Modelos Prácticos para Reservorios Geotérmicos Termoelásticos resueltos con Mathematica

Modeling the fluid flow in non-isothermal poroelastic aquifers using Mathematica

Mario-César Suárez Arriaga

Faculty of Physics and Mathematical Sciences
Michoacan University UMSNH, Mexico
e-mail: mcsa50@gmail.com

Resumen

Los modelos matemáticos, los métodos numéricos y la computación conforman la infraestructura profunda de la ciencia y la ingeniería. En este contexto el modelado matemático es una herramienta muy poderosa para estudiar sistemas complejos naturales, de ingeniería y hasta la misma sociedad humana. Este trabajo contiene una visión general de cómo resolver y aplicar modelos prácticos que van desde simples fórmulas algebraicas hasta ecuaciones fundamentales en Derivadas Parciales (EDP) al modelado de diversos fenómenos que se presentan en reservorios geotérmicos poroelásticos. Las capacidades gráficas programables y el alto nivel de cómputo del software llamado *Mathematica* (versión 7) son extremadamente útiles para la investigación práctica sobre el flujo de masa y energía en medios porosos deformables. Estos sistemas se presentan junto con procesos de mecánica de sólidos, vibraciones, transferencia de calor y flujo de fluidos en rocas porosas, que pueden ser modelados con EDP y resueltos con programas hechos a la medida en Mathematica. Los modelos son EDP clásicas, elípticas, parabólicas e hiperbólicas, o una mezcla de ellas. Estas EDP y las principales técnicas de solución son descritas en este trabajo en forma breve y heurística. Muestro que es posible comprender los complicados fenómenos naturales acoplados en reservorios geotérmicos y adquirir al mismo tiempo técnicas de modelado con EDP resolviéndolas utilizando Mathematica. Esta técnica abarca un gran grupo de diversas aplicaciones importantes, incluyendo la exploración de recursos naturales como son los acuíferos, la energía geotérmica, los yacimientos de petróleo y gas, así como la explotación de estos recursos.

Abstract

Mathematics is a universal language of both science and engineering. Differential equations, mathematical modeling, numerical methods, and computation form the deep infrastructure of sciences and engineering. In this context mathematical modeling using *Mathematica* (Wolfram Inc. version 7) is a very powerful tool for studying natural systems, and for applications to geothermal engineering. This paper contains a comprehensive overview of how to apply and solve practical models from simple algebraic formulae up to fundamental Partial Differential Equations (PDE) applied to coupled phenomena in non-isothermal poroelastic aquifers. The graphical and computational programmable capabilities of *Mathematica* are extremely useful for doing research and learning about the flow of mass and energy in deformable porous media. These systems exhibit coupled processes of solid mechanics, heat transfer, fluid flow and solute transport in pores that can be modeled with fundamental PDE and solved with *Mathematica*. These basic PDE are all classical and range from elliptic, parabolic and hyperbolic type, to a mixture of them. It is therefore possible to provide simultaneously an understanding of complicated coupled phenomena and their modeling with PDE by gradually approaching their solutions using *Mathematica.* This research technique covers a diverse group of very important applications, including natural resources exploration and exploitation of water resources and geothermal and petroleum reservoirs.

Introduction

This is an applied research project in progress. In the next sections, a brief introduction to the general theory of groundwater flow in non-isothermal deformable aquifers as PDE models is presented. At the same time we outline the main classical techniques to solve the simplified PDE. A didactic, self-contained *Mathematica* notebook programing these methods is included with practical examples. These codes will be free and available in the author's web page (http://www.fismat.umich.mx/~marioc/).

Modeling non-isothermal aquifers

Mathematical and numerical modeling is the process of obtaining approximate solutions to problems of scientific and/or engineering interest. There is an explosively growing enthusiasm in numerical modeling in general and in particular physical processes and its expansion to ever more sophisticated multi-physics to model coupled phenomena. The groundwater flow in non-isothermal aquifers is an example of complex coupled processes. The water and energy problems existing today can be explained using well founded mathematical theories and numerical tools that lead to modeling and solving them. The present knowledge on this matter provides a thorough understanding of the basic physical laws of the flow of fluids in poroelastic rocks under non-isothermal conditions.
The two main partial differential equations representing non-isothermal aquifers are the fundamental groundwater flow equation (1) and the general heat flow equation with conduction and convection (2). If there is a solute or contaminant substance included in the moving fluid, a third PDE is needed to model the solute transport (3). If there are waves or vibrations crossing the porous medium, a fourth PDE is needed to model these waves (4):

$$\partial_t \left(\varphi\, \rho_f\right) - \nabla \cdot \left(\frac{\rho_f}{\mu_f} \mathbf{K} \cdot \left(\nabla \mathbf{p}_f - \rho_f\, \mathbf{g}\right)\right) = \mathbf{q}_f$$

(* (1) General groundwater flow PDE *)

$$\rho\, \mathbf{c}_p\, \partial_t\, \mathbf{T} - \nabla \cdot \left(\mathbf{K}_T \cdot \nabla \mathbf{T}\right) + \nabla \cdot \left(\rho\, \mathbf{c}_p\, \mathbf{T}\, \mathbf{v}\right) = \mathbf{q}_H$$

(* (2) Heat conduction-convection PDE *)

$$\varphi\, \partial_t\, \mathbf{C} - \nabla \cdot \left(\varphi\, \mathbf{D} \cdot \nabla \mathbf{C}\right) + \nabla \cdot \left(\mathbf{C}\, \mathbf{v}\right) = \mathbf{C}_S\, \mathbf{q}_V$$

(* (3) Diffusion dispersion-advection PDE *)

$$\frac{1}{\rho\, \mathbf{v}^2} \partial_{t,t} \mathbf{p} - \nabla \cdot \left(\frac{1}{\rho} \nabla \cdot \mathbf{p}\right) = \mathbf{q}_S \qquad \text{(* (4) General Wave PDE *)}$$

Where φ [ad] is porosity; ρ_f [kg/m^3] is fluid density; μ_f [Pa·s] is fluid dynamic viscosity; **K** [m^2] is the absolute permeability tensor;

∇p_f [Pa/m] is the fluid pressure gradient; g [m/s^2] is the vector of gravity acceleration and q_f [kg/s/m^3] represents a source (> 0) or sink (< 0) term in the PDE (1). In the PDE (2) ρ is the density of the medium; C_p [J/kg/°C] is the isobaric rock specific heat capacity; *T* [°C] is the temperature; $\boldsymbol{K}_T$ [W/m/°C] is the rock thermal conductivity tensor; **v** [m/s] is the vectorial fluid velocity, and q_H [W/m^3] is the volumetric heat generation (or extraction). In the PDE (3) the scalar function C (x, y, z, t) is the solute concentration [moles/m^3]; the diffusion matrix **D** contains the eigenvalues D_X, D_Y, D_Z [m^2/s] or dispersion coefficients of the solute transport; **v** is Darcy's velocity; the term C_S is the solute concentration in the source fluid, and q_V [m^3/m^3/s] is the volume flow rate per unit volume of the source or sink in the aquifer. In the PDE (4) *p* represents pressure; *v* is the wave velocity; ρ is the density of the medium, and q_S [Pa·m^2/kg = m/s^2] is a source term. These PDE are parabolic, hyperbolic or of ellitic type (if $\partial_t = 0$). The first three PDE are known as diffusion equations (1-hydraulic, 2-thermal, 3-solute). The PDE (4) is the wave equation. They include several simpler models in one, two or three spatial dimensions. If the phenomenon they represent does not depend on time, they are called stationary and their partial derivatives with respect to time become zero. In this case, the same equations become of elliptic type and are called Laplace PDE or Poisson PDE.

Deformable Reservoirs

Rock compaction, fracturing, time dependent deformations, as well as creep and subsidence mechanisms are essentially produced by volcanic and tectonic activities, by lithostatic pressure and by fluid extraction/injection. Water contained in a rock, reduces its strength. The cohesive structure of rocks is weakened by the presence of a liquid. All the geomechanical parameters are influenced by this cohesion and are directly affected by the pressure and amount of liquid present in both pores and fractures. This is generally called the pore-fracture-water effect. In saturated rocks density and wave propagation speed are increased, while strength is reduced. In aquifers and in geothermal reservoirs, the different values in rock parameters are determined by the amount of liquid water, porosity ϕ, permeability, pressure and temperature. Compared with steam or with air, liquid water is almost incompressible, and this property tends to reduce both rock deformation and stiffness. The main theoretical part and background of this subject, was published in a previous issue of Geotermia Vol. 23, No. 2 (Suárez Arriaga, 2010).

Linear poroelastic theory

Fluid extraction from natural reservoirs of water, hydrocarbons or energy, causes the reduction of the internal pore-fracture pressure and of the effective aperture of pores and fissures. The main hypothesis in linear poroelasticity is that the fluids flow through a deformable porous rock whose solid skeleton can be deformed elastically (Biot, 1941). Assuming that rocks are only subjected to small deformations, Hooke's law can be applied to relate strains ϵ_{ij} and stresses σ_{ij}. We define a symmetric two-order tensor σ_T in four dimensions (Suárez, 2010), represented by a (4×4) matrix, which includes the bulk stress tensor σ_B acting in the porous rock and the fluid stress σ_F acting in the fluid filling up the pores, both influencing the bulk rock deformation. Let u_i be the three components of the displacement vector for the material particles forming the porous rock:

$$\sigma_T = \sigma_B + \sigma_F = \begin{pmatrix} \sigma_X & \sigma_{xy} & \sigma_{xz} & 0 \\ \sigma_{xy} & \sigma_Y & \sigma_{yz} & 0 \\ \sigma_{xz} & \sigma_{yz} & \sigma_Z & 0 \\ 0 & 0 & 0 & \sigma_f \end{pmatrix} =$$

$$\epsilon_B \begin{pmatrix} \lambda_U & 0 & 0 & 0 \\ 0 & \lambda_U & 0 & 0 \\ 0 & 0 & \lambda_U & 0 \\ 0 & 0 & 0 & -C \end{pmatrix} + 2\,G \begin{pmatrix} \epsilon_X & \epsilon_{xy} & \epsilon_{xz} & 0 \\ \epsilon_{xy} & \epsilon_Y & \epsilon_{yz} & 0 \\ \epsilon_{xz} & \epsilon_{yz} & \epsilon_Z & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} - \xi \begin{pmatrix} C & 0 & 0 & 0 \\ 0 & C & 0 & 0 \\ 0 & 0 & C & 0 \\ 0 & 0 & 0 & -M \end{pmatrix} \qquad (5)$$

Equation (5) is the general PDE of linear poroelasticity, where:

$$\zeta = \frac{C}{M}\,\epsilon_B + \frac{p_f}{M};\ \epsilon_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right);$$

$$\epsilon_i = \epsilon_{ii} = \frac{\partial u_i}{\partial x_i};\ i,\ j = 1,\ 2,\ 3;\ \begin{pmatrix} x_1 = x \\ x_2 = y \\ x_3 = z \end{pmatrix} \qquad (6)$$

σ_{ij} are the six applied stresses in [Pa]; ϵ_{ij} are the six strains [ad], describing the global elastic response of the rock. Coefficients λ_U, G, M and C are defined as special poroelastic parameters (see Suárez, 2010). The term $\epsilon_B = \epsilon_{kk}$ represents the volumetric deformation (strain) of the rock, p_f is fluid pressure and ζ is a strain variable describing the volumetric deformation of the fluid relative to the deformation of the solid grains. Applying the fundamental law of continuum mechanics to the poroelastic rock (Newton's second law), we obtain the partial differential equations for the variation of the fluid content ζ:

$$G\nabla^2 u_i + \left(\lambda_U + G\right)\frac{\partial^2 u_i}{\partial i\,\partial j} - C\frac{\partial \zeta}{\partial i} + F_i = 0;\ \{\forall\, i, j = x, y, z\} \qquad (7a)$$

Using equation (5) and the relationship between the undrained (λ_U) and drained Lame coefficients: $\lambda = \lambda_U - C\,b$, we obtain the corresponding group of partial differential equations for the fluid pressure:

$$G\nabla^2 u_i + \left(\lambda + G\right)\frac{\partial^2 u_i}{\partial i\,\partial j} - b\frac{\partial p_f}{\partial i} + F_i = 0;\ \{i, j = x, y, z\} \qquad (7b)$$

The fluid pressure p_f and the variation of the fluid content ζ are related through a simple diffusion partial differential equation when the gravity gradient is neglected (Bundschuh and Suarez, 2010):

$$\frac{\partial \zeta}{\partial t} = \frac{k}{\mu_f}\nabla^2 p_f \qquad (8)$$

Linear thermo-poroelastic equations

For non-isothermal processes occurring in the aquifer, we need to extend the previous theory when the temperature changes with time. We define the functions g_S, h_S, and S_S, as the volumetric Gibbs potential, the volumetric enthalpy and the volumetric entropy of the skeleton respectively. $g_S(\epsilon_{\text{ij}}, p_f, T)$ is the thermoporoelastic available enthalpy per unit volume [J/m^3]. We define the energy dissipation function as $\psi_S(\epsilon_{\text{ij}}, \sigma_{\text{ij}}, S_S, \text{T}, \text{p}, \varphi, g_S)$ in the skeleton. Using the Gibbs potential, the stresses, the porosity, the pore pressure and the density of entropy per unit volume of porous rock ψ_S is such that:

$$\frac{d\,\Psi_S}{d\,t} = \sigma_{ij}\,\frac{d\,\varepsilon_{ij}}{d\,t} - S_S\,\frac{d\,T}{d\,t} - \varphi\,\frac{dp}{d\,t} - \frac{dg_S}{d\,t} \;\geq\; 0 \qquad (9)$$

Assuming that there is no energy dissipation in the porous rock ($d\psi_S/dt = 0$) and for small changes in the rock, the Gibbs' potential describes the behavior of the skeleton. We deduced (Bundschuh and Suarez, 2010; Suárez, 2010) the thermoporoelastic matrix equations, which include the thermal tensions in the total stress tensor:

$$\boldsymbol{\sigma}_T = \boldsymbol{\sigma}_{\text{Bulk}} + \boldsymbol{\sigma}_{\text{Fluid}} + \boldsymbol{\sigma}_{\text{thermal}} =$$

$$= \begin{pmatrix} \sigma_x & \sigma_{\text{xy}} & \sigma_{\text{xz}} & 0 \\ \sigma_{\text{xy}} & \sigma_y & \sigma_{\text{yz}} & 0 \\ \sigma_{\text{xz}} & \sigma_{\text{yz}} & \sigma_z & 0 \\ 0 & 0 & 0 & \sigma_f \end{pmatrix} = \epsilon_B \begin{pmatrix} \lambda & 0 & 0 & 0 \\ 0 & \lambda & 0 & 0 \\ 0 & 0 & \lambda & 0 \\ 0 & 0 & 0 & -C \end{pmatrix} + 2\,G \begin{pmatrix} \epsilon_x & \epsilon_{\text{xy}} & \epsilon_{\text{xz}} & 0 \\ \epsilon_{\text{xy}} & \epsilon_y & \epsilon_{\text{yz}} & 0 \\ \epsilon_{\text{xz}} & \epsilon_{\text{yz}} & \epsilon_z & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} +$$

$$-\,(p - p_0) \begin{pmatrix} b & 0 & 0 & 0 \\ 0 & b & 0 & 0 \\ 0 & 0 & b & 0 \\ 0 & 0 & 0 & -\text{M}\,\frac{\zeta-\zeta_0}{p-p_0} \end{pmatrix} -$$

$$K_B\,(T - T_0) \begin{pmatrix} \gamma_B & 0 & 0 & 0 \\ 0 & \gamma_B & 0 & 0 \\ 0 & 0 & \gamma_B & 0 \\ 0 & 0 & 0 & \frac{\text{M}\,\varphi}{K_B}\,(\gamma_\varphi - \gamma_f) \end{pmatrix} \qquad (10)$$

Where λ is the drained Lame modulus, K_B is the bulk modulus, $b = \frac{M}{C}$ is the Biot-Willis (1957) coefficient, γ_B, γ_ϕ and γ_f are the bulk, pore and fluid thermal expansivities respectively. Equation (10) includes all of the thermal stresses acting in the skeleton and in the fluid contained in the porous rock. The extra terms σ_T^0, ζ_0, p_0 and T_0 represent the initial thermodynamic state of the porous rock at initial time $t_0 > 0$. The theory herein presented allows the approximate representation of compressible fluids migrating inside poroelastic rocks with non-isothermal conditions under the hypothesis of small rock deformations (Biot-Hooke's linear model). All these PDE can be simplified to study specific ideal problems in one or two dimensions, for transient or for stationary situations. In the next sections we introduce the main solution methods of these simplified but fundamental PDE programmed in *Mathematica* version 7.

Experimental parameters of Deformable Reservoirs

All geomechanical poroelastic parameters are influenced by the fluid pressure, temperature and amount of liquid present in both pores and fractures. We need about 20 poroelastic parameters to compute accurately all the geomechanical properties of rocks. But only three of these parameters are actually independent. Three basic parameters selected to constitute an experimental reference set were the drained bulk compressibility C_B, the expansion coefficient 1/H and the unconstrained specific storage 1/R (Wang, 2000). However, it is impossible to compute all the coefficients of the theory using only these three poroelastic constants, because at least five mixed coefficients are necessary for the whole poro-elastic coupling. A sufficient set of measured parameters is for example {E, G, ϕ, K_S, K_f}. With these moduli, we can compute the full set of poroelastic coefficients (Bundschuh and Suárez, 2010).

Code for the computation of the poroelastic coefficients

This cell computes automatically all the experimental coefficients of linear poroelasticity knowing an experimental set of five basic coefficients: Young modulus, shear coefficient, compressibility of the solid phase, porosity and bulk modulus of water:

```
E0 = 9.66; G = 4.2; KS = 42.0; Kf = 2.0; (* BULK Modulus of water GPa *)
```

$$\mu = 1787.0\ 10^{-6};\ \phi = 0.26;\ k = 1.5\ 10^{-12};$$

$$\nu = \frac{E0}{2\,G} - 1;\ KB = \frac{E0}{3\,(1-2\,\nu)};\ \text{(*While[True,If[KB>KS,Abort[]]];*)}$$

$$L = \frac{2\,G\,\nu}{1-2\,\nu};\ KB1 = L + \frac{2}{3}\,G;\ b = 1 - \frac{KB}{KS};\ KU = KB + \frac{b^2}{\phi/Kf + (b-\phi)/KS};$$

$$KU1 = KB + \frac{Kf}{\phi}\,b^2\left(1 + \frac{Kf}{KS}\left(\frac{b}{\phi} - 1\right)\right)^{-1};\ H = \frac{KB}{b};\ B = \frac{1 - KB/KU}{1 - KB/KS};\ C0 = \frac{1}{b}\,(KU - KB);$$

$$M = \frac{C0}{b};\ b1 = \sqrt{\frac{(KU-KB)}{M}};\ R = \frac{KB\,C0\,B}{(KU-KB)};\ B1 = \frac{R}{H};\ KS1 = \frac{KB}{1-b};\ LU = KU - \frac{2}{3}\,G;$$

$$L1 = LU - b\,C0;\ \nu U = \frac{3\,\nu + b\,B\,(1-2\,\nu)}{3 - b\,B\,(1-2\,\nu)};\ \phi 1 = \frac{Kf}{KS-Kf}\left(\frac{KU}{KS-KU} - \frac{KB}{KS-KB}\right)^{-1};$$

$$\phi 2 = \frac{Kf}{KS-Kf}\left(\frac{KS}{M\,(1-b\,B)} - \frac{b^2}{1-b} - b\right);\ CD = \frac{M\,(LU+2\,G) - C0^2}{LU+2\,G}\,\frac{k}{\mu};$$

```
Print["Experimental POROELASTIC COEFFICIENTS:"];
Print["Young modulus E [GPa]: ", E0]; Print["Shear modulus G [GPa]: ", G];
Print["porosity [%]: ", ϕ*100];
Print["solid modulus KS [GPa]: ", KS];
Print["fluid modulus Kf [GPa]: ", Kf]; Print["KB/KS = ", KB / KS]; Print[" "]
Print["COMPUTED POROELASTIC COEFFICIENTS:"];
Print["Poisson modulus ν, νU = ", ν, ", ", νU];
Print["Lame moduli λ, λ1, λU = ", L, ", ", L1, ", ", LU];
Print["Bulk Modulus KB, KB1 = ", KB, ", ", KB1, ", "];
Print["Undrained Bulk Modulus,KU, KU1 = ", KU, ", ", KU1];
Print["Coefficient of Skempton B, B1 = ", B, ", ", B1];
Print["Coefficients of Biot b,b1,C,M = ", b, ", ", b1, ", ", C0, ", ", M];
Print["Poroelastic Coefficients H, R = ", H, " ", R];
Print["Porosity confirmation ϕ1,ϕ2 = ", ϕ1, ", ", 100*ϕ2];
Print["Rigidity confirmation KS1 = ", KS1];
Print["Diffusivity Coefficient CD = ", CD]; Print[" "]
```

```
Experimental POROELASTIC COEFFICIENTS:

Young modulus E [GPa]: 9.66

Shear modulus G [GPa]: 4.2

porosity [%]: 26.

solid modulus KS [GPa]: 42.

fluid modulus Kf [GPa]: 2.

KB/KS = 0.109524


COMPUTED POROELASTIC COEFFICIENTS:

Poisson modulus ν, νU = 0.15, 0.316885

Lame moduli λ, λ1, λU = 1.8, 1.8, 7.26818

Bulk Modulus KB, KB1 = 4.6, 4.6,

Undrained Bulk Modulus,KU, KU1 = 10.0682, 10.0682

Coefficient of Skempton B, B1 = 0.609915, 0.609915

Coefficients of Biot b,b1,C,M = 0.890476, 0.890476, 6.14073, 6.89601

Poroelastic Coefficients H, R = 5.16578 3.15068

Porosity confirmation ϕ1,ϕ2 = 0.26, 26.

Rigidity confirmation KS1 = 42.

Diffusivity Coefficient CD = 3.76831 × 10^-9
```

```
(* This cell computes some Thermoporoelastic coefficients *)
ϕ = 0.0018; b = 0.3; B = 0.9; γB = 5.0 * 10^-5; γM = -46.0 * 10^-5; γU = 5.075 * 10^-5;
KB = 27.0 * 10^9; KU = 38.0 * 10^9; dζT = ϕ * γM - b * γB; dpTD = -B * KU * (ϕ * γM - b * γB) / b;
dpTU = -B * KB * ϕ * γM / b; dpTU0 = KB * (γU - γB) / b;
Print[" Thermal coefficients: "]; Print[" "]; Print["dζT = ", dζT];
Print["dpTD = ", dpTD]; Print["dpTU = ", dpTU]; Print["dpTU0 = ", dpTU0];
```

```
 Thermal coefficients:

dζT = -0.000015828

dpTD = 1.80439 × 10^6

dpTU = 67 068.

dpTU0 = 67 500.
```

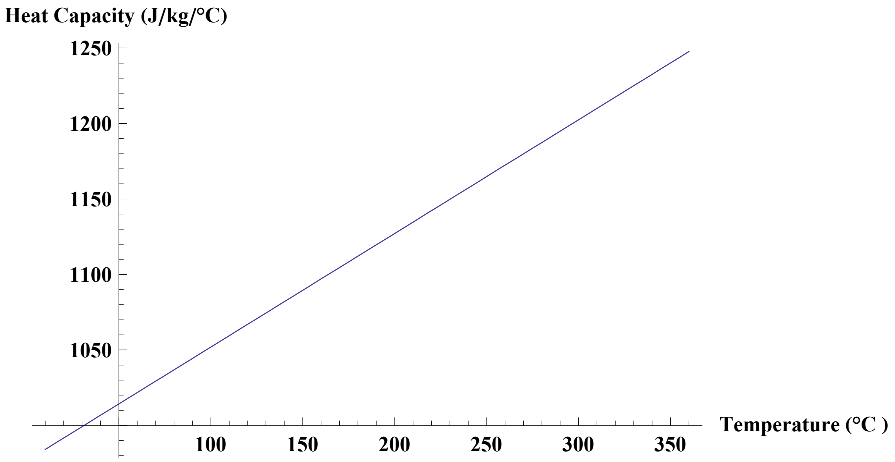
Code for the computation of thermal properties of rocks

The isobaric heat capacity and enthalpy of volcanic rocks in the range 20°C < T < 350°C can be approximated by the following polynomials. The models are included in the *Mathematica* command Plot:

In[26]:=

```
Plot[976.6065 + 0.752854 T, {T, 10.0, 360.0}, Axes → True,
 AxesLabel → {"Temperature  (°C ) ", "Heat Capacity (J/kg/°C) "},
 LabelStyle → Directive[Bold, FontFamily → "Times", 10]]
```

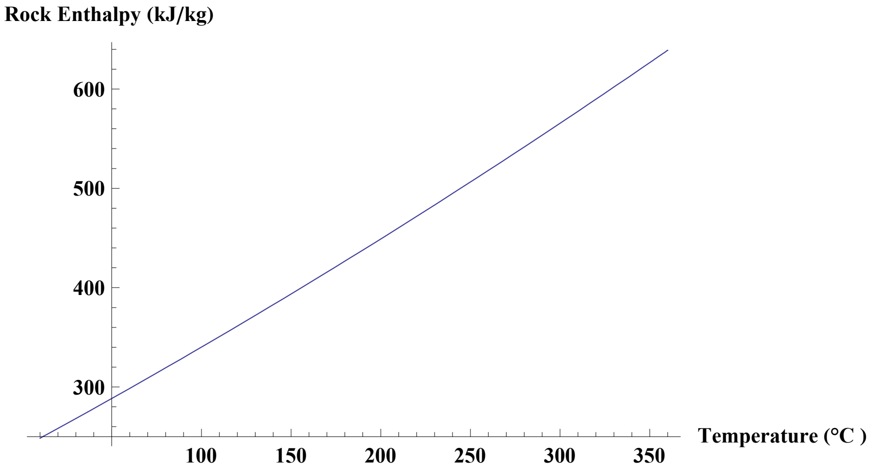
Out[26]=



In[27]:=

```
Plot[(238 674.5 + 976.6065 T + (0.752854/2) T^2) × 10^-3, {T, 10.0, 360.0},
 Axes → True, AxesLabel → {"Temperature  (°C ) ", "Rock Enthalpy (kJ/kg) "},
 LabelStyle → Directive[Bold, FontFamily → "Times", 10]]
```

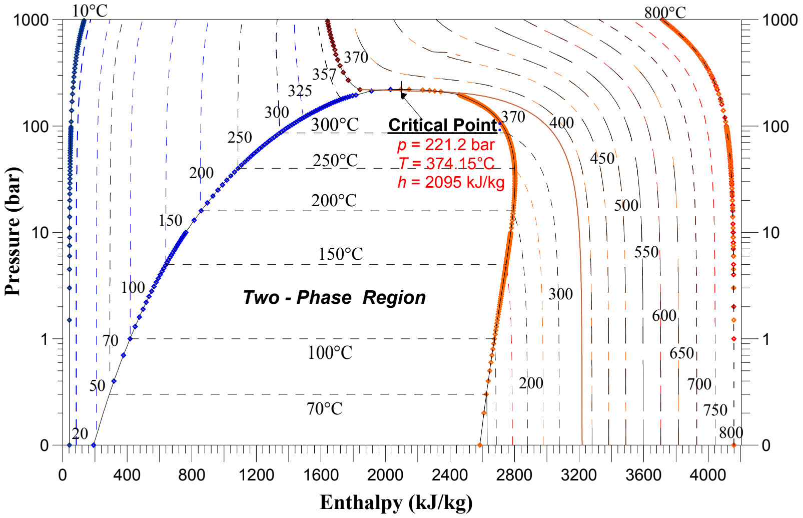
Out[27]=



Thermodynamical properties of Geothermal Water

Codes for the computation of the thermal properties of water

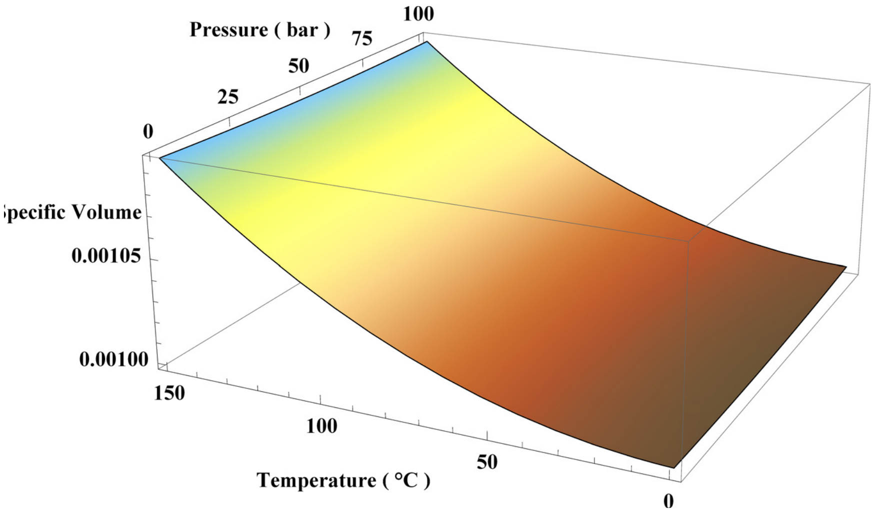
Water is not a simple substance, its general thermodynamic behavior is a very complex subject. The following graphic shows the full classical range of geothermal water without considering the effects of salts and gases (Bundschuh and Suárez, 2010):



However, it is possible to simplify the water complex behavior separating its full thermodinamycal domain in several subregions. This technique including all the models developed were published by the editorial Taylor & Francis in a recent book (Bundschuh and Suárez, 2010). The following cells compute automatically the density and enthalpy of low-enthalpy liquid water. The models are included in the *Mathematica* command Plot3D:

```
In[28]:= (* This cell computes the water specific volume v(p,T) in m^3/kg *)
        Plot3D[(999.8427563×10^-6 - 4.4888741×10^-8 p + 4.6382459×10^-8 T + 6.8717562×10^-10 p^2
           - 2.4966892×10^-10 p*T + 3.9625548×10^-9 T^2), {p, 0.01, 100.0}, {T, 0, 150.0},
         Axes → True, Mesh → None, ColorFunction → "SouthwestColors",
         AxesLabel → {"Pressure  ( bar )            ", "Temperature  ( °C )                      ",
           "    Specific Volume   "}, LabelStyle → Directive[Bold, FontFamily → "Times", 10],
         Ticks → {{0, 25, 50, 75, 100}, Automatic, Automatic}]
```

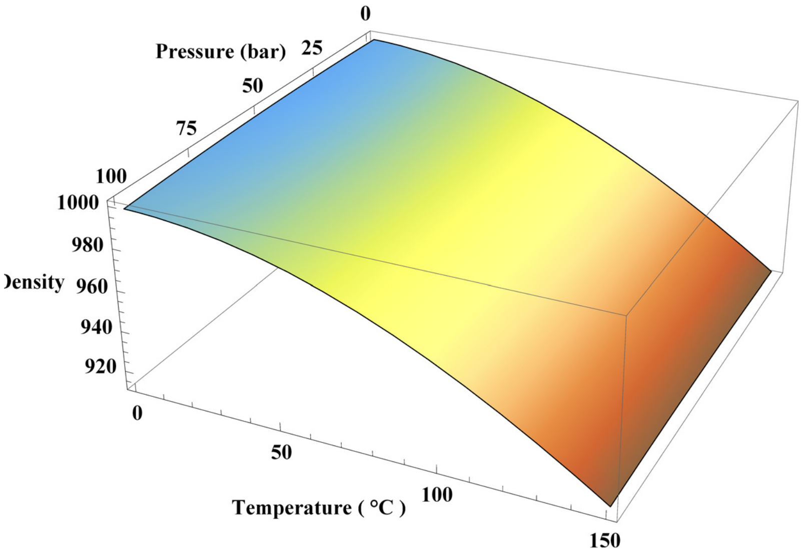
Out[28]=



In[29]:=

```
(* This cell computes the water density ρ(p,T) in kg/m^3 *)
Plot3D[(999.8427563×10^-6 - 4.4888741×10^-8 p + 4.6382459×10^-8 T + 6.8717562×10^-10 p^2
    - 2.4966892×10^-10 p*T + 3.9625548×10^-9 T^2)^-1, {p, 0.01, 100.0}, {T, 0, 150.0},
 Axes → True, Mesh → None, ColorFunction → "SouthwestColors",
 AxesLabel → {"Pressure (bar)            ", "Temperature ( °C )                         ",
   "  Density  "}, LabelStyle → Directive[Bold, FontFamily → "Times", 10],
 Ticks → {{0, 25, 50, 75, 100}, Automatic, Automatic}]
(* Thermal Properties of water *)
```

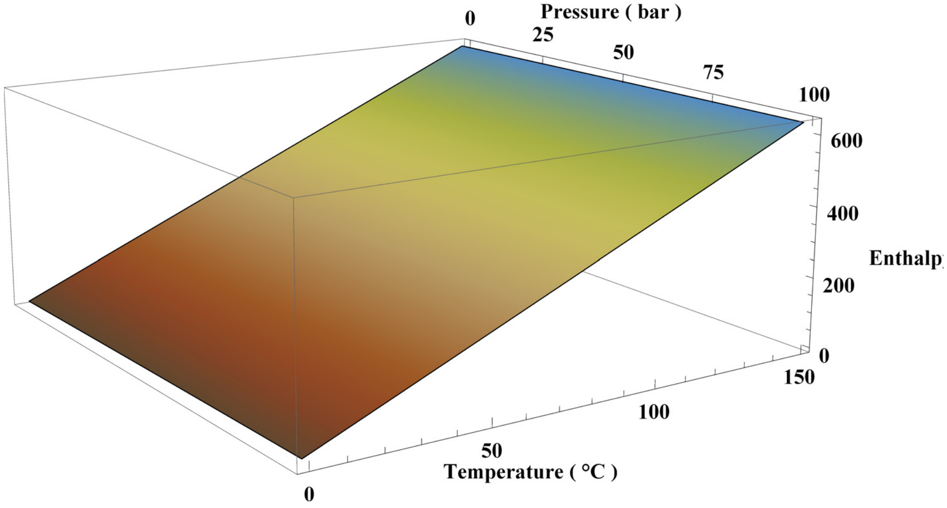
Out[29]=



In[31]:=

```
(* This cell computes the liquid water enthalpy h(p,T) in kJ/kg *)
Plot3D[0.103703286 + 0.104535515 p + 4.178354567 T - 3.243701×10^-4 p^2
  - 2.23906×10^-4 p*T + 9.70340×10^-5 T^2, {p, 0.0, 100.0}, {T, 0, 150.0},
 Axes → True, Mesh → None, ColorFunction → "SouthwestColors", PlotLabel → None,
 AxesLabel → {"Pressure ( bar )      ", "Temperature ( °C )                    ",
   " Enthalpy  "}, LabelStyle → Directive[Bold, FontFamily → "Times", 10],
 Ticks → {{0, 25, 50, 75, 100}, Automatic, Automatic}]
(* Thermal Properties of water *)
```

Out[31]=

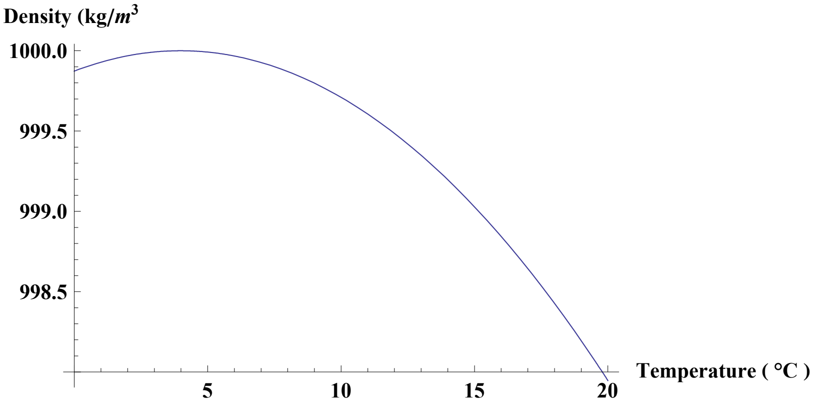


The behavior of low enthalpy water is simplified at low temperatures, because its thermodinamical dependence on pressure is very small. The following cells compute automatically the density and dynamic viscosity of low-enthalpy liquid water in the range [0, 250] °C. The models are included in the *Mathematica* command Plot:

In[32]:=

```
(* The cells compute water density as function of temperature ρ(T) in kg/m^3 *)
Plot[1000.0 (1.0 - 8×10^-6 ( T - 3.98)^2), {T, 0, 20.0},
 Axes → True, AxesLabel → {"Temperature ( °C ) ", "Density (kg/m^3 "},
 LabelStyle → Directive[Bold, FontFamily → "Times", 10]]
```

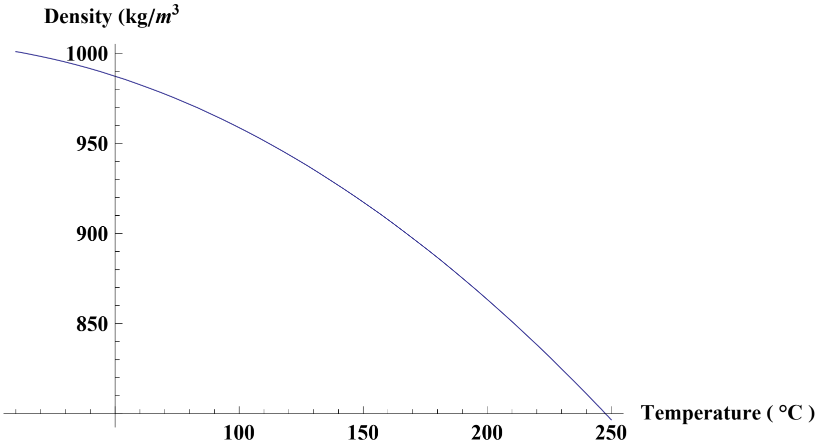
Out[32]=



In[33]:=

```
Plot[996.9 (1.0 - 3.17×10^-4 ( T - 25.0) - 2.56×10^-6 ( T - 25.0)^2), {T, 10, 250.0},
 Axes → True, AxesLabel → {"Temperature ( °C ) ", "Density (kg/m^3 "},
 LabelStyle → Directive[Bold, FontFamily → "Times", 10]]
```

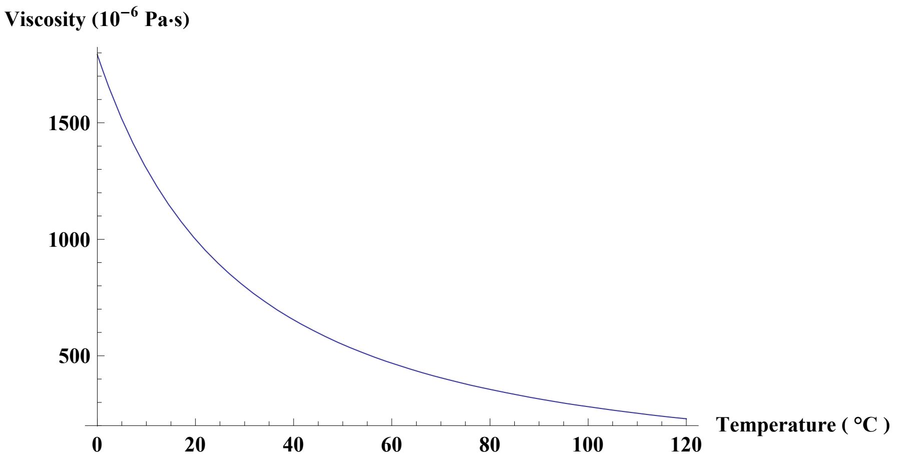
Out[33]=



In[34]:=

```
(* The cells compute water viscosity as function of temperature μ(T) in Pa·s *)
Plot[10^3 (1.0 + 0.015512 (T - 20.0))^-1.572, {T, 0, 120.0},
 Axes → True, AxesLabel → {"Temperature ( °C ) ", " Viscosity (10^-6 Pa·s) "},
 LabelStyle → Directive[Bold, FontFamily → "Times", 10]]
```

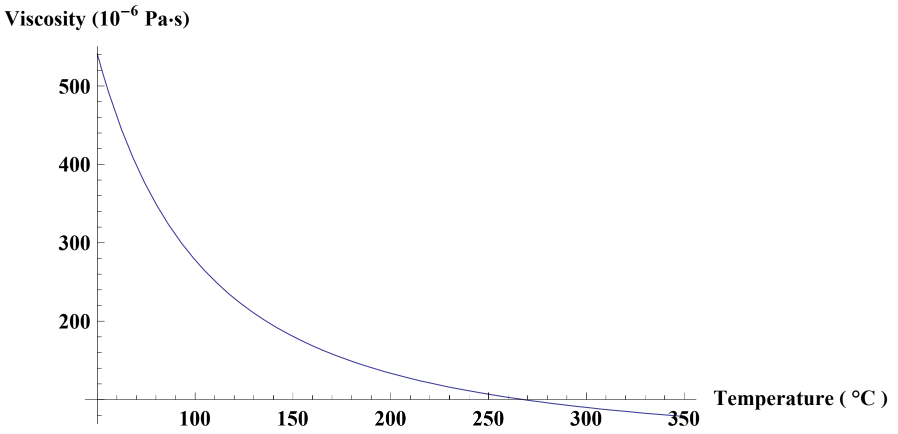
Out[34]=



In[35]:=

```
Plot[24.14 × 10^(247.8/(T+133.5)), {T, 50.0, 350.0},
 Axes → True, AxesLabel → {"Temperature ( °C ) ", " Viscosity (10^-6 Pa·s) "},
 LabelStyle → Directive[Bold, FontFamily → "Times", 10]]
```

Out[35]=



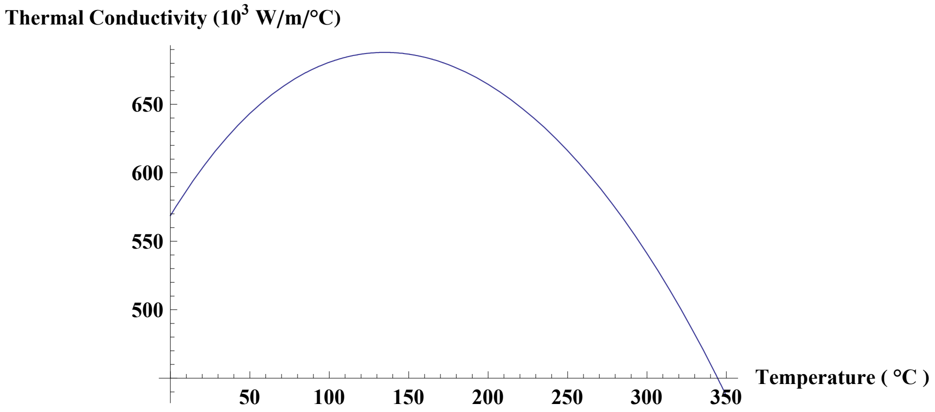
The thermal conductivity of water as a function of temperature in low, medium and high enthalpy systems can be estimated with this formula (10^3 W/m/°C) in the range [0, 350] °C. The model is included in the *Mathematica* command Plot:

In[36]:=

$$\texttt{Plot}\left[-922.47+2839.5\,\frac{T+273.15}{273.15}-1800.7\left(\frac{T+273.15}{273.15}\right)^{2}+525.77\left(\frac{T+273.15}{273.15}\right)^{3}-73.44\left(\frac{T+273.15}{273.15}\right)^{4},\ \{T,\,0.0,\,350.0\},\right.$$

```
 Axes → True, AxesLabel → {"Temperature  ( °C ) ", "  Thermal Conductivity (10^3 W/m/°C) "},
 LabelStyle → Directive[Bold, FontFamily → "Times", 10]]
```

Out[36]=



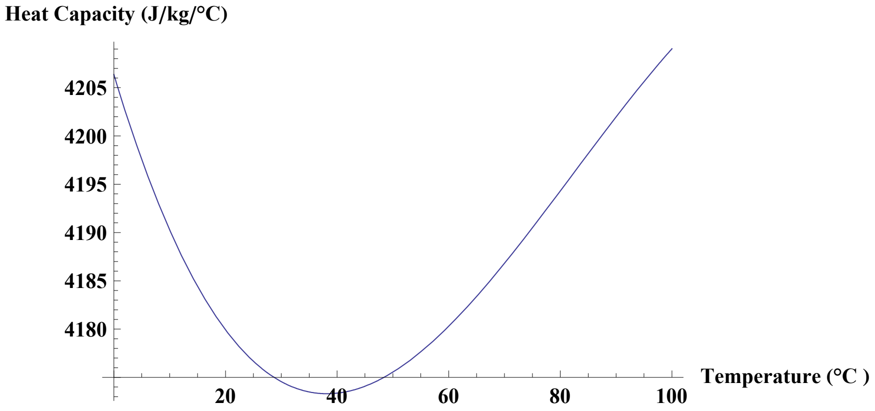
The isobaric heat capacity of water in the range 0°C < T < 350°C is approximated by the following polynomials (*J*/kg/°C) in the range [0, 100] °C. The models are included in the *Mathematica* command Plot:

In[37]:=

$$\texttt{Plot}\left[-1.3320081\times10^{-4}\,T^{3}+0.0328405\,T^{2}-1.9254125\,T+4206.3640128,\ \{T,\,0.0,\,100.0\},\right.$$

```
 Axes → True, AxesLabel → {"Temperature  (°C ) ", "  Heat Capacity (J/kg/°C) "},
 LabelStyle → Directive[Bold, FontFamily → "Times", 10]]
```

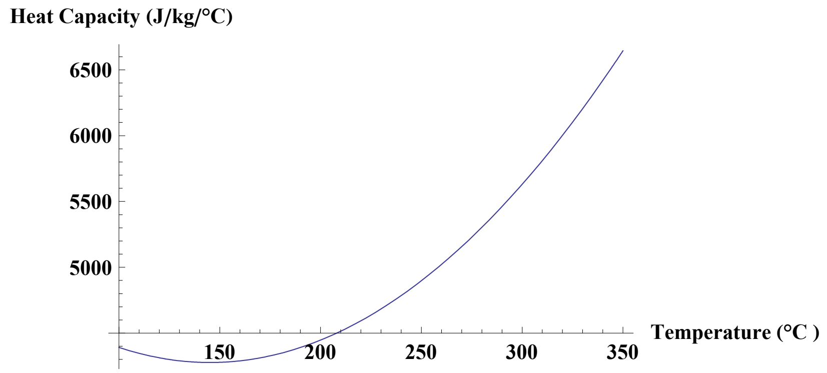
Out[37]=



In[38]:=

```
Plot[4187.6 (3.3774 - 1.12665×10^-2 (T + 273.15) + 1.34687×10^-5 (T + 273.15)^2), {T, 100.0, 350.0},
 Axes → True, AxesLabel → {"Temperature (°C ) ", "  Heat Capacity (J/kg/°C) "},
 LabelStyle → Directive[Bold, FontFamily → "Times", 10]]
```

Out[38]=



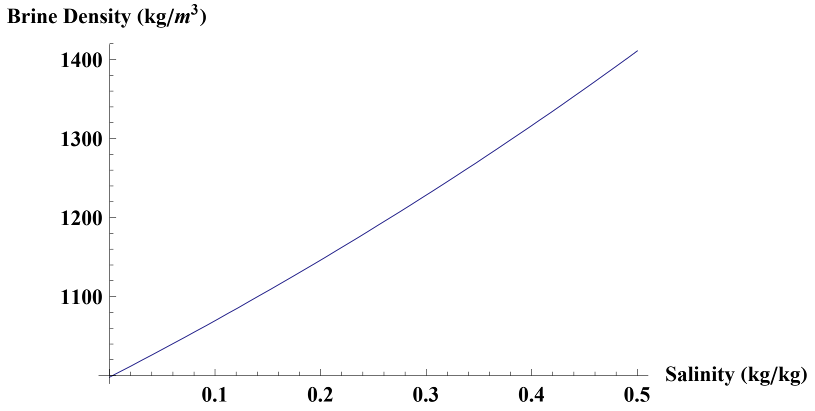
Code to compute the salinity of water (H_2O + NaCl)

For the water density dependence on salinity we consider the following correlations, where Cm is the mass fraction of NaCl in [kg/kg] . They are appropriate for aquifers at 20°C however, they could also be applicable approximately to low enthalpy systems The models are included in the *Mathematica* command Plot:

In[39]:=

```
Plot[998.0 * Exp[0.6923 Cm], {Cm, 0.0, 0.5}, Axes → True,
 AxesLabel → {"Salinity (kg/kg) ", "Brine Density (kg/m^3) "},
 LabelStyle → Directive[Bold, FontFamily → "Times", 10]]
```

Out[39]=

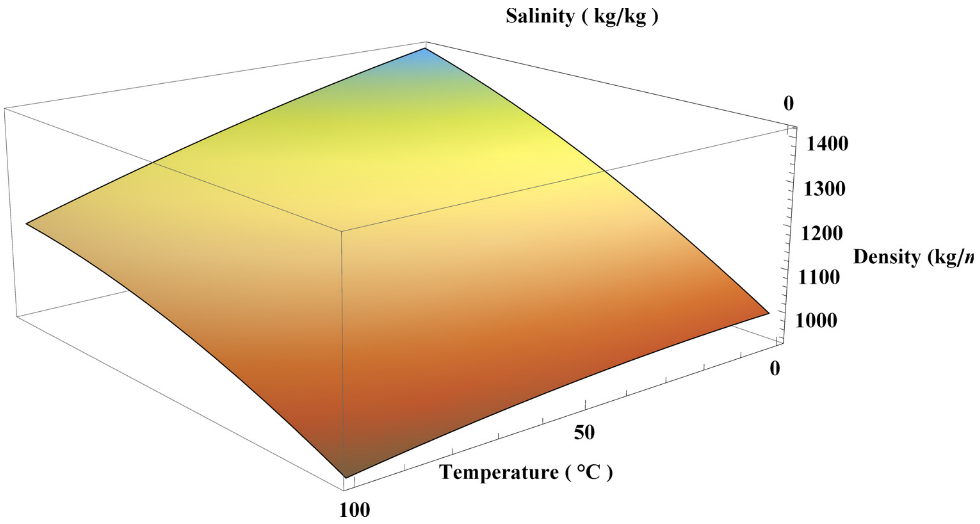


For the water density dependence on salinity Cm [kg/kg] and temperature [°C] a correlation is given by the following equation.It is appropriate for cold aquifers at [20, 50] °C however, it could also be applicable to low enthalpy systems The model is included in the *Mathematica* command Plot3D:

In[40]:=

```
Plot3D[998.0 (1.0 + 0.805 Cm - (T + 220.0 Cm - 4.0)^2 6.5×10^-6), {Cm, 0.0, 0.7}, {T, 0, 100.0},
 Axes → True, Mesh → None, ColorFunction → "SouthwestColors", PlotLabel → None,
 AxesLabel → {"Salinity  ( kg/kg )      ", "Temperature  ( °C )                     ",
   "   Density  (kg/m^3)   "},
 LabelStyle → Directive[Bold, FontFamily → "Times", 10],
 Ticks → {{0, 25, 50, 75, 100}, Automatic, Automatic}]
(* Thermal Properties of water *)
```

Out[40]=



Methods of Solution of fundamental PDE

Classical methods to solve the fundamental PDE

Modeling of groundwater flow and the transport of mass and heat in aquifers require solving partial differential equations or systems of differential equations, with initial and boundary conditions. These equations, in their most general form, were introduced in the two previous sections of this paper. In this section we mention the main methods currently used to solve the simplified forms of these PDE (Kythe et al, 2003).

1.- The method of separation of variables

This is one of the most efficient solution techniques for a certain class of PDE problems. It can be applied specifically to homogeneous PDE with homogeneous boundary conditions and to initial boundary value problems. In the *Mathematica* code we developed (FundamentalPDE.nb), we show applications to the Laplace, heat and wave equations.

2.- The method of Eigenfunctions expansion

The separation of variables cannot be used when the PDE and/or the boundary conditions are not homogeneous problems. The eigenfunctions expansion technique can be performed for non-homogeneous PDE with initial boundary value problems when the boundary conditions are homogeneous or can be transformed to homogeneous. In the *Mathematica* code we developed (EigenExpansionPDE.nb), we show applications to the Poisson, heat and wave equations.

3.- Espectral Methods (Fourier, Galerkin)

When the separation of variables or the eigenfunctions expansion techniques cannot be used to solve more sofisticated PDE, the espectral methods could be appropriate. Spectral Methods (SM) are high order techniques used to solve PDE either in their strong form or in their weak form (see subsection 4, FEM). What sets spectral methods apart from others like finite difference methods (which start from the PDE strong form) or finite element methods (which start from the weak form of the PDE) is that the solutions are approximated by high order orthogonal functions expansions. To build any spectral code only a few fundamental algorithms for interpolation, differentiation, Fast Fourier Transform and quadrature are needed, even for problems in complex geometries. In the *Mathematica* code we developed we show applications to the Poisson and diffusion equations.

4.- The Finite Element method (FEM)

The separation of variables and the eigenfunctions expansion techniques cannot be used when the PDE is non-linear, has variable coefficients, has non-homogeneous boundary conditions or when the boundaries geometry is of complicated nature. The Finite Element Method (FEM) is the most used numerical technique to solve approximately mathematical models expressed as PDE with arbitrary boundary conditions and for almost any initial boundary value problem. The FEM can be understood easily as a specifically sophisticated interpolation technique. The domain of the PDE is subdivided into a finite number of subdomains or elements of very simple geometry. The physical laws of the problem are applied to each element. The unknown variable, supposed to be a continuous function, can be approximated by interpolation functions in each element. For each one of these elements a matrix is obtained, which approaches the behaviour of the corresponding region. The accuracy of this matrix approach depends upon the size and complexity of the finite elements. The unknowns are the discrete values of the variable in the nodes linking the elements; all the elements are bound together to obtain a global matrix that represents the whole domain. In the stationary case, this process leads to a set of simultaneous linear algebraic equations, or to an ordinary differential system of equations in the transient case. In both cases, the solutions to these equations allow to approximate the unknown variable. The same basic procedure can be applied to an immense variety of problems. In the *Mathematica* code we are developing (FEM_PDE.nb), we will show applications to the Poisson, heat and wave equations.

The Method of Separation of Variables

The separation of variables can be used when the PDE and the boundary conditions are homogeneous or when the boundary conditions are not homogeneous but they can be transformed into homogeneous. In this *Mathematica* code we show applications to the Poisson, heat and wave equations.

The Difussion equation

We consider the homogeneus difussion equation with homogeneus Dirichlet conditions:

$$\begin{array}{ll} \partial_t U[x,\ t] = k\,\partial_{x,x} U[x,\ t] & \\ U[x,\ 0] = \phi[x] & BC : U[0,\ t] = 0 \\ U_t[x,\ 0] = \psi[x] & \qquad U[L,\ t] = 0 \end{array}$$

We build the solution as we did for the wave equation:

$$1)\ U[x, t] = X[x]\,T[t]$$

Plugging 1) into the difussion equation:

$$\frac{T'[t]}{k\,T[t]} = \frac{X''[x]}{X[x]} = -\lambda$$

Where λ again must be a constant. Therefore we solve the next two ODEquations:

$$T'[t]=\lambda k T[t]$$
$$X''[t]=\lambda X[x]$$

Where the solutions are:

$$X[x] = C\,\mathrm{Cos}\left[\sqrt{\lambda}\ x\right] + D\,\mathrm{Sin}\left[\sqrt{\lambda}\ x\right]$$
$$T[t] = A\,e^{-\lambda k t}$$

Using the boundary conditions on X [x] we get:

$$C = 0 \text{ and } 0 = D\,\mathrm{Sin}\left[\sqrt{\lambda}\ l\right],\ \text{therefore } \lambda_n = \left(\frac{n\pi}{L}\right)^2,$$
$$X[x] = \mathrm{Sin}\left[\frac{n\,\pi\,x}{L}\right] \text{ for } n = 1, 2, 3 \ldots$$

Once again we get an infinite number of Eigen- solutions for each n

$$U[x,\ t] = \sum_{n=1}^{\infty} A_n\,e^{-\left(\frac{n\pi}{L}\right)^2}\,\mathrm{Sin}\left[\frac{n\pi t}{L}\right]$$

Using now the initial conditions:

$$\phi[x] = \sum_{n=1}^{\infty} A_n\ \mathrm{Sin}\left[\frac{n\,\pi\,x}{L}\right]$$

With the Fourier expansion of the Sin series, we multiply by $\mathrm{Sin}\left[\frac{m\,\pi\,x}{L}\right]$ both sides of this equation:

$$\phi[x]\ \mathrm{Sin}\left[\frac{m\,\pi\,x}{L}\right] = \sum_{n=1}^{\infty} A_n\ \mathrm{Sin}\left[\frac{n\,\pi\,x}{L}\right]\ \mathrm{Sin}\left[\frac{m\,\pi\,x}{L}\right]$$

Then we integrate in the [0,L] interval

$$\int_0^L \phi[x]\ \mathrm{Sin}\left[\frac{m\,\pi\,x}{L}\right] dx = \int_0^L \sum_{n=1}^{\infty} A_n\ \mathrm{Sin}\left[\frac{n\,\pi\,x}{L}\right]\ \mathrm{Sin}\left[\frac{m\,\pi\,x}{L}\right] dx$$

$$\int_0^L \phi[x]\ \mathrm{Sin}\left[\frac{m\,\pi\,x}{L}\right] dx = \sum_{n=1}^{\infty} A_n\ \int_0^L \mathrm{Sin}\left[\frac{n\,\pi\,x}{L}\right]\ \mathrm{Sin}\left[\frac{m\,\pi\,x}{L}\right] dx$$

Observation:

$$\int_0^L \mathrm{Sin}\left[\frac{n\,\pi\,x}{L}\right]\ \mathrm{Sin}\left[\frac{m\,\pi\,x}{L}\right] dx = 0 \text{ if } n \neq m$$

Then:

$$\int_0^L \phi[x]\ \mathrm{Sin}\left[\frac{n\,\pi\,x}{L}\right] dx = A_m\ \int_0^L \mathrm{Sin}\left[\frac{n\,\pi\,x}{L}\right]^2 dx$$

Therefore:

$$A_m \;=\; \frac{2}{L}\int_0^L a[x]\,\mathrm{Sin}\left[\frac{m\,\pi\,x}{L}\right]\,dx$$

Finally, the solution of the difussion PDE is:

$$U[x,\,t] \;=\; \sum_{n=1}^{\infty} A_m\, e^{-\left(\frac{n\pi}{L}\right)^2}\,\mathrm{Sin}\left[\frac{n\pi t}{L}\right]$$

Solving the difussion PDE with Mathematica

```
Clear["Global`*"]
Print[
 Style["Difussion Equation with homogeneus boundary conditions:", 24, Bold]]
Print["∂tU[x,t]=k∂x,xU[x,t]"]
a[x_] = Input["IC for U[x,0]", ϕ[x]];
Print["IC: U[x,0]=", ϕ[x] , "        BC:U[0,t]=0"]
Print["             ",   "              U[L,t]=0"]
Print[""]
c = Input["Introduce value of c", c];
L = Input["Introduce value of l", l];
```

$$\mathbf{Print}\left[\text{"}A_m \;=\; \frac{2}{l}\int_0^l\text{"},\ a[x],\ \mathrm{Sin}\left[\frac{m\,\pi\,x}{L}\right],\ \text{"}dx\text{"}\right]$$

$$z = \frac{2}{L}\int_0^L a[x]\,\mathrm{Sin}\left[\frac{n\,\pi\,x}{L}\right]\,dx;$$

$$\mathbf{Print}\left[\text{"}U[x,t] \;=\; \sum_{n=1}^{\infty}\text{"},\ z\, e^{-\left(\frac{n\,\pi}{L}\right)^2}\,\mathrm{Sin}\left[\frac{n\,\pi\,t}{L}\right]\right]$$

$$U[x\_,\, t\_] := \sum_{n=1}^{20} z\, e^{-\left(\frac{n\,\pi}{L}\right)^2}\,\mathrm{Sin}\left[\frac{n\,\pi\,t}{L}\right]$$

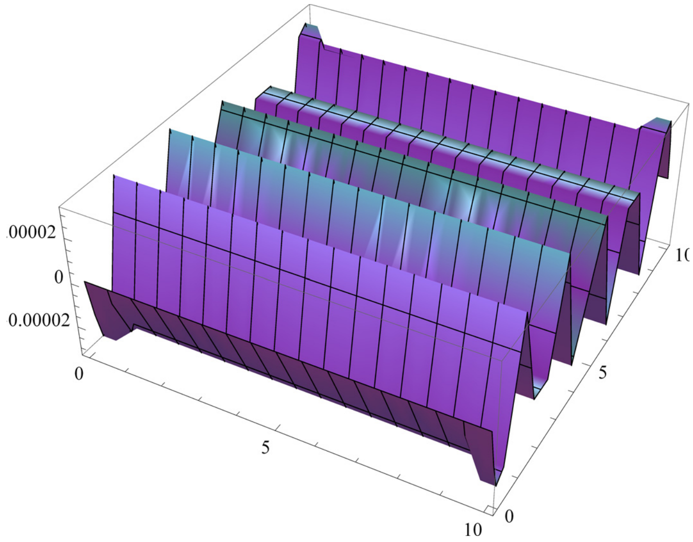
Difussion Equation with homogeneus boundary conditions:

```
∂tU[x,t]=k∂x,xU[x,t]

IC: U[x,0]=ϕ[x]       BC:U[0,t]=0

                        U[L,t]=0
```

$$A_m \;=\; \frac{2}{l}\int_0^1 -1. + x\mathrm{Sin}[3.14159\,m\,x]\,dx$$

$$U[x,t] \;=\; \sum_{n=1}^{\infty}\frac{2.\,e^{-9.8696\,n^2}\,(-0.31831\,n + 0.101321\,\mathrm{Sin}[3.14159\,n])\,\mathrm{Sin}[3.14159\,n\,t]}{n^2}$$

```
Plot3D[U[x, t], {x, 0, 10}, {t, 0, 10}]
```



The inhomogeneus Diffusion equation.

$$\partial_t u[x, t] = k\, \partial_{x,x} u[x, t] + q[x, t]$$

$$u(0, t) = u(L, t) = 0$$

$$u(x, 0) = f(x), \quad 0 < x < L$$

Initial conditions :

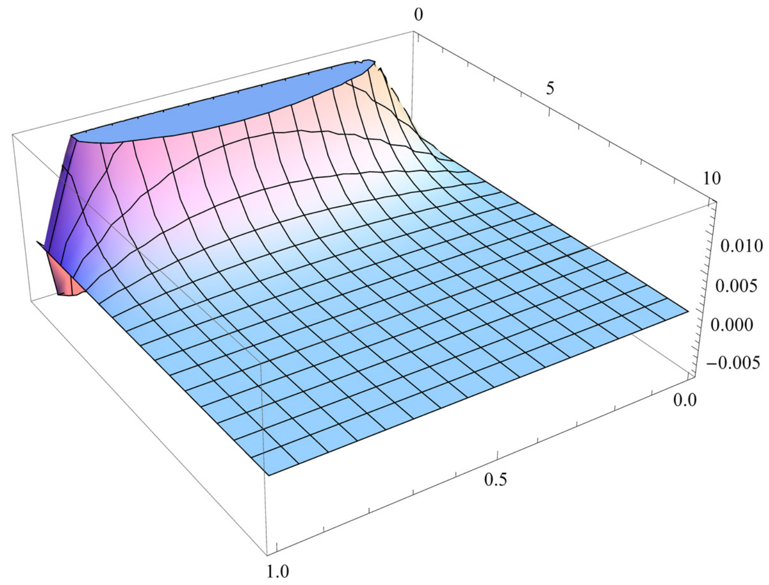
```
Clear["Global`*"]
k = 1; L = 1;
q[x_, t_] := x Exp[-t]
f[x_] := x - 1
i = 10;
```

Boundary conditions :

```
bc = ChoiceDialog["Condiciones de Frontera", {"Dirichlet" → 1, "Neuman" → 2}];

λ[n_] := (n Pi / L) ^2
ϕ[x_, n_] := If[bc == 1, Sin[Sqrt[λ[n]] x], Cos[Sqrt[λ[n]] x]]

Do[ Q[t] = 2 Integrate[q[x, t] ϕ[x, n], {x, 0, L}]; a = 2 Integrate[f[x] ϕ[x, n], {x, 0, L}];
  soln[n] = Flatten[DSolve[{c'[t] + λ[n] c[t] == Q[t], c[0] == a}, c[t], t]]; cc[n, t_] = c[t] /. soln[n]
   , {n, 0, i}]

u[x_, t_] := Sum[cc[n, t] ϕ[x, n], {n, 0, i}]

u[x, t]
```

Plot

```
Plot3D[u[x, t], {x, 0, 1}, {t, 0, 10}]
```



1.4 The Poisson equation

The Poisson PDE (non-homogeneous Laplace Equation)

$$\partial_{x,x}u[x, y] + \partial_{y,y}u[x, y] = q[x, y]$$

$$u\ (0, y) = u\ (L, y) = 0, \quad 0 < y < M$$

$$u\ (x, 0) = f_1\ (x), \quad u\ (x, M) = f_2\ (x), \quad 0 < x < L$$

Initial Conditions :

```
L = 1; M = 2;
q[x_, t_] := Pi^2 Sin[Pi x]
f1[x_] := 2 Sin[3 Pi x]
f2[x_] := -Sin[Pi x]
i = 10;
```

Boundary Conditions :

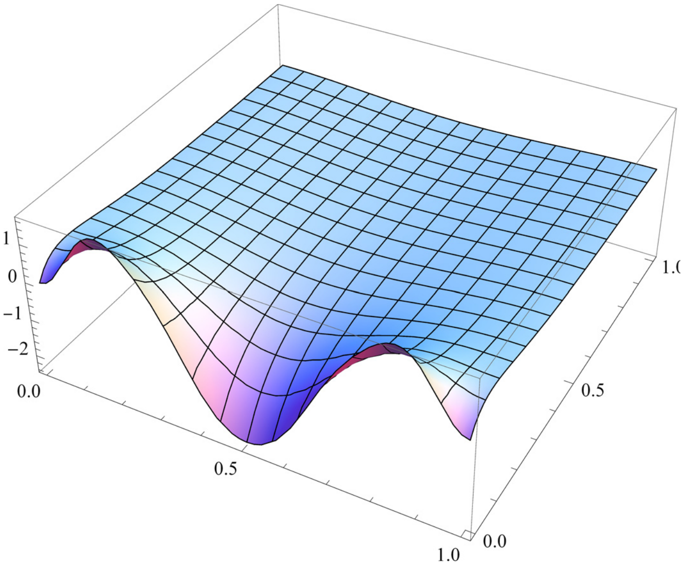
```
bc = ChoiceDialog["Condiciones de Frontera en x", {"Dirichlet" → 1, "Neuman" → 2}];

λ[n_] := (n Pi / L) ^2
ϕ[x_, n_] := If[bc == 1, Sin[Sqrt[λ[n]] x], Cos[Sqrt[λ[n]] x]]

Do[ Q[y] = 2 / L Integrate[q[x, y] ϕ[x, n], {x, 0, L}]; a1 = 2 / L Integrate[f1[x] ϕ[x, n], {x, 0, L}];
  a2 = 2 / L Integrate[f2[x] ϕ[x, n], {x, 0, L}];
  soln[n] = Flatten[DSolve[{c''[y] - λ[n] c[y] == Q[y], c[0] == a1, c[M] == a2}, c[y], y]]; cc[n, y_] = c[y] /. soln[n]
   , {n, 1, i}]

u[x_, y_] := Sum[cc[n, y] ϕ[x, n], {n, 0, i}]
```

```
FullSimplify[u[x, y]]
```

```
$Aborted
```

Plot

```
Plot3D[u[x, y], {x, 0, 1}, {y, 0, 1}, PlotRange → All]
```



References

[1] M.A. Biot, “General Theory of Three-Dimensional Consolidation”, Journal of Applied Physics, 12, 155-164, (1941).

[2] M.A. Biot, and D.G. Willis, “The Elastic Coefficients of the Theory of Consolidation, J. of Applied Mechanics, 24, 594-601, (1957).

[3] J. Bundschuh and M.C. Suárez, “Introduction to the Numerical Modeling of Groundwater and Geothermal Systems: Fundamentals of Mass, Energy and Solute Transport in Poroelastic Rocks”. Taylor & Francis CRC Press, ISBN: 978-0-415-40167-8 (2010).

[4] P. Kythe, P. Puri and M. Schäferkotter, "Partial Differential Equations and Boundary Value Problems with *Mathematica*". 2nd Edition, Chapmann & Hall / CRC, 418 pp. New York (2003).

[5] M.C.Suárez Arriaga, "La termoporoelasticidad en geotermia, definida en cuatro dimensiones". Geotermia Vol. 23, No. 2, pp. 41-50, ISSN: 0186-5897 (2010).