

Soluciones y comentarios

utrs2.m

```
1 function x = utrs2 (b, A)
2     N = size(A,1);
3     x = zeros(N,1);
4     x(N) = b(N)/A(N,N);
5     for j = N-1:-1:1
6         x(j) = ( b(j) - A(j,j+1)*x(j+1) )/A(j,j);
7     end
8 end
```

parmat.m

```
1 function B = parmat (A)
2     if 2*size(A,1) == size(A,2)
3         % Extrae de A las columnas impares y pares
4         col_imp = A(:, 1:2:size(A,2) );
5         col_par = A(:, 2:2:size(A,2) );
6         % Construye B
7         B = [col_imp', col_par'];
8     else
9         disp('Dimensiones incorrectas')
10        B = [];
11    end
12 end
```

matpar.m

```
1 function B = matpar (A)
2     if size(A,1) == 2*size(A,2)
3         % Extrae de A las filas impares y pares
4         fil_imp = A(1:2:size(A,1), : );
5         fil_par = A(2:2:size(A,1), : );
6         % Construye B
7         B = [fil_imp'; fil_par'];
8     else
9         disp('Dimensiones incorrectas')
10        B = [];
11    end
12 end
```

gra.m

```
1 x = linspace(2,6,200);
2 y = (x+10)./x.*sin(3/2*x) + exp(-x./(x+1) );
3 figure(1)
4 [mi,jmi] = min(y);
5 [ma,jma] = max(y);
6 plot(x,y, '-o', [x(jmi),x(jma)], [mi,ma], 'o')
```

```
graf.m
1 x = linspace(3/2,5,200);
2 y = exp(x./(x+4)).*sin(2*x) + log(1+x.^2);
3 figure(1)
4 [mi,jmi] = min(y);
5 [ma,jma] = max(y);
6 plot(x,y,'-',[x(jmi),x(jma)],[mi,ma],'o')
```

```
grafi.m
1 x = linspace(3,11/2,200);
2 y = cos(x./(x+4)).*sin(3*x) + (x.^2+1)./(x+1);
3 figure(1)
4 [mi,jmi] = min(y);
5 [ma,jma] = max(y);
6 plot(x,y,'-',[x(jmi),x(jma)],[mi,ma],'o')
```

En Moodle os he puesto comentarios personalizados. Aquí van algunos de los temas más comunes.

- Algunos olvidáis los puntos al definir la función de la gráfica. La penalización depende del número de los que falten. Seguramente uno solo es un despiste. Si faltan todos, es señal de que no se entiende bien.
- Si en `matpar` o `parmat` no se devuelve la matriz vacía sino una matriz nula, hay una penalización de $-0,5$.
- Cuando en la sustitución regresiva se recorren también los elementos nulos, lo he penalizado con $-0,2$ porque esto cambia la eficiencia de tiempo lineal a cuadrático en la dimensión. En la práctica, no he penalizado otras cosas finas relativas a la eficiencia.
- No he penalizado el cambio de orden de los argumentos de `utrs2`, pero me da un poco de rabia después de que hice hincapié en ello en las instrucciones que os di al principio.
- Algunos hacéis programas algo enrevesados. Un ejemplo muy repetido es crear un bucle ascendente y después definir una nueva variable para que se recorran los valores de forma descendente. Recordad que `n:-1:1` da los valores de `n` a `1`. No penaliza.