Lino Mediavilla
23/02/2017
Mecánica Computacional – Módulo 1
Prof. Juan Baus

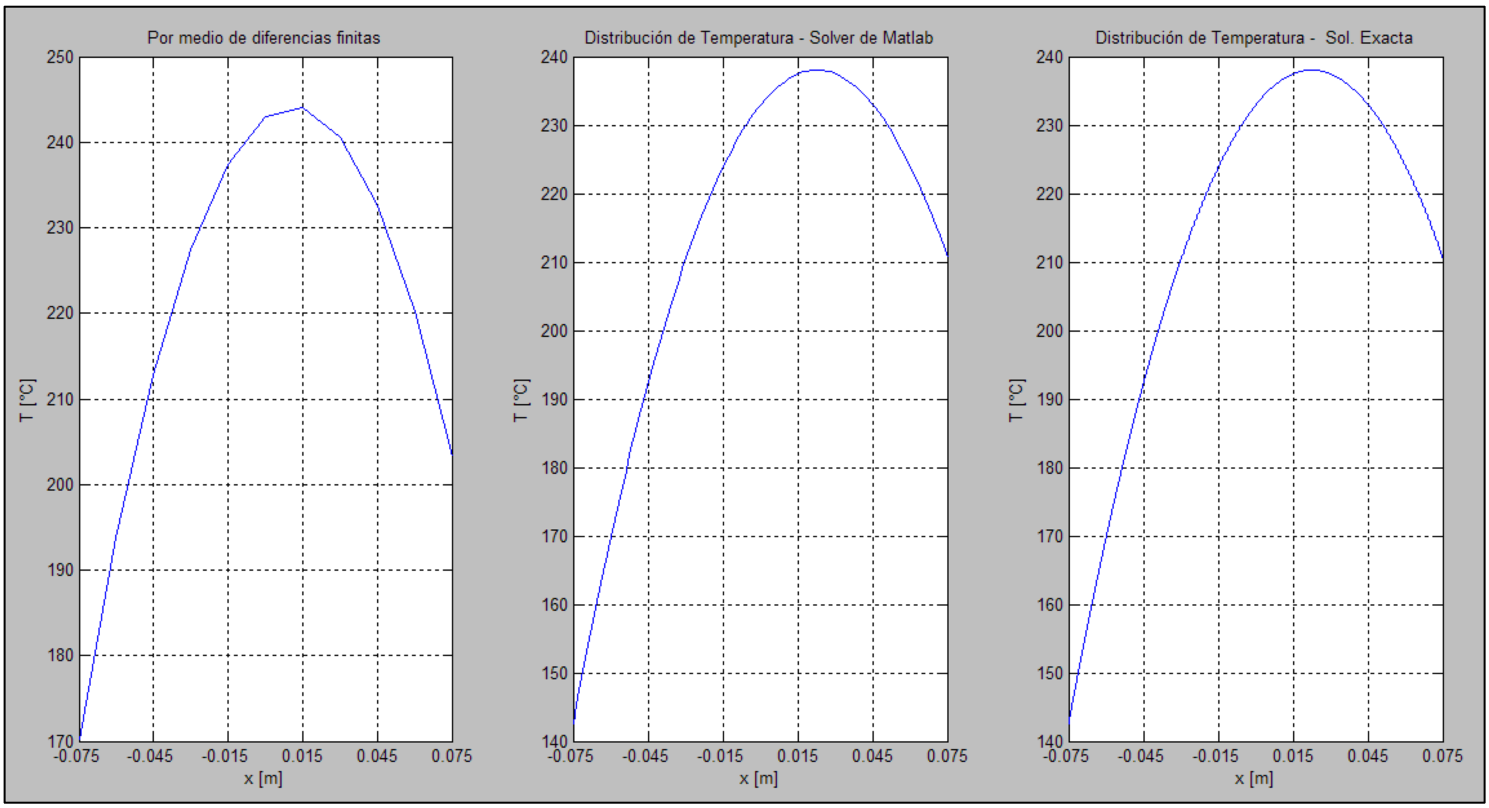
Deber # 4

Para las condiciones de la pared plana, graficar la distribución de temperatura (1D) obtenida a partir de los siguientes métodos:

1. Diferencias Finitas
2. Resolviendo la ecuación diferencial con una función de Matlab.
3. Con la solución a la ecuación diferencial directamente.

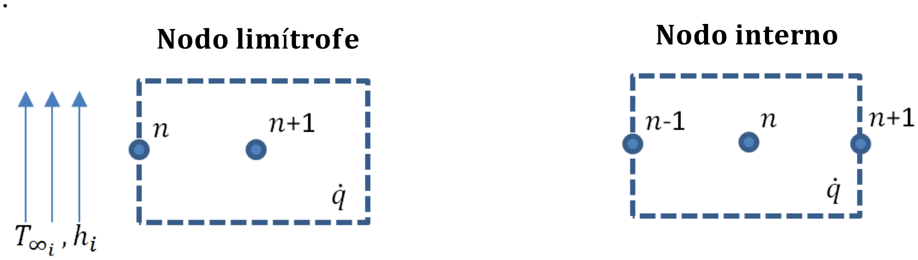
Resultados

En la siguiente gráfica se pueden apreciar los resultados de los 3 diferentes métodos solicitados.



Procedimiento

1. Para la resolución por diferencias finitas se realizó balance de energías en nodos internos y en nodos limítrofes en pared con convección. A partir de los siguientes diagramas, se obtuvieron las siguientes ecuaciones, que luego se insertaron en forma matricial en el código de MATLAB. Nota: se optó por escribir un programa que armara automáticamente el sistema de ecuaciones de temperaturas en los nodos, para poder estudiar el efecto de la disminución de Δx.



$$Nodo\ limítrofe: \quad \left(-h_i - \frac{k}{\Delta x}\right) T_n + \frac{k}{\Delta x} T_{n+1} = \frac{-\dot{q}\Delta x}{k} - h_i T_{\infty_i}$$

$$Nodo\ interno: \quad T_{n-1} - 2T_n + T_{n+1} = \frac{-\dot{q}\Delta x^2}{k}$$

2. Para la resolución con función de MATLAB se utilizó el solver de ecuaciones diferenciales con valores de frontera conocido como *bv4pc*.
   Dado que la ecuación diferencial a resolver era de 2do orden, era necesario reducir manualmente el sistema a uno de 1er orden, y crear dos funciones que serían pasadas como argumento a *bv4pc*. Estos archivos se llaman *bvp4bc.m* (con las condiciones de borde) y *bvp4ode.m* (con la ecuación diferencial). Otros argumentos requeridos fueron el dominio para la resolución (de –L a L) y un punto semilla (*initial guess*).

3. Para la solución exacta se usó la ecuación obtenida en clase para distribución 1-D de temperatura en pared plana;

$$T(x) = \frac{\dot{q}L^2}{2k}\left(1 - \frac{x^2}{L^2}\right) + \frac{T_{s_2} - T_{s_1}}{2} * \frac{x}{L} + \frac{T_{s_1} + T_{s_2}}{2}$$

   Donde,

$$T_{s_1} = \frac{\dot{q}L^2}{h_1} + T_{\infty_1}$$

$$T_{s_2} = \frac{\dot{q}L^2}{h_2} + T_{\infty_2}$$

Código (Implementado en MATLAB R2013b)

```
clear all
clc

%**** CONSTANTES DEL PROBLEMA ******
L = 0.075; %[m]
K = 75; % [W/m*K]
q = 1.5 * 10 ^6; %[W/m^3]
h_1 = 1000; %[W/m^2*K]
h_2 = 700; %[W/m^2*K]
T_fluido_1 = 30; %[°C]
T_fluido_2 = 50; %[°C]

%Cálculo de condiciones de borde
Ts_1 = (q*L/h_1) + T_fluido_1; %[°C]
Ts_2 = (q*L/h_2) + T_fluido_2; %[°C]
%**********************************


%******** Parte 1: RESOLUCIÓN POR DIFERENCIAS FINITAS ******
dx = 0.015/1;
nx = (2*L/dx)+1;
A = zeros(nx,nx);
B = zeros(nx,1);
```

```matlab
A(1,:) = [(-h_1-K/dx) K/dx zeros(1,nx-2)];
A(nx,:) = [zeros(1,nx-2) K/dx (-h_2-K/dx)];

B(1,1) = (-q*dx)-h_1*T_fluido_1;
B(nx,1) = (-q*dx)-h_2*T_fluido_2;

fila_A = [];

 for i=2:nx-1
     fila_A = [zeros(1,i-2) 1 -2 1 zeros(1,nx-1-i)];
     A(i,:) = fila_A;
     B(i,:) = [(-q*dx^2/K)];
 end

T_1 = A\B;
x_1 = (-L:dx:L);

subplot(1,3,1);
plot(x_1,T_1)
set(gca,'XTick',(-L:0.030:L));
xlim([-L L]);
title('Por medio de diferencias finitas');
grid on
xlabel('x [m]')
ylabel('T [°C]')
%*****************************************************************



%************* Parte 2: CON SOLVER DE MATLAB ***************

%Se utiliza el solver de ecuaciones diferenciales con valores
%de frontera conocido como "bv4pc".

solinit = bvpinit(linspace(-L,L,100),[1 -1]);
sol = bvp4c(@bvp4ode,@bvp4bc,solinit);

subplot(1,3,2);
x_2 = linspace(-L, L);
T_2 = deval(sol,x_2);
plot(x_2,T_2(1,:));

set(gca,'XTick',(-L:0.030:L));
xlim([-L L]);
title('Distribución de Temperatura - Solver de Matlab')
grid on
xlabel('x [m]')
ylabel('T [°C]')
%*****************************************************************



%***** Parte 3: MÉTODO ANALÍTICO - SOLUCIÓN EXACTA *********

%Se utiliza la solución obtenida en clase

T_3 = @(x) ((q*L^2)/(2*K))*(1-(x.^2)/(L^2)) + ...
    ((Ts_2 - Ts_1)/2) *(x./L)  +  ((Ts_1 + Ts_2)/2);

x_3 = (-L : L/10^6 : L);

subplot(1,3,3);
plot(x_3,T_3(x_3))
```

```
set(gca,'XTick',(-L:0.030:L));
xlim([-L L]);
title('Distribución de Temperatura -  Sol. Exacta')
grid on
xlabel('x [m]')
ylabel('T [°C]')
%***********************************************************
```

Funciones Adicionales creadas para usarse en la parte 2 (solver de MATLAB) :

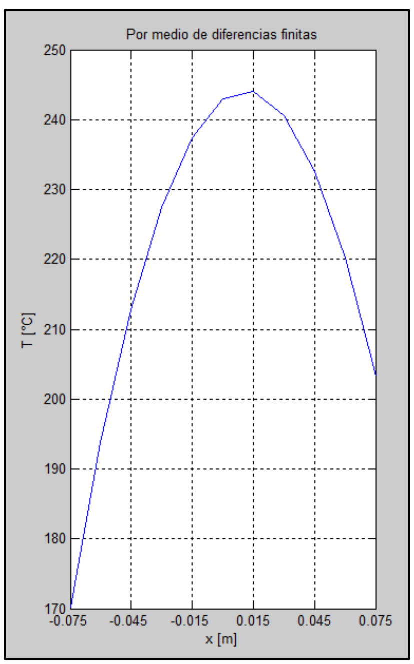
```
function res = bvp4bc(ya,yb)
res = [ya(1)-142.5000  yb(1)-210.7143];
```

```
function dydx = bvp4ode(x,y)
q = 1.5 * 10 ^6; %[W/m^3]
K = 75; % [W/m*K]
dydx = [ y(2)  -q/K];
```

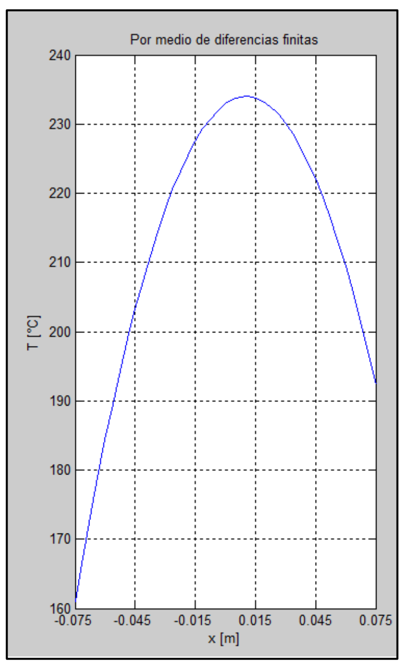
Conclusiones

Como se esperaba, los métodos de las partes 2 y 3 arrojan resultados muy similares, por lo que el paquete de MATLAB para solución numérica a ecuaciones diferenciales ha demostrado ser muy eficiente.

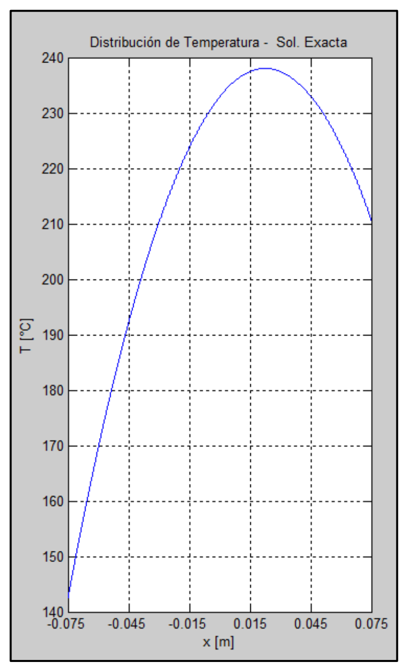
Al disminuir el paso (Δx) se logró que la gráfica de la distribución de temperatura tuviera un aspecto más ‘suave’, pero aún con un paso de 3.7 mm , es decir, generando una matriz de 41 ecuaciones (contra 11 ecuaciones a un paso de 15 mm), se presentan apreciables diferencias entre los valores numéricos de las curvas obtenidas por solver de MATLAB y la solución exacta.



$\Delta x = 15$ mm



$\Delta x = 3.7$ mm



$Sol. Exacta$

Referencias

Incropera, & F. P., Incropera, F. P. (2007). *Fundamentals of heat and mass transfer*. Hoboken, NJ: John Wiley.