Taller 6: Diferenciación e Integración Numérica

Valentina Cordova [*]   Emil Vega[†]

7 julio del 2016

1. Resumen

1.1. MÉTODO DE DIFERENCIAS FINITAS

Es un método que permite la resolución aproximada de ecuaciones diferenciales en derivadas parciales definidas en recintos finitos. Su expresión matemática esta dada de la forma $f(x+b)-f(x+a)$ que al dividirse por $b-a$ da como resultado un cociente diferencial, que se distingue porque se utilizan cantidades finitas en un lugar de infinitesimales (Boole, 1872). Se consideran normalmente tres formas que son las siguientes:

Diferencias adelanta

$$\Delta_h f(x) = \quad f(x+h) - f(x) \tag{1.1}$$

Diferencias atrasada

$$\nabla_h f(x) = f(x) - f(x-h) \tag{1.2}$$

Diferencias centrada

$$\delta_h f(x) = f(x+\frac{1}{2}h) - f(x-\frac{1}{2}h)) \tag{1.3}$$

[*]airina.cordova@yachaytech.edu.ec
[†]emil.vega@yachaytech.edu.ec

1.2. CONCEPTOS ADICIONALES

Condiciones de frontera de Dirichlet
Es un tipo de condición de frontera o contorno, que se utilizan cuando una ecuación diferencial ordinaria o una en derivadas parciales, se le especifican los valores de la solución que necesita la frontera del dominio.

Matriz Tridiagonal
Es una matriz cuyos elementos de su diagonal principal son los únicos distintos de cero.

Orden de convergencia de una solución numérica
Es la velocidad con la cual una sucesión converge a su límite.

Finalmente se observa que al utilizar la diferencia centrada con valores de n cada vez mayores, más se acerca la u aproximada a la u real, es decir la solución discretizada de la ecuación diferencial con las condiciones iniciales planteadas, se acerca más a la solución análitica.

2. PLANTEAMIENTO DEL PROBLEMA

2.1. EJERCICIO 1: ALGORITMO EN MATLAB

Cree un m-archivo con el nombre cds que calcule una solución aproximada para la ecuación unidimensional estacionaria de convección-difusión.

$$\frac{du}{dx} = \frac{1}{50}\frac{d^2u}{dx^2} \tag{2.1}$$

con condiciones de Dirichlet $u(0) y u(1) = 1$, utilizando el esquema centras de diferencias finitas de la Ecuación NÂř3 sobre n nodos equidistantes en el intervalo $[0,1]$ dados por $x_i = (i-1)h$ con
$h = \frac{1}{n-1}$ para $i = 1, 2, ..., n$.

En el ejercicio se crea un m-archivo haciendo uso de matlab llamado cds_EV que calcula la solución aproximada para la ecuación unidimensional.

2.2. EJERCICIO 2: SOLUCIÁ ANALÍTICA Y DISCRETA PARA LA ECUACIÃŞN DE CONVECCIÓN-DIFUSIÓN

a Verifique que la solución analítica de (1) es:

$$u(x) = \frac{exp(50)x - 1}{exp(50) - 1}$$

b Verifique que la solución discreta de (1) mediante el esquema central en diferencias finitas satisface el siguiente sistema de ecuaciones lineales:

$$\underbrace{\begin{pmatrix} \beta & \gamma & & & & \\ \alpha & \beta & \gamma & & & \\ & \alpha & \beta & \gamma & & \\ & & \alpha & \beta & \gamma & \\ & & & \alpha & \beta & \gamma \\ & & & & \alpha & \beta \end{pmatrix}}_{A} \underbrace{\begin{pmatrix} u_2 \\ u_3 \\ u_4 \\ \\ u_{n-2} \\ u_{n-1} \end{pmatrix}}_{\tilde{u}} = \underbrace{\begin{pmatrix} 0 \\ 0 \\ 0 \\ \\ \\ -\gamma \end{pmatrix}}_{b}$$

Donde los coeficientes de A y el vector b están dados por:
$alpha$=$1+25h$
$beta$= -2
y $gamma = 1-25h$.
El valor de u_i para $i = 2,3,4,...,n-2,n-1$ en el vector de incógnitas. Además $\tilde{u}$ representa la aproximación de $u(x_i)$.

2.3. EJERCICIO 3. ANÁLISIS DEL ERROR DE DISCRETIZACIÓN

a Graficar (2) y $\tilde{u}$ sobre los nodos $\{(x\_i,)\}\begin{matrix} n=1 \\ i=2 \end{matrix}$ para $n = 11,21,41,81$.
b Estime el orden de convergencia de la solución numérica en las normas infinito y $\|\cdot\|_{p(0,1)}$ con $p = 1,2$ para $n = 161,321,641,1281$.

Usando la siguiente fórmula:

$$\|f\|_{p(0,1)} \cong (\int_0^1 |f(x)|^P dx)^{\frac{1}{P}} \tag{2.2}$$

3. METODOLOGÍA

3.1. DESCRIPCIÓN DE LOS MÉTODOS NUMÉRICOS UTILIZADOS Y SUS RESPECTIVAS IMPLEMENTACIONES COMPUTACIONALES

EJERCICIO 1

cds_EV: Es el algoritmo creado y utilizado para que calcule la solución aproximada para la ecuación 4.

EJERCICIO 2

a. Verificación de solución analítica de (1)

Haciendo uso del comando de $Matlab$ llamado $dsolve$ se confirma que $u(x) = \frac{exp(50)x-1}{exp(50)-1}$ es solución de la ecuación NÂř4.

b. Verificación de solución discreta de (1)utilizando el esquema central de diferencias finitas

Se utiliza lo siguiente: con el objetivo de hallar una ecuación general que se pueda aplicar desde $x_2 hasta x(n-1)$

*$f(x) \cong \frac{(x+h)-(x-h))}{2h}$ (Ecuación NÂř3).

*$f''(x) \cong \frac{(x+h)+2f(x)+(x-h))}{h^2}$ (Segunda derivada de la Ecuación NÂř3)

a. Graficar (2) **y** $\tilde{u}$

Uso de MATLAB para gráficar uso el comando "$plot$" y uso las herramientas para que aparezca las cuatros gráficas que corresponden a los nodos:

Utilizando el algoritmo cds_EV y los nodos $xx = [x_2, ..., x_(n-1)]$. Se plotea $u = f(xx)$ que es la solución exacta, y $\tilde{u}$ en nuestro algoritmo es up que es la solución aproximada de Ecuación 1 para los mismos valores de n.

3.2. LISTADO DE LOS M-ARCHIVOS REALIZADOS:

Ejercicio 1:
* $cds_E V.m$
Ejercicio 2:
* $eval_f un.m$
* $T6_e jercicio2.m$
Ejercicio 3:
* $T6_e jercicio3_a.m$

[1]

4. DISCUSIÓN DE RESULTADOS

EJERCICIO 1: Para este ejercicio se utiliza el algoritmo llamado $cds\_EV.m$ el cual se denomina en $Matlab$

$$function[x,u,xx] = cds_E V(n)$$

para el cual el parámetro de entrada era n (nodos) que arrojará la solución de la ecuación diferencial.

EJERCICIO 2:

*a Se puede observar que al aumentar el valor de n la solución exacta y $\tilde{u}$ que en nuestro caso corresponde a up la soluciÃşn aproximada, se van acercando cada vez más en sus valores. Esto es por que la primera derivada de u es igual a
$\frac{1}{50}$ multiplicado por su segunda derivada. Como se muestra a continuación:

$$u(x) = \frac{e^{50x} - 1}{e^{50} - 1}$$

$u'(x) = \frac{1}{(e^{50}(-1))}(e^{50x}(50))$ Primera derivada de u

$u''(x) = \frac{1}{e^{50}-1}(e^{50x}(50)^2)$ Segunda derivada de u

$u' = \frac{1}{50}u''$ (Ecuación 2.1 en cual se reemplaza $u' y u''$)

-» $\frac{50e^{50x}}{e^{50}-1} = \frac{1}{50}\frac{(50)^2 e^{50x}}{(e^{50}-1)}$

-» $\frac{50e^{50x}}{e^{50}-1} = \frac{50e^{50x}}{(e^{50}-1)}$ (Se encuentra la igualdad).

Por lo tanto se puede decir analíticamente que u es solución de la ecuación diferencial es decir la Ecuación NÂř1.

b

Mediante el análisis hecho en clase, a partir de la siguiente deducción utilizando las Ecuación 3 y su segunda derivada $f''(x)$ se obtuvo:

1) Se hace un cambio de variable $x_i = u_i$ y se tiene:

$$x_2 = u_2' = \frac{u_3 - u_1}{2h};\ u_2'' = \frac{1}{50}\frac{u_3 - 2u_2 + u_1}{h^2}$$

$$\Rightarrow \left(\frac{1}{2h} - \frac{1}{50h^2}\right)u_3 - \left(\frac{1}{2h} + \frac{1}{50h^2}\right)u_1 = 0$$

[1] emil.vega@yachaytech.edu.ec
airina.cordova@yachaytech.edu.ec

$$\Rightarrow \frac{(25h-1)}{50h^2}u_3 + \frac{2u_2}{50h^2} = 0$$

$$\boxed{2u_2 + (25h-1)u_3 = 0}$$

$$x_3 = \frac{u_4 - U_1}{2h} = \frac{1}{50}(u_4 - 2u_3 + u_2)$$

$$\Rightarrow \left(\frac{1}{2h} - \frac{1}{50h^2}\right)u-4 - \left(\frac{1}{2h} - \frac{1}{50h^2}\right)u-2 + \frac{2}{50h_2}u-4$$

$$\left(\frac{25h^2-1h}{50h^2}\right)u_4 - \left(\frac{25h^2+1h}{50h^2}\right)U_2 + \frac{1}{25}u_3 = 0$$

$$\boxed{-(25h+1)u_2 + 2u_3 + (25h-1)u_4 = 0}$$

$$U_4 = \frac{u_5 - u_3}{2h} = \frac{1}{50h^2}(u_5 - 2u_4 + u_3)$$

$$\boxed{-(25h+1)u_3 + 2u_4 + (25h-1)u_5 = 0}$$

Generalizando:

$$\underbrace{-(25h+1)}_{\alpha}U_2 + \underbrace{2}_{\beta}U_3 + \underbrace{25h-1)}_{\gamma}U_4 = 0$$

$$\alpha U_3 + \beta U_4 + \gamma U_5 = 0$$

$$\alpha U_4 + \beta U_5 + \gamma U_6 = 0$$

$$\vdots$$

$$\alpha u_{n-3} + \beta U_{n-2} + \gamma u_{n-1} = 0$$

$$\alpha u_{n-2} + \beta u_{n-} + \gamma u_n = 0$$

$$\alpha u_{n-2} + \beta u_{n-1} \qquad = -\gamma$$

Además para ilustrar se toma un valor de n arbitrario para visualizar los resultados que pueden ser obtenidos después de hallar la ecuación generalizada anteriormente descrita:

$Para \quad n = 5$

$$A = \begin{pmatrix} 2.00 & 5.25 & 0 \\ -7.25 & 2.00 & 5.25 \\ 0 & -7.25 & 2.00 \end{pmatrix} \quad ; \quad x = \begin{pmatrix} 0 \\ 0,25 \\ 0,50 \\ 0,75 \\ 1,00 \end{pmatrix} \quad ; \quad u = \begin{pmatrix} -0.9030 \\ 0.3440 \\ -1.3708 \end{pmatrix}; \quad xx = \begin{pmatrix} 0.25 \\ 0.50 \\ 0.75 \end{pmatrix}$$

EJERCICIO 3:

a A continuación se muestran las cuatro gráficas o figuras que se obtuvieron para cada uno de los n.

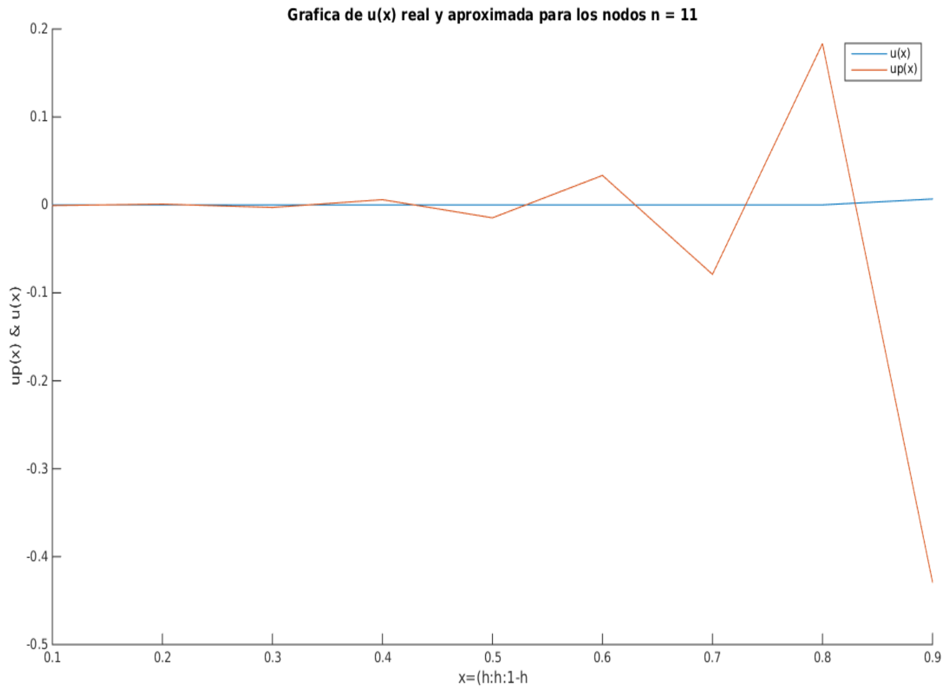

Figura 4.1: Figura 1 u(x) real, up(x) vs n=11)

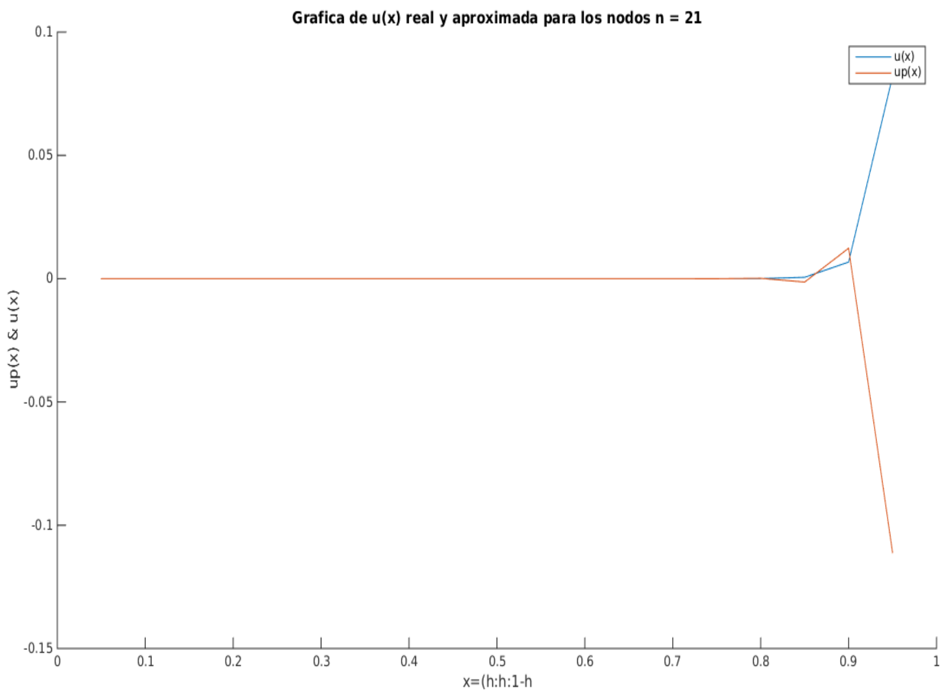

Figura 4.2: Figura 2 u(x) real, up(x) vs n=21

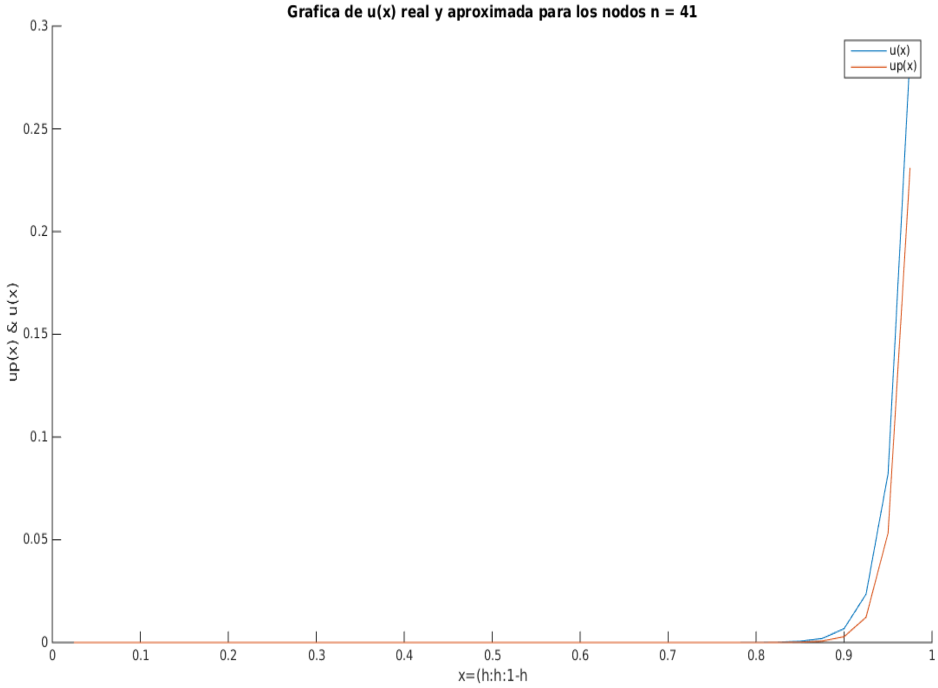

Figura 4.3: Figura 3 u(x) real, up(x) vs n=41

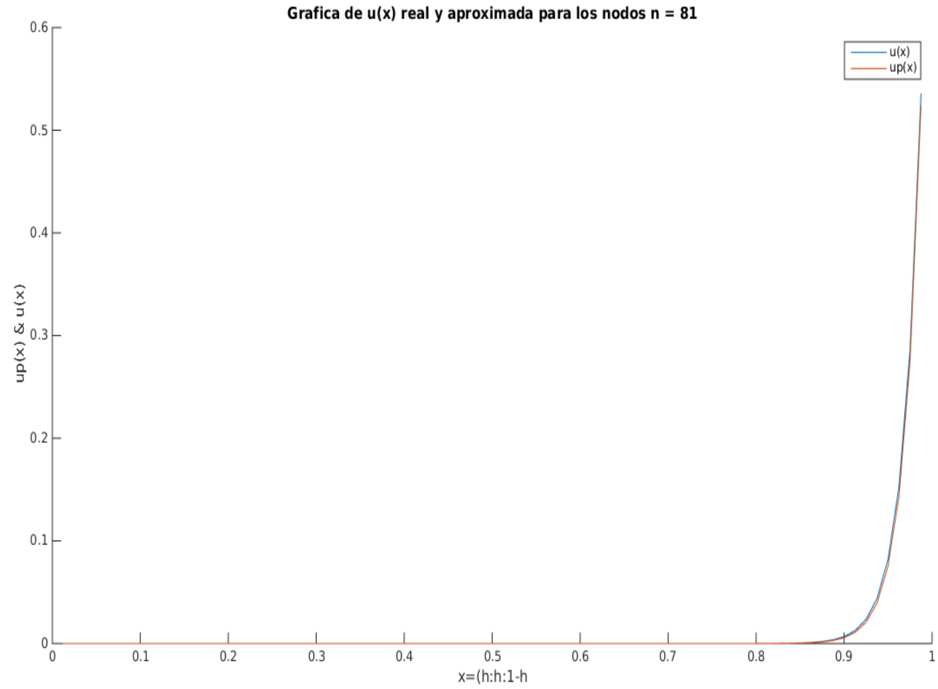

Figura 4.4: Figura 4 u(x) real, up(x) vs n=81

Al observar la Figura 1, la solución aproximada está aun muy alejada de la real. Pues las gráficas están bastante separadas una de otras. Cuando comparamos la Figura 1 con la Figura 2 se observa como la u discreta se aproxima un poco más, debido a que $n = 21$. Si continuamos con la comparación, en la Figura 4 la solución aproximada up estÃą muy junta a la solución real. Es decir la curva roja estÃą mucho más cerca de la curva de color azúl cuando $n = 81$.

-**b**: Se encontró la norma infinita y la norma Lp con $p = 1, 2$ para los $n = 161, 321, 641, 1281$ lo cual se muestra en el cuadro 1.

Tabla 4.1: Números de nodos con sus errores y convergencia

n	Norma Infinito	Norma Lp ; $p = 1$	Norma Lp ; $p = 2$	Orden de convergencia
161	0.003	0.00015563	0.00056882	N/A
321	0.00074896	0.000040122	0.00014354	1.9557
641	0.00018721	0.000010127	0.000035942	1.9862
1281	0.00004678	0.0000025392	0.0000089885	1.9957

Como se puede observar a medida que se aumentan los nodos, la norma infinito de los errores como la norma Lp disminuyen. Además, si comparamos las normas de los errores entre si, se nota que la norma Lp es menor que la norma infinito. Cabe seÃśalar que los errores para $p = 1$ y $p = 2$ son muy parecidos. Por lo tanto, en la convergencia para cada nodo se puede ver que se acerca mucho al grado 2 mientras aumentamos los nodos. Esto quiere decir que en el grado $p = 2$ la funciÃşn converge.

5. Conclusiones

- Se pudo aprender que la solución discreta encontrada mediante el esquema central en diferencias finitas, a medida que se aumenta el número de nodos se aproxima mucho más a la soluciÃşn analítica, es decir, para obtener la mejor aproximación, n tiene que ser muy grande.

- En las Figuras se pudo comprobar de manera visual que si se aumenta el nÃžmero de nodos, las funciones son similares.

- Los errores en la norma infinito y Lp disminuyen mientras se aumenta la cantidad de nodos. El error en la norma Lp es menor al error en la norma infinito.

- El uso de la regla del trapecio compuesto resultó ser muy útil y fácil de usar para encontrar una aproximación de la integral para hallar la norma de Lp.

6. Referencias Bibliografícas (Opcional)

Lista de las referencias citadas.

* Cuaderno de Métodos Numéricos. Cuarto Semestre. Yachay Tech.(2016). * Granville Sewell, The Numerical Solution of Ordinary and Partial Differential Equations, Academic Press (1988) * Cheng, A. and D. T. Cheng (2005). Heritage and early history of the boundary element method, Engineering Analysis with Boundary Elements, 29, 268âĂŞ302. * Boole, George, A Treatise On The Calculus of Finite Differences, 2Âł Ed., Macmillan and Company, 1872.

Apéndice A Lista de Programas Computacionales

Ejercicio 1:

```
% -------------------------------------------------------------------------
% Nombre del programa: cds_EV.m
% Autor(es): Valentina Cordova - Emil Vega
% Email del (los) autor(es): airina.cordova@yachaytech.edu.ec -
% emil.vega@yachaytrech.edu.ec
% Fecha de elaboracion: Julio 6 2016
% Breve descripcion del programa: calcule una solución aproximada para
%la ecuación unidimensional estacionaria de conveccit'on-difusión.
% Datos de entrada:
% ----------------
% n: nodos
%____________
% Datos de salida
% ---------------
% u: solución exacta
% x: intervalo [0,..,1]
% xx: intervalo [x_2,..,x_n-1]

% ------------------------------------------------------------------------
function [x,u,xx,A]=cds_EV(n)
h=1/(n-1);
x=[0:h:1];
xx= x(2:n-1);
x=x';
.00xx=xx';
alpha = -(25*h+1);
beta=2;
gamma=25*h-1;
A=diag(gamma*ones(1,n-3),1)+diag(alpha*ones(1,n-3),-1)+diag(beta*ones(1,n-2),0)
b=zeros(n-2,1);
b(n-2)=-gamma
u=A\b;
```

Ejercicio 2:

```
% -------------------------------------------------------------------------
% Nombre del programa: eval_fun.m
% Autor(es): Valentina Cordova - Emil Vega
% Email del (los) autor(es): airina.cordova@yachaytech.edu.ec -
% emil.vega@yachaytrech.edu.ec
% Fecha de elaboracion: Julio 6 2016
% Breve descripcion del programa: Evalua una función en un vector de valores.
% Datos de entrada:
% ----------------
% fun: función
% valor: vector
%____________
% Datos de salida
% ---------------
% u: vector solución
```

```
% ----------------------------------------------------------------------------
function u = eval_fun(fun,valor)
% Esta funciÃşn devuelve la evaluación de un vector de valores
% El parÃąmetro fun debe ser ingresado como string
f=inline(fun); %Convertimos la cadena de string en una función
n = length(valor);
u=zeros(n,1);
for i=1:n
    u(i)=f(valor(i)); % Evaluamos cada valor en la funciÃşn
end
```

Segundo algoritmo

```
% ----------------------------------------------------------------------------
% Nombre del programa: T6_ejercicio2.m
% Autor(es): Valentina Cordova - Emil Vega
% Email del (los) autor(es): airina.cordova@yachaytech.edu.ec -
% emil.vega@yachaytrech.edu.ec
% Fecha de elaboracion: Julio 6 2016
% Breve descripcion del programa: calcula 'u'
% Datos de entrada:
% ----------------
% n: nodos equidistantes
%_____________
% Datos de salida
% ---------------
% u : vector con los valores arrojados por la función

% ----------------------------------------------------------------------------
function [u,up]=T6_ejercicio2(n)
h=1/(n-1); % espacio entre puntos
x=[h:h:1-h]; % Puntos equidistantes
u = eval_fun('(exp(50*x)-1)/(exp(50)-1)',x); % Obtenemos la evalución de los puntos en l
[~,up,~]=cds_EV(n); % Obtenemos la evaluación de los puntos en la función aproximada
```

Ejercicio 3: Literal a

```
function T6_ejercicio3_a(n)
[~,up,xx]=cds_EV(n); % se obtiene los valores la funciÃşn aproximada y el vector x que v
u=T6_ejercicio2(n); % Se obtiene los valores en la funciÃşn real
hold on
plot (xx,u)
plot (xx,up)
str = sprintf('Grafica de u(x) real y aproximada para los nodos n = %i',n);
title(str)
xlabel('x=(h:h:1-h')
ylabel('up(x) & u(x)')
legend('u(x)','up(x)')
hold off
```

Literal b

```
% ----------------------------------------------------------------------------
% Nombre del programa: T6_ejercicio3_b.m
% Autor(es): Valentina Cordova - Emil Vega
```

```
% Email del (los) autor(es): airina.cordova@yachaytech.edu.ec -
% emil.vega@yachaytrech.edu.ec
% Fecha de elaboracion: Julio 6 2016
% Breve descripcion del programa: Halla la norma infinito y lp del error.
% Datos de entrada:
% ----------------
% n: valor de un nodo
% p: 1 o 2
%_____________
% Datos de salida
% ---------------
% eNorInf: Error en la norma infinito
% tC: Trapecio compuesto (Diferencias finitas)

% ---------------------------------------------------------------------------
function [eNorInf, tC]=T6_ejercicio3_b (n,p)
% Error en la norma infinito
[~,up,xx]=cds_EV(n); % se obtiene los valores la funciÃşn aproximada y el vector x que v
u=T6_ejercicio2(n); % Se obtiene los valores en la funciÃşn real
error=u-up; % Error absoluto
eNorInf = norm(error,inf); %norma infinito

%Error en la norma Lp
h=1/n;
f=abs((u-up)).^p; % f es el valor absoulto elevado a la p del error
tC=(h*(sum(f(2:n-3)))+((h/2)*(f(1)+f(n-2)))); % Aplicamos la regla del trapecio compuest
%eNorLp=tC^(1/p); % eNorInf es el error en la norma Lp
```

Orden de Convergencia

```
% ---------------------------------------------------------------------------
% Nombre del programa: convergencia.m
% Autor(es): Valentina Cordova - Emil Vega
% Email del (los) autor(es): airina.cordova@yachaytech.edu.ec -
% emil.vega@yachaytrech.edu.ec
% Fecha de elaboracion: Julio 6 2016
% Breve descripcion del programa: Calcula el orden de convergencia.
% Datos de entrada:
% ----------------
% n: vector de nodos
%_____________
% Datos de salida
% ---------------
% orden: Arroja el valor del orden de convergencia.

% ---------------------------------------------------------------------------
function orden = convergencia(n)
% Nos da el orden de convergencia
eN = zeros(length(n),1);
orden = zeros(length(n)-1,1);
for i=1:length(n)
    [~, eN(i)]=T6_ejercicio3_b (n(i),1); % Error en la norma Lp para cada n
end
for i=2:length(n)
    orden(i-1)= log2(eN(i-1)/eN(i)); % Orden de convergencia para cada error en la norma
end
```