

## Appendix 3. Implementation of the coefficient of thermal expansion and compressibility through the IFM interface manager for use with FEFOW 5 (Magri 2004)

Here the equations derived in Appendix 2 for the coefficient of thermal expansion and compressibility will be implemented in FEFLOW through the IFM interface. The definitions of the coefficients involved in the equations are given in Appendix 2.

### Goal

The goal is to incorporate in FEFLOW the following EOS

$$\rho^f = \rho_0^f \left( 1 - \bar{\beta}(T, p)(T - T_0) + \bar{\gamma}(T, p)(p - p_0) + \frac{\bar{\alpha}}{C_s - C_0}(C - C_0) \right) \quad (3.1)$$

in order to reproduce the fluid density for a wide range of temperature and pressure ( $p_{Sat} < p \leq 100$  MPa and  $0 \leq T \leq 350$  °C).

For this purpose Eq.(3.1) must include the equations for  $\bar{\beta}(T, p)$  and  $\bar{\gamma}(T, p)$  derived in Appendix 2 (Eq.(2.19) and Eq.(2.20)). From now on these equations are referred to freshwater condition ( $C_0=0$  g/l) at the atmospheric pressure ( $p_0=100$  kPa) and at triple point temperature ( $T_0=0$  °C), i.e.

$$\bar{\beta}(T, p) = - \frac{1}{\rho_0^f} \left[ T^5(g_0 + p_0g_1 + p_0^2g_2) + T^4(f_0 + p_0f_1 + p_0^2f_2) + T^3(e_0 + p_0e_1 + p_0^2e_2) + T^2(d_0 + p_0d_1 + p_0^2d_2) + T(c_0 + p_0c_1 + p_0^2c_2) + (b_0 + p_0b_1 + p_0^2b_2) \right] \quad (3.2)$$

and

$$\bar{\gamma}(T, p) = \frac{1}{\rho_0^f} \left[ (p - p_0)(a_2 + T^6g_2 + T^5f_2 + T^4e_2 + T^3d_2 + T^2c_2 + Tb_2) + a_1 + 2p_0a_2 + T^6(g_1 + 2p_0g_2) + T^5(f_1 + 2p_0f_2) + T^4(e_1 + 2p_0e_2) + T^3(d_1 + 2p_0d_2) + T^2(c_1 + 2p_0c_2) + T(b_1 + 2p_0b_2) \right] \quad (3.3)$$

where  $\rho_0^f(T_0, C_0, p_0) = 998.8396$  [g/l] and the coefficients are given in Table 2.1 (Appendix 2)

## Equation to implement

In FEFLOW 5 the following EOS for the fluid density is incorporated:

$$\rho^f = \rho_0^f \left( 1 - \bar{\beta}(T - T_0) + \frac{\bar{\alpha}}{C_s - C_0} (C - C_0) \right) \quad (3.4)$$

For achieving the cited goal, an external module coding the new expression of  $\bar{\beta}(T, p)$  (Eq.(3.2) and  $\bar{\gamma}(T, p)$  (Eq.(3.3)) has been implemented in the EOS Eq.(3.4). The implementation of  $\bar{\beta}(T, p)$  can be done directly by the use of the IFM. The IFM offers the users different Application Programming Interfaces (API) whereby external codes can be linked. For instance, the coefficient of thermal expansion can be implemented in FEFLOW via the API “IfmSetMatFlowExpansionCoeff”. On the other hand the EOS for the fluid density present in FEFLOW (Eq.4) does not take in account the coefficient of compressibility  $\bar{\gamma}(T, p)$ . Therefore no related API interface exists for this coefficient and it's not possible to implement directly the expression for  $\bar{\gamma}(T, p)$ (Eq.(3.2)) into Eq.(3.4). Therefore, to make up for this lack, a little trick is required:

Let's apply a simple factorisation in the EOS for the fluid density to be incorporated into FEFLOW (Eq.(3.1)):

$$\begin{aligned} \rho^f &= \rho_0^f \left( 1 - \bar{\beta}(T, p)(T - T_0) + \bar{\gamma}(T, p)(p - p_0) + \frac{\bar{\alpha}}{C_s - C_0} (C - C_0) \right) \\ &= \rho_0^f \left( 1 - \left( \bar{\beta}(T, p) - \bar{\gamma}(T, p) \frac{(p - p_0)}{(T - T_0)} \right) (T - T_0) + \frac{\bar{\alpha}}{C_s - C_0} (C - C_0) \right) \\ &= \rho_0^f \left( 1 - \bar{\beta}^*(T, p)(T - T_0) + \frac{\bar{\alpha}}{C_s - C_0} (C - C_0) \right) \end{aligned} \quad (3.5)$$

with

$$\underbrace{\bar{\beta}^*(T, p)}_{\substack{\text{Defined as BETASTAR} \\ \text{in the source code}}} = \left( \underbrace{\bar{\beta}(T, p)}_{\substack{\text{Defined as BETA} \\ \text{in the source code}}} - \underbrace{\bar{\gamma}(T, p) \frac{(p - p_0)}{(T - T_0)}}_{\substack{\text{Defined as GAMMASTAR} \\ \text{in the source code}}} \right) \quad T \neq T_0 \quad (3.6)$$

and  $\bar{\beta}(T, p)$ ,  $\bar{\gamma}(T, p)$  expressed in Eq.(3.3) and Eq.(3.4) respectively.

In this way, a new variable,  $\bar{\beta}^*(T, p)$  has been derived.  $\bar{\beta}^*(T, p)$  takes in account the coefficient of thermal expansion  $\bar{\beta}(T, p)$  and compressibility  $\bar{\gamma}(T, p)$ . This new variable can be directly implemented in the EOS of the fluid density present in FEFLOW (Eq.(3.4)) through the API “IfmSetMatFlowExpansionCoeff” leading to the EOS Eq.(3.1).

### **Description of the source code Beta\_Gamma.c**

The input parameters required for calculating  $\bar{\beta}^*(T, p)$  are the temperature (T) and pressure (p) of each element of the FE mesh. The temperature and pressure are nodal values therefore the code has to calculate the temperature and mean pressure of the grid element. This can be easily done by summing the nodal values of the considered physical parameters and then dividing this sum by the number of nodes of the element.

Below is attached the complete source code Beta\_Gamma.C. The proprietary code has been programmed in C++ language through the IFM tool.

The nodal values of temperature and pressure are loaded by the use of the API functions “IfmGetResultsTransportHeatValue”

and “IfmGetResultsFlowPressureValue” respectively.  $\bar{\beta}^*(T, p)$  is calculated for each mesh element by use of Eq.6. The resulting value is fully implemented in FEFLOW via the API function “IfmSetMatFlowExpansionCoeff”.

```

static void PostTimeStep (IfmDocument pDoc)
{
    int e, i;
    double T, p, A, B;
    double beta, gammastar, BETASTAR;

    /* Coefficients for the fitting and its derivatives */
    /* Useless a0 = 9.99792877961606e+02; */
    double a1 = 5.07605113140940e-04;
    double a2 = -5.28425478164183e-10;
    double b0 = 5.13864847162196e-02;
    double b1 = -3.61991396354483e-06;
    double b2 = 7.97204102509724e-12;
    double c0 = -7.53557031774437e-03;
    double c1 = 6.32712093275576e-08;
    double c2 = -1.66203631393248e-13;
    double d0 = 4.60380647957350e-05;
    double d1 = -5.61299059722121e-10;
    double d2 = 1.80924436489400e-15;
    double e0 = -2.26651454175013e-07;
    double e1 = 3.36874416675978e-12;
    double e2 = -1.30352149261326e-17;
    double f0 = 6.14889851856743e-10;
    double f1 = -1.06165223196756e-14;
    double f2 = 4.75014903737416e-20;
    double g0 = -7.39221950969522e-13;
    double g1 = 1.42790422913922e-17;
    double g2 = -7.13130230531541e-23;
    /*Reference pressure is 100kPa while REFERENCE TEMPERATURE IS 0!!!!!!!*/
    double p0 = 100.;

    /* Number of elements and number of nodes */
    int nElements = IfmGetNumberOfElements(pDoc);
    int nNodes = IfmGetNumberOfNodesPerElement(pDoc);

    /* Loop through all elements */
    for (e = 0; e < nElements; e++)
    {
        T = 0.;
        p = 0.;

        /* loop locally through all nodes of this element */
        for (i = 0; i < nNodes; i++)
        {
            /* Get global node index */
            int indNode = IfmGetNode(pDoc, e, i);
            /*T +=IfmGetResultsTransportHeatPreviousTimeValue(pDoc,indNode);*/
            T +=IfmGetResultsTransportHeatValue(pDoc,indNode);
            p +=IfmGetResultsFlowPressureValue(pDoc,indNode);
        }

        /* Solving the average physical properties (T and p) of this element */
        T /=(double)nNodes;
        p /=(double)nNodes;

        A = gam0+(gam1+(gam2+(gam3+(gam4+(gam5+gam6*T)*T)*T)*T)*T;
        B = a2+(b2+(c2+(d2+(e2+(f2+g2*T)*T)*T)*T)*T;

        /* Set BETA */
        beta = -(1/999.843633188666)*(bp0+(cp0+(dp0+(ep0+(fp0+gp0*T)*T)*T)*T)*T);

        /* Set GAMMA */
        gammastar = -(1/999.843633188666)*(A+B*(p-p0))/(T+.1);

        /* Set BETASTAR */
        BETASTAR=beta+gammastar;
        IfmSetMatFlowExpansionCoeff(pDoc, e, BETASTAR);
    }
}

```

Definition of the coefficients

Solving the average value for  
Temperature (T) and pressure (p)

Setting  $\bar{\beta}^{-1}(T, p)$  as expressed in  
Eq.6

## Validation of the code.

The equations for  $\bar{\beta}(T, p)$  and  $\bar{\gamma}(T, p)$  (Eq.(3.2) and Eq.(3.3) have been derived from the following polynomial fitting:

$$\rho^f(T, p) = a(p) + b(p)T + c(p)T^2 + d(p)T^3 + e(p)T^4 + f(p)T^5 + g(p)T^6 \quad \text{in } [\text{kg/m}^3] \quad (3.7)$$

$0 \leq T \leq 350$  and  $p_{\text{sat}} < p \leq 100$

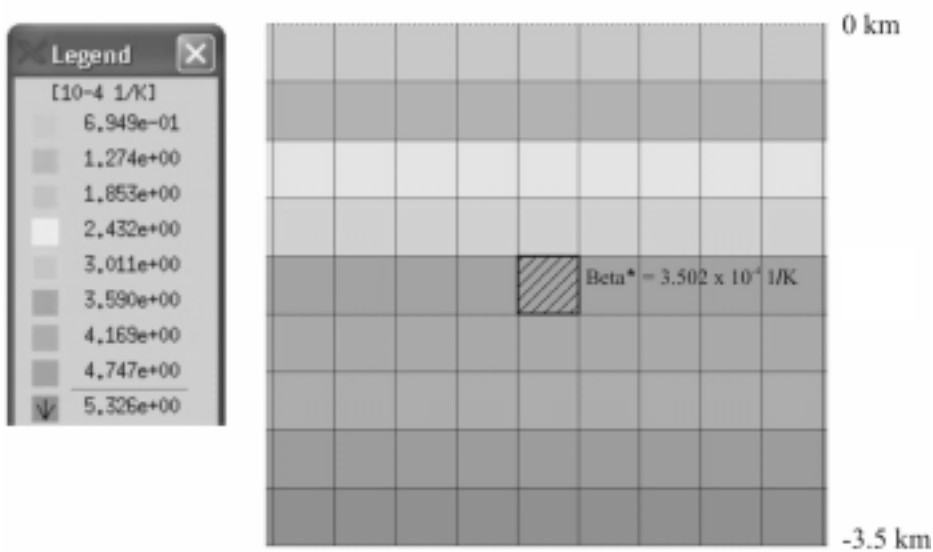
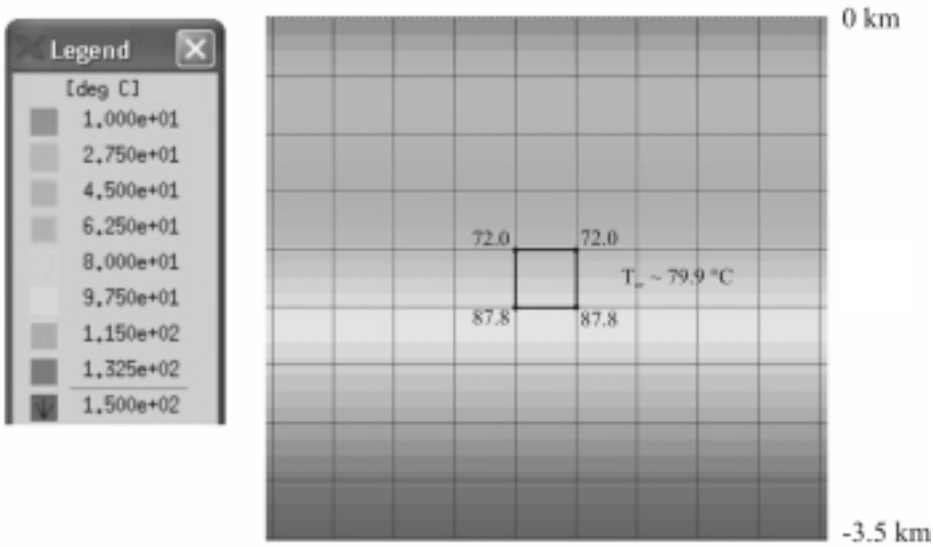
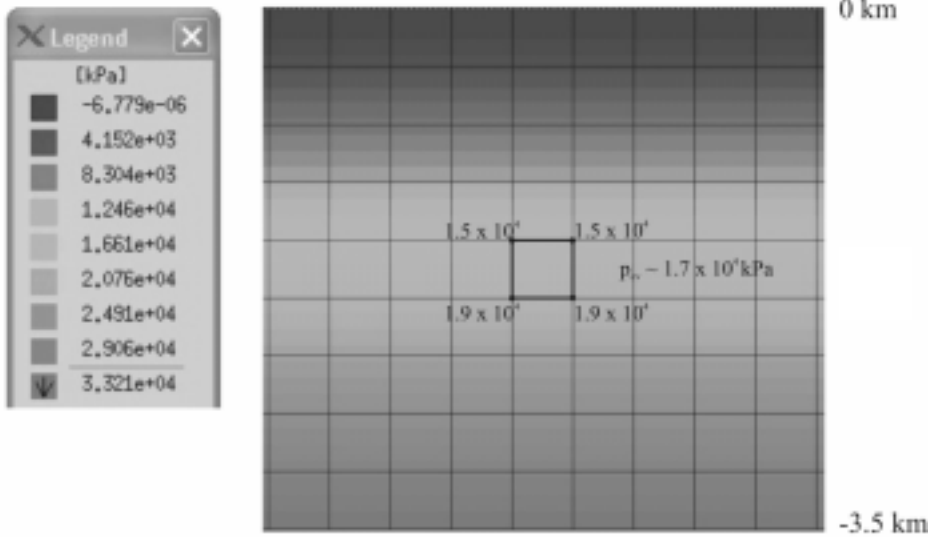
with temperature  $T$  in °C , the pressure  $p$  in kPa and the coefficients for fresh water are given in Table 2.1 (Appendix 2).

For validating the source code, the implemented Beta\_Gamma module has been tested on a vertical 2D coupled heat transport and fluid flow simulation (i.e  $\bar{\alpha} = 0$ ). In this way it is possible to compare the fluid density calculated from the simulation (Eq.5) with the one provided by the polynomial fitting (Eq.(3.7)). The code is valid only if these values coincide.

The conceptual model used for the coupled simulation is a 3.5 x 3.5 km square. The rectangular mesh is composed by 9 elements. At the top, a constant temperature of 10 °C and a head value of 0 m are set as boundary conditions while at the bottom a constant temperature of 150 °C is set.

The temperature and the pressure are calculated during the simulation process at each node of the considered element. The implemented Beta\_Gamma module derives the elemental mean value of these physical parameters and uses them as input data for returning the elemental value of BETASTAR. Results are shown in Fig 1.

For simplicity, only the central element of the mesh is considered. The mean temperature and pressure for this element are 79.9 °C and  $1.7 \times 10^4$  kPa respectively. The Beta\_Gamma module returns a value of BETASTAR equal to  $3.502 \times 10^{-4}$  1/K which leads to a fluid density equal to 979.24589 kg/m<sup>3</sup> (Eq.5) On the other hand, by replacing the (Eq 7.) the fluid density calculated by use of the polynomial fitting is equal to 979.24589 kg/m<sup>3</sup>. Since the numerical value of BETASTAR coincides with the value obtained from the polynomial fitting the code is validated.



**Fig.1** Calculated pressure (A) and temperature (B) input data and returned BETASTAR (C) by the use of the Beta\_Gamma module.