

1. Laboratorio 6: Integración Numérica

Funciones esenciales

- `@(x) f(x)` — funciones anónimas.
- `linspace(a,b,N)` — malla uniforme.
- `a:h:b` — malla con paso fijo h .

Reglas de cuadratura (implementadas por el alumno)

- `punto_medio(a,b,f,N)`
- `trapecio(a,b,f,N)`
- `simpson(a,b,f,N)`

Integración incorporada en Matlab

- `integral(f,a,b)`
- `integral2(f,xmin,xmax,ymin,ymax)`

Splines e interpolación

- `spline(x,y)` — spline cúbica.
- `ppval(S,x)` — evaluar la spline.

Gráficos básicos

- `plot(x,y), hold on, xlabel, ylabel, title, legend.`

2. Laboratorio 7: EDO de Primer Orden

Métodos de Euler (archivos del curso)

- `euler_explicito(t0,tf,y0,N,f)`
- `euler_implicito(t0,tf,y0,N,f)`

Solvers incorporados

- `[t,y] = ode45(f,[t0 tf],y0)`
- `[t,y] = ode15s(f,[t0 tf],y0)` — para problemas *stiff*.
- Uso con malla fija: `tspan = t0:h:tf; [t,y] = ode45(f,tspan,y0).`

Opciones de tolerancia

- `options = odeset('RelTol',1e-6,'AbsTol',1e-8);`
- `[t,y] = ode45(f,tspan,y0,options);`

3. Laboratorio 8: Sistemas y EDO de Segundo Orden

Métodos Euler para sistemas

- `euler_explicito_2orden`
- `euler_implicito_2orden`

Reducción a sistema de primer orden

Ejemplo:

$$y'' = -ky \quad \Rightarrow \quad \begin{cases} y'_1 = y_2, \\ y'_2 = -ky_1. \end{cases}$$

Vectores de estado

- Se usa `[y1; y2]` para representar el sistema.

4. Laboratorio 9: Sistemas de Ecuaciones Lineales I

Funciones necesarias (hechas por el alumno)

- `sustitucion_progresiva(L,b)`
- `sustitucion_regresiva(U,b)`
- `regla_de_cramer(A,b)`

Comandos esenciales

- `x = A \ b` — operador backslash.
- `[L,U,P] = lu(A)` — LU con pivoteo.
- `rand(n,m), zeros(n,m).`
- `norm(x,inf)` — norma infinito.
- `tic, toc` — medir tiempos.

5. Laboratorio 10: Sistemas de Ecuaciones II

Factorizaciones

- `[L,U,P] = lu(A)` — LU con pivoteo.
- `R = chol(A)` — Cholesky (A SPD).

Matrices y estructuras

- `issymmetric(A)` — verificar simetría.
- `eig(A)` — autovalores, útil para SPD.
- `spy(A)` — mostrar estructura dispersa.
- `nnz(A)` — número de elementos no nulos.

Matrices tridiagonales / banda

Construcción típica:

- `diag(a*ones(n,1)) + diag(b*ones(n-1,1),1) + diag(c*ones(n-1,1),-1)`