060	-.042
Row5	4	.030	.027	.061	.027	.061
Row6	5	.040	.040	.077	.040	.077
Row7	6	.050	.027	.061	.027	.061
Row8	7	.060	.048	.086	.049	.087
Row9	8	.070	-.067	-.051	-.067	-.050
Row10	9	.080	-.002	.026	-.002	.027
Row11	10	.090	-.007	.026	-.007	.027
Row12	11	.100	-.042	-.014	-.041	-.013
Row13	12	.110	-.074	-.029	-.074	-.028
Row14	13	.120	-.129	-.072	-.129	-.071
Row15	14	.130	-.035	.085	-.035	.086

```
There are     1 lines in DBG file
```

		Area	Tmean	
🟩	10	.00E+00	.00E+00	EMPTY
🟦	9	.00E+00	.00E+00	EMPTY
■	8	.00E+00	.00E+00	EMPTY
□	7	.00E+00	.00E+00	EMPTY
■	6	.00E+00	.00E+00	EMPTY
🟪	5	.00E+00	.00E+00	EMPTY
🟥	4	.00E+00	.00E+00	EMPTY
■	3	.20E+01	.20E+01	data4
🟩	2	.19E+01	.26E+01	data3

```
Step= 512 Dt=  .100E-01
2[data]:Rtdrom.txt     3[data]:Rtdrom.txt     Empty:  4  5  6  7  8  9 10
```

```
Session file TEST.SES    exists. If you want to OPEN it as an old problem press [Y].
>ROM
Reading 512 points done
>GTC;
Time curves - automatic scale according to selected curves for plotting i=2,...10
 2[data]:Rtdrom.txt     3[data]:Rtdrom.txt     Empty:  4  5  6  7  8  9 10
>ROM
```

```
AVAILABLE COMMANDS:
NEWPROB(clear database) FILE(read and interpret ses
control: #LABEL xxxx, #GOTO xxxx, #LOOP xxxx N (rep
windows: SCALE(range of model window) CLS(clear mo
export/import: WRITE (or W, text files) WRITEB,READB
groups:    EGROUP(or EG, element group) MPROP (or M
```

You should check if the row index of header and row index of the first data point are selected correctly.

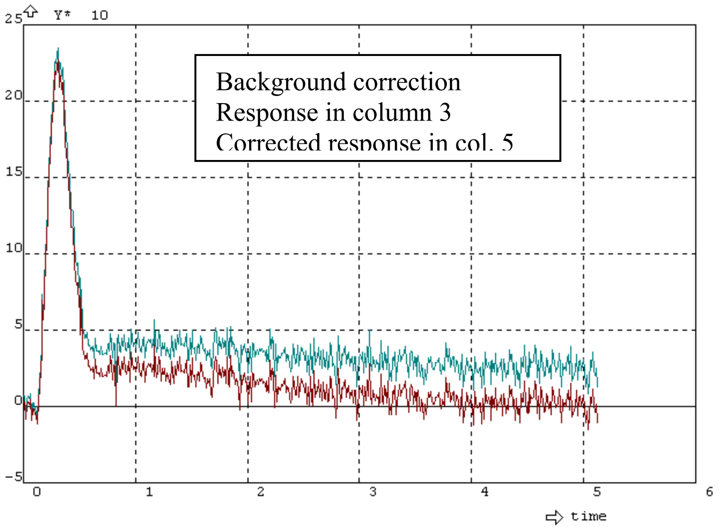
Select the columns in data file which are to be transferred. The first column (row index) is not important and therefore is skipped. Only the last two responses will be transferred into columns 2 and 3.

The data can be visualised either in form of graphs (command GTC) or as a table (command TCL).

We have extracted only the last two measured responses, and it is obvious, see figure, that both these curves are characterised by a rising background which is to be eliminated. This can be achieved by the following commands TCBGR, processing columns 2 and 3, and results, iteratively corrected responses, will be placed into columns 4 and 5, respectively.

```
TCBGR 2,4;
TCBGR 3,5;
```

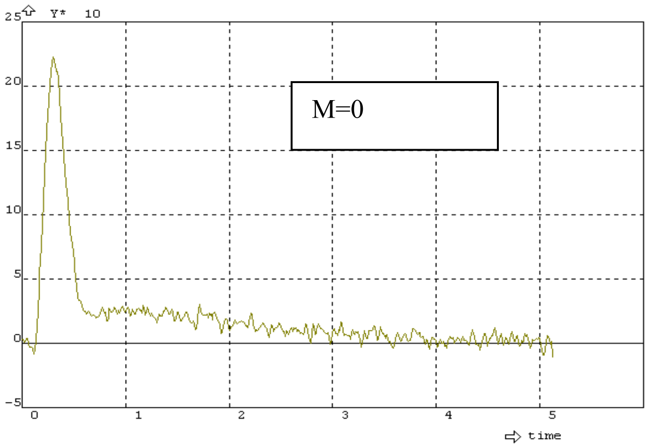
The corrected response in the column 5 is shown in the following figure (use TCP 5 for plotting):

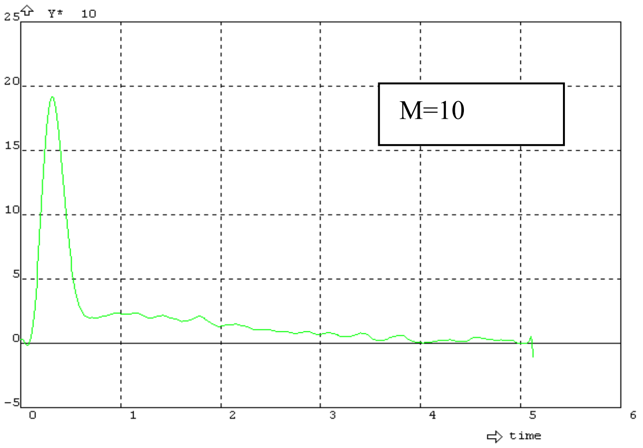


Using command SMOOTH a local smoothing can be performed either by local regression or by median method. The smoothing is controlled by one parameter M. For M=0 a linear and for M>0 a quadratic approximation is identified by regression of 2M+1 points, taking into account M points before and M points behind the processed point (the case M=0 is an exception, the linear smoothing is always identified from 4 points). A negative M would activate the median smoothing, again with 2|M|+1 points.

```
SMOOTH 5,6,0,1,511;
SMOOTH 5,10,10,1,511;
```

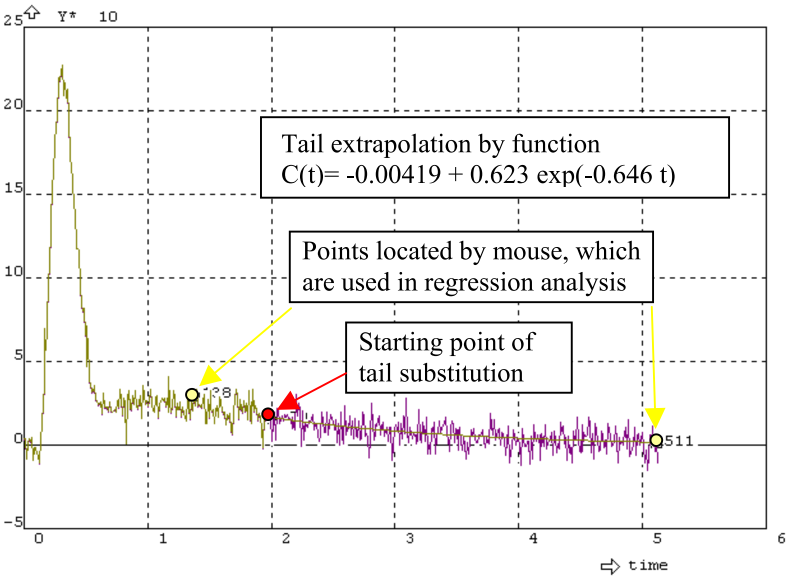
Smoothing response 5, results are in columns 6 (linear smoothing M=0) and 10 (quadratic smoothing M=10) respectively. The range of smoothed points 1 to 511 has been identified by mouse.





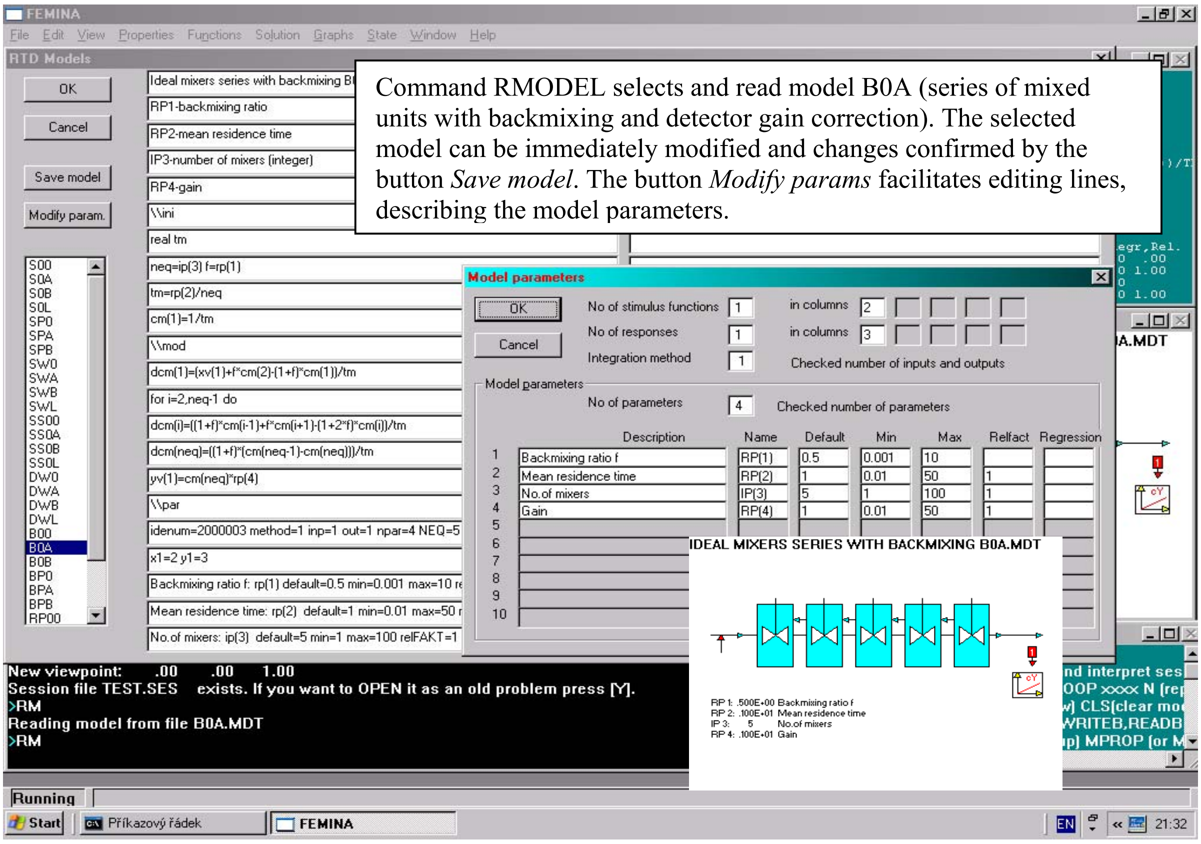
The last example demonstrates the exponential tail approximation of the response 5. Result will be in the column 6, points 138 up to 511 are used for nonlinear regression and tail approximation starts from the point 200. The identified background level is very low (-0.00419) which confirms, that the background raise correction was quite good.

```
TAIL 5,6,1,138,511,200;
```



2.4.2. Lumped parameter models of RTD – series of CSTR with backmixing

```
RMODEL B0A;
```



```
Ideal mixers series with backmixing B
RP1-backmixing ratio
RP2-mean residence time
IP3-number of mixers (integer)
RP4-gain
\\ini
real tm
neq=ip(3) f=rp(1)
tm=rp(2)/neq
cm(1)=1/tm
\\mod
dcm(1)=(xv(1)+f*cm(2)-(1+f)*cm(1))/tm
for i=2,neq-1 do
dcm(i)=((1+f)*cm(i-1)+f*cm(i+1)-(1+2*f)*cm(i))/tm
dcm(neq)=((1+f)*(cm(neq-1)-cm(neq)))/tm
yv(1)=cm(neq)*rp(4)
\\par
idenum=2000003 method=1 inp=1 out=1 npar=4 NEQ=5
x1=2 y1=3
Backmixing ratio f: rp(1) default=0.5 min=0.001 max=10 re
Mean residence time: rp(2)  default=1 min=0.01 max=50 r
No.of mixers: ip(3)  default=5 min=1 max=100 relFAKT=1
```

Model parameters

OK | Cancel

No of stimulus functions 1 in columns 2

No of responses 1 in columns 3

Integration method 1

Checked number of inputs and outputs

Model parameters

No of parameters 4 Checked number of parameters

	Description	Name	Default	Min	Max	Relfact	Regression
1	Backmixing ratio f	RP(1)	0.5	0.001	10		
2	Mean residence time	RP(2)	1	0.01	50	1	
3	No.of mixers	IP(3)	5	1	100	1	
4	Gain	RP(4)	1	0.01	50	1	
5							
6							
7							
8							
9							
10							

IDEAL MIXERS SERIES WITH BACKMIXING B0A.MDT

```
RP 1:  .500E+00 Backmixing ratio f
RP 2:  .100E+01 Mean residence time
IP 3:        5  No.of mixers
RP 4:  .100E+01 Gain
```

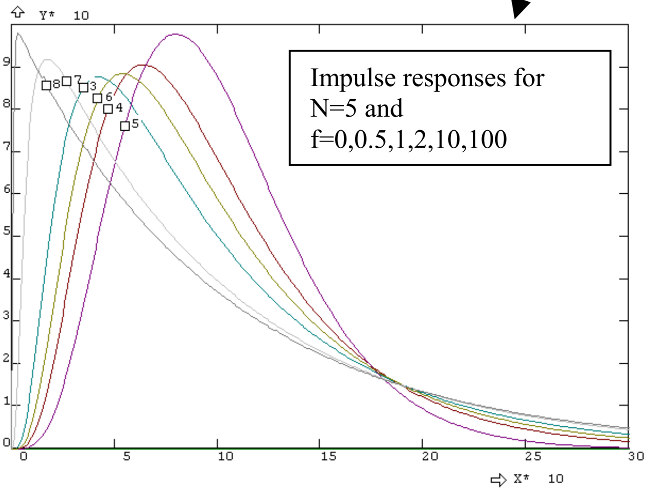
```
New viewpoint:     .00      .00     1.00
Session file TEST.SES    exists. If you want to OPEN it as an old problem press [Y].
>RM
Reading model from file B0A.MDT
>RM
```

Command RMODEL selects and read model B0A (series of mixed units with backmixing and detector gain correction). The selected model can be immediately modified and changes confirmed by the button *Save model*. The button *Modify params* facilitates editing lines, describing the model parameters.

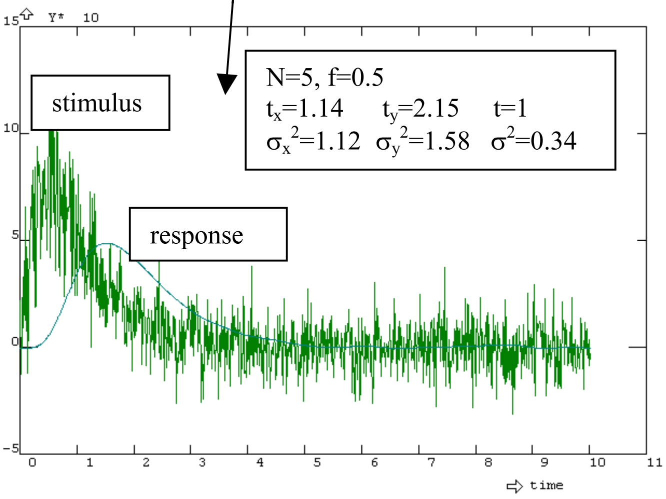
```
IMPULS -2,5,1000,.100E-01;
COPYTO 3,4;
PARSET 0,1,5,1;
…
IDMSER 2,2,1;
TCRND 2,2,1,.3;
TCRND 2,2,0,.100E+00;
PARSET .5,1,5,1;
CONVOL -2,5,1000,.100E-01;
MOMENT 2;
MOMENT 3;
```

Calculation of impulse response with default parameters. Result is in the 3rd column (could have been redirected using OUTPUT command). Instead of OUTPUT we simply copy the result to the 4,5,…th column for different values of backmixing parameter f, specified by PARSET.

Stimulus function in the 2nd column is generated by command IDMSER (series of 2 tanks) and noised. Response is calculated for f=0.5 and accuracy of computations is checked using moments: You see that the first moments agree well (t_x+t=t_y) but the sum of variances is not so good



Impulse responses for N=5 and f=0,0.5,1,2,10,100



stimulus

response

N=5, f=0.5

t_x=1.14 t_y=2.15 t=1

σ_x^2=1.12 σ_y^2=1.58 σ^2=0.34

2.4.3. FFT convolution, deconvolution, correlations

FFT will be used for identification of a system formed by two parallel series of mixed tanks (3+9), having the mean residence time 1. As a stimulus function the impulse response of 3-tanks having mean time 3 will be used (it should be noted that it is a rather difficult problem, because the stimulus function is 3-times longer than the identified impulse response resulting to a bad conditionality of the solution). The „theoretical“ response will be calculated by convolution using FFT, and a noise will be superposed to stimulus and also to the response function. Using these noised signals the impulse response will be identified using FFT with regularization.

```
C* Opening TEST date:12.11.02 at 11:43
TSTEP  500,0.02;
PASERI 3,3,9,.500E+00,3,1;
```

Impulse reponse of paralel series (3rd column) N_1=3, N_2=9, f=0.5, α=3

```
IDMSER 2,3,3;
```

Stimulus function in the 2nd column. Serie N_1=3 with mean time 3.

```
TCFFT  2,2,3,4,1;
```

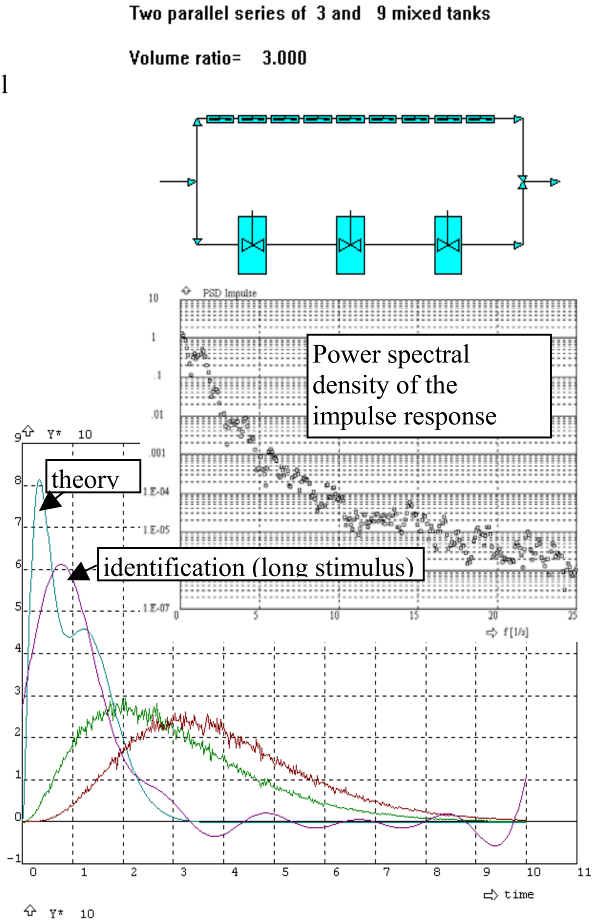
Convolution (2) of stimulus function (2nd column) and impulse response (3rd column). Resulting response will be in 4th column. The last parameter (0 - 1) sets the filtration level: if it is 1 the PSD graph will be shown and the treshold frequency can be selected by mouse.

```
TCRND  2,2,1,.05;
TCRND  4,4,1,0.05;
```

Noise added to the stimulus and response functions

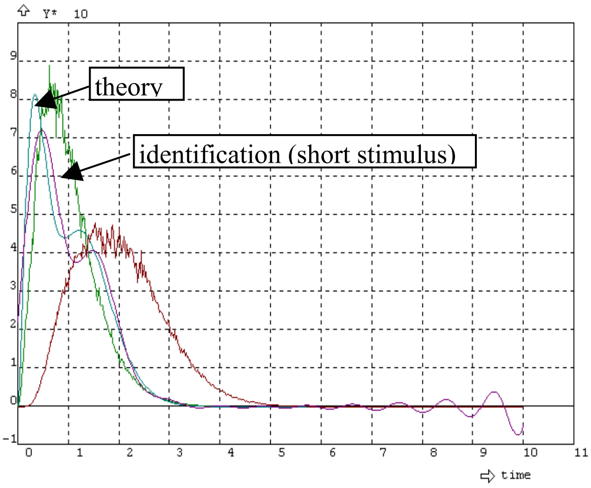
```
TCFFT  1,2,4,5,1,4,.100E-03;
```

Impulse response (5th column) calculated from stimulus (2nd column) and response (4th column). Filtration of noise using PSD graph (filtration level 1). Regularization is applied using a regularisation model of 4 tanks in serie, with regularisation weight w=10^{-4}.



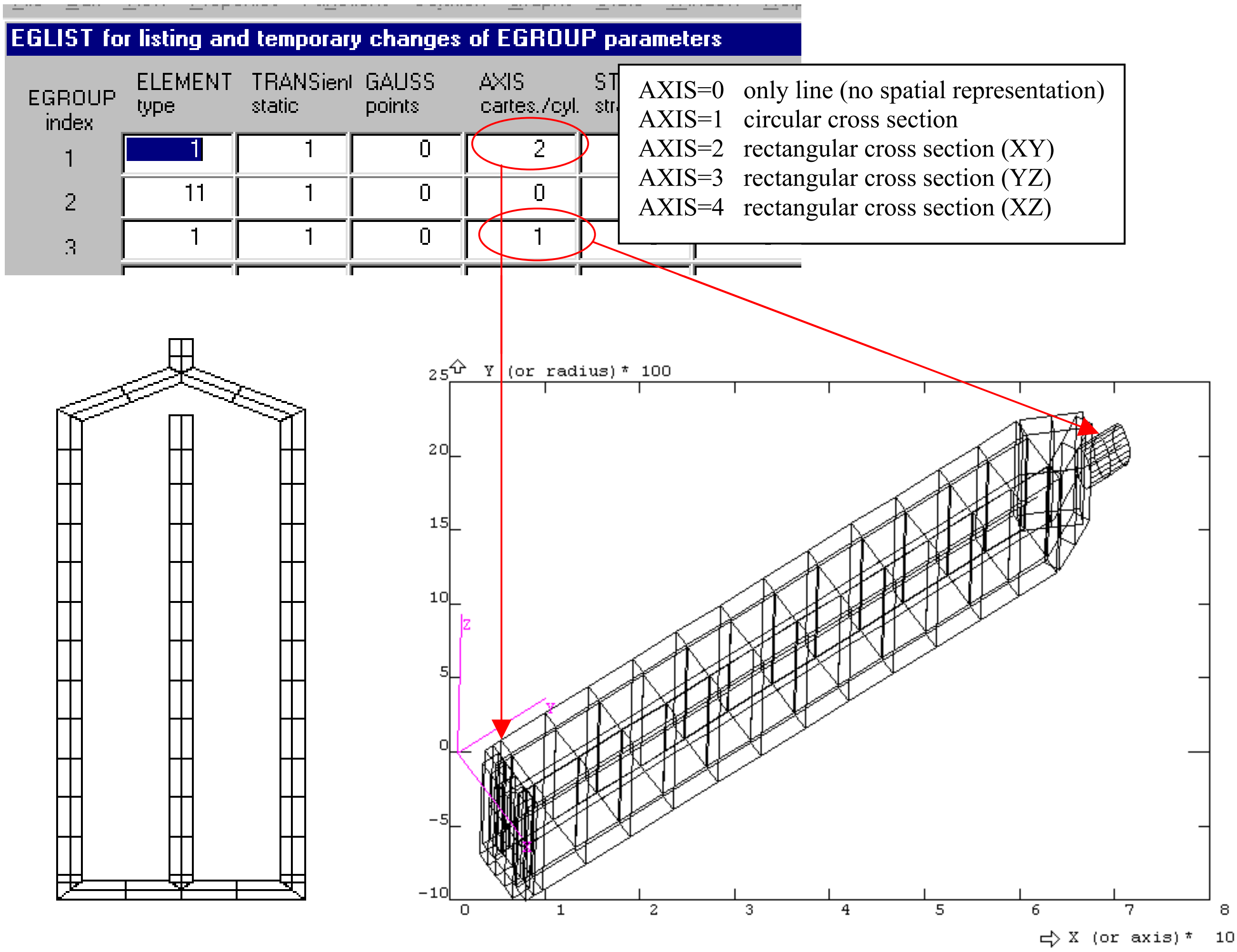
The whole procedure is repeated, this time for a shorter stimulus function (mean time 1). Accuracy is increased substantially.

```
IDMSER 2,3,1;
TCFFT  2,2,3,4,1;
TCRND  2,2,1,0.05;
TCRND  4,4,1,0.05;
TCFFT  1,2,4,5,1,4,.100E-03;
```



2.5. Detectors

FEM models created by FEMINA can be always considered as 3D models describing spatial distribution of tracer *c*(*t,x,y,z*) even if only 1D elements (for example pipes) or 2D elements (FLOW2D) are used. For example the model of continuous ohmic heater formed from 1D elements PIPE2D (elements are abscissas) has its 3D extensions, depending upon the specified cross section of pipes (parameters D and AREA in RCONST) and upon the parameter AXIS of EGROUP.



EGLIST for listing and temporary changes of EGROUP parameters

EGROUP index	ELEMENT type	TRANSienl static	GAUSS points	AXIS cartes./cyl.	ST str
1	1	1	0	2	
2	11	1	0	0	
3	1	1	0	1	

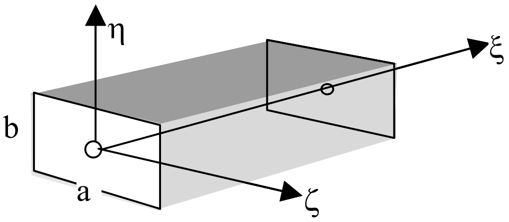
AXIS=0	only line (no spatial representation)
AXIS=1	circular cross section
AXIS=2	rectangular cross section (XY)
AXIS=3	rectangular cross section (YZ)
AXIS=4	rectangular cross section (XZ)

Geometry of spatial extension is defined by nodal coordinates (2 nodes of element PIPE2D) and by two RCONST parameters, D (hydraulic diameter) and AREA (of cross section). If the element is a circular pipe the EGROUP parameter AXIS=1 and diameter is D. In the case that the cross section of pipe is rectangular AXIS>1, and the parameters D, AREA define dimensions of sides according to

$$b = \frac{AREA}{D}(1+\sqrt{1-\frac{D^2}{AREA}}) \qquad \text{and} \qquad a = \frac{AREA}{b}$$

as follows from the definition of hydraulic diameter $D = \frac{4AREA}{perimeter}$.

Therefore the element PIPE2D looks in the local coordinate system ξ,η,ζ like a prismatic bar as soon as AXIS>1



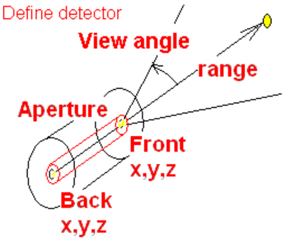
Remark: In fact there are two roots a_1,b_1, a_2,b_2 satisfying the area and the hydraulic diameter constraints. To avoid the ambiguity, the sign of parameter AREA is used. If AREA>0 then the dimension *a* is the smallest of the roots a_1,a_2 and reversely, if AREA<0 the side *a* is the greatest of a_1,a_2.Seems to be complicated? You need not worry about this, because dimension are specified by the command RCROSS and the sign encoding and decoding is carried out automatically. Therefore you can specify dimensions a,b always according to the fig.left.

The axis ξ is always the element axis fully determined by Cartesian coordinates of nodal points X,Y,Z, and axis η and ζ are located in the plane which is perpendicular to the element axis. FEMINA considers only three possible rotations of cross section around the axis ξ: AXIS=2 – axis η lies in the plane XY, AXIS=3 – axis η lies in the plane YZ, and AXIS=4 – axis η lies in the plane XZ.

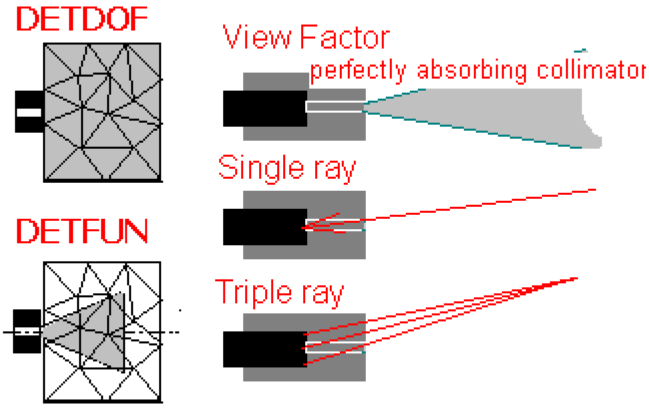
3D extension of 2D finite elements (e.g. PLANE2D or FLOW2D) is straightforward. For AXIS=1 (cylindrical coordinate system) elements defined in XY plane are represented by rings with axis X, while for AXIS=0 the region in the XY plane is extruded in the Z direction by a value given by RCONST parameter (height).

The 3D extension is a region, where a distribution of concentration (for example DOF(CN,3,XX,YY,ZZ)) or generally any function can be defined. In the same way, that is as a user function, the media attenuation is defined (more specifically attenuation is prescribed as a constant multiplied by a function of X,Y,Z).

This concentration field can be observed by one or more collimated detectors of radiation, characterised by position, dimensions and attenuation of a cylindrical collimator (see command DETDEF).



Responses of detectors can be evaluated by several different algorithms (view factor, single and triple ray methods, see theoretical manual of FEMINA) in the following steps:



If you specify the last parameter Row=0 all time steps from OUT file will be processed, Row>0 processes only current distribution.

1. Results of finite element analysis corresponding to time t=0, Δt, 2Δt,… are read from the file OUT into the zone of initial conditions (this transfer corresponds to the command LTC). It is also possible to skip reading from the OUT-file and process only the tracer distribution in the current time step – this option is useful for processing imported data for example from FLUENT.
2. Loop across all elements. Contribution of element to the detector response is calculated by numerical integration of selected DOF, usually concentration CN,CD, or CA (response to a unit concentration in integration points is evaluated by selected algorithm).
3. Evaluated responses of selected detectors are placed into specified columns of MOP (matrix of observation points).

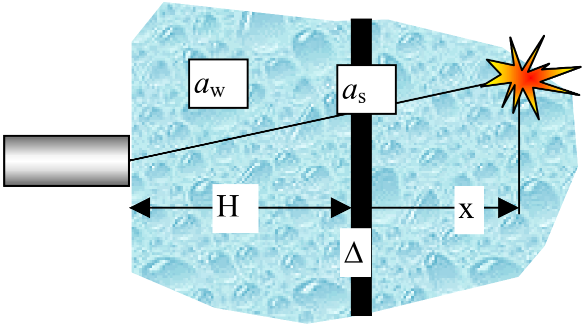
This run time consuming operation is carried out in the independent program RUNDET.EXE, called by the command DETDOF.

There exists also a different way of processing, where it is not a DOF, but a user defined function which describes the tracer distribution. Processing differs first of all by the integration method. Instead of integration over elements, the integration proceeds in regions defined by view angles and view ranges of specified detectors, and integrands can be any spatial functions. In the case that the functions require evaluation of concentrations, the Cartesian boxing has to be performed (see operation BOX). This second approach activated by the command DETFUN is more effective than the previous one as soon as the view range of detectors is relatively small.

Attenuation of media is also defined by a function, however this approach is not so straightforward, first of all because it would not be very effective to integrate attenuation of a radiation ray numerically along the whole ray (from a general point x,y,z to the detector), especially if there are thin but strongly absorbing metallic plates, walls of vessels, baffles and so on. It is in principle necessary to integrate

$$\int_0^L a(x)dx = a_m L \tag{1}$$

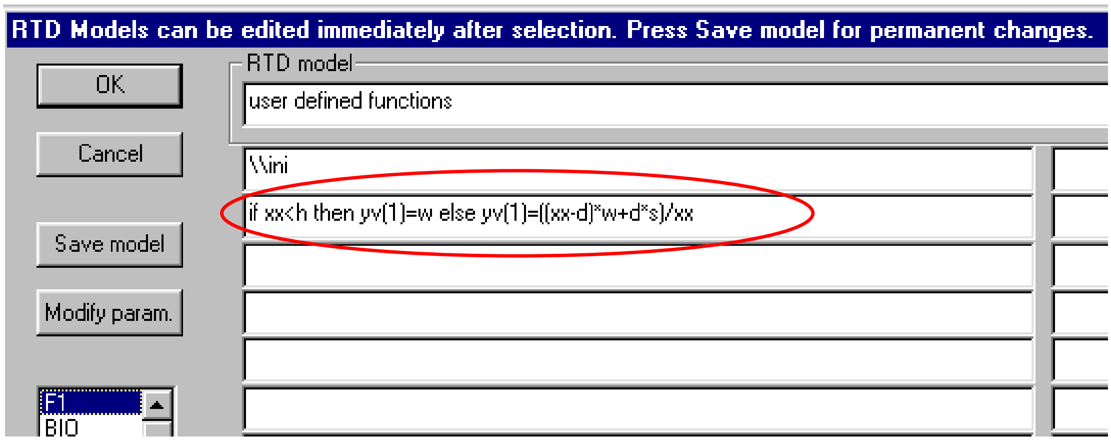
where a_m is the mean attenuation which depends on the mutual position of detector and the source of radiation at *x,y,z*, therefore a_m is not generally a constant but a function of *x,y,z*. And this is just this function which must be defined by user. Let us consider as an example a single metallic sheet between detector and the source of radiation



Therefore the function describing media attenuation should be written as

if x<H then am=aw else am=((x-d)*aw+d*as)/x

How to define this function in FEMINA? User defined functions are denoted by numbers (indices), and indices in the range –10 to 60 are reserved for simple functions defined either by a table (CURDEF) or by a single arithmetic expression (FUNDEF). Both these ways are not suitable for more complicated algorithms of attenuation and specific functions (with indices 101,102,…,105) must be defined using the script language in the same way as RTD models described in previous chapter 3. Besides standard RTD models there exists also a model F1 (use command RM and select F1 from list of models). Script, calculating mean attenuation as a function of coordinates XX,YY,ZZ, must be written into the \\INI section, section \\MOD and \\PAR can be omitted, see the following figure.


RTD Models can be edited immediately after selection. Press Save model for permanent changes.
OK
Cancel
Save model
Modify param.
F1
BIO
RTD model

user defined functions

```
\\ini
if xx<h then yv(1)=w else yv(1)=((xx-d)*w+d*s)/xx
```

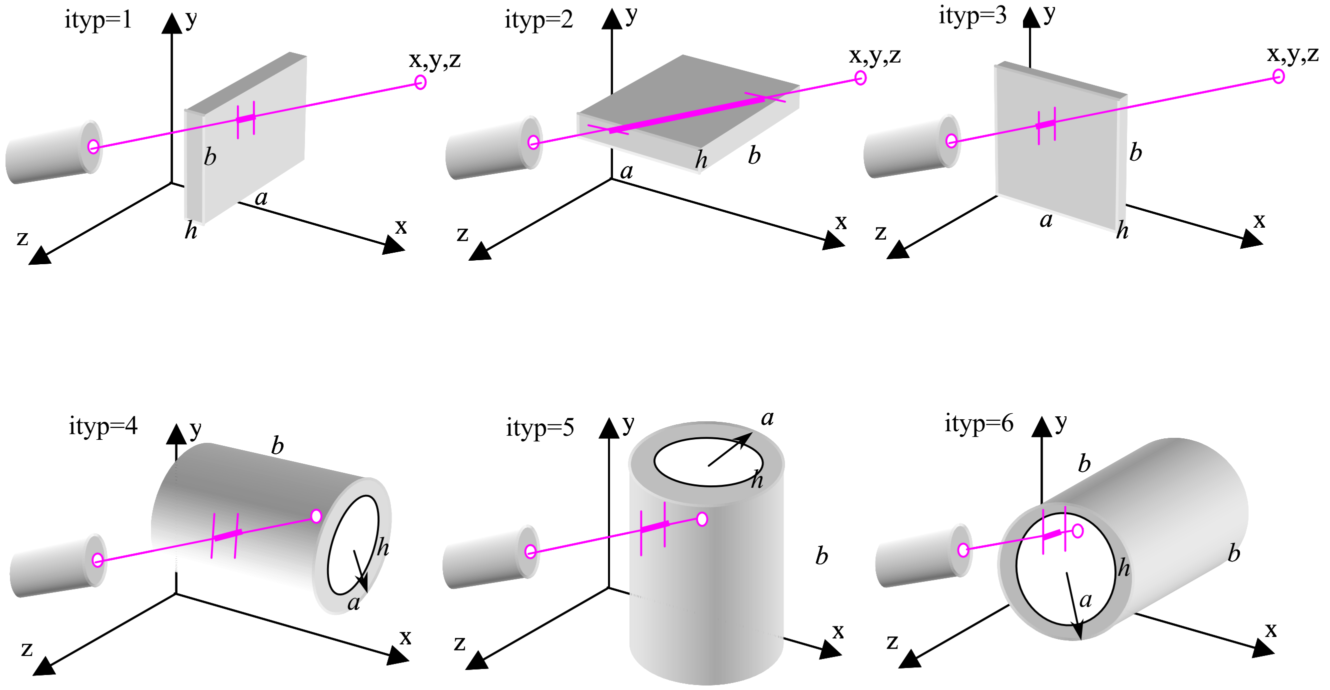
System variable yv(1) is used as the result returned by function 101 (variable yv(2) is function 102 …), H, D,W, S are user variables: distance, thickness, attenuation of water and steel. Use e.g. commands H=0.2 D=0.002 W=5.7 S=100 to assign proper values.

From now on the user function 101 is defined and can be referred in the commands DETDOF or DETFUN as an attenuation function.

This example was simple, however programming of more complicated arrangements is a more difficult task. To make the programming of attenuation easier the user function

ALEN(detector,x,y,z, xc,yc,zc, a,b,h, ityp)

returning thickness of a layer of material through which a ray is passing has been designed. Geometry of obstacle is selected by the parameter *ityp*: ityp 1-3 is a rectangular plate (centre *xc,yc,zc*, dimension *a* x *b* and thickness *h*), ityp=4-6 is a cylindrical shell (centre *xc,yc,zc*, radius *a*, length *b* and thickness *h*) and ityp=7-9 is a full cylinder.



The function ALEN enables to calculate for any ray (and for any collimated detector) the distance corresponding to the passage through metal, through water, air etc. and to calculate mean value of attenuation coefficient as a weighted average of attenuation coefficients (taken from tables) with weights given by calculated distances. Thickness of material is related to the length of ray calculated as a distance from the front face of detector to the point *x,y,z,* returned by function

RLEN(detector,x,y,z).

We shall try to demonstrate the whole procedure on example of mixer – cylindrical vessel filled by water with baffles, shaft and impeller, made from steel. We design the RTD model F1 in such a way that it can be used not only for one detector, but for multiple detectors in arbitrary positions (this is easy, because FEMINA sets the value of variable I to the index of currently processed detector[23]).

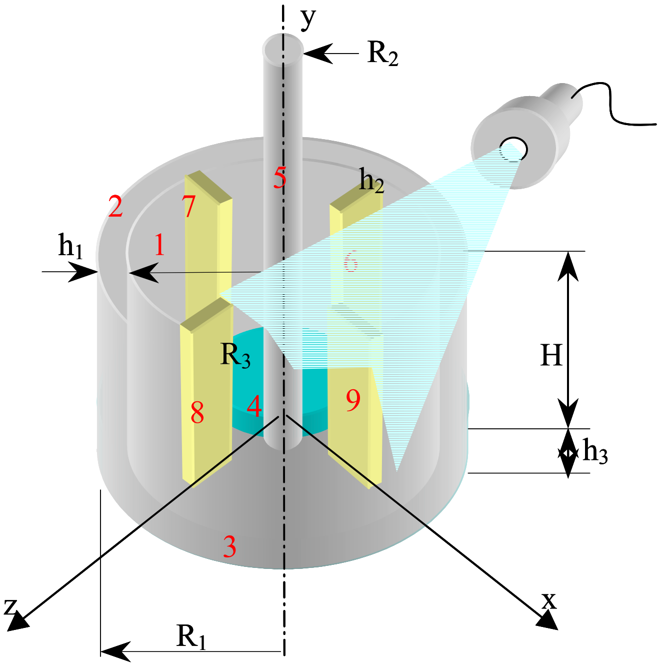
First step is to prepare a list of obstacles
having significant influence upon
attenuation of radiation.

1. water (cylinder height H, radius R_1-h_1)
2. cylindrical shell (height R, radius R_1, wall h_1)
3. bottom (cylinder height h_3, radius R_1)
4. impeller (cylinder height h_3, radius R_3)

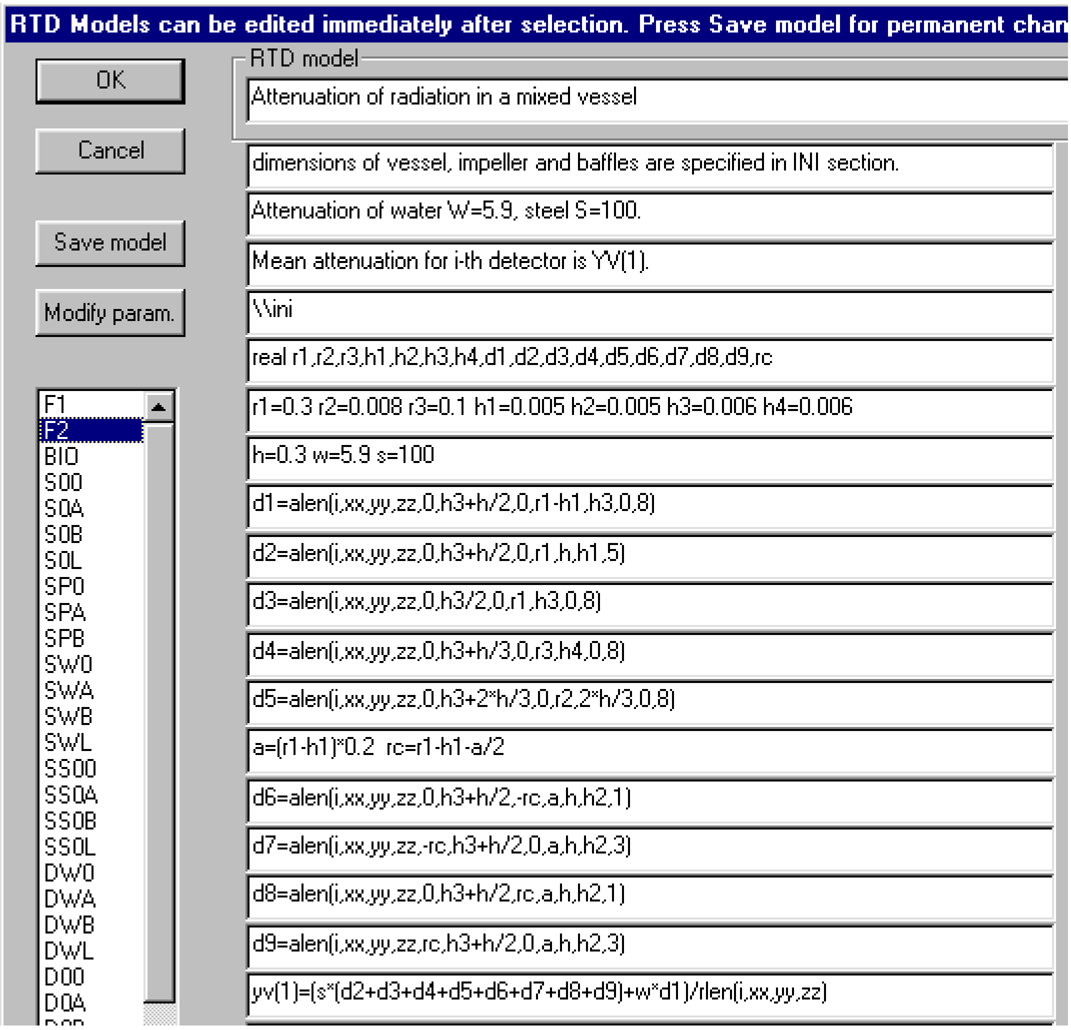
[23] One letter system variables (A,B,…,Z) are reserved for user as auxiliary variables and FEMINA never changes their values. Variable I is an exception, we have seen that for example in the command TCF value I is row index, and now during processing DETDOF or DETFUN the value I denotes processed detector.

5. shaft (cylinder height h_4, radius R_2)
6. baffle (plate Hxb, thickness h_2)
7. baffle (plate Hxb, thickness h_2)
8. baffle (plate Hxb, thickness h_2)
9. baffle (plate Hxb, thickness h_2)

It is assumed that the remaining length of rays (with the exception of items 1-9) is air and its attenuation can be neglected.



Using the list of obstacles it is possible to prepare a new RTD model, for example a model with the name F2



RTD Models can be edited immediately after selection. Press Save model for permanent chan

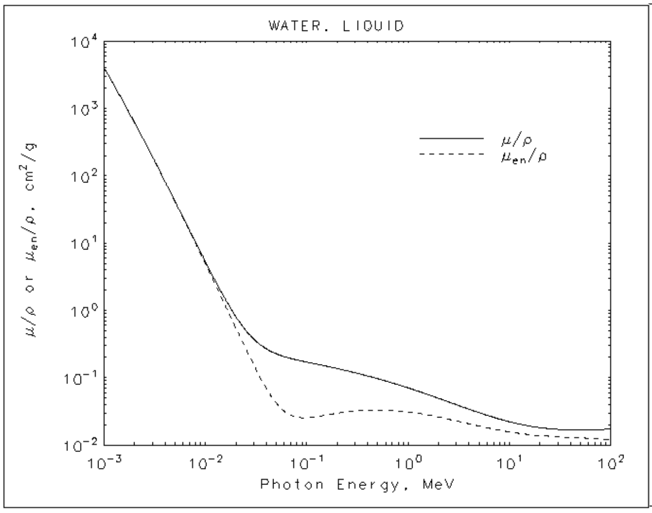
```
Attenuation of radiation in a mixed vessel
dimensions of vessel, impeller and baffles are specified in INI section.
Attenuation of water W=5.9, steel S=100.
Mean attenuation for i-th detector is YV(1).
\\ini
real r1,r2,r3,h1,h2,h3,h4,d1,d2,d3,d4,d5,d6,d7,d8,d9,rc
r1=0.3 r2=0.008 r3=0.1 h1=0.005 h2=0.005 h3=0.006 h4=0.006
h=0.3 w=5.9 s=100
d1=alen(i,xx,yy,zz,0,h3+h/2,0,r1-h1,h3,0,8)
d2=alen(i,xx,yy,zz,0,h3+h/2,0,r1,h,h1,5)
d3=alen(i,xx,yy,zz,0,h3/2,0,r1,h3,0,8)
d4=alen(i,xx,yy,zz,0,h3+h/3,0,r3,h4,0,8)
d5=alen(i,xx,yy,zz,0,h3+2*h/3,0,r2,2*h/3,0,8)
a=(r1-h1)*0.2  rc=r1-h1-a/2
d6=alen(i,xx,yy,zz,0,h3+h/2,-rc,a,h,h2,1)
d7=alen(i,xx,yy,zz,-rc,h3+h/2,0,a,h,h2,3)
d8=alen(i,xx,yy,zz,0,h3+h/2,rc,a,h,h2,1)
d9=alen(i,xx,yy,zz,rc,h3+h/2,0,a,h,h2,3)
yv(1)=(s*(d2+d3+d4+d5+d6+d7+d8+d9)+w*d1)/rlen(i,xx,yy,zz)
```

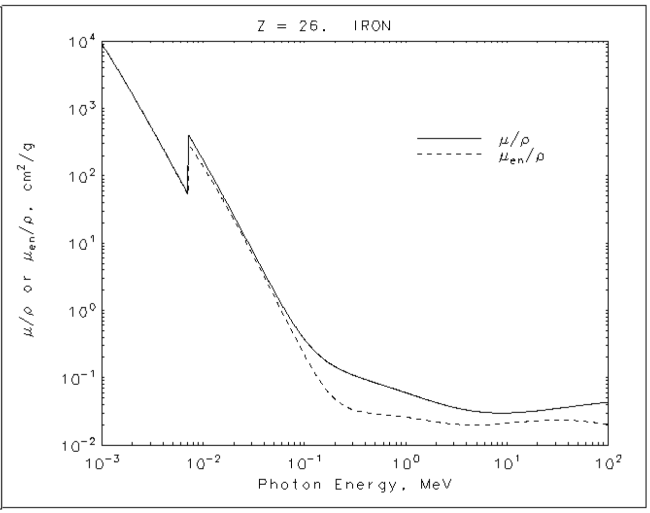
Again, reading this model (RM F2) the function number 101 describing attenuation will be defined. Remark: The functions ALEN and RLEN are useful not only for evaluation of CFD results, but also for calculation spatial distribution of density using methods of transmission tomography, see chapter 5.

Coefficient of attenuation *a* [1/m] is usually calculated from the mass attenuation coefficient *a*/ρ [m^2/kg] which is independent of material density. The mass attenuation is derived from the total cross section for an interaction by the photon taking into account atomic photoeffect, Rayleigh and Compton scattering and electron-positron production. Data which are freely available for most elements as well as for mixtures are available at Webb page of NIST

http://physics.nist.gov/PhysRefData/XrayMassCoef/ComTab/pmma.html

The relationship between energy of radiation and the mass attenuation coefficient (symbol μ is used instead of *a*) and mass energy absorption coefficients (μ_{en}) are summarised in tables and graphs. Examples for attenuation coefficients of water and iron are shown in the following figure:





Important commands

DETDEF (DD) *Detector number, x-front,x-back, y-front,y-back, z-front,z-back, aperture, view angle, range, attenuation*

definition of detector position using x,y,z coordinates of centreline points at the front and back of collimator hole. View angle and the range define a conical region, where integration is performed if the method DETFUN is chosen. Attenuation concerns material of collimator and this parameter is used only if the 1-ray or 3-rays algorithms of collimation are selected.

DETPLOT *Number of detectors* plots detectors

DETDOF (DETD) *dof (CN,CD,CA), method (1-view factor, 2-view factor with near distance correction, 3,4-single ray without and with correction, 5,6-triple ray without and with correction, 7-spherical detector), media attenuation, attenuation function, no.of detectors, detector 1, column of response 1, detector 2, column of response 2,...*

Responses of selected detectors for previously defined time steps (command T) calculated by RUNDET.EXE by integration of specified DOF at all elements. Parameter *media attenuation* is a constant multiplied by a *attenuation function* as soon as this function is defined.

DETFUN (DETF) *index of function, method (1-view factor, 2-view factor with near distance correction, 3,4-single ray without and with correction, 5,6-triple ray without and with correction, 7-spherical detector), media attenuation, attenuation function, no.of detectors, detector 1, column of response 1, detector 2, column of response 2,...*

Responses of selected detectors are calculated by integration of specified function within the detector view angles.

2.6. Optimisation and identification of a mathematical model

Previous examples have dealt with the solution of mathematical models. All these models are parameterised in the sense that the model predictions depend upon system variables and in the case that the model prediction can be compared with experimental data it is possible to optimise selected model parameters using operations OPTIMA or SOMA.

Examples of flow models:

- Identification of a RTD model described as a regression function with a set of unknown coefficients. These regression functions could represent impulse responses which can be identified by comparing with reference data in MOP (matrix of observation points). In this case the standard Levenberg Marquardt method is used and the range of available models is fixed.
- Identification of a lumped parameter model described by a system of ordinary differential equations. Model parameters are in this case flowrates, volumes of tanks, numbers of flow units and coefficients describing characteristics of detectors. Model predictions are responses to a simulated tracer injection, and these responses are automatically stored in MOP (matrix of observation points).
- Optimisation of 1D finite element models of pipelines or 2D models of simple flow systems, where optimised parameters are usually diameters of pipes and volumes of tanks (parameters RCONST) in 1D models, or geometry, for example diameters of nozzles, in 2D models. The model responses are in these cases time courses of tracer concentrations evaluated either at a nodal point or as an integral across a region. These time courses must be transferred into the MOP using operations LOADT or GTIME.

As a reference data the experimentally obtained responses are usually used, however, a numerical experiment can be used as well. So, the possible sources of reference data for a model identification are:

- Experiments – time responses of detectors, read by the command ROM into the matrix of observation points.
- Responses calculated by FEMINA using a 1D or 2D finite element model. The model simulates stimulus response experiment, starting from a steady solution of velocities, followed by a transient solution of mass transfer (concentration of tracer).
- Responses calculated by a CFD program, e.g. by Fluent or Cosmos. Time courses of concentrations must be transformed to a file which can be read to the matrix of observation points by the command ROM.
- CFD programs can be used only for calculation of a steady state flow field, i.e. velocities. These results can be imported into FEMINA, that simulates only the stimulus response experiment (FEMINA solves the transient concentration field and performs postprocessing using e.g. algorithms of collimated detectors).

The optimisation routines can be used also for the computerised tomography. The simplest applications will be demonstrated on examples, where the goal is identification of a *steady* distribution of matter (density) in a cross section or inside a vessel. The density distribution will be described by a simple function (e.g. as a polynomial of *x,y,z*) and the identification of its coefficients will be based upon comparison with steady signals from multiple detectors (transmission γ-tomography with several collimated detectors and electro-tomography based upon evaluation of signals from conductivity probes).

The two optimisation methods implemented in FEMINA, OPTIMA and SOMA, are the same from user's point of view. Both the methods accept models, defined either as a lumped parameter model activated by the command RMODEL, or models defined as batch files, which are activated by the command

OPEN *unit,file* the batch *file* will be opened on a *unit*. The unit cannot be 1, 2, 9,10,11,20 because these numbers are used for working files or opened windows. It is for example quite safe to use *unit*=7.

The batch file can describe for example a finite element model whose results are transferred to the matrix of observation points, or it can be only a function, e.g. quadratic polynomial with 3 unknown parameters, system variables A,B,C

TCF C2=A+B*C1+C*C1**2

The model parameters (e.g. A,B,C in the previous example) have to be defined in the same way as it was done at the lumped parameter models

PARDEF *No.of parameters,p_1-name,p_2-name,…*definition of model parameters, identified by names of arbitrary system variables or even by a vector element (e.g. RP(10), XPT(NPT-1),…). The parameters can be real or integer and their type will be identified automatically.

PARSET *p_1, p_2,….* set actual values of parameters.

PARLIM p_{1min}, p_{1max}, p_{2min}, p_{2max},... defines allowed range of parameters. This range is important first of all for efficiency of the SOMA method.

PARFIT *p_1, '0-supressed,1-linear,2-nonlin,3-search', ω_{p1}, p_2, '0-supressed, ...', ω_{p2},...*
For each model parameter the three values are specified: initial value, then an integer value determining optimisation (0-parameter is not optimised, 1-model is linear or nearly linear with respect to this parameter, 2-highly nonlinear parameter, 3-golden section search) and the last value ω_p is an underrelaxation factor (0,1) with the default value 1. The values ω_p<1 should be used only when problems with convergence are encountered.

Both the model prediction and the reference data must be in some columns of the observation point matrix MOP. It is necessary to specify *L* columns which are to be compared and used for calculation of a target function (for example the weighted sum of squares of differences):

$$s^2 = \sum_{j=1}^{L} \sum_{i=i_1}^{i_2} (y_{i,e_j} - y_{i,m_j})^2 y_{i,w_j}$$

CRITER *criterion*
It is possible to select another criterion for comparison than the sum of squares (it is only default). *Criterion*=0 sum of absolute values of differences, =1 sum of absolute values of relative differences, =2 sum of squares of differences (default), =3 sum of squares of relative differences, =4 integral of absolute values, =5 integral of absolute values of relative differences, =6 integral of squares, =7 integral of squares of relative differences.

COMPAR *Number of pairs, first row, last row, prediction$_1$, data$_1$, weight$_1$, prediction$_2$, ...*
Number of pairs equals the number of responses *L* which are used for optimisation (maximum is usually 4, because then you need 8 columns of MOP for measured data and model predictions and one column for weights), *first row i_1* and *last row i_2* define the range of MOP rows and therefore the number of points used for optimisation (the same for all responses). For each pair of responses the three parameters define columns of prediction, reference data and weighting coefficients, respectively. When *weight$_1$*=0, unit weights are assumed.

TCDEV calculates and displays deviation for actual settings (COMPAR and CRITER); result is in system variable SCOMP.

If the model is prepared and parameters are specified it is possible to start optimisation

OPTIMA *'MODEL 0,1 dif.eq., >1 file',Number of iterations, scaling, λ-increase, λ-decrease*

Deterministic algorithm of Levenberg Marquardt type modified also for integer parameters. The first parameter selects the optimised model: MODEL=0 is the active lumped parameter model calculating *impulse* responses (stimulus functions, inlets, are suppressed), MODEL=1 is also the lumped parameter model, calculating responses to inlet functions with zero initial conditions, MODEL=2 preserves initial condition as well as inlets. MODEL>2 is the unit, where a session file (model definition file) is prepared and opened by the command OPEN. The last three parameters concern the Levenberg Marquardt method and default values are optimal in most cases: *scaling*=1 (default) activates the so called scaling of normal equations system performed in each iteration, *λ-increase* (default 10) defines how many times the λ factor is increased in case that results of iteration worsens and on the other hand *λ-decrease* (default 0.5) determines speed of λ factor diminishing in case of success (λ =0 is an ideal case when the algorithm reduces to the fast Gauss method, while very large value of λ pushes the solution towards the steepest descent method, which is slower but more straightforward).

SOMA *'MODEL 0,1 dif.eq., >1 file',Number of iteration,Specimen,PRT,step,mass*

This method belongs to the group of stochastic algorithms (similar to genetic or evolutionary algorithms, random walk, Monte Carlo, simulated annealing and so on). The first two parameters are the same as in the command OPTIMA. The parameter *Specimen* defines number of randomly distributed trials (specimen) which are moving towards the best solution in each iteration, PRT is the perturbation of the search direction (PRT=1 no perturbation, in fact deterministic approach), *Step* and *mass* determine the length of specimen paths. For more details look at the chapter 4, but from practical point of view it is usually sufficient to accept default values of parameters.

The OPTIMA and SOMA are main tools for optimisation, while the regression algorithms (fit of parameters of several predefined models LINREG, NELREG, NLR2) are of only limited use. In fact any task which can be solved by regression can be also solved by OPTIMA or SOMA, but not so quickly, and so easily. On the other hand OPTIMA/SOMA cannot be processed as a batch only as a part of a macro. This procedure will be demonstrated in the following paragraphs.

System variables frequently used in optimisation procedures are:

KCOMP	criterion of optimisation, see command CRITER
SCOMP	calculated norm of deviations
NCOMP	number of compared pairs of MOP columns (experiment-prediction)
ICOMP(i)	compared pairs of columns and weights in MOP, see

ICOMP(1)	column of 1st prediction
ICOMP(2)	column of 1st experiment
ICOMP(3)	column of weights for 1st pair
ICOMP(4)	column of 2nd prediction
ICOMP(5)	column of 2nd experiment
ICOMP(6)	column of weights for 2nd pair
…	…

2.6.1. Model identification by regression analysis

We shall assume, that the impulse response of a system is known and represented by a table of points stored in the observation matrix. These data can be read from a data file (command ROM), or created artificially, for example by the following sequence of commands

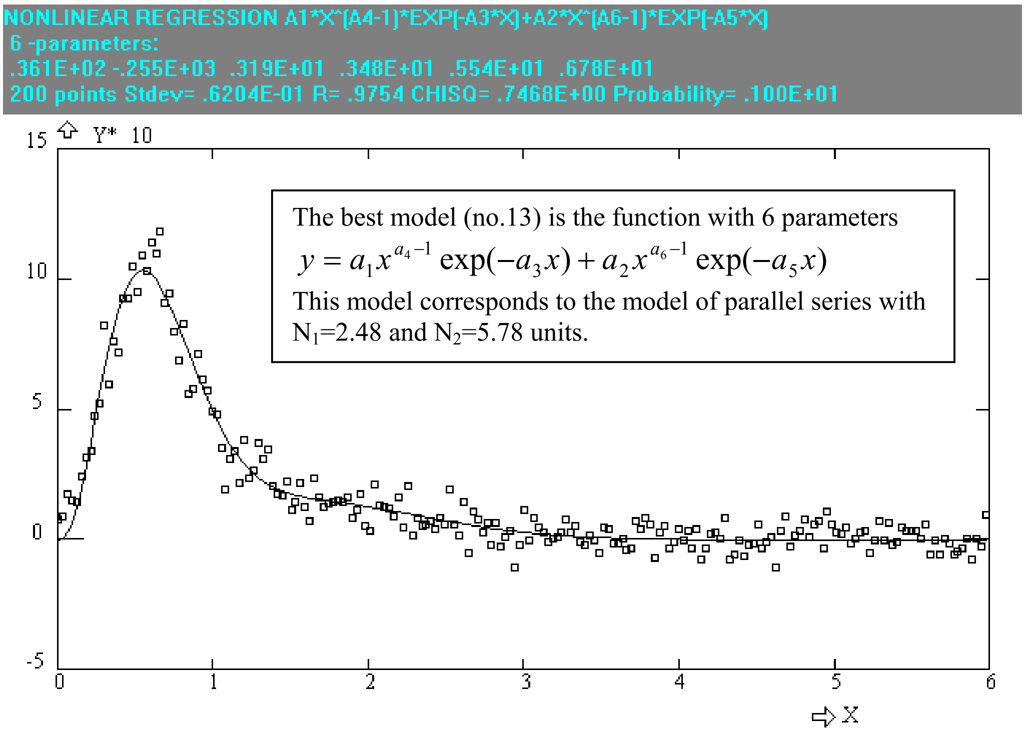
```
TSTEP 200,0.03
PASERI 3,1.5,6,.500E+00,.5,1
TCRND 3,3,1,.100E+00
TCRND 3,3,0,.05
TCFUN C4=1
```

200 points (rows of the observation matrix) with equidistant time step 0.03. The first column is time, the second column will be later used for model prediction, the third column represents measured concentration (formed as an impulse response of Bishoff ‘s model of two parallel series with noise superposed) and the fourth column is set to 1 (estimated standard deviation of measured points).

Regression analysis is activated by command

```
NELREG 0,1,3,2,4;
```

The first parameter (0) is an index of selected model (today 13 models are implemented). We usually do not know which model is the best, and in this case we specify 0, which means that all models will be tested and the best one selected automatically. The following parameters identify columns in MOP, independent variable (time) in the 1st column, experimental data in 3rd column, model prediction in the 2nd and weights in the 4th column.



```
NONLINEAR REGRESSION A1*X^(A4-1)*EXP(-A3*X)+A2*X^(A6-1)*EXP(-A5*X)
 6 -parameters:
 .361E+02 -.255E+03  .319E+01  .348E+01  .554E+01  .678E+01
 200 points Stdev= .6204E-01 R= .9754 CHISQ= .7468E+00 Probability= .100E+01
```

The best model (no.13) is the function with 6 parameters

$$y = a_1 x^{a_4 - 1} \exp(-a_3 x) + a_2 x^{a_6 - 1} \exp(-a_5 x)$$

This model corresponds to the model of parallel series with N_1=2.48 and N_2=5.78 units.

The regression can be repeated for specified models, e.g. for model 6 (series) and 8 (axial dispersion).

```
NELREG 6,1,3,2,4;
NELREG 8,1,3,2,4;
```

Important statistics are summarised in the following table

Model	χ^2	R	s	identified parameters
13 - Parallel series	0.7468	0.9754	0.06204	N_1=2.48 N_2=5.78
8 – Axial dispersion	1.071	0.9646	0.07373	
6 - Series	1.106	0.9634	0.07492	N=2.56

2.6.2. RTD model identification

In this example we shall demonstrate several methods, how to identify parameters of a general RTD model using the same data as previously

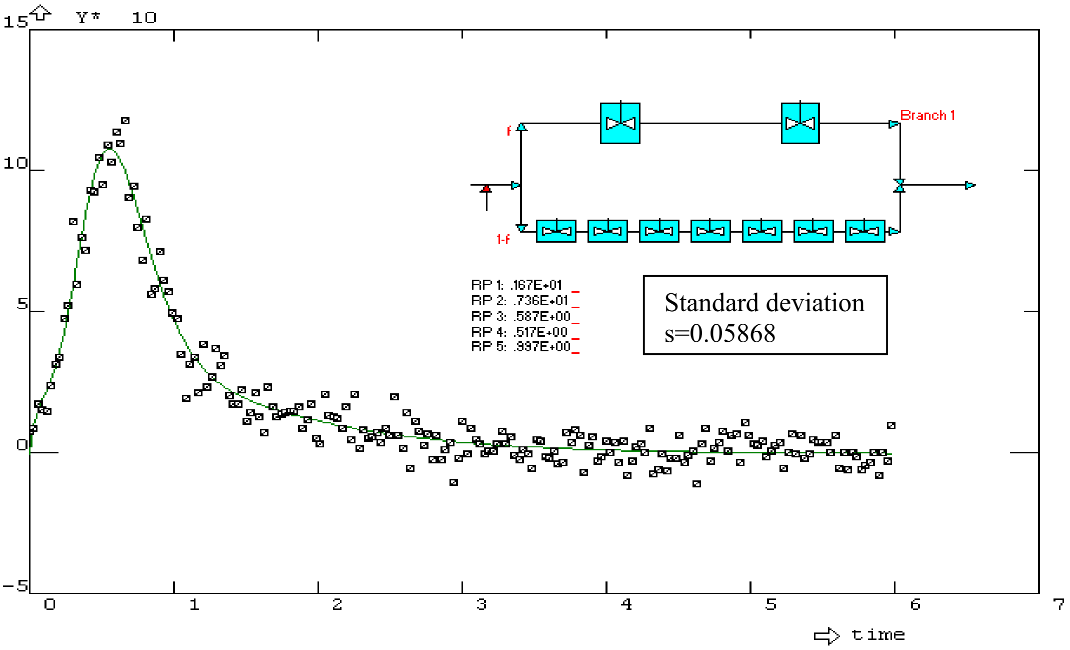
```
TSTEP 200,0.03
PASERI 3,1.5,6,.500E+00,.5,1
TCRND 3,3,1,.100E+00
TCRND 3,3,0,.05
TCFUN C4=1
```

RTD model could be defined as a sequence of commands in a session file, calculating in this example values of model response in the second column of the observation matrix. We can use the Bishoff's model, because its impulse response can be directly calculated by the command PASERI. Therefore the model file can be formed from the single command and saved under name e.g. PASERI

```
RECORD PASERI
PASERI 2,a,b,c,d,e;
ENDREC
```

This model has 5 parameters, A,B,C,D,E which must be described by the following commands

```
PARDEF 5,A,B,C,D,E
PARLIM .1,3,3,10,.1,.9,.1,.9,.1,5
PARFIT 1,2,1,5,2,1,.7,2,1,.7,2,1,1.3,1,1
COMPAR 1,1,200,2,3,4
OPEN 7,Paseri
OPTIMA 7,10
```



We could also try the SOMA method and test results obtained with different values of operating parameters (specimen, PRT, step and mass). To make the testing easier it is suitable to prepare a macro for initial assignment of model parameters, with the name for example ASSGN

```
RECORD ASSGN
A=0.5
B=4
C=0.8
D=0.8
E=1.5
ENDREC
```

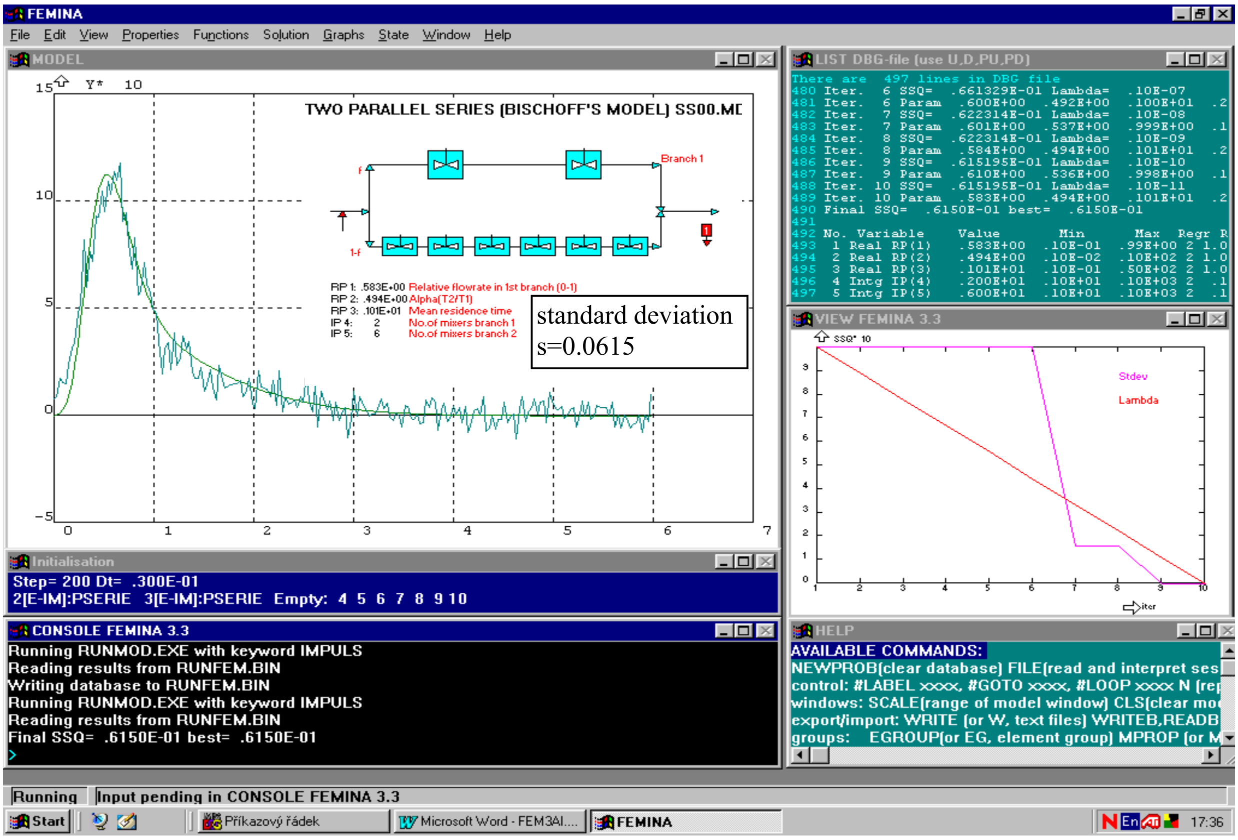
Now it is possible to repeat the sequence

```
        ASSGN
        SOMA 7,100,5,0.5,0.3,1.6
```

and check resulting standard deviations at different choices of operating parameters. You will probably confirm our experience, that the stochastic method SOMA is much less effective than the deterministic algorithm OPTIMA, and even one hundred of iterations with different values of operating parameters was not sufficient to achieve comparable accuracy as OPTIMA in only ten iterations. However SOMA should be fail-proof because it is non-derivative and very simple method.

The previous example is not a typical one and such a straightforward approach can be used only with two standard models: series and parallel series of ideally mixed tanks. Generally it is necessary to use RTD lumped parameter models defined in text files *.MDT activated by the command RMODEL. This procedure is even simpler, because the sequence of commands ROM, RMODEL, COMPAR and OPTIMA is usually enough. We shall illustrate this procedure using the same data (in the 3rd column of MOP) and the model SS0.mdt – two parallel series, however the number of flow units cannot be real, but an integer parameter. In this case it is necessary to use the command PARFIT to select type of parameters (all 5 parameters are marked by 2 as strongly non-linear) and rather small under-relaxation parameters ω_p=0.1 had to be set for integer parameters, number of flow units. The command OUTPUT is used for redirecting model prediction to the 2nd column of MOP (the default value 3 would rewrite the experimental data).

```
RMODEL SS00;
PARFIT .500E+00,2,1,.500E+00,2,1,1,2,1,3,2,.1,7,2,.1;
OUTPUT 1,2;
COMPAR 1,1,200,2,3,4;
OPTIMA 0,10;
```



RP 1: .583E+00 Relative flowrate in 1st branch (0-1)
RP 2: .494E+00 Alpha(T2/T1)
RP 3: .101E+01 Mean residence time
IP 4: 2 No.of mixers branch 1
IP 5: 6 No.of mixers branch 2

standard deviation
s=0.0615

```
There are  497 lines in DBG file
480 Iter.  6 SSQ=  .661329E-01 Lambda=  .10E-07
481 Iter.  6 Param  .600E+00  .492E+00  .100E+01  .2
482 Iter.  7 SSQ=  .622314E-01 Lambda=  .10E-08
483 Iter.  7 Param  .601E+00  .537E+00  .999E+00  .1
484 Iter.  8 SSQ=  .622314E-01 Lambda=  .10E-09
485 Iter.  8 Param  .584E+00  .494E+00  .101E+01  .2
486 Iter.  9 SSQ=  .615195E-01 Lambda=  .10E-10
487 Iter.  9 Param  .610E+00  .536E+00  .998E+00  .1
488 Iter. 10 SSQ=  .615195E-01 Lambda=  .10E-11
489 Iter. 10 Param  .583E+00  .494E+00  .101E+01  .2
490 Final SSQ=  .6150E-01 best=  .6150E-01
491
492 No. Variable    Value      Min      Max  Regr R
493  1 Real RP(1)   .583E+00  .10E-01  .99E+00 2 1.0
494  2 Real RP(2)   .494E+00  .10E-02  .10E+02 2 1.0
495  3 Real RP(3)   .101E+01  .10E-01  .50E+02 2 1.0
496  4 Intg IP(4)   .200E+01  .10E+01  .10E+03 2  .1
497  5 Intg IP(5)   .600E+01  .10E+01  .10E+03 2  .1
```

```
Step= 200 Dt=  .300E-01
2[E-IM]:PSERIE  3[E-IM]:PSERIE  Empty:  4  5  6  7  8  9 10
```

```
Running RUNMOD.EXE with keyword IMPULS
Reading results from RUNFEM.BIN
Writing database to RUNFEM.BIN
Running RUNMOD.EXE with keyword IMPULS
Reading results from RUNFEM.BIN
Final SSQ=  .6150E-01 best=  .6150E-01
>
```

```
AVAILABLE COMMANDS:
NEWPROB(clear database) FILE(read and interpret ses
control: #LABEL xxxx, #GOTO xxxx, #LOOP xxxx N (rep
windows: SCALE(range of model window) CLS(clear mo
export/import: WRITE (or W, text files) WRITEB,READB
groups:   EGROUP(or EG, element group) MPROP (or M
```

Running | Input pending in CONSOLE FEMINA 3.3

Comment: Comparing results obtained by using different methods with the same data reveals that the regression model seems to be less accurate than the model PASERI and even than the model SS00. Maybe an error in the regression analysis?

2.6.3. ODE model identification

It is possible to define and optimise an arbitrary system of ordinary differential equations, and it need not be RTD models. However, the procedure is similar as will be demonstrated on the problem of fouling described by the rate equation

$$\frac{dC}{dt} = Ae^{-E/(RT)} - B \tag{2.6.3-1}$$

where C represents the rate of fouling [kg/(m^2.s)] on a wall, A,B,E are model parameters and T is known temperature, which might be a function of time. Let us assume that the experimental data are represented by two sets of measurements at two different temperatures $T_1(t)$ and $T_2(t)$. The both experiments were carried out in such a way that the temperature of material rose linearly until the time t_1 and t_2, and afterwards temperature remained constant, T_1, T_2, respectively. Two experimentally determined time courses $C_1(t)$ and $C_2(t)$ (deposited mass of fouled material) are in the 2nd and 3rd columns of data file FOUL.DAT. Model can be created for example as the following file FOUL.MDT

```
Regression of fouling model
1-A, 2-B, 3-E, 4-R initial rate,
RP(5)-T0, RP(6)-T1, RP(7)-T2,
RP(8)-time1, RP(9)-time2
\\ini
real t0,t1,t2,time1,time2,b1,b2,te1,te2
t0=rp(5)+273 t1=rp(6)+273 t2=rp(7)+273 time1=rp(8) time2=rp(9)
b1=(t1-t0)/time1  b2=(t2-t0)/time2
cm(1)=r  cm(2)=r
\\mod
te1=min(t1,t0+b1*time) te2=min(t2,t0+b2*time)
dcm(1)=a*exp(-e/te1)-b dcm(2)=a*exp(-e/te2)-b
yv(1)=cm(1) yv(2)=cm(2)
\\par
IDENUM=0 METHOD=-2 NEQ=2 NPAR=9 INP=0 OUT=2
Y1=4  Y2=5
A:A DEFAULT=2e4 MIN=1E-4 MAX=1e6 RELFAKT=1 REGRES=2
B:B DEFAULT=2e-7 MIN=1E-10 MAX=1 RELFAKT=1 REGRES=2
E activ.energy/R:E DEFAULT=5000 MIN=10 MAX=30000 RELFAKT=1 REGRES=2
Initial resistance:R DEFAULT=0 MIN=0 MAX=1 RELFAKT=1 REGRES=2
Initial temperature:RP(5) DEFAULT=50
Final temperature T1:RP(6) DEFAULT=70
Final temperature T2:RP(7) DEFAULT=85
Heating time 1:RP(8) DEFAULT=800
Heating time 2:RP(9) DEFAULT=1000
```

The lines up to \\INI are only text and can be omitted.

Variables declared in the INI section are auxiliary. The only important commands are cm(1)=r and cm(2)=r defining initial conditions for the two experiments C(time=0).

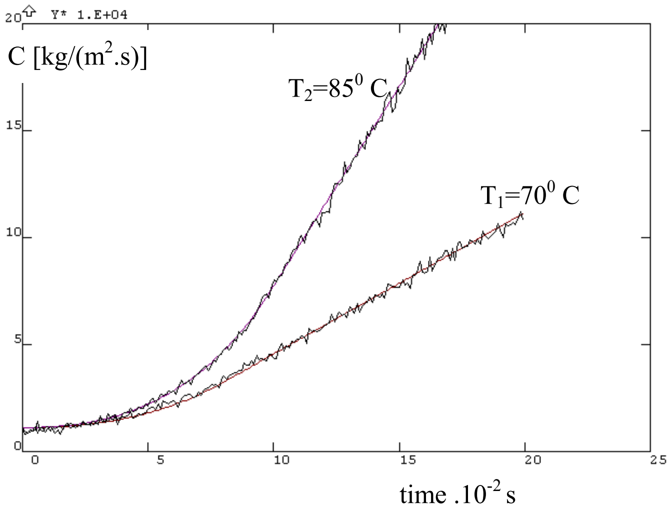
\\MOD section describes differential equations, dcm(1), dcm(2) are first derivatives (rates of fouling), yv(1),yv(2) recorded responses, in this case time courses of C(t).

\\PAR section describes numerical parameters which can be later changed in dialogs.

Next it is necessary to add the name of our new model FOUL.MDT into the list of existing models $FEMINA.CMD and even then it is possible to solve the problem

```
R 6,FOUL.DAT
RM FOUL
COMPARE 2,1,200,4,2,0,5,3,0;
OPTIMA 0,20
GTC;
```

Data and prediction are shown in the figure right. Evaluation of experiments with arbitrary temperature courses $T(t)$ is nearly the same– in this case these courses will be defined as stimulus functions (using system variables $x(t)$).



The following example is a case study, experimental determination of heat transfer coefficient at a surface of warm packed food inserted into a cooling box (cooling cabinet). Heat transfer coefficient at natural convection is usually assumed in the form of a general correlation, written in terms of Nusselt number and Rayleigh number

$$Nu = kRa^m, \quad Nu = \frac{\alpha H}{\lambda}, \qquad Ra = \frac{g\rho c_p H^3 (T - T_e)}{\nu\lambda}, \qquad (2.6.3\text{-}2,3,4)$$

where α [W.m^{-2}.K^{-1}] is heat transfer coefficient, H is height of sample, λ [W.m^{-1}.K^{-1}] is thermal conductivity, g=9.81 acceleration, c_p [J.kg^{-1}.K^{-1}] heat capacity, ν [m^2.s^{-1}] kinematic viscosity. $T(t)$- is surface temperature and $T_e(t)$ is ambient temperature of air. However, it is not possible to describe changes of Nu by this single (and simple) relationship in the whole range of Rayleigh number Ra and therefore a more general model is to be used

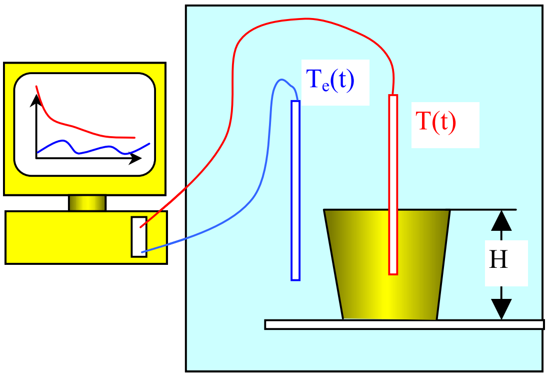
$$Nu = k_1 Ra^{m_1} \quad \text{for Ra<Ra}_{\text{crit}}\text{, and } Nu = k_2 Ra^{m_2}. \qquad (2.6.3\text{-}5)$$

Experimental determination of parameters k_1, k_2, exponents m_1, m_2 and Ra$_{\text{crit}}$ is based upon recording time course of temperature $T(t)$ of a metallic model and the temperature of air $T_e(t)$ inside the cooling box. The model predicting temperature of metallic sample is based upon single ordinary differential equation

$$Mc_p \frac{dT}{dt} = -\alpha S(T - T_e) \quad \text{or using Nusselt number} \quad \frac{dT}{dt} = -\frac{\lambda NuS}{HMc_p}(T - T_e), \qquad (2.6.3\text{-}6)$$

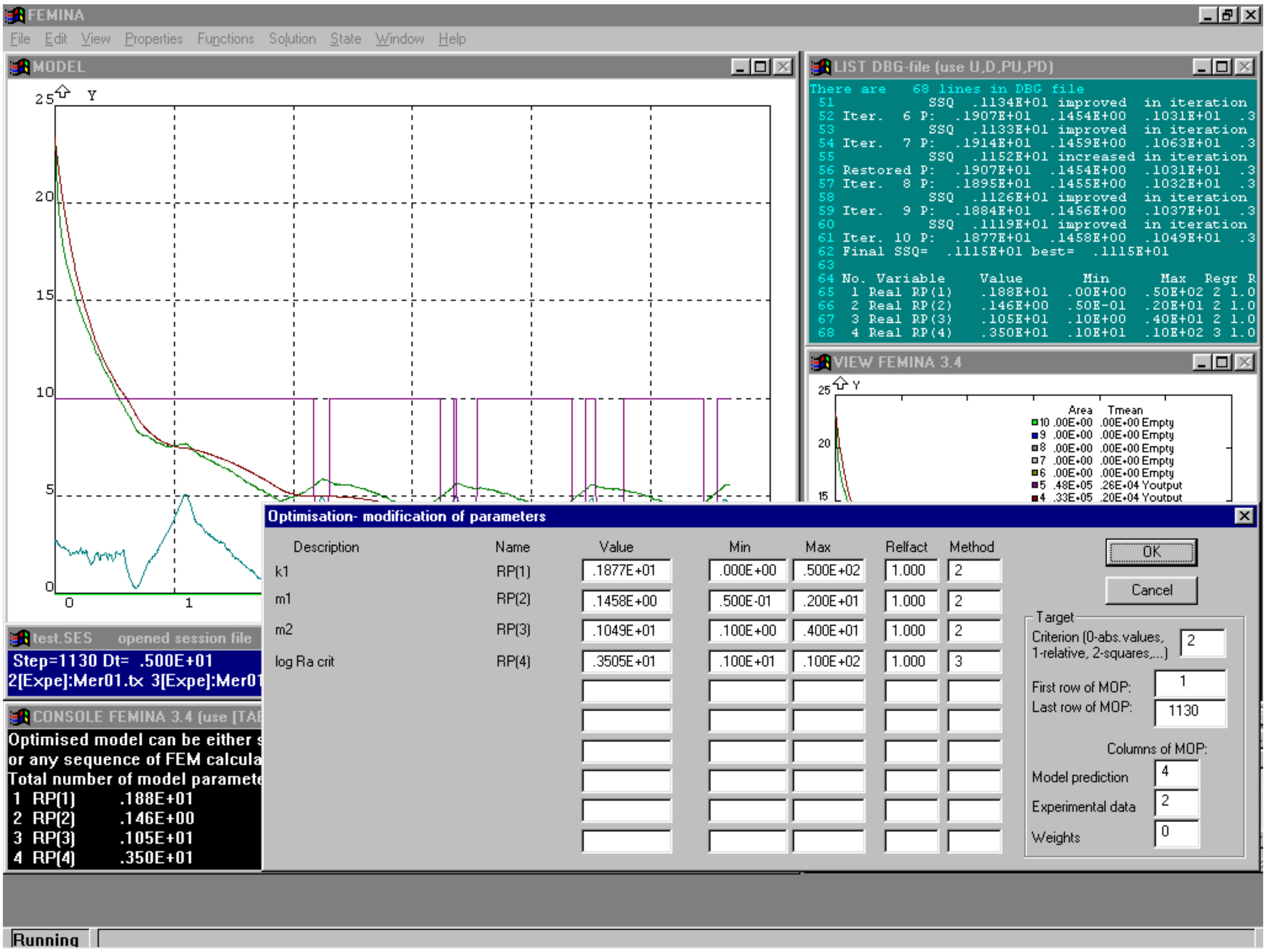
where M [kg] is mass, and S [m^2] surface of sample. Solution of this equation cannot be obtained in an analytical form first because $T_e(t)$ is not a constant and also because λ, Nu, c_p are temperature dependent parameters. This is why the equation is defined in FEMINA as the following model (F4)

```
COOLING CABINET
Determination of heat transfer coefficient
from measured temperature of environment (xv(1))
assuming metalic sample. Model parameters
RP(1) - k1  (from Nu=k1.Ra^m1)
RP(2) - m1
RP(3) - m2  (from Nu=k2.Ra^m2)
RP(4) - Rac  critical Rayleigh number.
\\ini
real ra,lambda,ro,gama,ny,smcp,t1,t2,t3,nu
real k1,k2,m1,m2,rcrit
h=0.03 smcp=0.000063234
k1=rp(1) m1=rm(2) m2=rp(3) rcrit=rp(4)
k2=k1*rcrit**(m1-m2)
cm(1)=c2(1)
\\mod
t1=xv(1)
lambda=.0241+0.000075*t1
c=1005
ro=1.276-0.004365*t1
gama=1/(t1+273)
ny=1.3e-5+9e-8*t1
ra=9.81*gama*h**3*abs(cm(1)-t1)*ro*c/(ny*lambda)
if ra<rcrit then nu=k1*ra**m1 else nu=k2*ra**m2
dcm(1)=lambda*nu/h*smcp*(t1-cm(1))
yv(1)=cm(1)
\\par
IDENUM=       0 METHOD=-2 NEQ=  1 NPAR= 4 INP=1 OUT=1
X1=3  Y1=4
k1:RP(1) DEFAULT=1 MIN=0 MAX=50 RELFAKT=1 REGRES=2
m1:RP(2) DEFAULT=.2 MIN=.1 MAX=1 RELFAKT=1 REGRES=2
m2:RP(3) DEFAULT=1 MIN=.1 MAX=3 RELFAKT=1 REGRES=2
Ra crit:RP(4) DEFAULT=1e4 MIN=100 MAX=1E8 RELFAKT=1 REGRES=3
```



Data are in the file mer01.txt – first column is time, second column temperature *T*e, third column measured temperature of sample. Use the command ROM to read the data, then activate model F4 (using command RM) and calculate theoretical prediction by command INTEGR. Optimised values of k_1, k_2, exponents m_1, m_2 and $\mathrm{Ra}_{\mathrm{crit}}$ can be found by OPTIMA. Comparison of column 3-experimental data and column 4-calculated temperature, is specified by the command COMPAR.

The following figure demonstrates, that majority of important values can be specified in a dialog panel after issuing the command OPTIMA, for example: criterion of optimisation (least sum of absolute values, least sum of squares, least integral deviation,…), range of MOP rows (matrix of observation points) and columns of MOP which are used for comparison between model prediction and measured temperature of sample. Further on an optimisation method can be selected for each parameter separately (0-fixed value, 1-linear parameter, 2-nonlinear regression, 3-golden section method), underrelaxation parameter and prescribed minima and maxima of searched parameters.



```
There are   68 lines in DBG file
 51              SSQ  .1134E+01 improved  in iteration
 52 Iter.  6 P:  .1907E+01 .1454E+00 .1031E+01  .3
 53              SSQ  .1133E+01 improved  in iteration
 54 Iter.  7 P:  .1914E+01 .1459E+00 .1063E+01  .3
 55              SSQ  .1152E+01 increased in iteration
 56 Restored P:  .1907E+01 .1454E+00 .1031E+01  .3
 57 Iter.  8 P:  .1895E+01 .1455E+00 .1032E+01  .3
 58              SSQ  .1126E+01 improved  in iteration
 59 Iter.  9 P:  .1884E+01 .1456E+00 .1037E+01  .3
 60              SSQ  .1119E+01 improved  in iteration
 61 Iter. 10 P:  .1877E+01 .1458E+00 .1049E+01  .3
 62 Final SSQ=  .1115E+01 best=  .1115E+01
 63
 64 No. Variable    Value      Min      Max  Regr R
 65  1 Real RP(1)   .188E+01  .00E+00  .50E+02 2 1.0
 66  2 Real RP(2)   .146E+00  .50E-01  .20E+01 2 1.0
 67  3 Real RP(3)   .105E+01  .10E+00  .40E+01 2 1.0
 68  4 Real RP(4)   .350E+01  .10E+01  .10E+02 3 1.0
```

```
 Step=1130 Dt=  .500E+01
2[Expe]:Mer01.tx 3[Expe]:Mer01
```

```
Optimised model can be either s
or any sequence of FEM calcula
Total number of model paramete
 1  RP(1)      .188E+01
 2  RP(2)      .146E+00
 3  RP(3)      .105E+01
 4  RP(4)      .350E+01
```

Optimisation- modification of parameters

Description	Name	Value	Min	Max	Relfact	Method
k1	RP(1)	.1877E+01	.000E+00	.500E+02	1.000	2
m1	RP(2)	.1458E+00	.500E-01	.200E+01	1.000	2
m2	RP(3)	.1049E+01	.100E+00	.400E+01	1.000	2
log Ra crit	RP(4)	.3505E+01	.100E+01	.100E+02	1.000	3

Target

Criterion (0-abs.values, 1-relative, 2-squares,...) 2

First row of MOP: 1

Last row of MOP: 1130

Columns of MOP:

Model prediction 4

Experimental data 2

Weights 0

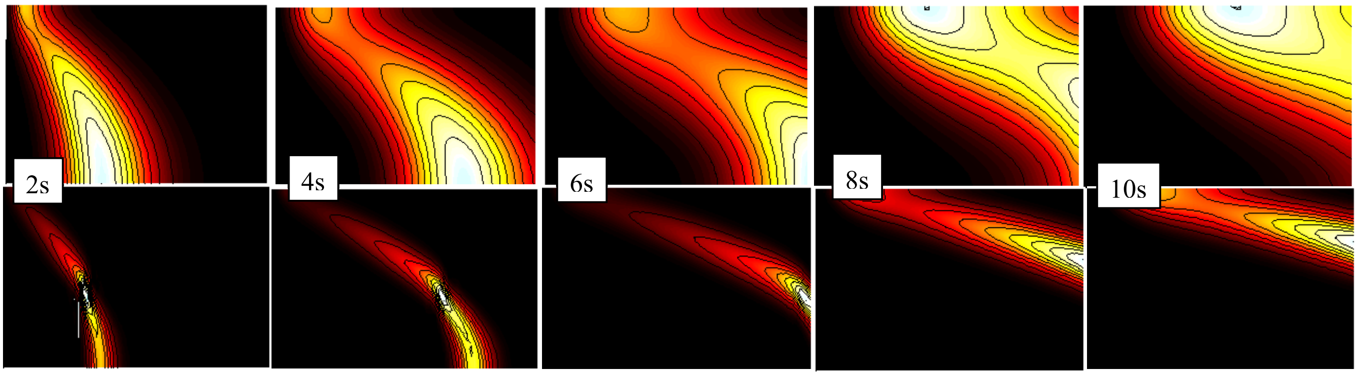
2.6.4. FEM model optimisation

We have found in previous chapters that the 1D model of axial dispersion is not suitable for description of responses in laminar flows if the diffusion coefficient is very small. In this case an empirical formula for the coefficient of axial dispersion

$$D_e = D + \frac{u^2 \tau}{\pi}$$

has been suggested, where τ is an unknown model parameter. This parameter can be identified by comparison with a correct response obtained either from experiment or numerically. As a theoretical response we are going to use results of 2D model of fully developed laminar flow in a straight pipe analysed in chapter 2.2.2. – the following figure (contours of tracer concentration in five time steps up to the mean residence time) illustrates significant influence of diffusion coefficient.

Mesh 60 x 40, Δt=1 s, upwind, D=10^{-6} (upper row), 10^{-7} (bottom row) m^2/s



Let us assume that the responses calculated by the 2D program is saved in the last (10th) column of MOP and the matrix is in file PIPEOPT.DAT. Then we can write program calculating response from a 1D model of axial dispersion with parameter τ (TAUD) and this value will be optimised.

```
READ 6,PIPEOPT.DAT
H=1
R=0.01
U=0.1
W=3*H/U
L=200
PT 1,0,0;
PT 2,H,0;
CR2PT 1,1,2;
CURDEF -1,0,4,0,0,W/L,1,2*W/L,1,3*W/L,0,W,0;
EGROUP 1,PIPE2D,1,0,0;
RCONST 1,PIPE2D,2*R,.100E-03,0,1E-3;
MCR 1,30,1,2,1,0,1;
NFPT 2,PRES,-1,0;
NFPT 1,PRES,-2,8;
SOLVE 0,1,.750E-01,0,0,4;
NFPT 1,CN,-1,1;
DN(1)=1E-7
PARDEF 1,TAUD;
PARLIM 0.1,20;
PARFIT 10,3,1;
COMPAR 1,1,60,3,10,0;
OPEN 7,PIPEOPT2.GEO;
```

Geometry is defined using H-length, R-radius, U-mean velocity, W-maximum time, L-number of time steps.
Duration of stimulus function (function number -1 define by table) is 3W/L.

30 elements PIPE2D are used, and pressure at inlet is specified to 8 Pa and zero at outlet. This pressure difference ensures mean velocity 0.1 m/s as it should be. SOLVE is directed to solve only pressures and velocities, therefore only 1 time step is sufficient.

Optimised parameter TAUD definition, allowed limits τ_{min}=0.1 s and τ_{max}=20 s. PARFIT specifies the optimisation option 3 – this is simple method of one dimensional search based upon *Golden Section* method. COMPAR defines columns for comparison: 60 points, prediction of 1D model in 3rd column and reference response in 10th column. OPEN model definition file on unit 7.

The model definition file PIPEOPT2.GEO should solve transient concentration field in L time steps, evaluate results in such a way, that the response, concentrations in the last node of pipe, will be placed into the 3rd column and afterward normalised to unit area.

```
RECORD PIPEOPT2.GEO
      INI CN,0
      SOLVE 0,1,w/1,1,0,0,0,1,0
      I=0
      T=0
      NT=60
      #LOOP LAB,NT;
      I=I+1
      LOADT T;
      C1(I)=T
      C3(I)=V3(LPU(CN)+ND)
      T=T+0.5
      #LABEL LAB;
      NORM 3,3,0
ENDREC
```

Initial condition for concentration is zero.

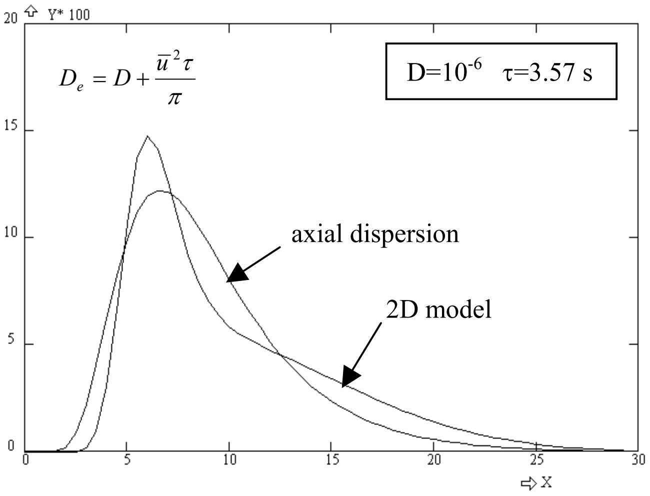
Finite element solution in L-time steps.

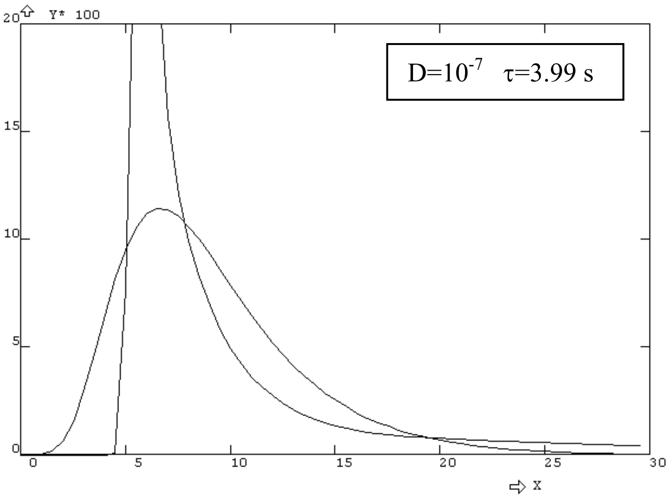
This parameter 1 ensures preservation of element parameters (flowrates calculated previously). Therefore it is not necessary to repeat calculation of flow, and only one iteration for concentration is sufficient in each time step.

Value of concentration in node ND in time T.

```
OPTIMA 7,10
```

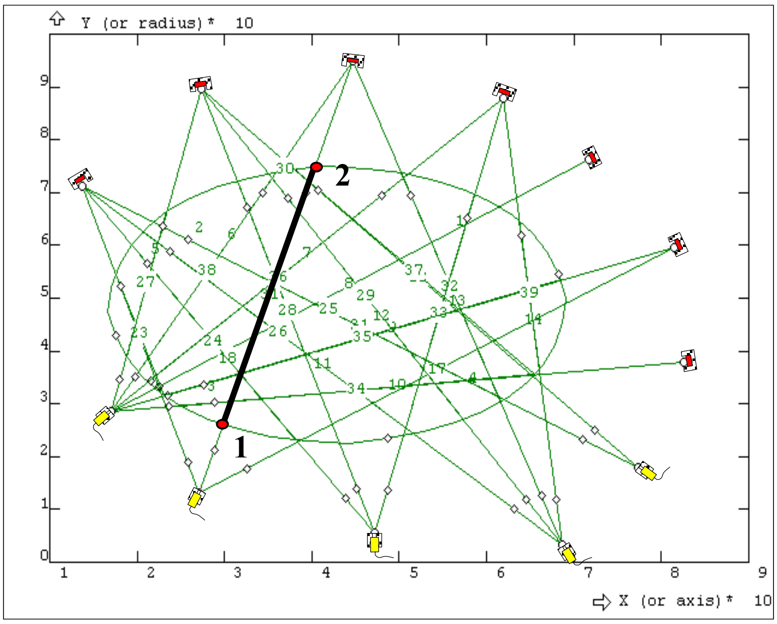
Optimisation of TAUD. Because the direct search option (3) has been selected 10 iterations means in fact 10 initial trials inside the specified range of TAUD, followed by several steps of golden section method. This is sufficient for 1% accuracy of calculated optimal value TAUD.





2.6.5. Transmission tomography

Optimisation methods can be applied also for the transmission tomography, aiming to identify distribution of density in a cross-section of an apparatus. The simplest arrangement consists in using several shielded γ-emitters and collimated detectors, monitoring attenuation of rays due to absorption in material. Such a situation is shown in the following figure (circular cross-section, surrounded by 7 emitters and 5 detectors).



Let us assume, that the measured signal of an arbitrary pair emitter-detector can be expressed as a mean value of unknown density ρ(x,y) along the ray (e.g. along the abscissa 1-2 in the figure),

$$\rho_m = \frac{1}{L_{12}} \int_1^2 \rho(x,y)dl \tag{2.6.5-1}$$

Experimental data are therefore represented by M values ρ_{m1}, ρ_{m2}, …, ρ_{mM} , which enable to identify a model of density distribution ρ(x,y), for example in the form of a quadratic polynomial

$$\rho(x,y) = A + B((x-0.5)^2 + (y-0.5)^2) + C(x-0.4)(y-0.4) \tag{2.6.5-2}$$

The function (2) has been selected accidentally and has only 3 parameters which are to be identified.

Solution proceeds in the following steps:

First the measured abscissas are generated: We start with definition of outline (a circle in this case) and then we locate emitters and detectors as points PT using mouse (or by using the command READ 5, if you have prepared the coordinates in an ASCII file with the postfix PCS). The connecting lines emitter-detector are defined using CR2PT, and points of intersection with the cross-section perimeter are calculated using PTCR2. These new points define abscissas (use the command CR2PT again), along which integrals of function ρ(x,y) will be calculated.

```
PT 1,.428E+00,.489E+00;       PT 7,.813E+00,.598E+00;       PT 19,.471E+00,.572E-01;
PT 2,.690E+00,.502E+00;       PT 8,.824E+00,.381E+00;       PT 20,.137E+00,.713E+00;
PT 3,.170E+00,.287E+00;       PT 9,.772E+00,.181E+00;       PT 21,.273E+00,.897E+00;
PT 4,.446E+00,.949E+00;       PT 10,.686E+00,.360E-01;      CR2PT 5,3,21;
PT 5,.618E+00,.879E+00;       CIRCLE 1,1,2,4;               CR2PT 6,3,4;
PT 6,.716E+00,.764E+00;       PT 18,.271E+00,.133E+00;      CR2PT 7,3,5;
```

```
CR2PT 8,3,6;
CR2PT 9,3,7;
CR2PT 10,3,8;
CR2PT 11,10,20;
CR2PT 12,10,21;
CR2PT 13,10,4;
CR2PT 14,10,5;
CR2PT 15,18,20;
CR2PT 16,18,4;
CR2PT 17,18,7;
CR2PT 18,19,20;
CR2PT 19,19,21;
CR2PT 20,19,5;
CR2PT 21,9,20;
CR2PT 22,9,21;
PTCR2 1,22;
PTCR2 1,20;
PTCR2 1,21;
PTCR2 1,19;
PTCR2 1,13;
PTCR2 1,7;
PTCR2 1,8;
PTCR2 1,9;
PTCR2 1,14;
PTCR2 2,6;
PTCR2 2,16;
PTCR2 2,19;
PTCR2 2,21;
PTCR2 2,15;
PTCR2 2,12;
PTCR2 3,8;
PTCR2 3,9;
PTCR2 3,10;
PTCR2 3,16;
PTCR2 3,18;
PTCR2 3,19;
PTCR2 4,20;
PTCR2 4,11;
PTCR2 4,12;
PTCR2 4,13;
PTCR2 4,21;
PTCR2 4,22;
PTCR2 4,10;
PTCR2 4,17;
PTCR2 3,5;
PTCR2 2,18;
PTCR2 2,11;
PTCR2 2,22;
PTCR2 3,6;
PTCR2 3,7;
PTCR2 4,14;
PTCR2 34,3;
CR2PT 23,32,35;
CR2PT 24,49,38;
CR2PT 25,31,44;
CR2PT 26,50,41;
CR2PT 27,13,48;
CR2PT 28,30,39;
CR2PT 29,33,42;
CR2PT 30,51,28;
CR2PT 31,52,45;
CR2PT 32,29,37;
CR2PT 33,23,43;
CR2PT 34,22,40;
CR2PT 35,27,53;
CR2PT 36,54,24;
CR2PT 37,36,46;
CR2PT 38,35,26;
CR2PT 39,24,53;
CR2PT 40,51,46;
CR2PT 41,28,52;
CR2PT 42,27,54;
```

Next step is the definition of function ρ(*x,y*) with parameters *A,B,C*, according to Eq. (2)

```
FUNDEF 1,A+B*((XX-.5)**2+(YY-.5)**2)+C*(XX-.4)*(YY-.4);
```

The length of abscissa, which is used for normalisation of integral (1), can be calculated from coordinates of end-points, or (and this is only a self-indulgence) as a line integral of a function which is identically equal one

```
FUNDEF 2,1;
```

The "measured" abscissas are curves CR starting from the index 23 in this specific case. As a mathematical model we suggest procedure, calculating integrals (1) along eighteen curves 23,24,…40 and storing these 18 values into the 2nd column of MOP (in the 3rd column will be measured values). A line integrals can be calculated using command INTGCR and so that we would not have to mechanically retype this command 18-times, we make use the command #LOOP. The sequence of commands will be recorded into the file *tomof.geo*

```
RECORD TOMOF.GEO
      I=22
      #LOOP Lab1,18;
      I=I+1
      INTGCR I,1;
      R=INTEGRAL
      INTGCR I,2
      C2(I-22)=R/INTEGRAL
      #LABEL Lab1;
ENDREC
```

Simulated experimental data will be prepared just by using this model for selected values of parameters *A,B,C* and results will be transferred to the 3rd column of MOP. In a real case the experimentaly obtained values (mean densities

```
A=0.01
B=.5
C=.5
TSTEP 18,1;
```

```
F tomof.geo
TCFUN C3=C2;
```

Further on it is necessary to describe optimised parameters (PARDEF, PARFIT, PARLIM), to identify compared columns of MOP (COMPAR) and activate the mathematical model *tomof.geo* (opening the file on unit 7)

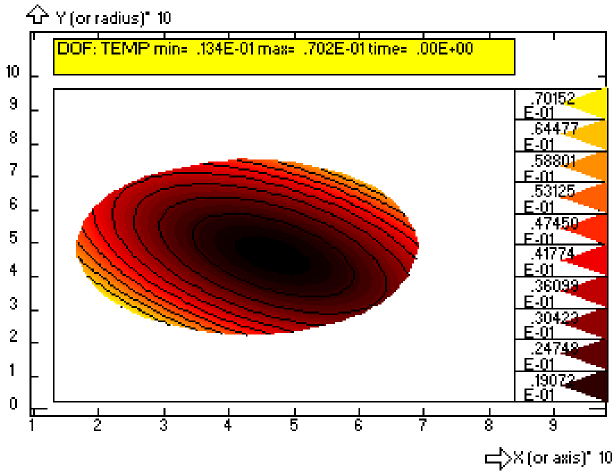
```
PARDEF 3,A,B,C;
PARFIT .02,1,1,.45,1,1,.6,1,1;
PARLIM 0,1,0,1,0,1;
COMPAR 1,1,20,2,3,0;
OPEN 7,Tomof.geo;
```

From now on it is possible to start up OPTIMA or SOMA optimisation. When using OPTIMA it is sufficient to specify model number (this is number of unit 7) and number of iterations (e.g. 3)

```
OPTIMA 7,3;
```

Graphical presentation of results, for example contours of density ρ(x,y) (this is function number 1 with evaluated parameters *A,B,C*) is not so easy. It is necessary to mesh the cross section by finite elements – the cross-section must be defined as a surface (using SFCR) and meshed by the command MSF. The actual values of the function number 1 are transferred to the zone of initial conditions as an arbitrary nodal parameter (for example temperature) using command INI, and even then the contours can be plotted using GD2.

```
SFCR 1,1,2;
EGROUP 1,PLANE2D,0,0,0,0,0;
MSF 1,30,30,1,1,4;
INITIA TEMP,1;
GD2 TEMP,3
```



This example describes only a principle of the method, however, real applications differ only in details. First of all processing of experimental data and evaluation of measured mean density ρ_{mi} along a specific ray must be performed: Assuming linear relationship between density and linear attenuation coefficient the detected intensity of radiation along the *i*-th ray can be expressed by

$$I(\rho_m, L_i) = I_{i0} e^{-a\rho_m L_i} \qquad (2.6.5\text{-}3)$$

where the coefficient *a* is determined from independent experiments or theoreticaly from tabulated absorption coefficients and densities of individual components of e.g. solid-liquid mixtures[24]. The incident intensity I_{i0} is a parameter, which is to be calculated for each ray independently using results of calibration measurement when material density ρ_{ref} is uniform and constant (for example measurement with empty apparatus or with apparatus completely filled by a liquid) – this calibration experiment yields values I_{refi}. Using these values the mean density along the i-th ray can be evaluated from the detected signal I_i as follows

[24] This is the simplest formula, but in exactly the same way general functions of the type $I_i \exp(-f(\rho)L_i)$, with more independently determined parameters, can be used.

$$\rho_{mi} = \rho_{ref} - \frac{1}{aL_i} \ln \frac{I_i}{I_{refi}} \qquad (2.6.5\text{-}4)$$

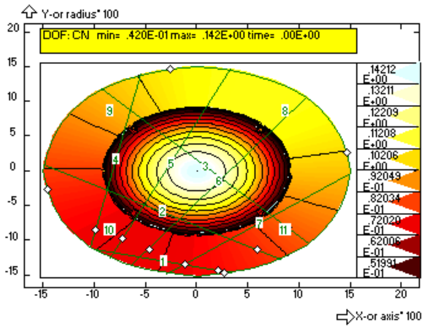
Now the whole procedure is easy; it is convenient to prepare experimental data in form of a text file (free format can be used)

I_{ref1}	I_1
I_{ref2}	I_2
…	
I_{refN}	I_N

and import them into (for example) the 4th (I_{refi}) and 5th (I_{i}) columns of MOP by using the command ROM. Let us assume, that the rays are curves CR starting from 1. The mean densities are then calculated from the values recorded by detectors I_{i} and I_{refi} according to Eq.(4) as follows (assuming for example ρ_{ref}=1000 kg/m^3 and a=0.01 m^2/kg)

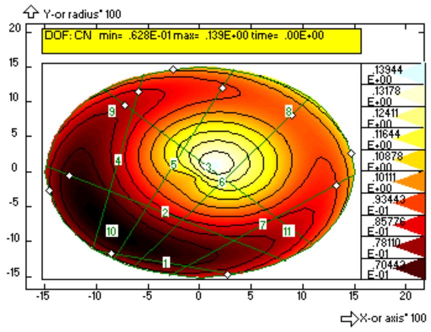
```
TCF C3=1000-LOG(C5/C4)/(0.01*CLEN(I))
```

Another difference concerns the model, describing distribution of density (function number 1 in the previous example). Only very simple functions can be defined by single expression, usually it is necessary to use the more general functions 101,102,.. defined by scripts (see previous examples). In this case it is possible to define even discontinuous functions, for example 6-parametric model (optimised parameters *a,b,c,d,e,f*) suitable for evaluation of radial density distribution in a draft tube mixer. Remark: number of parameters can be reduced, if the radius of draft tube is known.



Asymmetry inside the annular region

```
q=xx**2+yy**2 t=sqr(q) if t>0 then [s=xx/t  p=yy/t]
if Q<a then yv(1)=b+c*q else yv(1)=d+e*s+f*p yv(2)=1
```



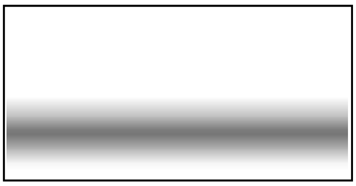
Asymmetry of distribution

```
q=xx**2+yy**2
f=max(sqr(q),1e-6) h=xx/f g=yy/f
yv(1)=a+q*(b+c*q)+d*h+e*g yv(2)=1
```

Model based upon Bessel's functions

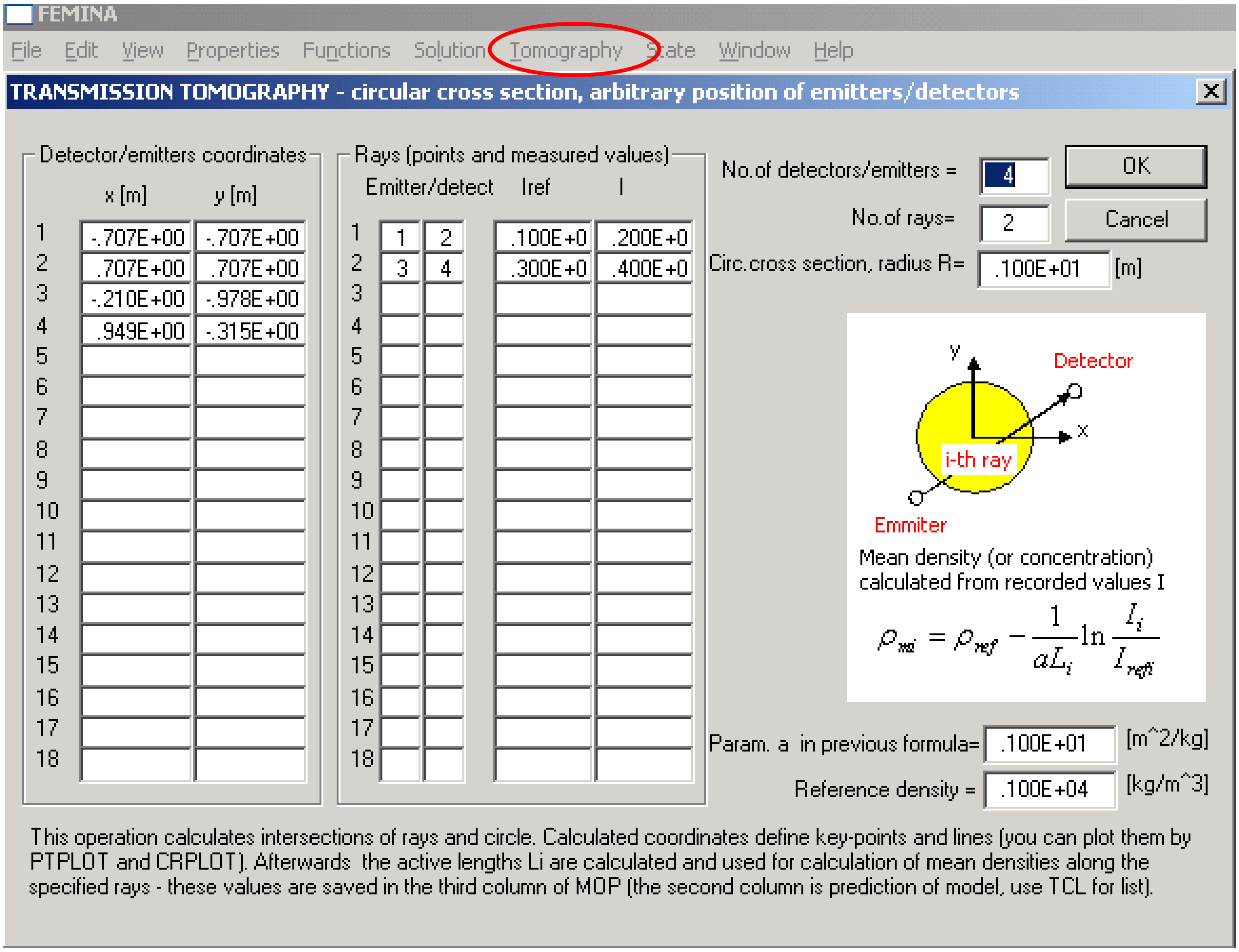
```
q=xx**2+yy**2
if Q<a then yv(1)=b+c*q else yv(1)=d+e*jn(0,f*q)
yv(2)=1
```

In the case of multiphase flows in a rectangular cross section it possible to use for example models (6 parameters *a,b,c,d,e,f*)



if yy<a then yv(1)=b+c*yy+d*exp(-e*yy) else yv(1)=f

While this approach can be applied for quite arbitrary shape of cross section or even for an 3D case, the most frequently encountered geometries, i.e. circular and rectangular cross sections can be processed more easily using the dialog panel accessed from the menu bar Tomography -> Circular cross section, shown in the following figure. In this panel we fill the X,Y positions of emitters and detectors (assuming that the origin of the cartesian coordinate system is in the center of cross section), radius of circle or lengths of sides. At the same time measured values I_i and I_{iref} are specified for each ray (the dialog panel is restricted to 18 rays) together with the reference density ρ_{ref} and parameter a in Eq.(2.6.5-4). This operation calculates intersections of rays with perimeter of cross-section, effective lengths L_i of rays and mean densities ρ_{mi} calculated from Eq.(2.6.5-4) are placed into the 3rd column of MOP.


FEMINA
File Edit View Properties Functions Solution Tomography State Window Help
OK
Cancel
Y
X
Detector
i-th ray
Emmiter

TRANSMISSION TOMOGRAPHY - circular cross section, arbitrary position of emitters/detectors

Detector/emitters coordinates

	x [m]	y [m]
1	-.707E+00	-.707E+00
2	.707E+00	.707E+00
3	-.210E+00	-.978E+00
4	.949E+00	-.315E+00
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		
15		
16		
17		
18		

Rays (points and measured values)

	Emitter/detect		Iref	I
1	1	2	.100E+0	.200E+0
2	3	4	.300E+0	.400E+0
3				
4				
5				
6				
7				
8				
9				
10				
11				
12				
13				
14				
15				
16				
17				
18				

No.of detectors/emitters = 4

No.of rays= 2

Circ.cross section, radius R= .100E+01 [m]

Mean density (or concentration) calculated from recorded values I

$$\rho_{mi} = \rho_{ref} - \frac{1}{aL_i} \ln \frac{I_i}{I_{refi}}$$

Param. a in previous formula= .100E+01 [m^2/kg]

Reference density = .100E+04 [kg/m^3]

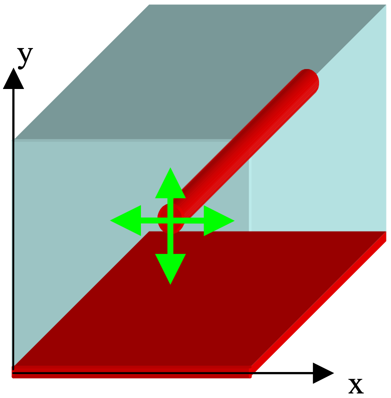
This operation calculates intersections of rays and circle. Calculated coordinates define key-points and lines (you can plot them by PTPLOT and CRPLOT). Afterwards the active lengths Li are calculated and used for calculation of mean densities along the specified rays - these values are saved in the third column of MOP (the second column is prediction of model, use TCL for list).

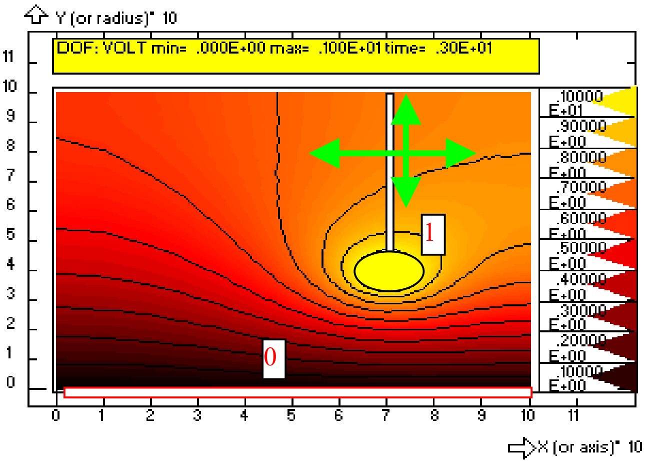
From now on the solution proceeds as previously, it means by the model selection and optimisation of its parameters.

Remark: The dialog box defines new keypoints (PT 1,2,…) and new curves (CR 1,2,…,N), beams, restricted to the interior of cross-section. Besides this the circle or the rectangular boundary is defined using several subsequent points and curves. Be aware of the fact that the old points and curves are deleted! Matrix of observation points is also created with N-row (N-beams) and the old content is rewritten. To save the data for a following session, use for example the command WB (Write dataBase).

2.6.6. Electrotomography

Much more complicated is electro-tomography, based upon measurement of effective electrical conductivity of a volume of material which is in a contact with measuring electrodes. Results are information about spatial distribution of electrical conductivity and inferred density distribution. There exist many different arrangements of electrodes, but in this example we shall analyse configuration with only one submerged moving electrode and the second electrode fixed at the bottom of vessel.





Solution is based upon computation of electric potential distribution for a specific geometry (position of the moving probe) and calculating electrical current flowing through electrodes. For each observation point it is therefore necessary to use finite element method with different mesh, afterwards integrate electrical current and save it to MOP. Instead the integration the current density it is more suitable (and more accurate) to calculate the total electric power dissipated in the measured volume

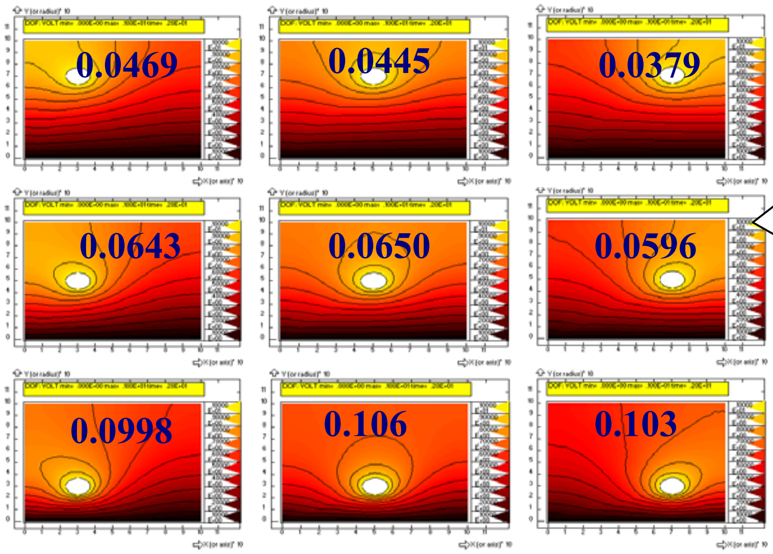
$$P = U \cdot I = \int_{\Omega} \kappa (\nabla U)^2 \, d\Omega \qquad (2.6.6\text{-}1)$$

where κ is specific electric conductivity in a point *x,y,z*.

We shall approximate the distribution of electrical conductivity as a bilinear polynomial of coordinates x,y

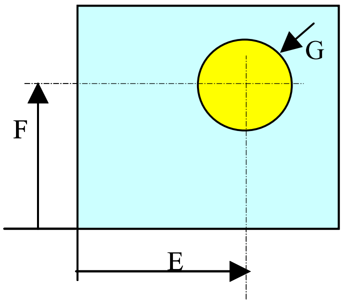
$$\kappa(x, y) = A + Bx + Cy + Dxy\,. \qquad (2.6.6\text{-}2)$$

Let us assume that for identification of four unknown coefficients A,B,C,D it will be sufficient to measure electric current at 9 different positions of the moving cylindrical probe

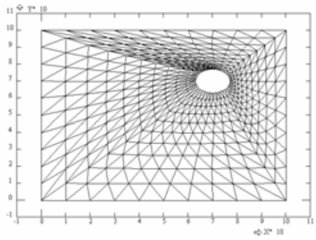


These graphs illustrate asymmetric distribution of electric potential due to asymmetry of electric conductivity. The numbers are calculated electric currents – proportional to measured signals.

Mathematical model will be represented by a command-file (named for example TOMODEL.GEO), describing geometrical model, mesh generation, boundary conditions (zero potential at bottom and unit potential at cylindrical probe) and solution of electric potential by finite element method for variable material parameter κ(x,y) defined by Eq.(2). The operation SOLVE ensures not only solution but also post-processing, whose result, total dissipated power (1) is transferred from the system variable POWERE into appropriate row of MOP. The session file must include solutions for all nine positions of measuring probe and this repetition will be again ensured using command #LOOP:

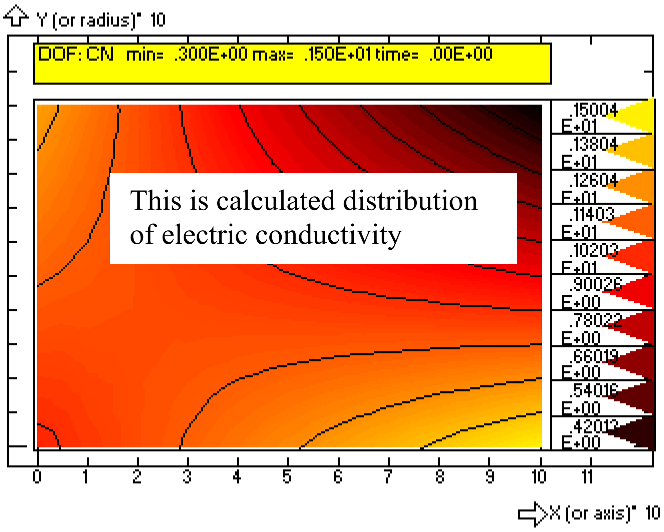


```
G=0.05
E=.1
I=0
#LOOP E,3
E=E+.2
F=.1
#LOOP F,3
F=F+.2
NPT=0
NCR=0
NSF=0
PT 1,.000E+00,.000E+00;
PT 2,.100E+01,.000E+00;
PT 3,.100E+01,.100E+01;
PT 4,.000E+00,.100E+01;
PT 5,E,F;
PT 6,E+G,F+G;
CIRCLE 5,5,6,4;
PT 14,.5,0;
PT 15,E/2,F/2;
PT 16,(E+1)/2,F/2;
PT 17,0,0.5;
PT 18,E/2,(F+1)/2;
PT 19,.5,1;
PT 20,(E+1)/2,(F+1)/2;
PT 21,1,.5;
SF8PT 1,1,2,12,10,14,16,11,15;
SF8PT 2,2,3,6,12,21,20,13,16;
SF8PT 3,3,4,8,6,19,18,7,20;
SF8PT 4,4,1,10,8,17,15,9,18;
NE=0
ND=0
MSF 1,10,10,1,.5,3;
MSF 2,10,10,1,.500E+00,3;
MSF 3,10,10,1,.500E+00,3;
MSF 4,10,10,1,.500E+00,3;
NMERGE ;
NFCR 5,VOLT,-1,0,0,0;
NFCR 1,VOLT,-1,1,1,1;
NFCR 2,VOLT,-1,1,1,1;
NFCR 3,VOLT,-1,1,1,1;
NFCR 4,VOLT,-1,1,1,1;
SOLVE 1,1,1,0,1,0,0,0,0;
I=I+1
C2(I)=POWERE
#LABEL F
#LABEL E
```



This model definition file can be used for identification of optimal parameters A,B,C,D. As a reference data we shall use results of simulation for A=1, B=0.5, C=0.3, D=-1.5, transferred to the 3rd column of MOP and optimisation will be performed with different initial values of A,B,C,D.

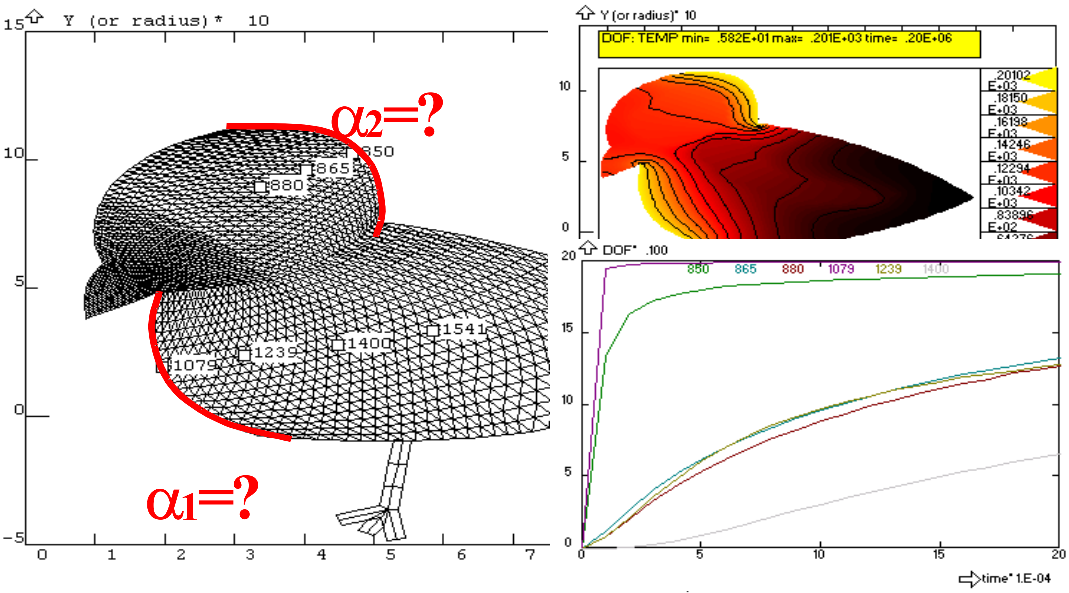
```
ANALYS 2;
EGROUP 1,PLANE2D,0,1,0,0,0;
MPROP 1,PLANE2D,.600E+00,0,4200,0,998,0,.400E-01,1,.210E+12,0,.280E+00,0,.120E-
04,0;
FUNDEF 1,A+B*XX+C*YY+D*XX*YY;
A=1
B=.5
C=.3
D=-1.5
F TOMODEL.GEO
TSTEP 9,1;
TCFUN C3=c2;
PARDEF 4,A,B,C,D;
PARFIT .7,2,1,.7,2,1,.5,2,1,-1.3,2,1;
PARLIM 0,2,0,1,0,1,-2,2;
COMPAR 1,1,9,2,3,0;
OPEN 7,Tomodel.geo;
OPTIMA 7,3
INITIA CN,1;
GD2 CN,3;
```



This is calculated distribution of electric conductivity

2.6.7. Heat transfer identification

This example represents a different kind of applications: Identification of unknown heat transfer coefficients on basis of measured time courses of temperature in points inside a body. Procedure will be demonstrated upon a model of a „two dimensional“ chicken which is heated or cooled. So that the model would be as realistic as possible, also its inner structure, bones with different thermophysical properties will be considered. The model of chicken is shown in the following figure



Geometrical and finite element model of chicken will be created by the following sequence of commands (file *tkure.geo*). Notice some specific features not used previously: combination of different kinds of elements, two dimensional „PLANE2D“ and one-dimensional „TRUSS2D“ (for modelling the chicken’s claws). Further on it is a surface SF having triangular topology (chicken’s beak) and associated meshing, and finally the way haw intestines of chicken are modelled (operation ERMSF).

```
EGROUP 1,PLANE2D,1,1,0,0,0;
PT 1,.876E-01,.384E+00;
PT 2,.143E+00,.438E+00;
PT 3,.191E+00,.486E+00;
PT 4,.152E+00,.541E+00;
PT 5,.112E+00,.598E+00;
PT 6,.876E-01,.505E+00;
PT 7,.979E-01,.725E+00;
PT 8,.114E+00,.867E+00;
PT 9,.261E+00,.704E+00;
PT 10,.354E+00,.535E+00;
PT 11,.265E+00,.502E+00;
PT 12,.188E+00,.103E+01;
PT 13,.313E+00,.113E+01;
PT 14,.470E+00,.106E+01;
PT 15,.504E+00,.761E+00;
PT 16,.444E+00,.638E+00;
PT 17,.209E+00,.136E+00;
PT 18,.335E+00,-.728E-01;
PT 19,.540E+00,-.909E-01;
PT 20,.726E+00,-.547E-01;
PT 21,.527E+00,.296E+00;
PT 22,.679E+00,.922E+00;
PT 23,.936E+00,.103E+01;
PT 24,.104E+01,.734E+00;
PT 25,.109E+01,.242E+00;
PT 26,.943E+00,.904E-01;
PT 27,.819E+00,.574E+00;
PT 28,.520E+00,-.359E+00;
PT 29,.450E+00,-.399E+00;
PT 30,.488E+00,-.450E+00;
PT 31,.531E+00,-.475E+00;
SF6PT 1,1,3,5,2,4,6;
SF8PT 2,3,10,8,5,11,9,7,4;
SF8PT 3,10,15,13,8,16,14,12,9;
SF8PT 4,3,18,20,10,17,19,21,11;
SF8PT 5,10,20,25,15,21,26,27,16;
PT 32,.168E+00,.801E+00;
PT 33,.261E+00,.619E+00;
PT 34,.422E+00,.831E+00;
PT 35,.316E+00,.104E+01;
PT 36,.185E+00,.689E+00;
PT 37,.362E+00,.692E+00;
PT 38,.395E+00,.970E+00;
PT 39,.236E+00,.943E+00;
PT 40,.789E+00,.211E+00;
PT 41,.840E+00,.293E+00;
PT 42,.613E+00,.378E+00;
PT 43,.370E+00,.641E+00;
PT 44,.921E+00,.263E+00;
MSF 1,20,1,3;
MSF 2,20,20,1,1,3;
MSF 3,20,20,1,1,3;
MSF 4,20,20,1,1,3;
MSF 5,20,20,1,1,3;
EGROUP 2,TRUSS2D,1,0;
RCONST 2,TRUSS2D,0,0,.1,.1;
MPROP 2,TRUSS2D,
2,0,2000,0,1500,0,0,0,0,0,0,0;
CR2PT 15,19,28;
```

```
CR2PT 16,28,29;
CR2PT 17,28,30;
CR2PT 18,28,31;
MCR 15,3,1,2;
MCR 16,1,1,2;
MCR 17,1,1,2;
MCR 18,1,1,2;
MPROP 3,PLANE2D,
10,0,2000,0,800,0,0,0,0,0,0,0,0,0;
SF8PT 6,33,34,35,32,37,38,39,36;
SF8PT 7,10,40,41,16,21,44,42,43;
ERMSF 6,1,1,3;
ERMSF 7,1,1,3;
NMERGE ;
```

Mathematical model consists in solution of transient heating of chicken from an initial constant temperature, while the ambient temperature is maintained constant (but different). Let us assume that there are different values of heat transfer coefficients α on the chicken surface - in our example only two different and unknown values α_1=A, α_2=B will be considered on curves 10 and 8, respectively. Results of solution, temperatures at all nodes and in all times are saved in the OUT-file and can be later on retrieved to the zone of initial conditions (zone 3), processed, and finally transferred to the matrix of observation points as a model prediction. The sequence of commands, defining the mathematical model for the case when two thermocouples located at coordinates (0.4,0.9) and (0.4,0.5) record 20 time steps with Δt=10000 s, and for values of heat transfer coefficients in variables A,B, can be written into the file *kurmodel.geo*

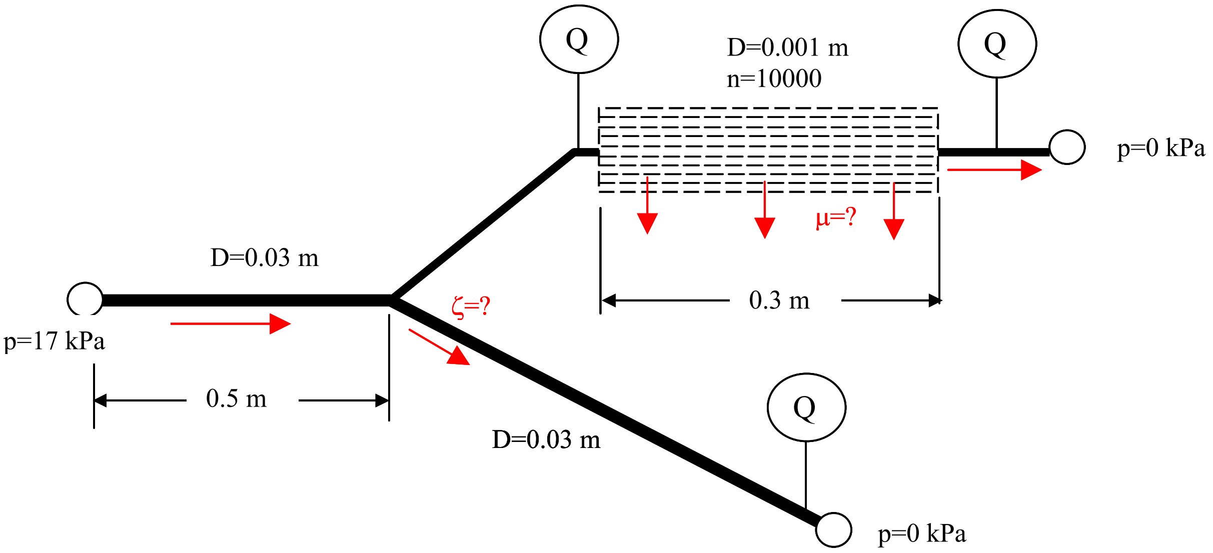
```
RECORD KURMODEL.GEO
      NFCR 10,TEMP,21,A,A,A;
      NFCR 8,TEMP,21,B,B,B;
      INITIA TEMP,0;
      SOLVE 0,20,.100E+05,0,0,0,1,0,0;
      T=0
      I=0
      #LOOP T,20;
      I=I+1
      T=T+10000
      LOADT T
      C1(I)=T
      C2(I)=DOF(TEMP,3,.4,.9,0.)
      C3(I)=DOF(TEMP,3,.4,.5,0.)
      #LABEL T;
ENDREC
```

It is necessary to issue the command BOX so that the function DOF could be used for evaluation of predicted temperatures in a general point *x,y* (which is not coincident with any node generally). Simulated experimental data will be located in columns 4 and 5, while the model prediction will be in columns 2 and 3 (column 1 is time).

```
TE(1)=200
BOX .5
TSTEP 20,.100E+05;
PARDEF 2,A,B;
PARLIM 100,3000,100,3000;
PARFIT 1000,2,1,1500,2,1;
COMPAR 2,1,20,2,4,0,3,5,0;
OPEN 7,KURMODEL.GEO
OPTIMA 7,3
```

2.6.8. Wall permeability identification

We shall consider a pipeline network with a section, consisting of several parallel capillaries with permeable walls. Permeability of wall μ and the local loss coefficient ζ are unknown parameters that are to be identified from experiment, where the flowrate before and behind the section of parallel capillaries is measured.



Model corresponding to the previous scheme is defined by the following batch (session file)

```
A=1E-12
Z=1
PT 1,.000E+00,.500E+00;
PT 2,.500E+00,.500E+00;
CR2PT 1,1,2;
PT 3,.600E+00,.800E+00;
CR2PT 2,2,3;
PT 4,.900E+00,.800E+00;
CR2PT 3,3,4;
PT 5,.100E+01,.800E+00;
CR2PT 4,4,5;
PT 6,.500E+00,.500E+00;
PT 7,.100E+01,.000E+00;
CR2PT 5,6,7;
RCONST 1,PIPE2D,.03;
MCR 1,3,1,2,0,0,1;
MCR 2,3,1,2,0,0,1;
MCR 4,2,1,2,0,0,1;
RCONST 2,PIPE2D,.03;
MCR 5,5,1,2,0,0,1;
EGROUP 2,PIPE2D,0,0,0,0,0,0,0,0,0,0,0,10000;
RCONST 3,PIPE2D,.100E-02,0,0,.01,0,0,0,0,0,0,A;
MCR 3,5,1,2,0,0,1;
NMERGE ;
ERMEL 9,1,4,1;
RCONST 4,PIPE2D,.03,0,Z;
NFPT 1,PRES,-1,17E3;
NFPT 7,PRES,-1,0;
NFPT 5,PRES,-1,0;
SOLVE 0,1,1,1,0,20;
```

Variable A will be permeability μ, while Z is ζ

The commands PT, CR2PT can be substituted by the single command CRS (simultaneous generation of points and curves using mouse – record of this operation is transformed to the following sequence PT,PT,CR2PT,PT….)

EGROUP number 2 will be associated with parallel pipes (the last parameter defines number of pipes in the bundle, n=10000).

RCONST number 3 concerns also the parallel pipes – the last parameter A is permeability of wall (thickness of wall h=0.01 must be specified, because overall permeability is proportional to μ/h)

ERMEL command (Egroup, Rconst, Material for an ELement) changes Rconst group number 4 of the element where ζ is to be specified.

NFPT – prescribed pressures at endpoints.

This batch performs computation for specified parameters A and Z. Calculated flowrates corresponding to experimental data from flowmeters are parameters of elements 14 and 19 (first and last element in the parallel pipes section) and in the element 9 (flowrate in the second branch). These predicted flowrates should be compared with the experimental data in the matrix of observation points (MOP), where the model prediction is in the second and experiment is in the third column (for example). So that the model parameters A and Z could be optimised automatically by OPTIMA, it is necessary to prepare the following model (file name is for example perm0.geo)

```
RECORD PERM0.GEO
PERM(3)=A
ZETA(4)=Z
NT=2
SOLVE 1,1,1,1,0,20;
C2(1)=(E1(IEP(EQ,14))-E1(IEP(EQ,19)))*10000
C2(2)=E1(IEP(EQ,9))
ENDREC
```

This model sets the parameters PERM and ZETA of the RCONST group number 3 (bundle of pipes) and 4 (pipe behind branching) and performs solution (20 iterations of flow). Difference of flowrates along the bundle is placed into 1st row, and flowrate in the element 9 to the second row of second column.

You can check the model by F PERM0.GEO and TCL (look at MOP – there should be two rows (NT=2) and two calculated flowrates in the second column. Experimental data can be typed into the 3-rd column by TCI. Even than the optimisation can be carried out by 4 commands in a similar way as previously (remark: MOP is simplified version of OPEN)

```
MOP perm0.geo
PARDEF 2,A,Z;
COMPAR 1,1,2,2,3,0;
OPTIMA
```