

Máster Universitario en Ingeniería Matemática y Computación

- OPTIMIZACIÓN -

Actividad 1: Optimización de funciones

Alumno: Pablo Ruiz Molina

1. Selección de intervalo de longitud 1 de las funciones test

En el primer apartado de esta actividad se busca seleccionar un intervalo de longitud 1, en el cual la función sea unimodal y contenga el mínimo localizado en el valor positivo más pequeño.

Para entender qué tipo de funciones hay que enfrentar, el primer paso es visualizarlas. Se han escogido rangos ciertamente amplios para entender las funciones de manera general. El código usado para ello es el siguiente:

```
% Visualización inicial de las funciones

% Limpieza de la consola, variables y figuras
clc; clear; close all;

% Definición de las funciones objetivo
f1 = @(x) (x.^6) - 7*(x.^3) + 7;
f2 = @(x) log(x) - sin(x);
f3 = @(x) exp(1).^cos(x) - x;
f4 = @(x) funcionrr(x);
%f4 = @(x) x;

%% Visualización inicial
x1_values = linspace(-10, 10, 1000); y1_values = f1(x1_values);
x2_values = linspace(0, 10, 1000); y2_values = f2(x2_values);
x3_values = linspace(-10, 10, 1000); y3_values = f3(x3_values);
x4_values = linspace(-10, 10, 1000); y4_values = f4(x4_values);

figure (1);
subplot(2, 2, 1), plot(x1_values, y1_values, 'b-', 'LineWidth', 2);
title('f1(x) = x^6 - 7*x^3 + 7');
ylim([-10, 50]);
xlabel('x');
ylabel('f(x)');
legend('f1(x)');
hold on;
grid on;

subplot(2, 2, 2), plot(x2_values, y2_values, 'b-', 'LineWidth', 2);
title('f2(x) = ln(x) - sin(x)');
ylim([-5, 5]);
```

```

xlabel('x');
ylabel('f(x)');
legend('f2(x)');
hold on;
grid on;

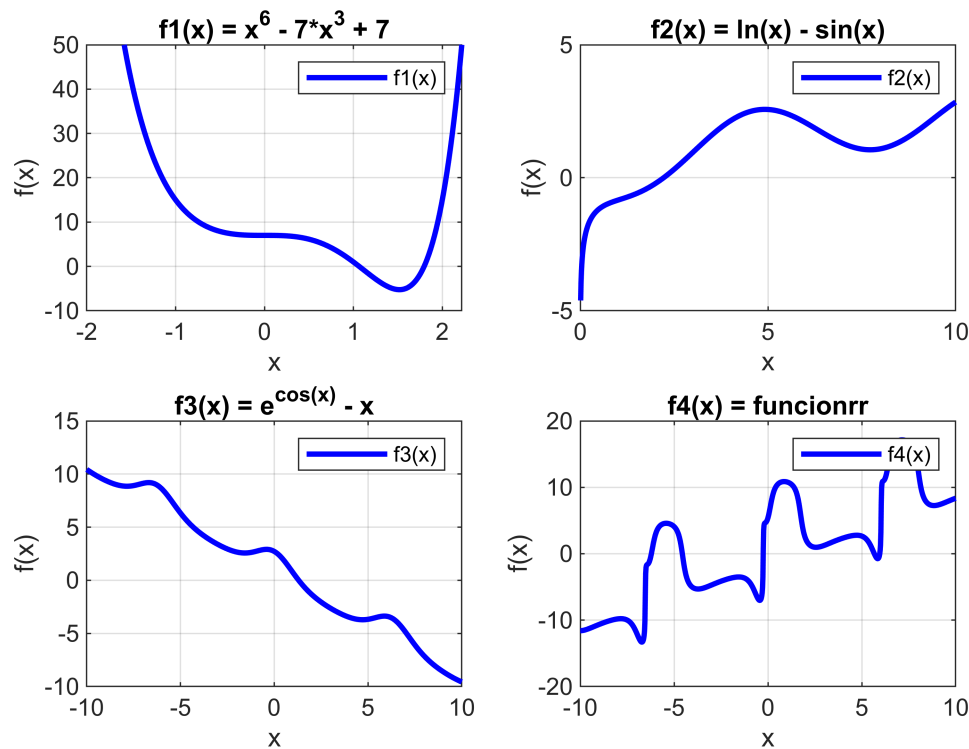
subplot(2, 2, 3), plot(x3_values, y3_values, 'b-', 'LineWidth', 2);
title('f3(x) = e^{cos(x)} - x', 'Interpreter', 'tex');
xlabel('x');
ylabel('f(x)');
legend('f3(x)');
hold on;
grid on;

subplot(2, 2, 4), plot(x4_values, y4_values, 'b-', 'LineWidth', 2);
title('f4(x) = funcionrr');
xlabel('x');
ylabel('f(x)');
legend('f4(x)');
hold on;
grid on;

sgtitle('Visualización inicial de las funciones');

```

Visualización inicial de las funciones



A destacar cómo la segunda función (arriba a la derecha) cuenta con un límite de $-\infty$ en $x = 0$. Es de hecho esperable según el rango donde se define la función en el enunciado: $(0, \infty)$.

Es fácil seleccionar para cada función un intervalo unidad en el que la función sea unimodal, con un mínimo, y que a su vez sea el más cercano a 0 por la parte positiva.

```
%% Selección del intervalo de búsqueda de cada función
% Calcula intervalos de manera automática
intervalo_f1 = [1, 2];
intervalo_f2 = [7.25, 8.25];
intervalo_f3 = [4.25, 5.25];
intervalo_f4 = [2, 3];

figure (2);
subplot(2, 2, 1), plot(x1_values, y1_values, 'b-', 'LineWidth', 2);
subplot(2, 2, 1), xline(intervalo_f1(1), '--r', 'a', 'LabelOrientation',
'horizontal', 'LineWidth', 1.5);
subplot(2, 2, 1), xline(intervalo_f1(2), '--g', 'b', 'LabelOrientation',
'horizontal', 'LineWidth', 1.5);
legend('f1(x)', 'a', 'b');
title('f1(x) = x^6 - 7*x^3 + 7');
xlim([0, 3]);
ylim([-30, 30]);
xlabel('x');
ylabel('f(x)');
hold on;
grid on;

subplot(2, 2, 2), plot(x2_values, y2_values, 'b-', 'LineWidth', 2);
subplot(2, 2, 2), xline(intervalo_f2(1), '--r', 'a', 'LabelOrientation',
'horizontal', 'LineWidth', 1.5);
subplot(2, 2, 2), xline(intervalo_f2(2), '--g', 'b', 'LabelOrientation',
'horizontal', 'LineWidth', 1.5);
legend('f2(x)', 'a', 'b');
title('f2(x) = ln(x) - sin(x)');
xlim([6, 9]);
xlabel('x');
ylabel('f(x)');
hold on;
grid on;

subplot(2, 2, 3), plot(x3_values, y3_values, 'b-', 'LineWidth', 2);
subplot(2, 2, 3), xline(intervalo_f3(1), '--r', 'a', 'LabelOrientation',
'horizontal', 'LineWidth', 1.5);
subplot(2, 2, 3), xline(intervalo_f3(2), '--g', 'b', 'LabelOrientation',
'horizontal', 'LineWidth', 1.5);
legend('f3(x)', 'a', 'b');
title('f3(x) = e^{cos(x)} - x', 'Interpreter', 'tex');
xlim([3.25, 6.25]);
xlabel('x');
ylabel('f(x)');
hold on;
grid on;
```

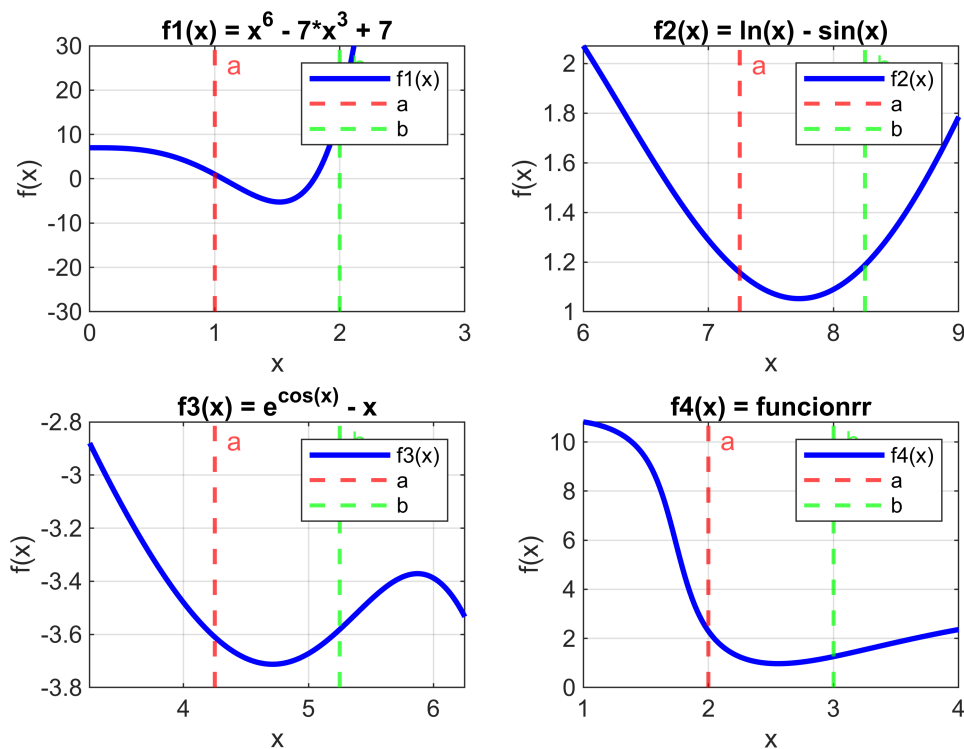
```

subplot(2, 2, 4), plot(x4_values, y4_values, 'b-', 'LineWidth', 2);
subplot(2, 2, 4), xline(intervalo_f4(1), '--r', 'a', 'LabelOrientation',
'horizontal', 'LineWidth', 1.5);
subplot(2, 2, 4), xline(intervalo_f4(2), '--g', 'b', 'LabelOrientation',
'horizontal', 'LineWidth', 1.5);
legend('f4(x)', 'a', 'b');
title('f4(x) = funcionrr');
xlim([1, 4]);
xlabel('x');
ylabel('f(x)');
hold on;
grid on;

sgtitle('Selección de intervalos');

```

Selección de intervalos



2. Algoritmos de búsqueda: Búsqueda dicotómica

Una vez los intervalos de búsqueda están seleccionados, se van a desarrollar y probar los diferentes algoritmos de búsqueda objetivo de la actividad.

2.1 Algoritmo de tipo búsqueda dicotómica (*budi.m*)

La búsqueda dicotómica es un método simple y eficiente diseñado para encontrar el mínimo de una función unimodal dentro de un intervalo dado. Este tipo de función tiene una sola tendencia descendente seguida de una ascendente (o viceversa), garantizando que exista un único mínimo local dentro del intervalo considerado. El algoritmo es iterativo y trabaja reduciendo progresivamente el tamaño del intervalo donde se encuentra el

mínimo, basándose en evaluaciones estratégicas de la función en puntos seleccionados dentro del intervalo. El código implementado ha sido el siguiente:

```
function [x_min, iter] = budi(f, a, b, delta)
    % Método de búsqueda dicotómica
    % Entradas:
    %   f: función objetivo
    %   a, b: extremos iniciales del intervalo unimodal
    %   tol: tolerancia para la convergencia
    %
    % Salidas:
    %   x_min: punto donde se alcanza el mínimo
    %   iter: número de iteraciones realizadas

    % Implementación del método

    % valores iniciales
    eps = delta/10; % Siempre más pequeño que la tolerancia.
    % Distancia entre los puntos (epsilon < delta)

    % Inicialización de las variables
    iter = 0; % Contador de iteraciones
    max_iter = 100; % Número máximo de iteraciones

    % mientras (b - a) >= tol
    while (b - a) > delta && iter < max_iter
        iter = iter + 1;

        % Calcular xa y xb
        x = (a + b) / 2;
        xa = x - eps / 2;
        xb = x + eps / 2;

        % Evaluar la función en los extremos del nuevo intervalo
        if f(xa) < f(xb)
            b = xb;
        else
            a = xa;
        end
    end

    % Cálculo del mínimo
    x_min = (a + b) / 2;
end
```

Los resultados obtenidos una vez se pasa el algoritmo a todas las funciones son los siguientes:

```
% Parámetros de la búsqueda dicotómica
delta = 1e-6; % Tolerancia (delta)

% Cálculo de las iteraciones mínimas de Budi
```

```

iter_min_budy = log(delta)/log(.5);
disp(['Número de iteraciones mínimas para búsqueda dicotómica: iter_min = ',
num2str(iter_min_budy)])

```

Número de iteraciones mínimas para búsqueda dicotómica: iter_min = 19.9316

```

% Llamadas a los métodos
tic;
[x1_budi, iter1_budi] = budi(f1, (1), intervalo_f1(2), delta);
tiempo1_budi = toc * 1000;

tic;
[x2_budi, iter2_budi] = budi(f2, intervalo_f2(1), intervalo_f2(2), delta);
tiempo2_budi = toc * 1000;

tic;
[x3_budi, iter3_budi] = budi(f3, intervalo_f3(1), intervalo_f3(2), delta);
tiempo3_budi = toc * 1000;

tic;
[x4_budi, iter4_budi] = budi(f4, intervalo_f4(1), intervalo_f4(2), delta);
tiempo4_budi = toc * 1000;

figure (3);
f1_budy = f1(x1_budi);
subplot(2, 2, 1), plot(x1_values, y1_values, 'b-', 'LineWidth', 2);
hold on;
subplot(2, 2, 1), plot(x1_budi, f1_budy, 'r*', 'MarkerSize', 10, 'MarkerFaceColor',
'r'); % Punto óptimo
subplot(2, 2, 1), xline(intervalo_f1(1), '--r', 'a', 'LabelOrientation',
'horizontal', 'LineWidth', 1.5);
subplot(2, 2, 1), xline(intervalo_f1(2), '--g', 'b', 'LabelOrientation',
'horizontal', 'LineWidth', 1.5);
legend('f1(x)', 'a', 'b', 'min');
title('f1(x) = x^6 - 7*x^3 + 7');
xlim([0, 3]);
ylim([-30, 30]);
xlabel('x');
ylabel('f(x)');
grid on;

f2_budy = f2(x2_budi);
subplot(2, 2, 2), plot(x2_values, y2_values, 'b-', 'LineWidth', 2);
hold on;
subplot(2, 2, 2), plot(x2_budi, f2_budy, 'r*', 'MarkerSize', 10, 'MarkerFaceColor',
'r'); % Punto óptimo
subplot(2, 2, 2), xline(intervalo_f2(1), '--r', 'a', 'LabelOrientation',
'horizontal', 'LineWidth', 1.5);
subplot(2, 2, 2), xline(intervalo_f2(2), '--g', 'b', 'LabelOrientation',
'horizontal', 'LineWidth', 1.5);

```

```

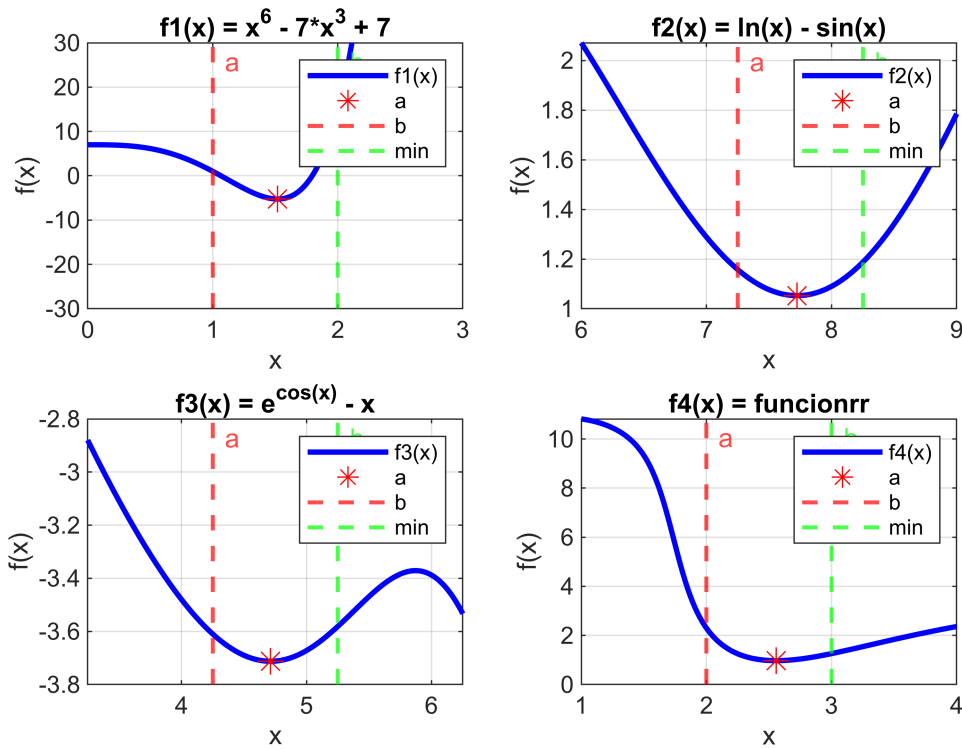
legend('f2(x)', 'a', 'b', 'min');
title('f2(x) = ln(x) - sin(x)');
xlim([6, 9]);
xlabel('x');
ylabel('f(x)');
grid on;

f3_budy = f3(x3_budi);
subplot(2, 2, 3), plot(x3_values, y3_values, 'b-', 'LineWidth', 2);
hold on;
subplot(2, 2, 3), plot(x3_budi, f3_budy, 'r*', 'MarkerSize', 10, 'MarkerFaceColor',
'r'); % Punto óptimo
subplot(2, 2, 3), xline(intervalo_f3(1), '--r', 'a', 'LabelOrientation',
'horizontal', 'LineWidth', 1.5);
subplot(2, 2, 3), xline(intervalo_f3(2), '--g', 'b', 'LabelOrientation',
'horizontal', 'LineWidth', 1.5);
legend('f3(x)', 'a', 'b', 'min');
title('f3(x) = e^{cos(x)} - x', 'Interpreter', 'tex');
xlim([3.25, 6.25]);
xlabel('x');
ylabel('f(x)');
grid on;

f4_budy = f4(x4_budi);
subplot(2, 2, 4), plot(x4_values, y4_values, 'b-', 'LineWidth', 2);
hold on;
subplot(2, 2, 4), plot(x4_budi, f4_budy, 'r*', 'MarkerSize', 10, 'MarkerFaceColor',
'r'); % Punto óptimo
subplot(2, 2, 4), xline(intervalo_f4(1), '--r', 'a', 'LabelOrientation',
'horizontal', 'LineWidth', 1.5);
subplot(2, 2, 4), xline(intervalo_f4(2), '--g', 'b', 'LabelOrientation',
'horizontal', 'LineWidth', 1.5);
legend('f4(x)', 'a', 'b', 'min');
title('f4(x) = funcionrr');
xlim([1, 4]);
xlabel('x');
ylabel('f(x)');
grid on;

sgtitle('Búsqueda dicotómica: Mínimos encontrados.');
```

Búsqueda dicotómica: Mínimos encontrados.



En la siguiente tabla, se detallan los valores obtenidos para el valor x_{\min} , para el valor que cobra la función, las iteraciones necesarias, así como el tiempo de ejecución:

```
% Definir datos para las columnas
funciones = {'f1(x)'; 'f2(x)'; 'f3(x)'; 'f4(x)'};
x_min = [x1_budi; x2_budi; x3_budi; x4_budi];
f_x_min = [f1(x1_budi); f2(x2_budi); f3(x3_budi); f4(x4_budi)];
iter = [iter1_budi; iter2_budi; iter3_budi; iter4_budi];
times = [tiempo1_budi; tiempo2_budi; tiempo3_budi; tiempo4_budi];

% Crear una tabla
T = table(funciones, x_min, f_x_min, iter, times, 'VariableNames', {'Funciones', 'x min', 'f(x min)', 'Iteraciones', 'Tiempos (msec)'});

% Mostrar la tabla en el Live Editor
disp(T);
```

Funciones	x min	f(x min)	Iteraciones	Tiempos (msec)
{'f1(x)'}	1.5183	-5.25	21	1.9877
{'f2(x)'}	7.7242	1.0528	21	2.0157
{'f3(x)'}	4.7124	-3.7124	21	1.2384
{'f4(x)'}	2.5574	0.97258	21	198.37

ghjk