



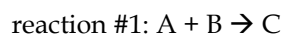
## EXAMPLE 32

## FREE RISE FOAMING

## DESCRIPTION

In Example 31, we have learned how to construct a chemically-reacting flow problem with constant fluid properties. In this example, we further investigate how to make the fluid properties dependent on field variables. To this end, two new keywords, PMAT and LSPM, are introduced.

Let us consider the polyurethane foam system again as described in Example 31. This system involves six species and two exothermic chemical reactions, which we repeat for convenience:



where A, B, C, D, E and F refer to resin, isocyanate, urethane, water, carbon dioxide and urea respectively.

The process of interest is that of the free rise foaming. In order to build a suitable mathematical model, we treat the multi-component system as a compressible homogeneous fluid. By compressibility, we mean here that the overall density is a function of mixture composition. It is thus anticipated that the chemically-induced variations in overall density will have a dramatic impact on the flow kinematics.

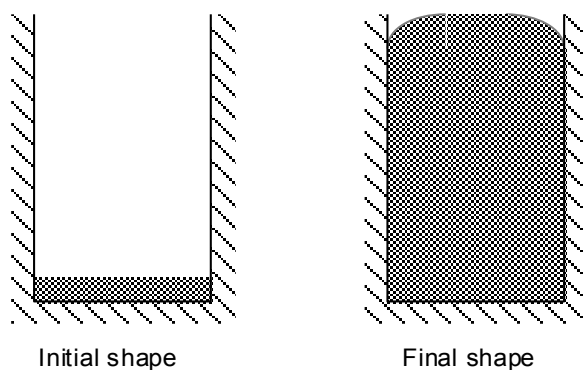


Fig. 1 Schematic of the free rise foaming



A schematic of the free rise foaming process is depicted in Fig. 1. The reactants are pre-mixed rapidly and introduced at the bottom of a thermally-insulated, cylindrical reactor. Foaming is then allowed to occur freely. Production of carbon dioxide leads to a continuous decrease in overall density, which results in an increase in the foam height. The reactor temperature rises by about 150 degrees due to the exothermic nature of the chemical reactions.

The flow problem is considered two-dimensional axisymmetric, time-dependent, and involves a free surface, i.e. the front of the rising foam. The boundary of the computational domain consists of four parts. The foam velocity vanishes at the bottom of the reactor. The foam is assumed to slip (either perfectly or partially) along the reactor sidewalls. Free surface boundary conditions are specified on top, and values of the contact angle are imposed at the free surface end-points. Conditions of perfect thermal/solutal insulation are imposed along the whole boundary.

As far as initial conditions are concerned, we specify homogeneous values for the reactant mass fractions as well as room temperature over the initially-flat computational domain. It is worth mentioning, in relation with the experiments, that the simulation starts only once the so-called *cream time* has been reached (about 12 seconds). As the simulation proceeds, the foam front moves vertically due to the production of the blowing agent; this is accompanied by continuous deformation of the finite element mesh.

## KEYWORDS

derivation of a 1D model: fitting of the kinetics, flow driven by density changes, field dependence (PMAT), species

## FILENAMES

foam.mdf, foam.msh, foam.dat, foam.cons, foam.lst, foam.res, foam.cfx.res

## DERIVATION OF A 1-D MODEL AND FITTING OF THE KINETICS

This section is not essential to the basic understanding of the polyurethane foam system, and can be skipped. But we urge you to work through it.

The free rise foaming process can conveniently be used to study the kinetics of a particular polyurethane foam formulation. Along experimental measurements of temperature and foam height evolution profiles, a one-dimensional mathematical model can be derived, that allows for the prediction of species mass fractions, temperature and foam height as a function of time. Kinetic data are then gathered using best-fit values, and may further be used in POLYFLOW.



The one-dimensional model is developed along lines similar to those of Lefebvre and Keunings [1]. Here, we assume that the flow is driven only by density changes. It is considered perfectly adiabatic, spatially homogeneous and uni-directional with a linear elevation velocity. Dynamical forces are neglected altogether.

For each of the species involved, we define a mass fraction  $X_i$ , a density  $\rho_i$ , a specific heat capacity  $cp_i$ , and a mole mass  $m_i$ . Using the transport properties of the constituent species, the transport properties of the fluid are calculated from (semi) empirical mixture rules.

In this example, it is assumed that overall density and heat capacity relate to composition by the following algebraic relationships:

$$1/\rho = X_A/\rho_A + X_B/\rho_B + X_C/\rho_C + X_D/\rho_D + X_E/\rho_E + X_F/\rho_F$$

$$cp = cp_A X_A + cp_B X_B + cp_C X_C + cp_D X_D + cp_E X_E + cp_F X_F$$

Note that we use a reciprocal form for the density for convenience. The conservation equations reduce to:

$$\text{Species balances } \rho \frac{\partial \omega_i}{\partial t} = S_i \quad (i = A \text{ to } F)$$

$$S_A = m_A (-r_1)$$

$$S_B = m_B (-r_1 - 2r_2)$$

$$S_C = m_C (r_1)$$

$$S_D = m_D (-r_2)$$

$$S_E = m_E (r_2)$$

$$S_F = m_F (r_2)$$

$$r_1 = K_1 \exp\left(\frac{-E_{a1}}{RT}\right) \left(\frac{\rho \omega_A}{m_A}\right) \left(\frac{\rho \omega_B}{m_B}\right)$$

$$r_2 = K_2 \exp\left(\frac{-E_{a2}}{RT}\right) \left(\frac{\rho \omega_B}{m_B}\right) \left(\frac{\rho \omega_D}{m_D}\right)$$

$$\text{Energy balance } \rho c_p \frac{\partial T}{\partial t} = H_1 r_1 + H_2 r_2$$

$$\text{Kinematic equation } \frac{\partial h}{\partial t} - v = 0$$

$$\text{Overall mass balance } \frac{\partial \rho}{\partial t} + \rho \frac{\partial v}{\partial h} = 0 \rightarrow \frac{\partial v}{\partial h} = -\frac{1}{\rho} \frac{\partial \rho}{\partial t} = \rho \sum_i \left( \frac{1}{\rho_i} \frac{\partial \omega_i}{\partial t} \right) = \sum_i \left( \frac{S_i}{\rho_i} \right) \rightarrow v = \sum_i \left( \frac{S_i}{\rho_i} \right) h$$



They transform into a set of eight coupled, non-linear ordinary differential equations, which may be integrated by means of a standard Runge-Kutta numerical procedure. There are 24 parameters:

- 6 densities:  $\rho_A$  to  $\rho_F$
- 6 heat capacities:  $cp_A$  to  $cp_F$
- 6 mole masses:  $m_A$  to  $m_F$
- 6 chemical data:  $K_1$  to  $K_2$ ,  $Ea_1$  to  $Ea_2$ ,  $H_1$  to  $H_2$

Enclosed in the \$POLYFLOW/test/foam directory is a set of FORTRAN formatted files which are aimed at solving the polyurethane foam system using the Runge-Kutta procedure. They have been designed to serve as a general tool and can easily be extended to new situations.

The filenames are:

foam.com	→	defines common blocks
foam.db	→	input file, contains all material data, to be customized
foam.f	→	source code
foam.readme	→	program usage

Please note an executable must be produced first. To use the executable, type the following UNIX command:

```
% name_of_the_executable < foam.db
```

On execution, you will get a number of ASCII files: A, B, C, D, E, F, t, h and ro, i.e. the evolution of the species mass fractions, foam temperature, foam height and foam density as a function of time. These files, in turn, can be imported into the curve section of the POLYPLOT program for visualization purposes.

## MESH GENERATION

Following are some considerations that you may find helpful in generating the mesh. The mesh is made of 5X5 elements with refinements at the wall (geometric progression, ratio is 0.3) and at the free surface (Chebyshev distribution) to capture the foam front curvature more accurately (Fig. 2). On Fig. 3, we display the boundary sets segmentation.

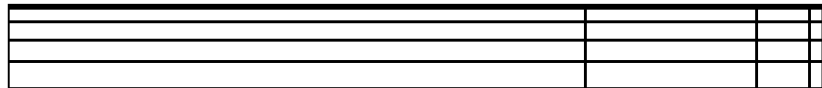


Fig. 2 Finite element discretization.

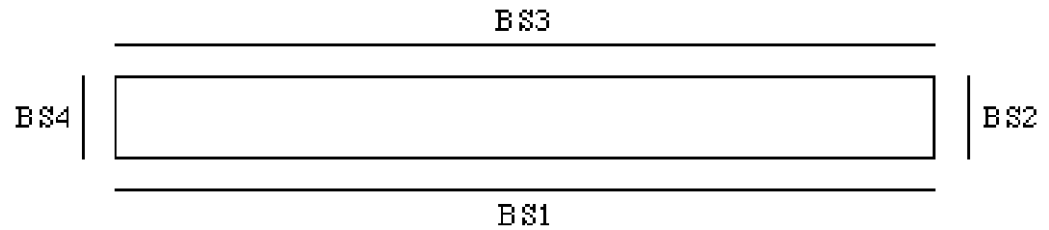


Fig. 3 Boundary sets. (BS1 = bottom, BS2 = wall, BS3 = top, BS4 = axis of symmetry)

## FLUID PROPERTIES

Throughout this section, we use the MKS system of units. Fluid properties have been chosen as follows:

- foam

dynamic viscosity:  $\text{fac} = 1 \text{ [Pa} \cdot \text{s]}$

specific mass:  $\text{density} = \text{value} \cdot F$ , where

$\text{value} = 1$

$1/F = X_A/\rho_A + X_B/\rho_B + X_C/\rho_C + X_D/\rho_D + X_E/\rho_E + X_F/\rho_F$

thermal conductivity:  $K = 10^{-8} \text{ [W} \cdot \text{m}^{-1} \cdot \text{K}^{-1}]$

heat capacity:  $C_p = \text{value} \cdot F$ , where

$\text{value} = 1$

$F = c_{pA} \cdot X_A + c_{pB} \cdot X_B + c_{pC} \cdot X_C + c_{pD} \cdot X_D + c_{pE} \cdot X_E + c_{pF} \cdot X_F$

- species

Species	specific masses [Kg $\cdot$ m-3]	heat capacities [J $\cdot$ Kg-1 $\cdot$ K-1]	mole masses [Kg $\cdot$ mole-1]
A	1100	2301	1.000
B	1206	1757	0.087
C	1200	1214	1.087
D	1000	4184	0.018
E	2	825	0.044
F	1323	1549	0.0148

diffusivities [ $\text{m}^2 \cdot \text{s}^{-1}$ ] (all species)  $D = 0$

- reaction #1:  $A + B \rightarrow C$

pre-exponential factor = 150

activation energy = 46000

enthalpy = 84000

order = 2

→

$\text{prefac} = 1.0174002 \cdot 10^{-6}$  at  $T_r = 294.15$

→

$E_a/R = 5532.6365$

→

$\text{enthalp} = 84000$

→

power of  $c(A)$  = power of  $c(B)$  = 1



- reaction #2:  $2 B + D \rightarrow E + F$

pre-exponential factor = 1800	→	prefac = $1.2208803 \cdot 10^{-5}$ at $T_r = 294.15$
activation energy = 46000	→	$E_a/R = 5532.6365$
enthalpy = 165000	→	entalp = 165000
order = 2	→	power of c(B) = power of c(D) = 1

## POLYDATA SESSION

```
- Read a mesh: foam.msh
- Create a new task: time-dependent, 2D axisymmetric, 'Task 1'
  - Define species
    Create a new species
      name = resin
      nickname = A
      mole mass = 1 [kg/mol]
    Create a new species
      name = isocyanate
      nickname = B
      mole mass = 0.087 [kg/mol]
    Create a new species
      name = urethane
      nickname = C
      mole mass = 1.087 [kg/mol]
    Create a new species
      name = water
      nickname = D
      mole mass = 0.018 [kg/mol]
    Create a new species
      name = carbon_dioxide
      nickname = E
      mole mass = 0.044 [kg/mol]
    Create a new species
      name = urea
      nickname = F
      mole mass = 0.148 [kg/mol]
  - Define reactions
    Create a new reaction: A + B → C
      Domain of reaction: whole mesh
      Equation name: polymerization
      Rate 'constant'
        Ta = 0 [K]
        Tr = 294.15 [K]
        prefac = 1.0174002 10-6 [m3/mol/s]
        beta = 0 [-]
```



**$E_a/R = 5532.6365$  [K]**  
 **$\text{entalp} = 84000$  [J/mol]**  
 Integer orders  
 Power of  $c(A) = 1$   
 Power of  $c(B) = 1$   
 Power of  $c(C) = 0$   
 Create a new reaction:  **$2 B + D \rightarrow E + F$**   
 Domain of reaction: whole mesh  
 Equation name: expansion  
 Rate 'constant'  
 **$T_a = 0$  [K]**  
 **$T_r = 294.15$  [K]**  
 **$\text{prefac} = 1.2208803 \cdot 10^{-5}$  [m<sup>3</sup>/mol/s]**  
 **$\beta = 0$  [-]**  
 **$E_a/R = 5532.6365$  [K]**  
 **$\text{entalp} = 165000$  [J/mol]**  
 Integer orders  
 Power of  $c(B) = 1$   
 Power of  $c(D) = 1$   
 Power of  $c(E) = 0$   
 Power of  $c(F) = 0$   
 - Create a sub-task: generalized Newtonian non-isothermal flow, 'Foam flow'  
 Domain of the sub-task: whole mesh  
 Material data  
 Constant viscosity:  **$\text{fac} = 1$  [Pa s]**  
 Density  
 **$\text{density} = 1$  [kg/m<sup>3</sup>]** *see note 1*  
 with PMAT on density: f's independent for the moment  
 **$\text{average density} = 1130$**   
 Thermal conductivity:  **$K = 10^{-8}$  [W/m/K]**  
 Heat capacity per unit mass [J/kg/K] *see note 1*  
 **$C_p = 1$ ,**  
 with PMAT on  $C_p$ : f's independent for the moment  
 Average temperature:  **$t_{\text{init}} = 294.15$  [K]**  
 Flow boundary conditions *see note 2*  
 BS1:  $(V_n, V_s) = (0, 0)$   
 BS2: **slip conditions  $(V_n, F_s) = (0, -f_{\text{slip}} * V_s)$**   
 Navier's law:  **$f_{\text{slip}} = 1.2$**   
 BS3: free surface  
 Surface tension:  **$\gamma = 0.001$**   
 Boundary conditions  
 **$\text{Angle} = -30^\circ$  at the intersection with BS2**  
 **$\text{Angle} = 180^\circ$  at the intersection with BS4**  
 Imposed director  **$(D_x, D_y) = (0, 1)$**  along the whole surface  
 Upwinding in the kinematic equation  
 BS4: axis of symmetry



Thermal boundary conditions	<i>see note 2</i>
BS1: insulated	
BS2: insulated	
BS3: insulated	
BS4: axis of symmetry	
Global remeshing:	
method of spines on whole mesh:	
inlet: BS4	
outlet: BS2	
Interpolation: quadratic coordinates	<i>see note 3</i>
- Create a sub-task: transport of species, 'Transport of resin'	<i>see note 4</i>
Domain of the sub-task: whole mesh	
Material data	
Average concentration: <b>cinit = 0.6276 [-]</b>	
<b>Diffusivity = 0 [m<sup>2</sup>/s]</b>	
Concentration boundary conditions	
BS1: insulated	
BS2: insulated	
BS3: insulated	
BS4: axis of symmetry	
- Create a sub-task: transport of species, 'Transport of isocyanate'	<i>see note 4</i>
Domain of the sub-task: whole mesh	
Material data	
Average concentration: <b>cinit = 0.3414 [-]</b>	
<b>Diffusivity: 0 [m<sup>2</sup>/s]</b>	
Concentration boundary conditions	
BS1: insulated	
BS2: insulated	
BS3: insulated	
BS4: axis of symmetry	
- Create a sub-task: transport of species, 'Transport of urethane'	<i>see note 4</i>
Domain of the sub-task: whole mesh	
Material data	
Average concentration: <b>cinit = 0 [-]</b>	
<b>Diffusivity = 0 [m<sup>2</sup>/s]</b>	
Concentration boundary conditions	
BS1: insulated	
BS2: insulated	
BS3: insulated	
BS4: axis of symmetry	
- Create a sub-task: transport of species, 'Transport of water'	<i>see note 4</i>
Domain of the sub-task: whole mesh	
Material data	
Average concentration: <b>cinit = 0.031 [-]</b>	
<b>Diffusivity = 0 [m<sup>2</sup>/s]</b>	
Concentration boundary conditions	





- BS1: insulated  
 BS2: insulated  
 BS3: insulated  
 BS4: axis of symmetry
- Create a sub-task: transport of species, 'Transport of carbon\_dioxide' *see note 4*  
 Domain of the sub-task: whole mesh  
 Material data  
 Average concentration: **cinit = 0 [-]**  
**Diffusivity = 0 [m<sup>2</sup>/s]**  
 Concentration boundary conditions  
 BS1: insulated  
 BS2: insulated  
 BS3: insulated  
 BS4: axis of symmetry
  - Create a sub-task: transport of species, 'Transport of urea' *see note 4*  
 Domain of the sub-task: whole mesh  
 Material data  
 Average concentration: **cinit = 0 [-]**  
**Diffusivity = 0 [m<sup>2</sup>/s]**  
 Concentration boundary conditions  
 BS1: insulated  
 BS2: insulated  
 BS3: insulated  
 BS4: axis of symmetry
  - **LSPM** *see note 5*  
 Functional dependence of density:  
 $F(f_1, f_2, \dots) = 1 / (f_1 + f_2 + \dots)$  *see note 6*  
 $f_1 = a + b \cdot X_1 + c \cdot X_1^{**2} + d \cdot X_1^{**3}$   
 $X_1 = \text{resin}$   
 $(a, b, c, d) = (0, 1/1100, 0, 0)$   
 $f_2 = a + b \cdot X_1 + c \cdot X_1^{**2} + d \cdot X_1^{**3}$   
 $X_1 = \text{isocyanate}$   
 $(a, b, c, d) = (0, 1/1206, 0, 0)$   
 $f_3 = a + b \cdot X_1 + c \cdot X_1^{**2} + d \cdot X_1^{**3}$   
 $X_1 = \text{urethane}$   
 $(a, b, c, d) = (0, 1/1200, 0, 0)$   
 $f_4 = a + b \cdot X_1 + c \cdot X_1^{**2} + d \cdot X_1^{**3}$   
 $X_1 = \text{water}$   
 $(a, b, c, d) = (0, 1/1000, 0, 0)$   
 $f_5 = a + b \cdot X_1 + c \cdot X_1^{**2} + d \cdot X_1^{**3}$   
 $X_1 = \text{carbon\_dioxide}$   
 $(a, b, c, d) = (0, 1/2, 0, 0)$   
 $f_6 = a + b \cdot X_1 + c \cdot X_1^{**2} + d \cdot X_1^{**3}$   
 $X_1 = \text{urea}$   
 $(a, b, c, d) = (0, 1/1323, 0, 0)$   
 Functional dependence of C<sub>p</sub> (parameter a):



```

F(f1, f2, ...) = f1 + f2 + ...
f1 = a+b*X1+c*X1**2+d*X1**3
    X1 = resin
    (a, b, c, d) = (0, 2301, 0, 0)
f2 = a+b*X1+c*X1**2+d*X1**3
    X1 = isocyanate
    (a, b, c, d) = (0, 1757, 0, 0)
f3 = a+b*X1+c*X1**2+d*X1**3
    X1 = urethane
    (a, b, c, d) = (0, 1214, 0, 0)
f4 = a+b*X1+c*X1**2+d*X1**3
    X1 = water
    (a, b, c, d) = (0, 4184, 0, 0)
f5 = a+b*X1+c*X1**2+d*X1**3
    X1 = carbon_dioxide
    (a, b, c, d) = (0, 825, 0, 0)
f6 = a+b*X1+c*X1**2+d*X1**3
    X1 = urea
    (a, b, c, d) = (0, 1549, 0, 0)
- Numerical parameters
  Numerical parameters for iterations
  Convergence test = 0.01
  Transient iterative parameters
  Initial value of Time = 12
  Upper limit of Time = 200
  Initial value of delta-Time = 0.1
  Min value of delta-Time = 0.1
  Max value of delta-Time = 0.1
  Tolerance = 1
  Max number of successful steps = 400
  Implicit Euler method
- Outputs
  - Output triggering
    Every 10 valid time steps
  - Default output : CFD-Post
  - System of units for CFD-POST: metric_MKSA+Kelvin
  - Probe
    Probe 1: prefix = foam_1, (x, y) = (0, 0)
- Filename syntax
  - Prefix: foam
- Save and exit
  - Filenames:
    Mesh file:      foam.msh
    Data file:      foam.dat
    Result file:    foam.res
    Restart file:   foam.rst

```

*see note 7**see note 8*



CFD-Post: foam.cfx.res

### **Note 1: Density and heat capacity per unit mass**

We use the PMAT keyword to make the density dependent on the species mass fractions. Please note that, at this stage, we do not yet complete the details of the functional dependence of the density. We merely cause a dependence to be created.

In the event of a dependence, POLYFLOW treats the density as an intermediate variable (i.e. there will be a density field). The initial value is 1130; it corresponds to the value of the density before the reactions take place as per the formula given above. Back to the original menu, we click on (or type) PMAT again to clear the PMAT flag. Similarly, we make the heat capacity dependent on the species mass fractions using the PMAT keyword.

### **Note 2: Flow and thermal boundary conditions**

The velocity vanishes identically on BS1. The fluid is assumed to slip along BS2 according to the Navier's law of slipping. Slip coefficient is 1.2 (in MKS units). BS3 is assigned a free surface, on which the surface tension coefficient is  $0.001 \text{ [N} \cdot \text{m}^{-1}\text{]}$ . In the presence of surface tension, additional boundary conditions must be imposed on both sides of the free surface. At the intersection with BS2, the free surface boundary condition corresponds to an angle of  $-30^\circ$ . At the intersection with BS4, we impose an angle equal to  $180^\circ$ . Those angles deal with the direction of the force which must be applied onto the free surface in order to compensate for the surface tension traction. We also constrain the surface displacement to be aligned with the vertical direction. To do so, we prescribe  $D = (0, 1)$  over the whole moving surface. We finally account for upwinding in the kinematic equation to ensure that irregularities in the mesh will be smoothed out over space. The reactor is assumed to be perfectly insulated.

### **Note 3: Interpolation**

Since the problem includes significant surface tension effect, we presently advise the interpolation shape functions be the same for both the velocity and the coordinates fields. We therefore turn the interpolation type for the coordinates into quadratic.

### **Note 4: Create sub-tasks for species**

We now create transport equations for species one by one. Please note you do not have access to the density as density was previously defined in the sub-task relative to the flow problem. Initial conditions are 0.6276, 0.3414 and 0.031 for species A, B and D respectively, while being zero for products C, E and F. Diffusivities are set to zero. We select the option 'Insulated boundary' for all boundaries.

### **Note 5: LSPM**

We then update the field dependencies. The reason it could not be processed sooner is that access to field variables can only be performed after all field equations exist. Please note you



do not need to move through the tree structure towards the density and/or heat capacity options to review the dependencies. Just type the LSPM keyword (this keyword may be entered from anywhere unless otherwise specified). You get a summary of the field dependencies that you can work on directly.

**Note 6: Functional dependence of the density**

The overall density reads:

$$1/\rho = X_A/\rho_A + X_B/\rho_B + X_C/\rho_C + X_D/\rho_D + X_E/\rho_E + X_F/\rho_F$$

This form can be regarded as a sum of elementary functions of the form:

$$f_i = X_i/\rho_i$$

which are all linear in species. Two concepts hold: (i) a functional dependence and (ii) a field dependence. A functional dependence is no more than a combination of elementary functions that you would build separately to make the overall form more readable. A field dependence is a function of none, one or two field variables that can be chosen from a list (temperature, pressure, species, if any). This feature has been designed to let you construct complex dependencies without the need of re-programming the source code.

Here, the overall density may be regarded as a reciprocal sum of six elementary functions. First, we select 'Reciprocal sum', then we create elementary functions. For each of them, we select the polynomial function, specify the field, and set the numerical parameters as required.

Note that in programming, there are usually several ways for expressing a dependence. So after you complete this section, you may find that you will be able to solve many of the dependencies differently, or in fewer instructions, than we have shown in our examples.

**Note 7: Functional dependence of the specific heat**

The overall specific heat is expressed as:

$$cp = cp_A*X_A + cp_B*X_B + cp_C*X_C + cp_D*X_D + cp_E*X_E + cp_F*X_F$$

It may be regarded as a sum of six elementary functions. First, we select 'Sum', then we create elementary functions. For each of them, we select the polynomial function, specify the field, and set the numerical parameters as required.

**Note 8: Numerical parameters**

We review the numerical parameters which control the time-marching scheme. We choose:

initial time value = 12



upper time limit = 200  
initial value of the time-step = 0.1  
min value of the time-step = 0.1  
max value of the time-step = 0.1  
tolerance = 1  
max number of successful steps = 400

Please note that we explicitly force the time-step to be equally-spaced. Every 10 steps will correspond to one second real time. Convergence test for iterations is set to 0.01.

## RUNNING POLYFLOW

The input file for POLYFLOW is 'foam.dat'. The primary unknowns are the velocity, pressure, temperature, species mass fractions and density fields. Please note that completion of the calculation may take much CPU time.

## GRAPHIC POST-PROCESSING

Two cases have been studied: (i) perfect slip conditions at the vertical reactor wall (the free surface remains flat), and (ii) partial slip (the free surface exhibits a curvature).

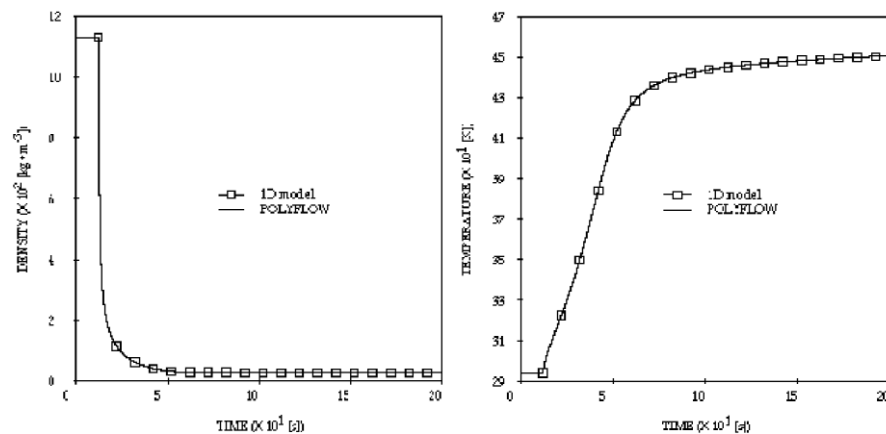


Fig. 4. Time evolution of foam density (left) and foam temperature (right).

When perfect slip occurs at the vertical wall, the mathematical model is exactly described by the above one-dimensional model. This provides a non-trivial test problem for POLYFLOW. We compare in Fig. 4 the predicted time-dependent foam density and temperature obtained by means of the one-dimensional model and by use of POLYFLOW. Agreement is excellent. While the foam density no longer varies after a minute or so, we observe that the temperature does not yet reach a final, constant value at  $t = 200$  seconds. This is a consequence of the widely different rates of reaction as can be seen in Fig. 5 (left) where mass fractions as a



function of time are plotted. The model predictions clearly demonstrate the faster kinetics of the blowing reaction relative to those of the gelling reaction.

The POLYFLOW time history curves are obtained using the Probe output. This output extract pairs of the form  $(t, f(t))$ . These pairs are organized in curve files that can be displayed e.g. with POLYCURVE.

Fig. 5 (middle) shows successive foam fronts, as well as the final, steady-state finite element mesh when partial slip applies (right). In the present case, the free surface is no longer flat in view of the viscous forces acting near the wall (based on an number-mean elevation velocity of about 0.001, the capillary number  $Ca^{-1}$  is computed to be 1). Except in a small zone near the moving contact point, we find out that the computed temperature and mass fraction fields remain essentially homogeneous in space. Density goes to 25.87 ( $t = 200$  seconds) from 1130.42 ( $t = 12$  seconds). Note that while foaming is completed in a matter of minutes, the computer simulation may take many hours CPU-time.

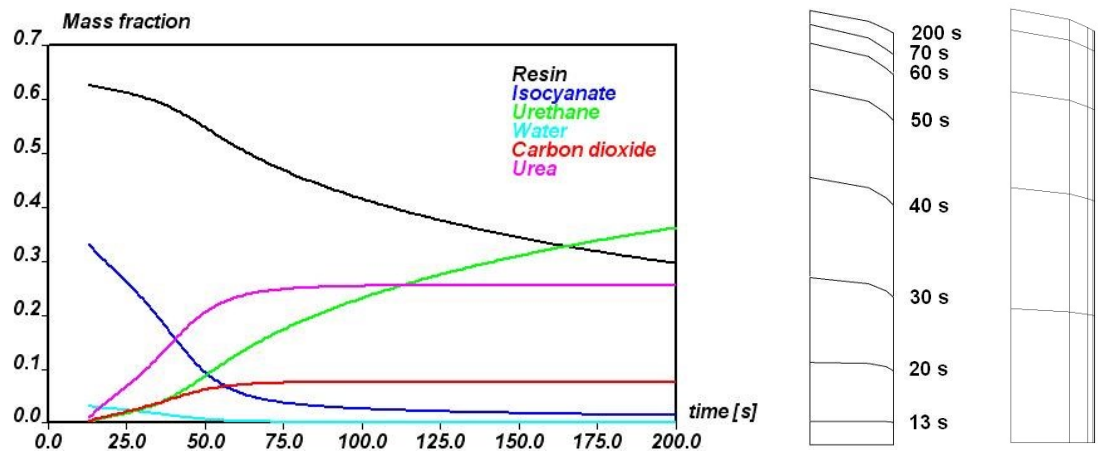


Fig. 5. Mass fractions as a function of time (left), evolution of foam flow front as a function of time (middle), and final steady-state finite element mesh (right).

## REFERENCES

1. L. Lefebvre and R. Keunings in: M. Cross et al. (Eds.), Mathematical Modeling in Material Processing, Oxford University Press, 1991.