Algoritmos Numéricos por Computadora

COM - 14105

"Actually, a person does not really understand something until he can teach it to a computer"
Donald Knuth, 1974

Objetivos

- Solucionar sistemas de ecuaciones lineales y ecuaciones diferenciales de forma numérica
- Entender el funcionamiento de diversos métodos numéricos
- Entender los errores numéricos de las soluciones computacionales
- Familiarizarse con el modelado matemático de sistemas físicos
- Usar un lenguaje de programación matricial de manera eficiente

Temario

1. Introducción
    1. Modelado de sistemas dinámicos
    2. Redondeo y truncamiento
    3. Raíces de funciones y optimización
    4. Números complejos
2. Sistemas lineales
    1. Valores y vectores propios
    2. Eliminación de Gauss
    3. Factorizaciones
    4. Métodos iterativos
    5. Sistemas no lineales
3. Ecuaciones diferenciales ordinarias
    1. Interpolación e integración
    2. Soluciones analíticas sencillas
    3. Problemas con valor inicial
    4. Sistemas de ecuaciones lineales de primer orden
    5. ODE de orden superior
    6. Métodos de paso variable, implícitos y multipasos
    7. Problemas con valores en la frontera
4. Ecuaciones diferenciales parciales (lineales de segundo orden)

Evaluación

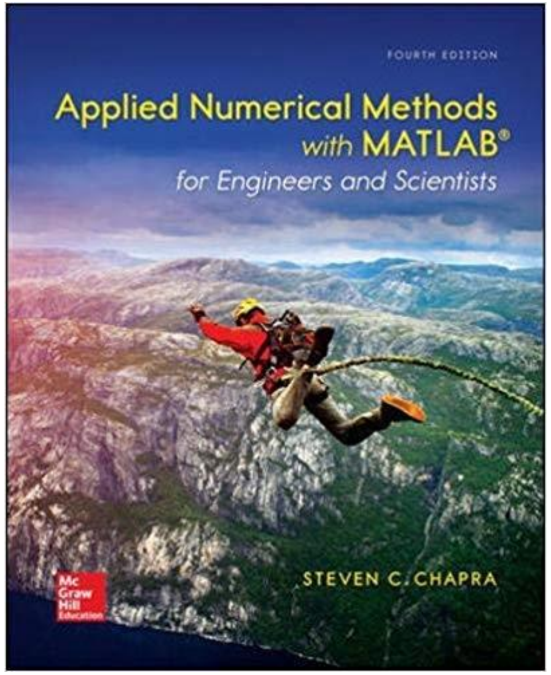
• Tareas	10%
• Proyectos	20%
• Primer examen parcial	20%
• Segundo examen parcial	20%
• Examen final	30%

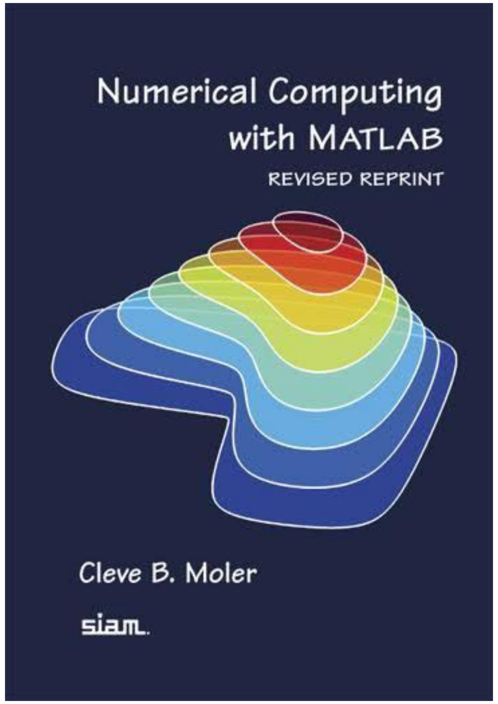
Student Outcome - ABET

An ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics.

Bibliografía

Steven Chapra, *Applied Numerical Methods with MATLAB for Engineers and Scientists*, McGraw- Hill, Fourth edition, 2018.



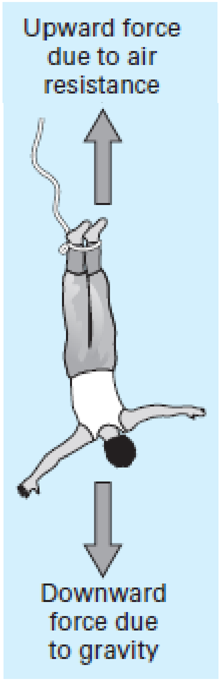


Cleve Moler, *Numerical Computing with MATLAB*, SIAM, 2008.

Introducción

Modelado de sistemas dinámicos

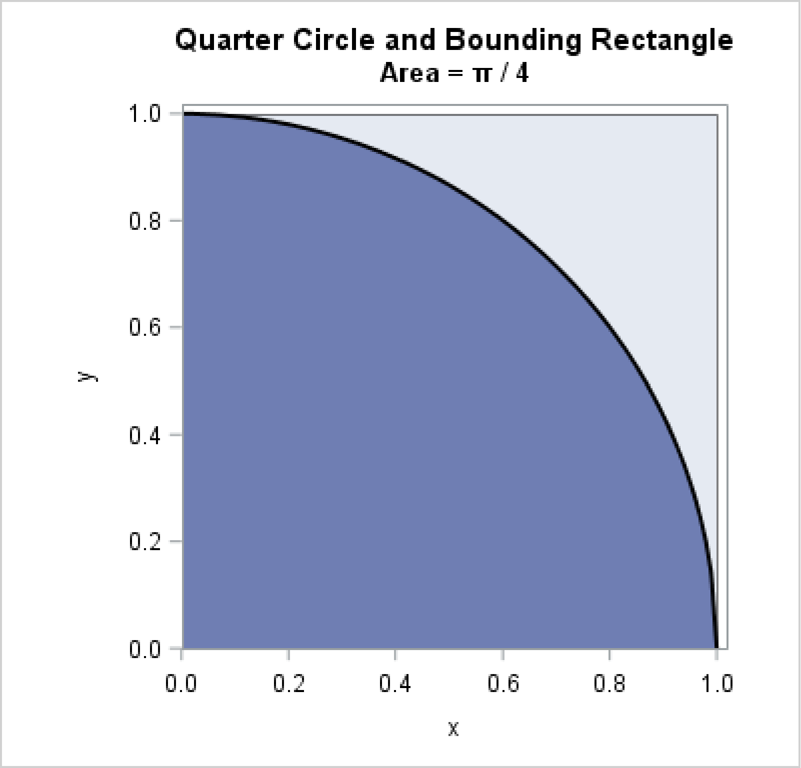
1. Fuerzas que actúan en caída libre



Introducción

Modelado de sistemas dinámicos

2. Montecarlo



https://blogs.sas.com/content/iml/2016/03/14/monte-carlo-estimates-of-pi.html

Introducción

Modelado de sistemas dinámicos

3. Satélite geoestacionario / Luna / + Sol

Introducción

Redondeo y truncamiento

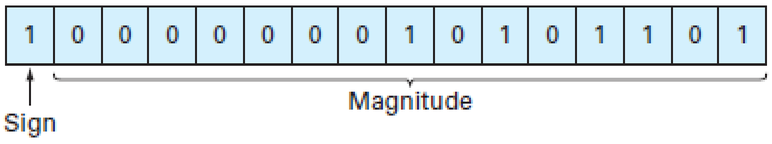
Double-precision floating-point numbers

ANSI/IEEE Standard 754

Introducción

Redondeo

¿-173 en binario?



¿Números decimales?

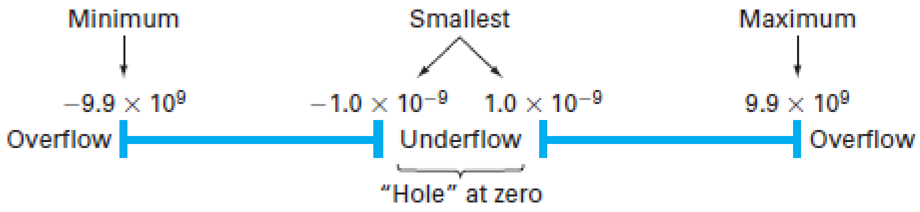
Punto flotante

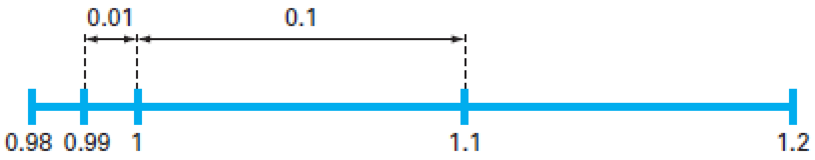
Normalización

Ejemplo

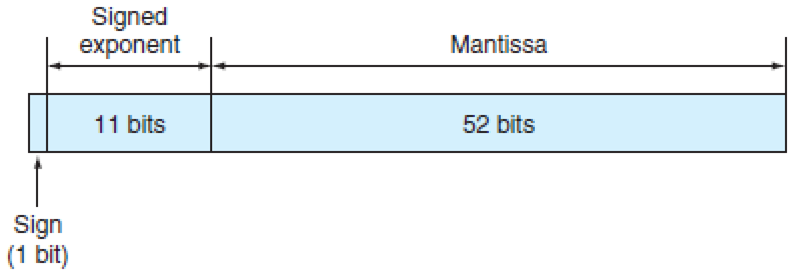
Base-10 computer – 5-digit word size

$s_1d_1d_2 \times 10^{s0d0}$





IEEE double-precision format



$$x = \pm(1 + f) \cdot 2^e$$

$$0 \leq f < 1$$

$$f = (\text{integer} < 2^{52})/2^{52}$$

$$-1022 \leq e \leq 1023$$

$$e = \text{integer}$$

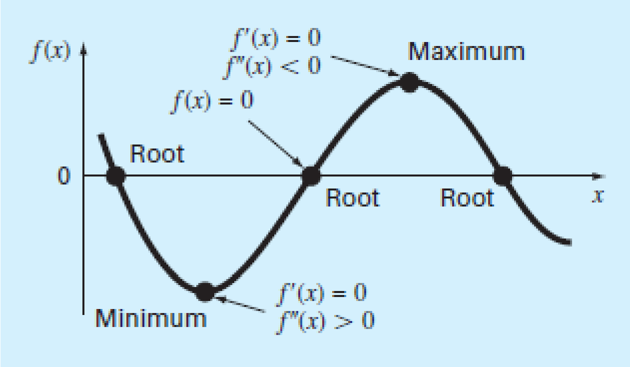
Finite f implies finite *precision*.

Finite e implies finite *range*

Floating point numbers have discrete spacing, a maximum and a minimum.

Introducción

Raíces y optimización

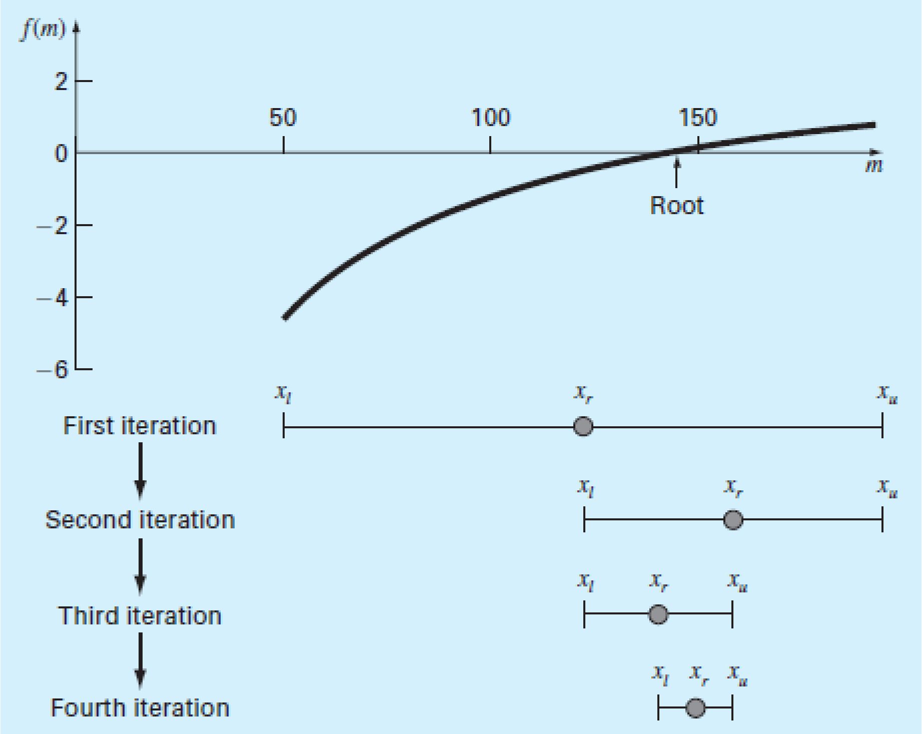


Raíces:

Métodos cerrados: bisección, interpolación lineal

Métodos abiertos: Newton-Raphson, secante

Bisección



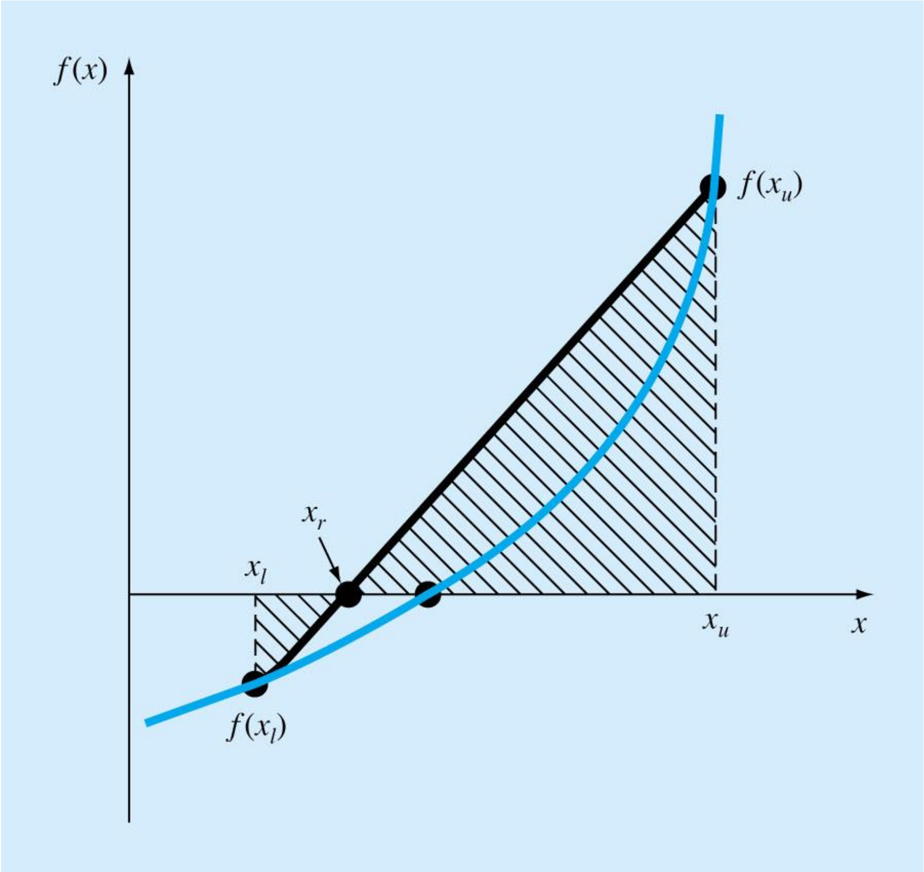
Convergencia lineal

Rapidez de convergencia

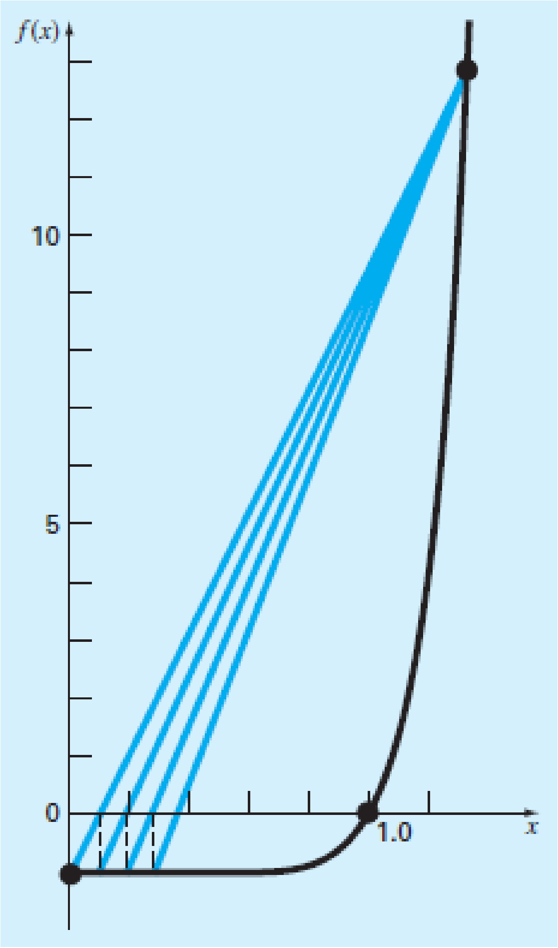
La sucesión p_n converge a p con orden α y una constante de error asintótica λ si

$$\lim_{n\to\infty} \frac{|p_{n+1} - p|}{|p_n - p|^{\alpha}} = \lambda.$$

Interpolación lineal (falsa posición)



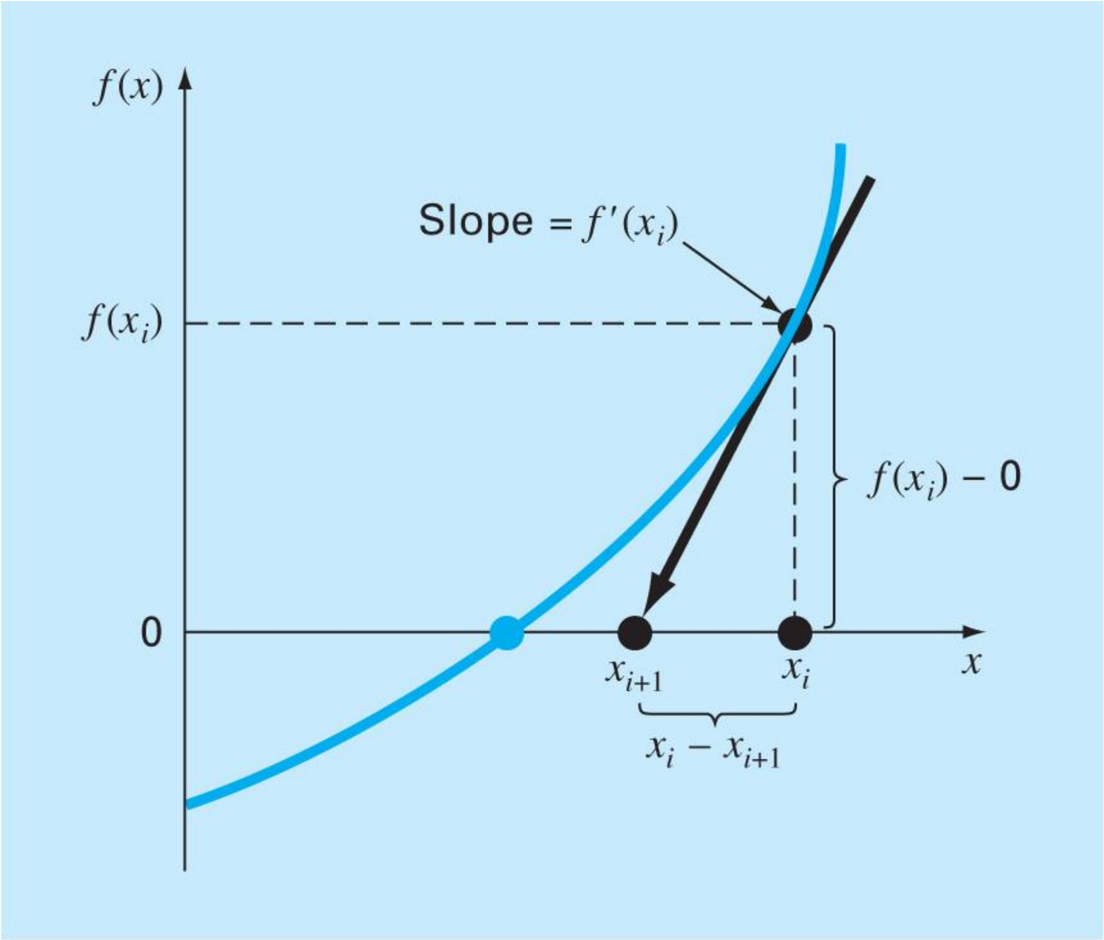
Interpolación lineal (falsa posición)



$f(x) = x^{10} - 1$

$f(x) = x^2 - M$

Newton-Raphson



$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

Newton-Raphson (raíz cuadrada de M)

$$f(x) = x^2 - M$$

$$\begin{aligned} x_{n+1} &= x_n - \frac{x_n^2 - M}{2x_n} \\ &= \frac{1}{2}\left(x_n + \frac{M}{x_n}\right) \end{aligned}$$

Newton-Raphson (convergencia cuadrática)

$$f(x_r) = f(x_n) + f'(x_n)(x_r - x_n) + \frac{f''(\xi)}{2}(x_r - x_n)^2 = 0$$

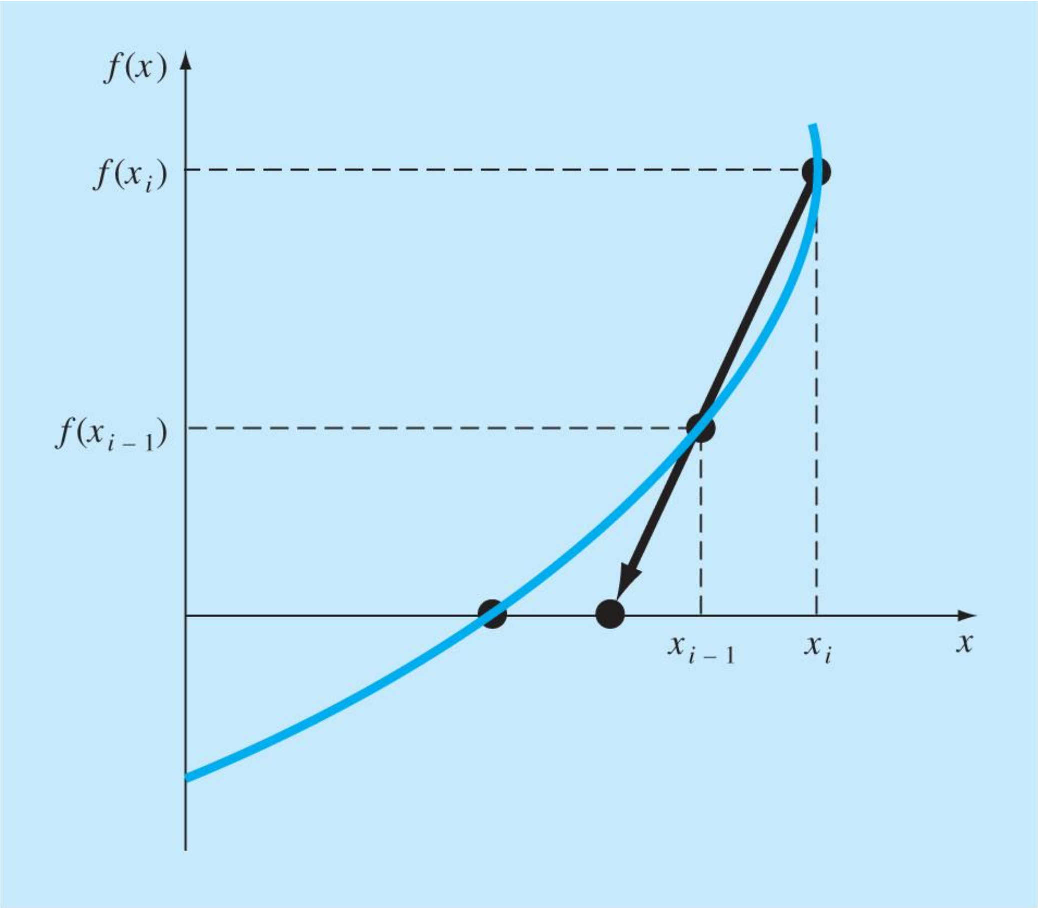
$$f(x_n) = f'(x)(x_n - x_{n+1})$$

$$e_{n+1} = -\frac{1}{2}\frac{f''(\xi)}{f'(x_n)}e_n^2$$

$$e_{n+1} = O(e_n^2)$$

Secante

La derivada se aproxima con una diferencia finita hacia atrás:



$$s_n = \frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}}$$

$$x_{n+1} = x_n - \frac{f(x_n)}{s_n}$$

Problema

6.22 You are designing a spherical tank (Fig. P6.22) to hold water for a small village in a developing country. The volume of liquid it can hold can be computed as

$$V = \pi h^2 \frac{[3R - h]}{3}$$

where V = volume [m^3], h = depth of water in tank [m], and R = the tank radius [m].

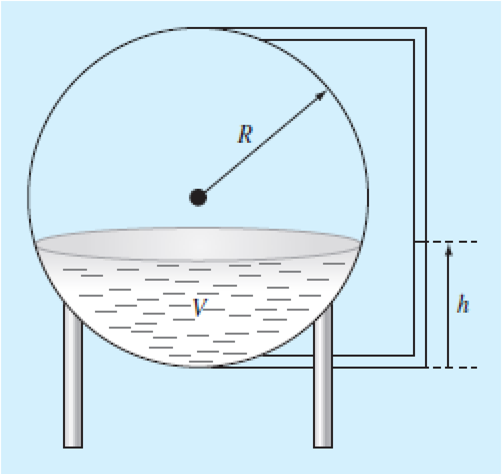
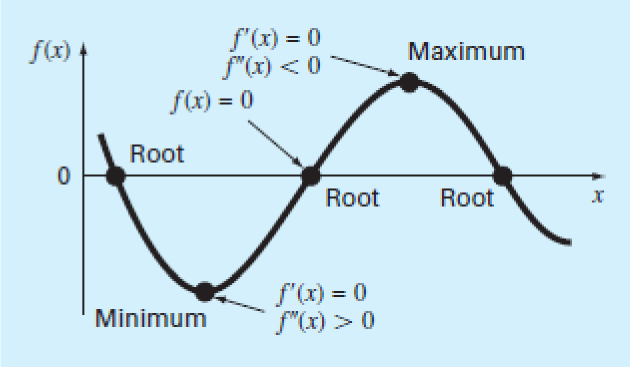


FIGURE P6.22

If $R = 3$ m, what depth must the tank be filled to so that it holds 30 m^3? Use three iterations of the most efficient numerical method possible to determine your answer. Determine the approximate relative error after each iteration. Also, provide justification for your choice of method. Extra information: **(a)** For bracketing methods, initial guesses of 0 and R will bracket a single root for this example. **(b)** For open methods, an initial guess of R will always converge.

Introducción

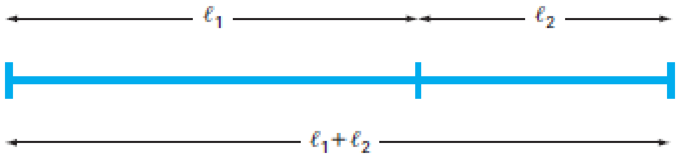
Raíces y optimización



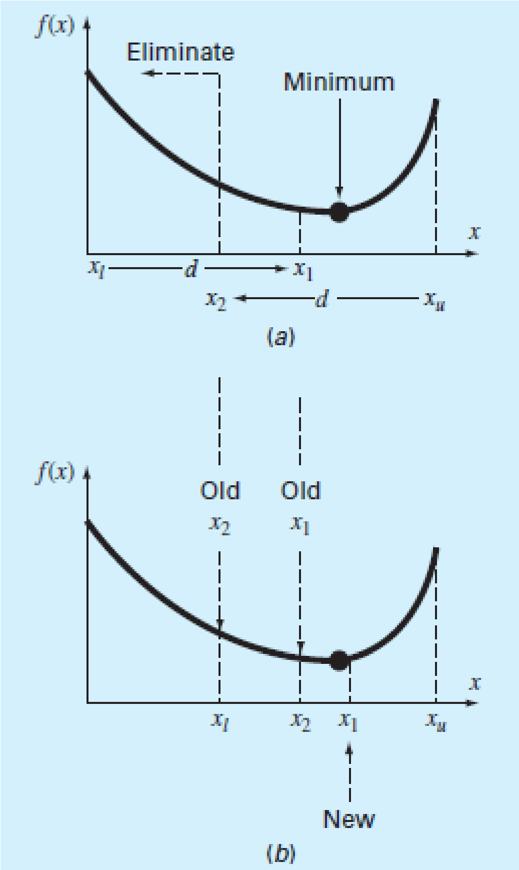
Optimización:

- Proporción áurea (golden ratio)
- Interpolación parabólica

Proporción áurea



$$\frac{l_1+l_2}{l_1} = \frac{l_1}{l_2} = \emptyset$$

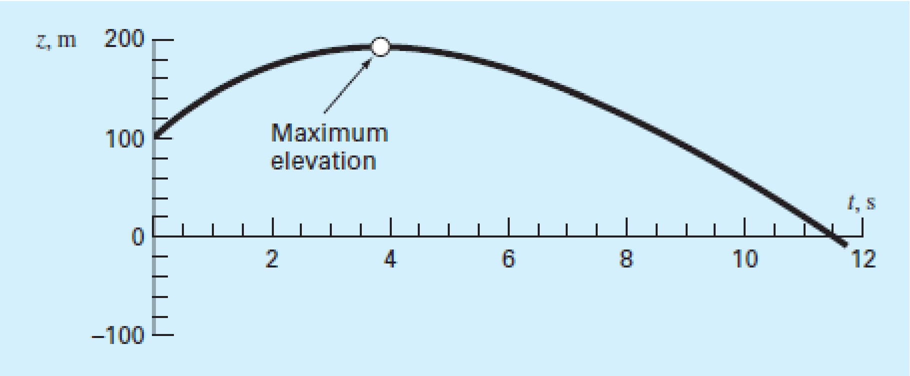


$$x_1 = x_l + d$$
$$x_2 = x_u - d$$

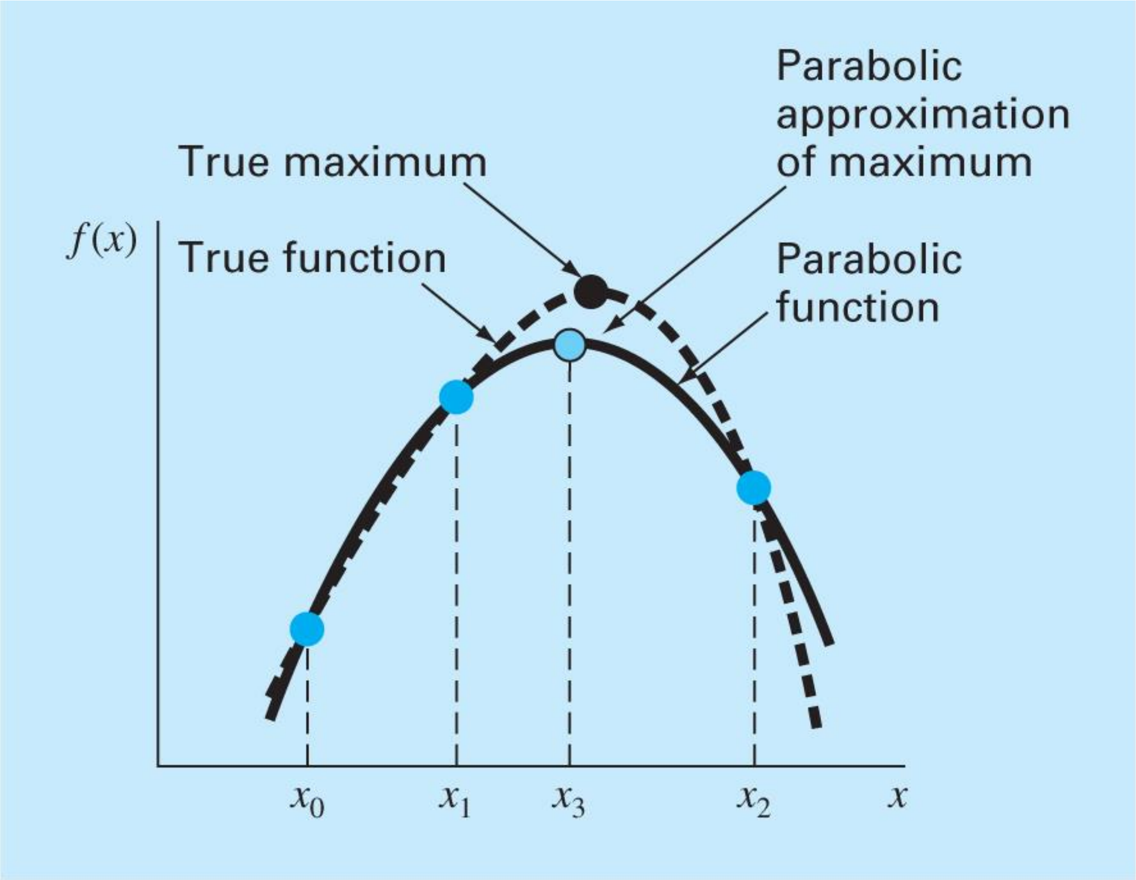
$$d = (\emptyset - 1)(x_u - x_l)$$
$$d = g\,(x_u - x_l)$$

Problema

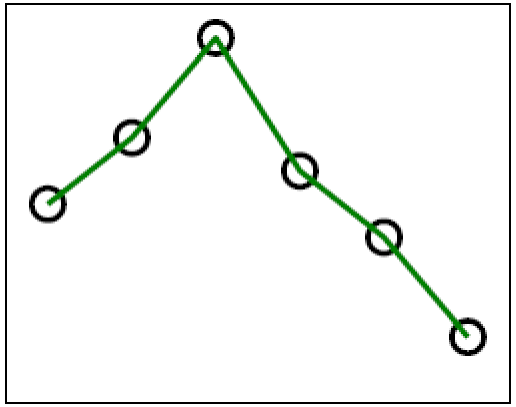
Elevación de un objeto proyectado inicialmente hacia arriba con una velocidad inicial (resistencia lineal)

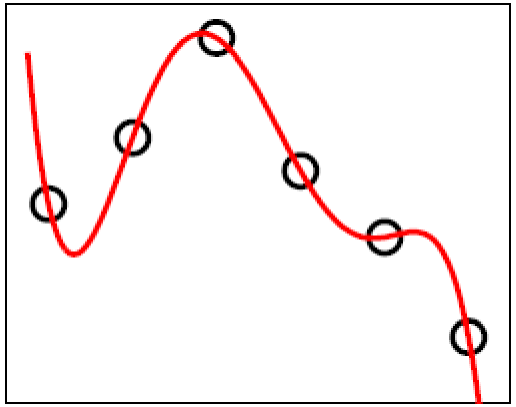


Interpolación parabólica



Interpolación





- Lagrange

$$P(x_k) = y_k, \ \ k = 1, \ldots, n$$

$$P(x) = \sum_k \left(\prod_{j \neq k} \frac{x - x_j}{x_k - x_j} \right) y_k$$

- Newton