Algoritmos Numéricos por Computadora

Examen final - Otoño 2020

Sube tu archivo resultado a Canvas → Examen Final antes de las 15:30 horas.

Cada pregunta vale 5 puntos.

Los exámenes son trabajos individuales. Está estrictamente prohibido dar o recibir ayuda de cualquier persona.

El artículo 29 del Reglamento de Alumnos establece que "se calificará como no acreditada (N.A.) cualquier materia cuando el alumno incurra en alguna práctica fraudulenta".

Para resolver el examen puedes usar cualquier función vista en clase o que tenga MATLAB.

1. Entender los errores numéricos de las soluciones computacionales.

1.1 El epsilon de la máquina es el número más pequeño que cuando se suma a 1 resulta en un número más grande que 1. ¿Cuánto vale el epsilon de la máquina?

```
format long
%El epsilon de la máquina vale
eps
```

```
ans =
     2.220446049250313e-16
```

1.2 Determina el número real positivo (no normalizado) más pequeño *mrp*. MATLAB no puede distinguir entre 0 y un número que sea menor a *mrp*.

```
% Smallest non zero
% 0 00000000000 0000000000000000000000000000000000000000000000000001
smallestNonZero = (2^-52)*(2^-1022)  % 2^-1074
```

```
smallestNonZero =
    4.940656458412465e-324
```

```
eps*realmin
```

```
ans =
    4.940656458412465e-324
```

1.3 ¿Cuál es el valor real máximo que puede representar MATLAB? Si cualquier cálculo produce un valor mayor, el resultado sera Inf (error de overflow).

```
% Largest value
%+(1.f)2^1023
maxNumPos = +(1+maxF)*2^1023
```

```
maxNumPos =
```

```
    1.797693134862316e+308
```

```
realmax
```

```
ans =
    1.797693134862316e+308
```

1.4 Determina, con un pequeño programa, cuál es el menor entero positivo *mep* que no puede representarse en MATLAB de manera exacta.

```
% El menor entero positivo que tenemos es el número 1, pero si podemos
% representarlo en Matlab, supongo que la pregunta se refiere al Menor de
% los enteros que ya no puede presentarse (que sería el siguiente número al
% máximo) por lo que tenemos que:

mep = realmax+1
```

```
mep =
    1.797693134862316e+308
```

1.5 Usa aproximaciones de diferencias finitas $O(h)$ hacia adelante

$$f'(x) \approx \frac{f(x+h) - f(x)}{h}$$

y hacia atrás

$$f'(x) \approx \frac{f(x) - f(x-h)}{h}$$

y una aproximación $O(h^2)$ centrada

$$f'(x) \approx \frac{f(x+h) - f(x-h)}{2h}$$

para estimar la primera derivada de la función

$$f(x) = 25x^3 - 6x^2 + 7x - 88$$

Evalua la derivada en $x = 2$ usando un paso de $h = 0.25$. Compara tus resultados con el valor verdadero de la derivada (error relativo). Interpreta los resultados.

```
format short
f = @(x) (25*x^3) - (6*x^2) + (7*x) - 88;

h = 0.25;
x = 2;

% Aproximación hacia adelante
fp_1 = (f(x+h) + f(x)) / h
```

```
fp_1 = 1.1366e+03
```

```
% Aproximación hacia atrás
```

```
fp_2 = (f(x) - f(x-h)) / h
```

```
fp_2 = 248.5625
```

```
% Aproximación centrada
fp_3 = (f(x+h) - f(x-h)) / (2*h)
```

```
fp_3 = 284.5625
```

```
% Derivada calculada con Matlab
fsym = sym(f);
dfs1 = diff(fsym,1);
df_1 = matlabFunction(dfs1)
```

```
df_1 = function_handle with value:
    @(x)x.*-1.2e+1+x.^2.*7.5e+1+7.0
```

```
df_1(x)
```

```
ans = 283
```

```


% Podemos observar los errores de redondeo al tratar de evaluar la
% función en cada uno de los casos anteriores ya que perdemos decimales o
% valores al tratar de hacer las divisiones y las evaluaciones de la
% función con números pequeños o decimales. Es por eso que en la evaluación
% de Matlab tenemos un valor entero, ya que evita esos errores relativos
```

2. Usar un lenguaje de programación matricial de manera eficiente.

2.1 En los métodos multipasos para resolver ecuaciones diferenciales ordinarias se utilizan los últimos *k* valores calculados de la función para determinar el valor de la variable en el siguiente punto. Para no guardar todos los valores calculados de la función, puede usarse una cola de tamaño k.

De manera eficiente agrega 60 a la siguiente cola.

```
cola = [30,35,40,45,50,55];
cola = [cola 60]
```

```
cola = 1×7
    30    35    40    45    50    55    60
```

```
% la cola debe quedar [35,40,45,50,55,60];
```

2.2 A partir de la matriz A

```
A = magic(5)
```

```
A = 5×5
    17    24     1     8    15
    23     5     7    14    16
     4     6    13    20    22
    10    12    19    21     3
    11    18    25     2     9
```

calcula de la manera más compacta posible:

a) La mediana de A

```
M = median(A,'all')
```

```
M = 13
```

b) Cuántos elementos de A son menores a la mediana y cuántos son mayores

```
% Elementos menores
B = A < M
```

```
B = 5×5 logical array
   0   0   1   1   0
   0   1   1   0   0
   1   1   0   0   0
   1   1   0   0   1
   1   0   0   1   1
```

```
% Total de elementos menores a la mediana
nnz(B)
```

```
ans = 12
```

```
% Elementos mayores
C = A > M
```

```
C = 5×5 logical array
   1   1   0   0   1
   1   0   0   1   1
   0   0   0   1   1
   0   0   1   1   0
   0   1   1   0   0
```

```
% Total de elementos mayores a la mediana
nnz(C)
```

```
ans = 12
```

c) Suma ahora la mediana calculada a los elementos que son menores que la mediana

```
A = A + (B*M)
```

```
A = 5×5
    17    24    14    21    15
    23    18    20    14    16
    17    19    13    20    22
    23    25    19    21    16
    24    18    25    15    22
```

d) Cuántos elementos de A son ahora menores a la mediana original (debe ser cero)

```
% Elementos menores
B = A < M
```

```
B = 5×5 logical array
```

```
     0     0     0     0     0
     0     0     0     0     0
     0     0     0     0     0
     0     0     0     0     0
     0     0     0     0     0
```

```
% Total de elementos menores a la mediana
nnz(B)
```

```
ans = 0
```

3. Solucionar sistemas de ecuaciones lineales.

Resuelve el siguiente sistema de ecuaciones utilizando la factorización LU

3x1 - 2x2 + x3 = -10

2x1 + 6x2 - 4x3 = 44

-x1 - 2x2 + 5x3 = -26

3.1 Escribe el sistema de forma matricial Ax=b

```
syms x1 x2 x3
A = [3 -2 1 ; 2 6 -4 ; -1 -2 5]
```

```
A = 3×3
     3    -2     1
     2     6    -4
    -1    -2     5
```

```
x = [x1;x2;x3]
```

x =

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$

```
b = [-10;44;-26]
```

```
b = 3×1
   -10
    44
   -26
```

```
A*x
```

ans =

$$\begin{pmatrix} 3\,x_1 - 2\,x_2 + x_3 \\ 2\,x_1 + 6\,x_2 - 4\,x_3 \\ 5\,x_3 - 2\,x_2 - x_1 \end{pmatrix}$$

3.2 Factoriza la matriz A

```
[L,U,P] = lu(A)
```

```
L = 3×3
    1.0000         0         0
    0.6667    1.0000         0
   -0.3333   -0.3636    1.0000
U = 3×3
    3.0000   -2.0000    1.0000
         0    7.3333   -4.6667
         0         0    3.6364
P = 3×3
     1     0     0
     0     1     0
     0     0     1
```

3.3 comprueba que PA=LU

```
LU = L*U
```

```
LU = 3×3
     3    -2     1
     2     6    -4
    -1    -2     5
```

```
PA = P*A
```

```
PA = 3×3
     3    -2     1
     2     6    -4
    -1    -2     5
```

```
isequal(PA,LU)
```

```
ans = logical
   1
```

3.4 Calcula Pb =P*b

```
Pb = P*b
```

```
Pb = 3×1
   -10
    44
   -26
```

3.5 Calcula x usando L y U

```
x_1 = U\(L\b)
```

```
x_1 = 3×1
    1.0000
    5.0000
   -3.0000
```

```
% Comprobando
A * x_1
```

```
ans = 3×1
  -10.0000
```

```
   44.0000
  -26.0000
```

4. Solucionar ecuaciones diferenciales de forma numérica.

Resuelve el siguiente problema de valor inicial

y'= (1 + 2x) sqrt(y); y(0)=1

sobre el intervalo de x = 0 a 1.

4.1 Analíticamente.

```
% Solucion analitica
syms x y(x)
eqn = diff(y,x) == (1+2*x)*sqrt(y)
```

```
eqn(x) =
```

$$\frac{\partial}{\partial x}\; y(x) = \sqrt{y(x)}\;\;(2\,x+1)$$

```
yG = dsolve(eqn)
```

```
yG =
```

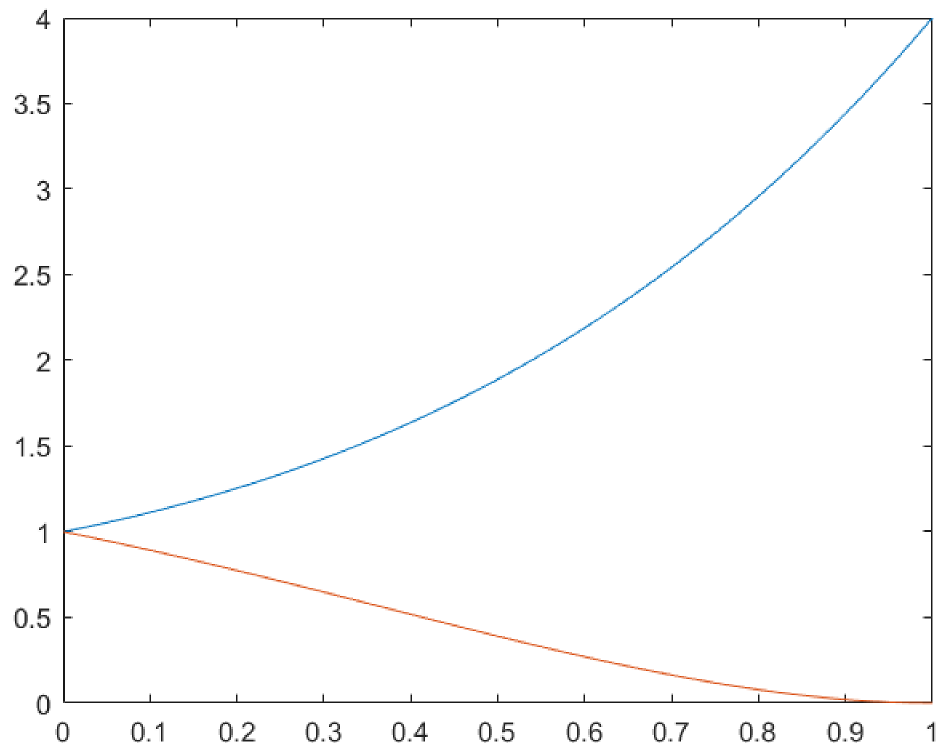
$$\begin{pmatrix} 0 \\ \frac{(C_1 + x\;(x+1))^2}{4} \end{pmatrix}$$

```
cond = y(0) == 1;
yP = dsolve(eqn,cond)
```

```
yP =
```

$$\begin{pmatrix} \frac{(x\;(x+1)+2)^2}{4} \\ \frac{(x\;(x+1)-2)^2}{4} \end{pmatrix}$$

```
fplot(yP, [0 1]);
hold off;
```



4.2 Usando el método de Euler.

```
% Ecuación
f = @(x,y) (1+2*x)*sqrt(y);

% Variables
% Valores iniciales
y0 = 1;
t0 = 0;
tf = 1;
h = .25;

% Solución por euler
[t_e,y_e] = ivps(f, y0, t0, tf, h, 'Euler')
```

```
t_e = 1×5
         0    0.2500    0.5000    0.7500    1.0000
y_e = 1×5
    1.0000    1.2500    1.6693    2.3153    3.2663
```

4.4 Usando el método RK de cuarto orden.

```
% Solución por rk4
[t_r,y_r] = ivps(f, y0, t0, tf, h, 'rk4')
```

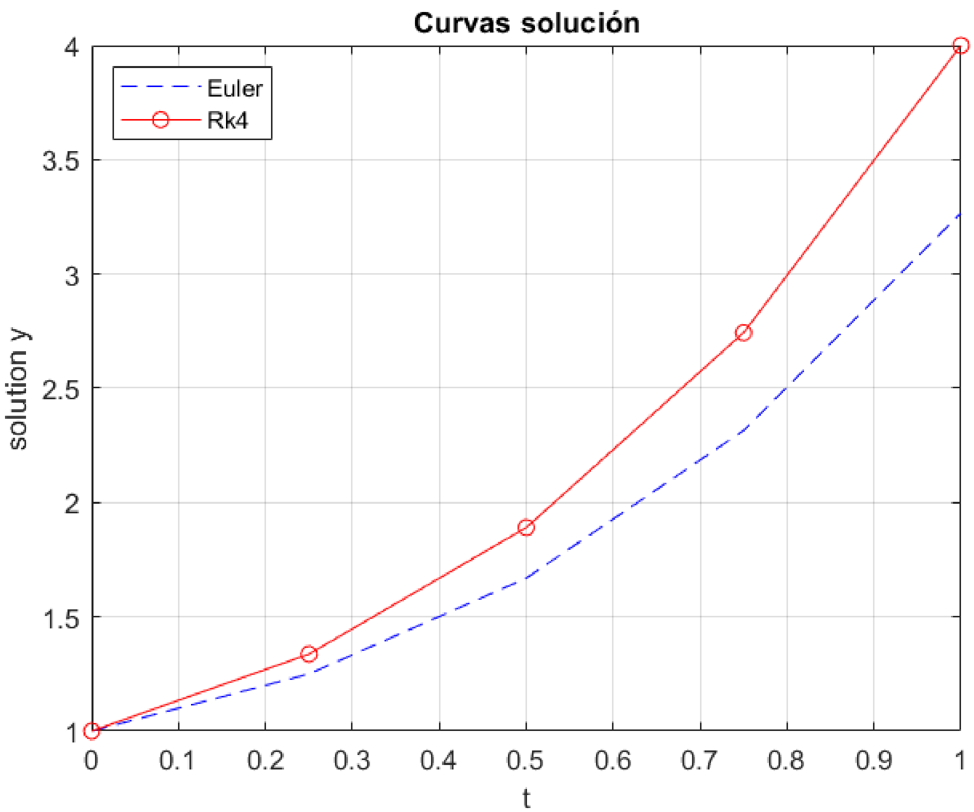
```
t_r = 1×5
         0    0.2500    0.5000    0.7500    1.0000
y_r = 1×5
```

```
    1.0000    1.3369    1.8906    2.7431    3.9998
```

Para los métodos numéricos usa un tamaño de paso de 0.25.

4.5 Despliega todos tus resultados en una misma gráfica (y usa legend para identificar cada solución).

```
% Graficamos
plot(t_e,y_e, '--b')
hold on
grid on
plot(t_r,y_r,'-or')

xlabel('t')
ylabel('solution y')
title('Curvas solución')
legend({'Euler','Rk4'},'Location','northwest')
hold off
```



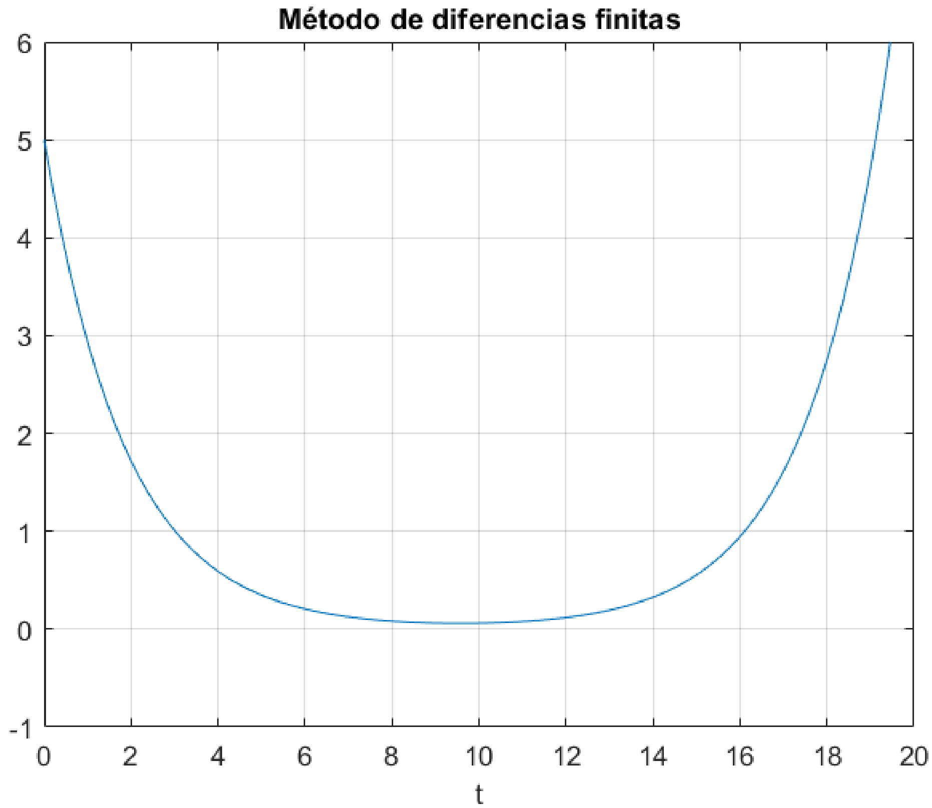
5. Entender el funcionamiento de diversos métodos numéricos.

Resuelve el siguiente problema de valores en la frontera

$7y''- 2 y' - y + x = 0; \quad y(0) = 5, y(20) = 8$

```
a=0;
b=20;
```

```
ya=5;
yb=8;

N=100;
h=(b-a)/(N-1);

bv=zeros(N-2,1);
bv(1) = -ya;
bv(N-2) = -yb;

diagonal = -(2+(2/7)*h^2)*ones(N-2,1);
subdiagonal = ones(length(diagonal),1);
superdiagonal = subdiagonal;

w = Triadiag(subdiagonal,diagonal,superdiagonal,bv);

y=zeros(N,1);
y(1) = ya;
y(N) = yb;
y(2:N-1) = w;

t=a:h:b;
plot(t,y)
title('Método de diferencias finitas');
xlabel('t');
axis([a b -1 6]);
grid on;
```



Solución analítica solicitada

Para el examen final pueden resolver la siguiente ecuación diferencial de segundo orden no homo
con coeficientes constantes:

y'' + 5y' + 6y = 4e(−x) + 5sin x

Sea $y'' + 5y' + 6y = 4e^{-x} + 5\sin x$

Solución $= X_h + X_p$

Resolviendo la homogenea

$$y'' + 5y' + 6y = 0$$

Sea $\lambda^2 + 5\lambda + 6 = 0 \Rightarrow \lambda_1 = -2 \quad \lambda_2 =$

$$\therefore X_h(x) = K_1 e^{-2x} + K_2 e^{-3x}$$

Resolviendo la particular

Proponemos

$$X_p(x) = Ae^{-x} + B\cos(x) + C\,\text{sen}(x)$$

$$\dot{X}_p(x) = -Ae^{-x} - B\,\text{sen}(x) + C\cos(x)$$

$$\ddot{X}_p(x) = Ae^{-x} - B\cos(x) - C\,\text{sen}(x)$$

$$\Rightarrow Ae^{-x} + B\cos(x) + C\,\text{sen}(x) - 5Ae^{-x} - 5B$$

$$+ 5C\cos(x) + 6Ae^{-x} - 6B\cos(x) - 6C\,\text{se}$$

$$\Rightarrow \quad 2Ae^{x} - 5B\cos(x) - 5C\,\text{sen}(x) - 5B\,\text{sen}(x)$$
$$+ 5C\cos(x) = 4e^{-x} + 5\sin x$$

Sacando el sistema

1) $2A = 4$

$A = \frac{4}{2}$

$A = 2$

2) $-5B + 5C = 0$

$-5B = -5C$

$B = C$

3) $-5C - 5(C) = 5$

$-10C = 5$

$C = \frac{5}{-10}$

$C = -\frac{1}{2}$

$$\therefore \; X_p(x) = 2e^{-x} - \frac{1}{2}\cos(x) - \frac{1}{2}\,\text{sen}(x)$$

$$\therefore \; X(x) = X_h(x) + X_p(x)$$

```
function [t,y] = ivps(f, y0, t0, tf, h, type)
    t = t0:h:tf;
    n = length(t);
    y = zeros(1,n);
    y(1) = y0;
    for i=1:1:n-1
        switch type
            case 'Euler'
                phi = f(t(i),y(i));
            case 'rk4'
                k1 = f(t(i),y(i));
                k2 = f( (t(i)+(0.5*h)),y(i)+(0.5*k1*h) );
                k3 = f( (t(i)+(0.5*h)),y(i)+(0.5*k2*h) );
                k4 = f( (t(i)+h),(y(i)+(k3*h)) );
                phi = (1/6)*(k1+(2*k2)+(2*k3)+k4);
        end
        y(i+1) = y(i) + (phi*h);
    end
end

function x = Triadiag(e,f,g,r)

    n=length(f);

    % Eliminación hacia adelante
    for k = 2:n
        factor = e(k)/f(k-1);
        f(k) = f(k) - factor*g(k-1);
        r(k) = r(k) - factor*r(k-1);
    end

    % Eliminación hacia atrás
    x(n) = r(n)/f(n);
    for k = n-1:-1:1
        x(k) = (r(k)-g(k)*x(k+1))/f(k);
    end

end
```