FITH2 A SET OF FORTRAN INTERFACES TO PROCESS TOUGH2 INFORMATION, DATA AND RESULTS

Mario-César Suárez Arriaga(1) and Fernando Samaniego Verduzco(2)

(1) UNAM, UMSNH & CFE, e-mail: msuarez@zeus.ccu.umich.mx
(2) UNAM & PEMEX, e-mail: fsv@chall.fi-p.unam.mx

ABSTRACT

Graphic presentation of numerical results, mesh build up and basic data pre-processing are activities routinely carried out in geothermal reservoir simulations. To face this problem in a systematic way, we have developed FITH2 (Fortran Interfaces to process TOUGH2 information), a set of FORTRAN programs having the capabilities to pre-process the data required by TOUGH2 and to post-process its numerical results. FITH2 is external to TOUGH2; not forming part of its original architecture, it performs several useful actions. FITH2 is able to create meshes of simple geometry, including boundaries; it can interpolate thermodynamic and petrophysical variables, assigning values to every element in the mesh. It uses Tchebyschev polynomials for simple spatial interpolation, or universal kriging for estimating non-stationary parameters. This program generates grids and calculates 3D coordinates of every center and nodal point conforming the elements. These are fundamental operations because in this way, many geometric theorems could be applied to the mesh in vectorial form, transforming the original grid as desired. F ITH2 creates ASCII files ready to be included in any graphic commercial software to produce attractive presentations of great visual quality. Another important option is that FITH2 is able to interact with *Mathematica* system for analyzing and plotting TOUGH2 results. In this graphic atmosphere partial or final outcomes can be quickly visualized, in the form of contours and three-dimensional surfaces. It is also possible to create animations or virtual motions of those surfaces in order to see the evolution of the reservoir thermodynamic properties. All this work can be done in a 486 or Pentium PC, 16 Mb RAM and enough free disk space.

INTRODUCTION

TOUGH2 (Pruess, 1991) contains technical and scientific knowledge needful to model and study the behavior of hydrothermal reservoirs. But data processing and presentation of numerical outcomes obtained from simulations are problems not still solved in a general way. In a typical geothermal project, the basic conceptual model data, as well as the results produced by the code, should be seen on a summarized manner by a wide variety of individuals: scientists, technicians, officials, investors and politicians. That is why it is very important to introduce this information in a clear and precise way and at the same time in an elegant, convincing and simple form. The enormous volume of typical numeric columns of simulation outcomes, could be simple, evident and routinely read by the expert, but it does not produce the same effect on the general public.

An expensive and long term solution is the creation of specific software to visualize the information required by TOUGH2 and the corresponding numerical results. But it is difficult to develop a good graphic software and it takes too much time to arrive at the degree of specialization required by TOUGH2, e.g. Sato et al. (1995), Sullivan (1995), GeoCad (1996), Hardeman & Swenson (1998). In the international market there are graphic software able to make great quality graphics in 2 and 3 dimensions, although not related to this code. Such commercial software can be easily coupled to TOUGH2, by developing FORTRAN interfaces to read initial data, final results and to write compatible ASCII files.

On the other hand, TOUGH2 needs continuous spatial distribution of reservoir parameters. Optimum spatial interpolation of geothermal measurements is also necessary for the approximated calculation of non-measured parameters. We introduce a practical, general methodology covering the aforementioned aspects. The origin of the program described herein, was inspired by the first version of a subroutine created by K. Pruess in 1987 to build MESH file automatically which was included in one of the old versions of MULKOM, and corresponds to the present 'XYZ' option in the MESHMAKER module (Pruess, 1991).

BRIEF DESCRIPTION OF FITH2

The program FITH2 is a code written in FORTRAN-77, externally coupled to TOUGH2 through files MESH, INCON and OUTPUT. It is planned as an integrated software system, with a main reading data program and control of options. It is formed by 6 subroutines: MXYZG3, TRANSF2, TCHEBY3, IRFk2, GRAFEL2 and MALLAF3. The first subroutine creates the mesh; the second modifies the grid plane by plane, by means of three optional rigid transformations; the third and fourth subroutines interpolate data and provide estimations of non-measured parameters. Those four subroutines can create the MESH and INCON files required by TOUGH2 (Pruess, 1987). The last two subroutines read the simulation outcomes and generate ASCII files ready to be used by any graphic commercial software.

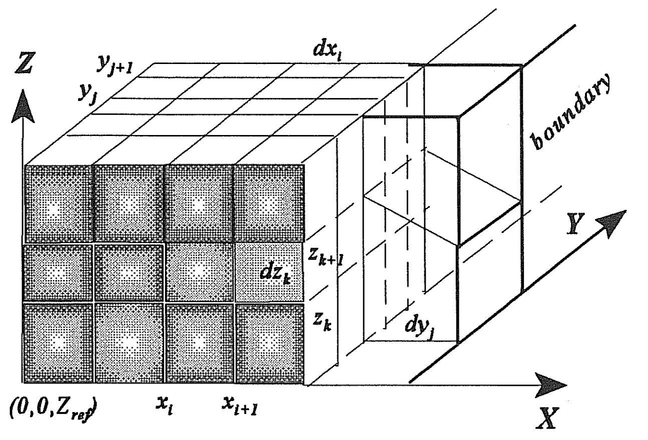


Fig. 1.- A mesh with boundaries created by MXYZG3

At each plane Z, FITH2 creates rectangular elements inside the mesh and rectangles or trapeziums as boundary elements (Fig. 1). These shapes suffice to represent the majority of geothermal reservoir geometries observed in Mexico. Certainly this basic mesh could be modified later on by hand in order to characterize local or regional details of great complexity. FITH2 is neither an expert system nor totally automatic. It requires human judgements and the constant interaction with the user.

FITH2 is open to future innovations and changes. This is an ongoing work in process. The different modules of FITH2 are coupled to TOUGH2 interacting in several ways. General interaction and coupling are illustrated in figure 2.

DESCRIPTION OF THE SUBROUTINES

FITH2.FOR- Is the main program that requests initial information, reading options and calling subroutines in the order wanted by the user. This is still an experimental version and is not documented enough.

MXYZG3- Is the main subroutine, its functions are to build up the mesh starting from few data and to calculate all the coordinates of the elements. It has two initial options: The first one uses a pre-existent file containing basic grid information, the same way as MESHMAKER does. The second option can calculate the whole mesh using only the coordinates of the producing and injection wells, and constructing one element per well. It also requires reservoir's long and width. If the second choice is used the primary distances $dx_i = (x_{i+1} - x_i)$, $dy_j = (y_{j+1} - y_j)$ are calculated at each stratum $dz_k = (z_{k+1} - z_k)$ (Fig. 1).

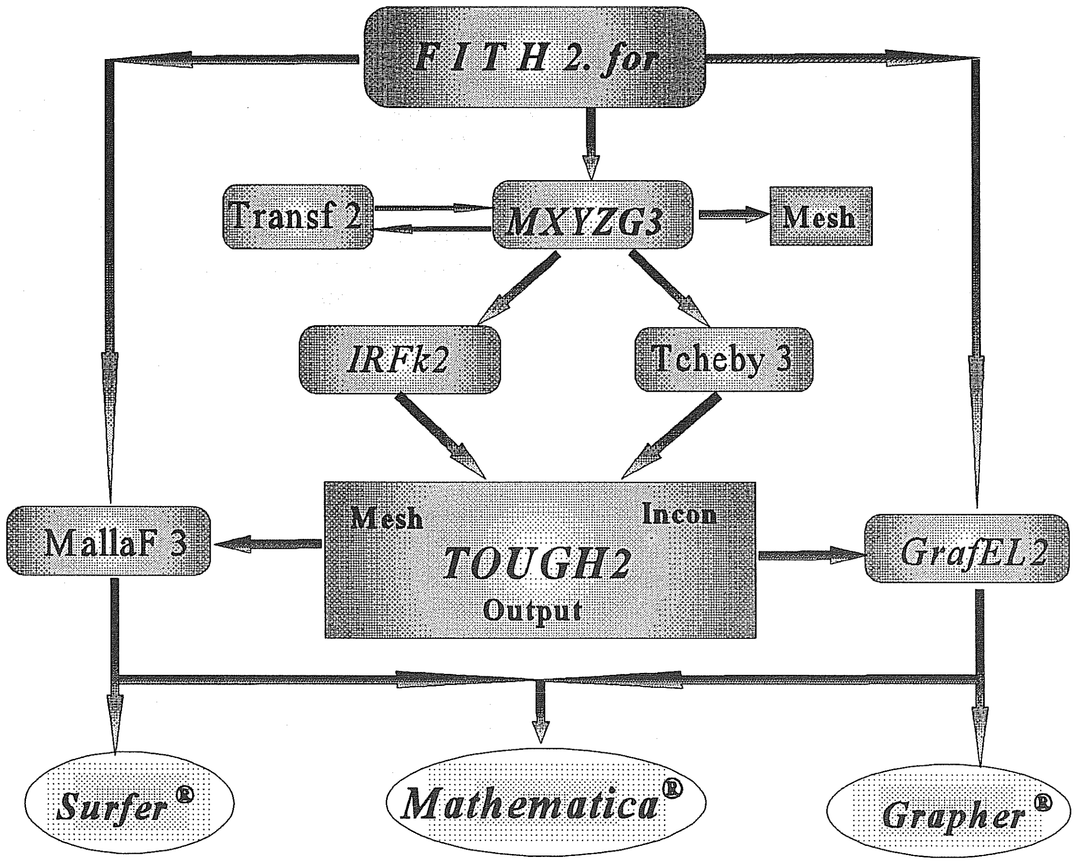


Figure 2.- **Interactions between FITH2 and TOUGH2**

Once defined the basic distances in the three axes (X, Y, Z) the code calculates 3D coordinates of all nodal points conforming the mesh, determines the centers of all the elements and makes user/machine interactive calculations to construct the boundary elements, including the possibility of building blocks of bigger size connected to two or more reservoir elements. Vertical coordinates are referred to a Z_{ref} reference level defined by the user. For example, $Z_{ref} = 0$ means that the deepest reservoir stratum is at sea level. The absolute grid origin corresponds to $(0, 0, Z_{ref})$, where (0, 0) is the starting point from which the first distance (dx_1, dy_1) is measured. Coordinates calculation for each nodal point in the mesh follows the next algorithm (MARCE-1):

```
$k = 0, NZ$
        $j = k*(NY+1),\ (k+1)*NY+k$
                $i = j*(NX+1),\ (j+1)*NX+j$
                $X_i = X_0$                        (If $i \le NX$)
                $X_i = X(i-NX-1)$                  (If $i > NX$)
                $X_0 = X_0 + dX_{i+1}$
        $Y_i = Y_0$                                (If $j \le NY$)
        $Y_i = Y(i-(NY+1)*NX-NY-1)$                (if $j > NY$)
        $Y_0 = Y_0 + dY_{j+1}$
$Z_i = Z_0$
$Z_0 = Z_0 + dZ_{k+1}$
```

The geometric center of each element (XC_i, YC_i, ZC_i) is calculated with a slightly different algorithm. (MARCE-2):

```
$k = 1, NZ$
        $j = (k-1)*NY+1,\ k*NY$
                $i = (j-1)*NX+1,\ j*NX$
                $XC_i = X_0 + dX_i/2$              (if $i \le NX$)
                $XC_i = XC(i-NX)$                  (if $i > NX$)
                $X_0 = X_0 + dX_i$
        $YC_i = Y_0 + dY_j/2$                      (if $j \le NY$)
        $YC_i = YC(i-NY*NX)$                       (If $j > NY$)
        $Y_0 = Y_0 + dY_j$
$ZC_i = Z_0 + dZ_k/2$
$Z_0 = Z_0 + dZ_k$
```

NX, NY and NZ are the number of distances d_N in each one of the axes respectively. The next actions are to give a name to every element, to calculate areas, distances between connections and volumes. These operations are similar to those performed by MESHMAKER (Pruess, 1987). It is also possible to assign each element to different rock pre-defined types, for example as function of depth, lithology, fault's elements or fractured zones with high permeability.

After introducing the total number of boundary elements connected to the mesh, euclidean distances between centers of boundary blocks and interior elements are calculated:

$$d(\vec{C}_F, \vec{E}_i) = \sqrt{(C_X - X_i)^2 + (C_Y - Y_i)^2 + (C_Z - Z_i)^2} \qquad (1)$$

(C_X, C_Y, C_Z) are the coordinates of the geometric center of the boundary block and (X_i, Y_i, Z_i) are the coordinates of the element interface.

TRANSF2- This subroutine interacts directly with MXYZG3 and is used to transform the mesh. It only carries out two-dimensional rigid transformations. At each plane Z= constant, it can make (Figs. 3 & 4):

a).- Transfers $T(\vec{P}) = \vec{OP} + \vec{v};\ \forall$ *vector* $\vec{v}, \vec{OP} = (x,y)$,
b).- Rotations around axis X: $R(\vec{P}) = x\vec{U} + y\vec{U}^{\perp}$,
c).- Reflections with respect to L: $F(\vec{P}) = x\vec{U} - y\vec{U}^{\perp}$
where vector: $\vec{U} = (u_1, u_2),\ \vec{U}^{\perp} = (-u_2, u_1)$,

$$\|T,R,F(\vec{P} - \vec{Q})\| = \|\vec{P} - \vec{Q}\| \qquad \textbf{(2)}$$

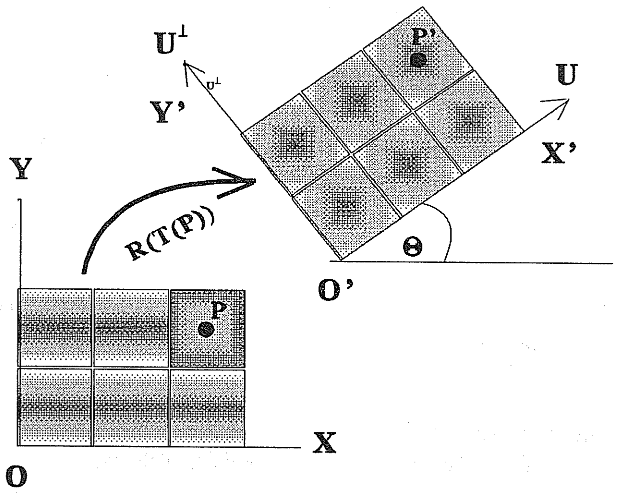


Fig.3.- Transfer & rotation performed by Transf2

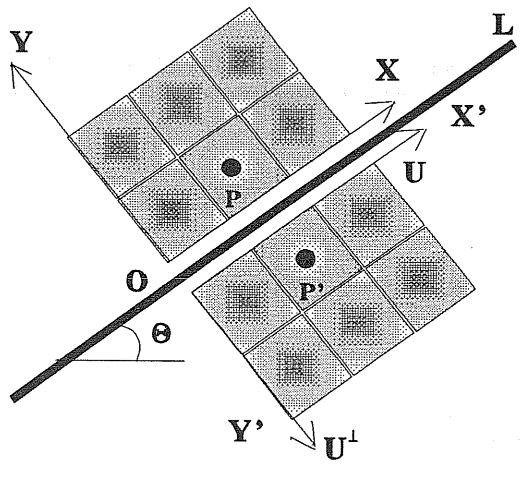


Fig. 4.- Reflection of the mesh around line L

This last transformation is equal to a 180° rotation of the plane containing the OP vector, around line L. This subroutine is useful when the user wants to test several effects produced by different mesh orientations, or to create fictitious wells in order to simulate boundary effects such as natural recharge and/or discharge.

TCHEBY3.- This subroutine interpolates in one, two or three dimensions, considering Tchebyschev polynomials $\{T_m(t)\}$ as the interpolation basis. It is well known that for a high degree of the basis involving powers of order greater than 6, simple polynomials oscillate producing numerical instability. The use of Tchebyschev polynomials avoids this problem. In 1D, these functions are defined by the recurrent relationship (Legras, 1971): $T_{m+1}(t) - 2t\,T_m(t) + T_{m-1}(t) = 0$, with the initial values $T_0(t) = 1$, $T_1(t) = t$. If we know m of its values, any function f (t) is thus interpolated by:

$$f_m(t) \approx \sum_{j=0}^{m} c_j T_j(t) \;;\quad \forall t \; real: \qquad (3)$$
$$-1 \leq t \leq 1 \;;\; t = \cos\theta : \quad T_m(t) = \cos(m\theta)$$

To contain any real value of t inside the interval [-1, +1] the following change of variable is made:

$$\tau = \frac{a+b}{2} + \frac{b-a}{2} t \;\Rightarrow\; t = \frac{2\tau - a - b}{b-a} \qquad (4)$$
$$Obviously\ in\ all\ cases: \; -1 \leq T_m(t) \leq +1$$

This last property is responsible for the stable calculations provided by Tchebyschev polynomials. Tcheby3 can also perform least squares approximations in 1, 2 or 3 dimensions:

$$f(\vec{r}) \approx \sum_{j=0}^{N} c_j b_j(\vec{r}) \;;\quad \forall \vec{r} = (x,y,z) \qquad (5)$$
$$b_j(\vec{r}) = T_p(x)\, T_q(y)\, T_r(z) \,,\; \forall\, p,q,r = 0,1,2,\ldots,N$$

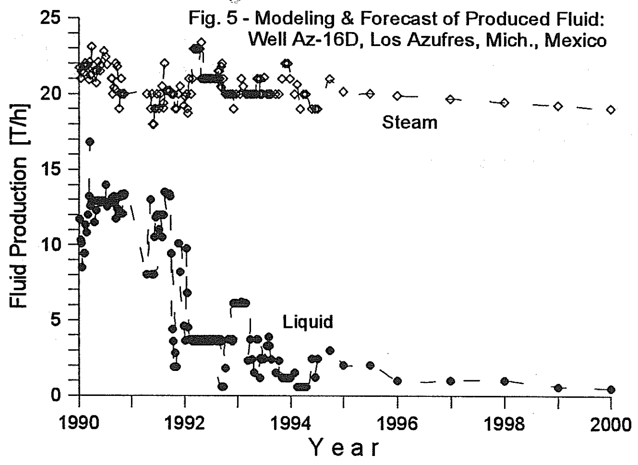
Where N is the number of basis functions. If n samples are known, the unknown coefficients c_j are solution of the system:

$$\sum_{k=0}^{N} c_k \langle b_k, b_j \rangle = \langle f, b_j \rangle \;;\; j = 0,n \qquad (6)$$
$$\langle b_k, b_j \rangle = \sum_{i=0}^{n} b_k(\vec{r}_i)\, b_j(\vec{r}_i) \,,\; \langle f, b_j \rangle = \sum_{i=0}^{n} f(\vec{r}_i)\, b_j(\vec{r}_i)$$

We have used this technique to solve several practical geothermal reservoir engineering problems: interpolation and extrapolation of petrophysical parameters, forecast models for injection/extraction rates, efficient low degree polynomials for the calculation of thermodynamic properties of single phase or two-phase water, integration of very irregular functions, e.g. total fluid rate during several years of production/reinjection history (Suárez, 1985). An integration algorithm based on Tchebyschev polynomials is straightforward and powerful. Any real function can be integrated approximatelly by the following formula (Suárez, 1984):

$$\int_a^b f(x)\,dx \approx \frac{b-a}{n+1} \sum_{j=0}^{Na} \tau_j f_j \qquad (7)$$
$$f_j = f\left(\frac{a+b}{2} + \frac{b-a}{2} y_j\right) \;;\; y_j = \cos\left(\frac{2j+1}{2n+2}\pi\right)$$
$$\tau_j = 1 + 2\sum_{k=1}^{n/2} \frac{T_{2k}(y_j)}{1-4j^2} \;;\; T_{k+1}(y_j) = 2y_j T_k(y_j) - T_{k-1}(y_j)$$

T_k are Tchebyschev polynomials, a and b can be any real numbers, even equal to $-\infty$ or $+\infty$, *Na* is the order of the approximation. Tcheby3 can call internally a real function TI performing the integration defined by (7). Figure 5 shows an application of this method.



IRFk2- This subroutine uses a technique of non-stationary kriging, based on the theory of intrinsic random functions of order k > 0 (Matheron, 1973), in order to perform optimum interpolation, estimate non-measured parameters and assign values to all the elements in the mesh. It is particularly useful in the spatial modeling of geothermal variables with drift (Suárez & Samaniego, 1998). IRFk2 solves the non-stationary kriging system, also known as universal kriging:

$$P_0 \approx L(P_0) = \sum_{i=1}^{n} \beta_i P_i \,,\; n \leq N$$
$$\sum_{i=1}^{n} \beta_i K_{ij} + \sum_{l=1}^{m_k} \mu_l b_l^j = K_{0j} \qquad (8)$$
$$\sum_{i=1}^{n} \beta_i b_j^i = b_j^0 \quad,\quad j = 1,n$$

Where $L(P_0)$ is an optimum linear estimator of any unknown parameter P_0 in the reservoir; β_i are unknown coefficients in the linear estimation of P_0; $b_j^{\,l} = b_j(\mathbf{r}_l)$ are polynomial basis functions, $\mathbf{r}_l=(x_l, y_l\, z_l, t)$ is a cartesian position vector at time t of sample P_i, μ_l are the classical lagrange multipliers; $K_{ij} = K(\mathbf{r}_j - \mathbf{r}_i)$ is a generalized covariance measuring the spatial correlation between random variables R at $\mathbf{r}_i$, $\mathbf{r}_j$. This portion of the code could pre-process all parameters required by TOUGH2: density, porosity, thermal conductivity, permeability, rock specific heat, pressure, temperature, etc. Also provides an estimate of the uncertainties at every interpolated parameter, through the variance σ^2 of the kriging error:

$$\sigma^2 = \sum_{i=1}^{n}\sum_{j=1}^{n}(\beta_i \beta_j K_{ij} - 2\beta_i K_{i0}) + K_{00} \qquad (9)$$

Table 1 shows estimations of spatial distribution of porosity using this technique. The values were measured at wells of the Los Azufres, México geothermal field. The corresponding generalized covariance for these data is:

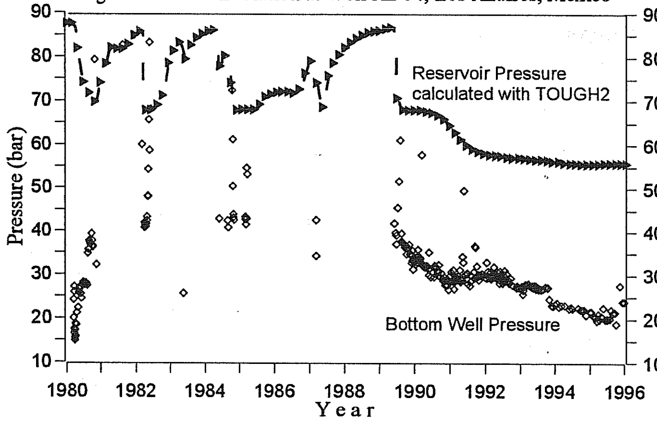
$$\vec{h} = \vec{r}_j - \vec{r}_i \Rightarrow K(\vec{h}) = 32.5630\ \delta(\vec{h}) + 0.23474\ 10^{-8}\ |\vec{h}|^3$$

Table 1.- Estimation of Porosity by Point kriging

X	Z	Φ_0	$L(\Phi_0)$	σ^2
3987.30	2900.00	22.7	22.70	-0.3211E-05
2369.50	3004.50	16.7	16.70	0.8326E-06
2389.50	3014.50	?	15.20	0.8631E+01
2422.10	2865.00	11.7	11.70	-0.3871E-05
2432.10	2865.00	?	11.86	0.4041E+01
1793.10	2730.00	7.1	7.10	-0.4480E-05
1803.10	2730.00	?	9.73	0.3951E+01
3100.00	2600.00	?	6.23	0.7847E+02
111.20	2654.00	1.4	1.40	-0.2227E-06
201.20	2744.00	?	42.03	0.5913E+03

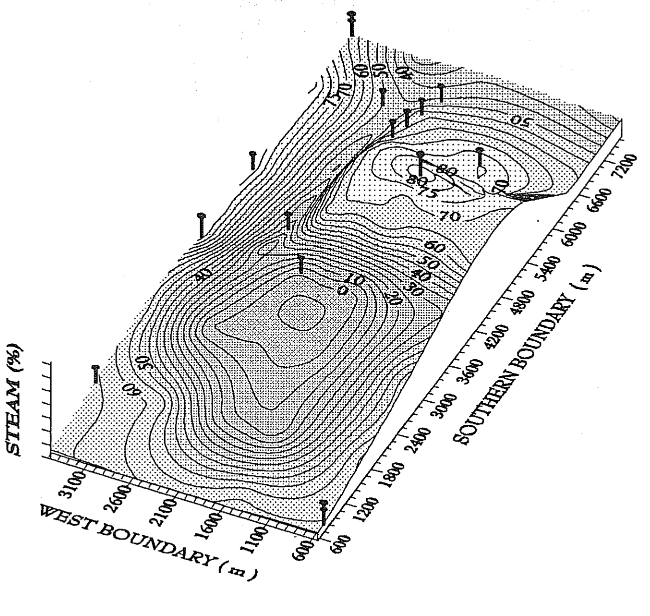
GRAFEL2- This subroutine processes the OUTPUT file to plot TOUGH2 results in the form: variable vs. time. Variable could be pressure, temperature, steam saturation, etc. This part of the code separates the outcomes of every selected element in chronological order, generating one ASCII file per each element to be plotted. The format of these files is totally compatible with graphic commercial software. GrafEL2 could even generate its own graphs through the graphic subroutines built in the FORTRAN compiler. The program reads the names of the chosen elements in an auxiliary file. Then opens the OUTPUT file and reads it sequentially looking for several key words. The first one is **MESH**, where it reads the total number of elements and begins to count the simulation time steps. The next key word is **TOTAL TIME**. Here it reads: SUMTIM. Then finds all the **ELEMk** elements to be plotted, reading: ELE, N, PNE, TNE, SNE, etc. The next key word is **ELEMENT SOURCE**, reading the data to plot fluid rates and enthalpies: eleh(k), well, NP, Qf, hf (k). Finallly it writes the properties in columns, as functions of time with the format: write (k, '(6F8.2, 2x, a5)') ST(NT), P(k)/ 1.e5, T(k), S(k)*100., P2(k)/1.e5, hf(k)/1.e3, eleh(k). The iteration ends when the reading operation finds "**END OF TOUGH**" or "**WRITE FILE**". Figure 6 shows an example.

Fig. 6.- Pressure Evolution at Well Az-04, Los Azufres, Mexico



MALLAF3- Reads the final or partial results of TOUGH2 and generates compatible ASCII files to draw 2D contours and 3D surfaces of high visual quality, to include them in final reports or for public presentations. This subroutine reads the mesh data in the file created by MXYZG3: elem(k), ie(k), x(k), y(k), z(k). The code reads sequentially the OUTPUT file looking for different key words. The first one is "**ELEM**", then it reads the same lines as in GrafEL2. A corresponding ASCII file is written following the format: write(1,5050) elem(k), ie(k), x(k), y(k), z(k), PNE/1.e5, TNE, SNE*100. This file is compatible with SURFER® (1995) software. Contours of steam saturation at Los Azufres reservoir were obtained with this part of the code (Fig. 7).

Fig. 7.- Steam Saturation Distribution at 1750 masl Los Azufres, Mexico geothermal reservoir.



Interfacing with *Mathematica*

Mathematica (Wolfram, 1992), is a commercial software defined as a completely integrated system and general computer language capable to perform symbolic, numerical and graphic mathematical computations at different levels of complexity. MallaF3 also can interact with the *Mathematica* program, through special instructions provided by this software (Wolfram, 1992). In this graphic environment different TOUGH2 outcomes could be visually analyzed, including simple f (t), 2D or 3D graphs. For example the following command:

ReadList["Arch1.dat", Number, RecordLists -> True],

allows *Mathematica* to read numeric data in file "Arch1.dat" created by FITH2, returning a list of those numbers into a *Mathematica* format and making a separate list (vector) of each line in that file. A main advantage is that a whole list of numbers in *Mathematica* can be treated as a single object. If Arch1.dat are composed of two columns, the expression:

ListPlot [$\{x_i, y_i\}$ PlotJoined -> True],

will produce a similar graphic object as in figure 6, if the same data are used. To plot 2D contours or 3D surfaces with *Mathematica* from results processed by MallaF3, the next instructions are used:

ListContourPlot [*vector*] and **ListPlot3D** [*vector*]

Where *vector* means an array of properties (heights) P (x,y), readed by **ReadList**. Other special options could be used with *Mathematica* in order to visualize final or partial simulation results immediately. It is possible to run external programs, such as FITH2, from inside *Mathematica,* which controls the external program and analyzes the generated output.

The cybernetic environment offered by *Mathematica* is useful to analyze data and, particularly, in the dynamic visualization of graphical outcomes. It is also possible to present animations or virtual motions of graphic objects from file OUTPUT simulating the evolution of thermodynamical reservoir properties. First, it is necessary to create a sequence $\{S_i\}$ of graphic objects. Then the instruction:

ShowAnimation [$\{S_1, S_2, S_3, ..., S_N\}$],

will display the animated graphics (Wolfram, 1992).

CONCLUSION

From the results of this work we conclude that FITH2 code has the capabilities to pre-process the data required in typical simulations and to post-process the numerical outcomes produced by TOUGH2. The numeric files created by FITH2 can be used in a wide variety of graphic environments.

REFERENCES

GeoCad for MS-Windows (1996). Software for preparing models of groundwater, geothermal and other muti-phase flows. GERD, Industrial Research Ltd., New Zealand.

Hardeman, B. & Swenson, D. (1998). A Geometric Modeling Framework for the Numerical Analysis of Geothermal Reservoirs. Proceedings, 23rd Workshop on Geothermal Reservoir Engineering, Stanford University.

Legras, J. (1971). Méthodes et Techniques de l'Analyse Numérique. (324 pp.), Ed. Dunod, Paris, France.

Matheron, G., 1973. The Intrinsic Random Functions and their Applications. Adv. In Applied Probability. Vol. 5, No. 2 (pp. 439-468).

Pruess, K. (1991). TOUGH2 - A General-Purpose Numerical Simulator for Multiphase Fluid and Heat Flow. Earth Sciences Division, Lawrence Berkeley National Laboratory, University of California, LBL-29400, UC-251.

Sato, S., Okabe, T., Osato, K. & Takasugi, S. (1995). Graphical User Interface for TOUGH/TOUGH2 - Development of Database, Pre-processor and Post-processor. Proceedings of the TOUGH Workshop'95. Lawrence Berkeley National Laboratory (pp. 271-276), Berkeley, California, March 20-22, 1995.

Suárez, M.C. (1984). Técnicas Avanzadas de Interpolación y Aproximación. Vol. II. Textbook notes written for the Michoacán University (150 pages), Morelia, Mich., Mexico.

Suárez, M.C. (1985). Cálculo Analítico de Propiedades Térmicas del Agua Pura. Geotermia, Vol.1, No.1, (p. 74-77).

Suárez M.C. & Samaniego, F. (1998) Intrinsic Random Functions of high order and their application to the Modeling of non-stationary Geothermal Parameters. Proceedings, 23rd Workshop on Geothermal Reservoir Engineering, Stanford University.

Sullivan, M. & Bullivant, D. (1995). A Graphical Interface to the TOUGH Family of Flow Simulators. Proceedings of the TOUGH Workshop'95. Lawrence Berkeley National Laboratory (pp. 90-95), Berkeley, Ca., March 20-22, 1995.

SURFER® for Windows (1995). User's Guide. Contouring and 3D Surface Mapping. Version 6. Golden Software Inc.

Wolfram, S. (1992). ***Mathematica***. A System for Doing Mathematics by Computer. Wolfram Rersearch, 2nd Edition, Addison-Wesley Publishing Company (961 pp.), New York.