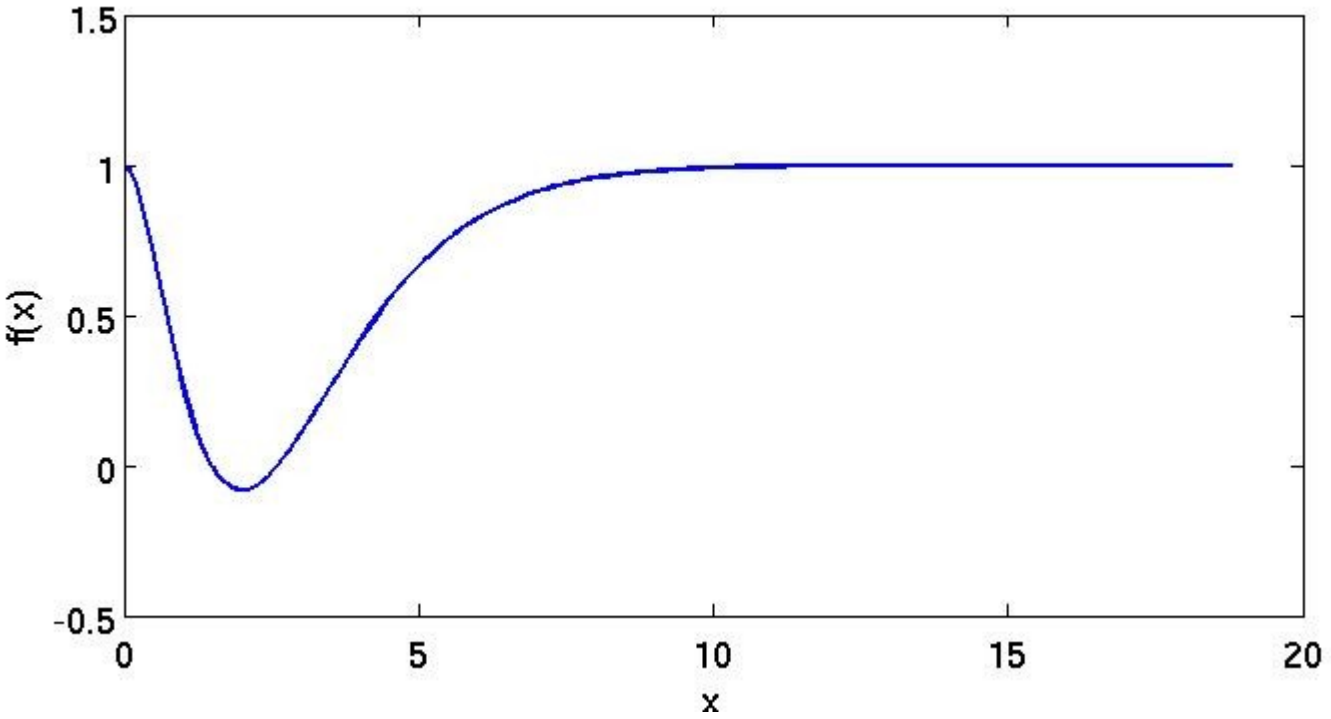
Clase 1:

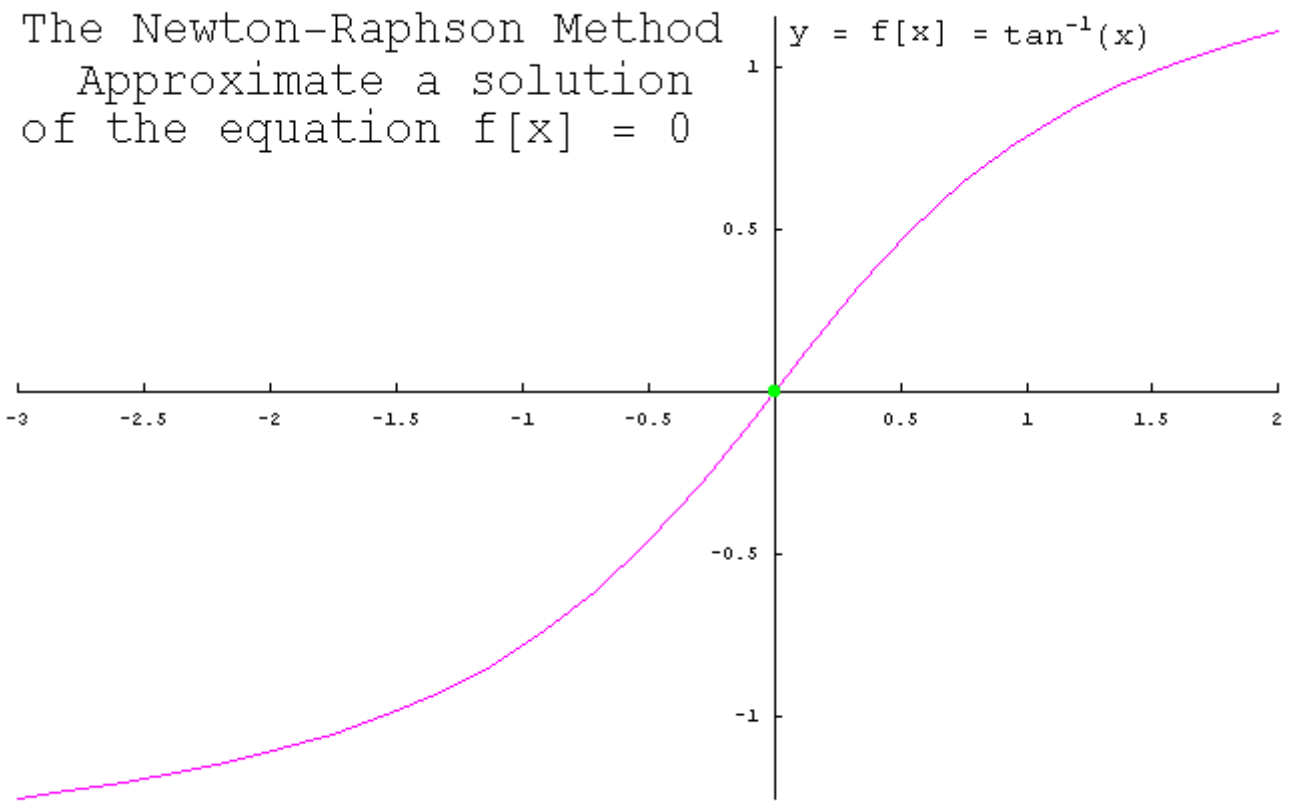


1) Cálculo de r
Suponga que s
Por ejemplo:
f(x)= sen(x)
f(x)=1-x²exp(-x,
Encontrar las raices es el problema
f(x)=0
O sea, encontrar los puntos donde la función corta al eje x.

- En algunos casos el problema puede ser resuelto en forma analítica. Ej:
- $f(x)=ax^2+bx+c=0$
- $f(x)=sen(x)=0$, etc...
- En la mayoria de los casos reales, sin embargo no hay solución analítica.
- Luego, queremos una *solución numérica*

- Método de Newton.
- Consiste en una iteración en la cual se busca la raiz partiendo desde un punto arbitrario cercano a la raiz usando las tangentes a la curva.
- Es necesario conocer la derivada de la función para calcular la tangente.

http://math.fullerton.edu/mathews/a2001/Animations/RootFinding/NewtonMethod/Newtoncc.html

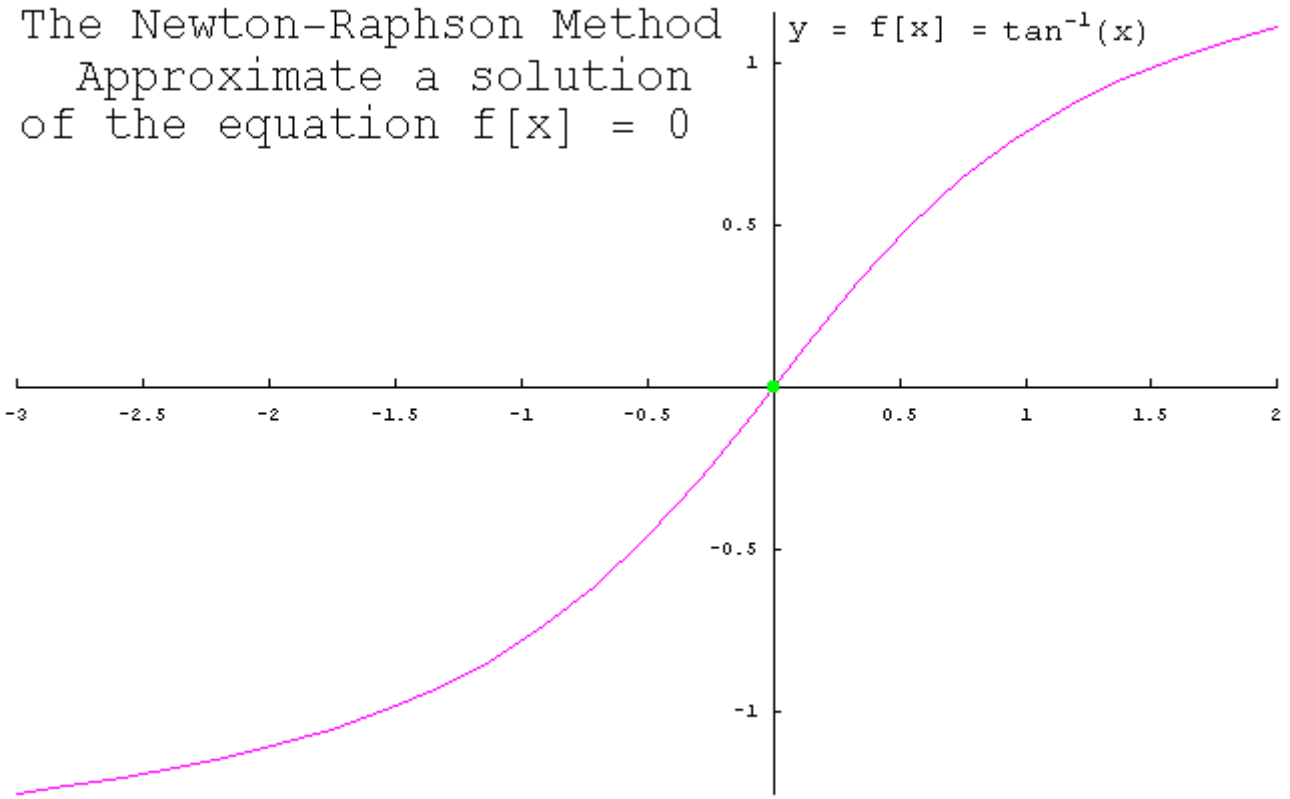


Animación que muestra cómo funciona el método de Newton

- El algoritmo es el siguente. Primero damos una raiz tentativa x_0. La interseccion de la tangente a f(x) en $x=x_0$ estará dada por:
- $x_1 = x_0 - f(x_0)/f'(x_0)$
- En donde $f'(x_0)$ es la derivada de la función en x_0. Luego, encontrado x_1, se vuelve a repetir el proceso. Al cabo de n+1 veces tendremos:
- $X_{n+1} = x_n - f(x_n)/f'(x_n)$

- Para que el algoritmo funcione deben cumplirse ciertas condiciones, por ejemplo que la derivada en todos los puntos evaluados sea distinta de cero.
- A veces tambien la solución nunca converge, por elegir un punto inicial no adecuado.

http://math.fullerton.edu/mathews/a2001/Animations/RootFinding/NewtonMethod/Newtonff.html



Ejemplo de fallo del método de Newton

- Matlab trae incorporado una implementación del método de Newton, la cual puede ser llamada con la función fsolve. Ej: se desea resolver para h la ecuación:
- $$-mgh + k(\sqrt{h^2 + L^2} - L)^2 = 0$$
- Dados los parámetros m=80 kg, L=10 m, k=10 N/m y g=9.8 m/s².

Código:

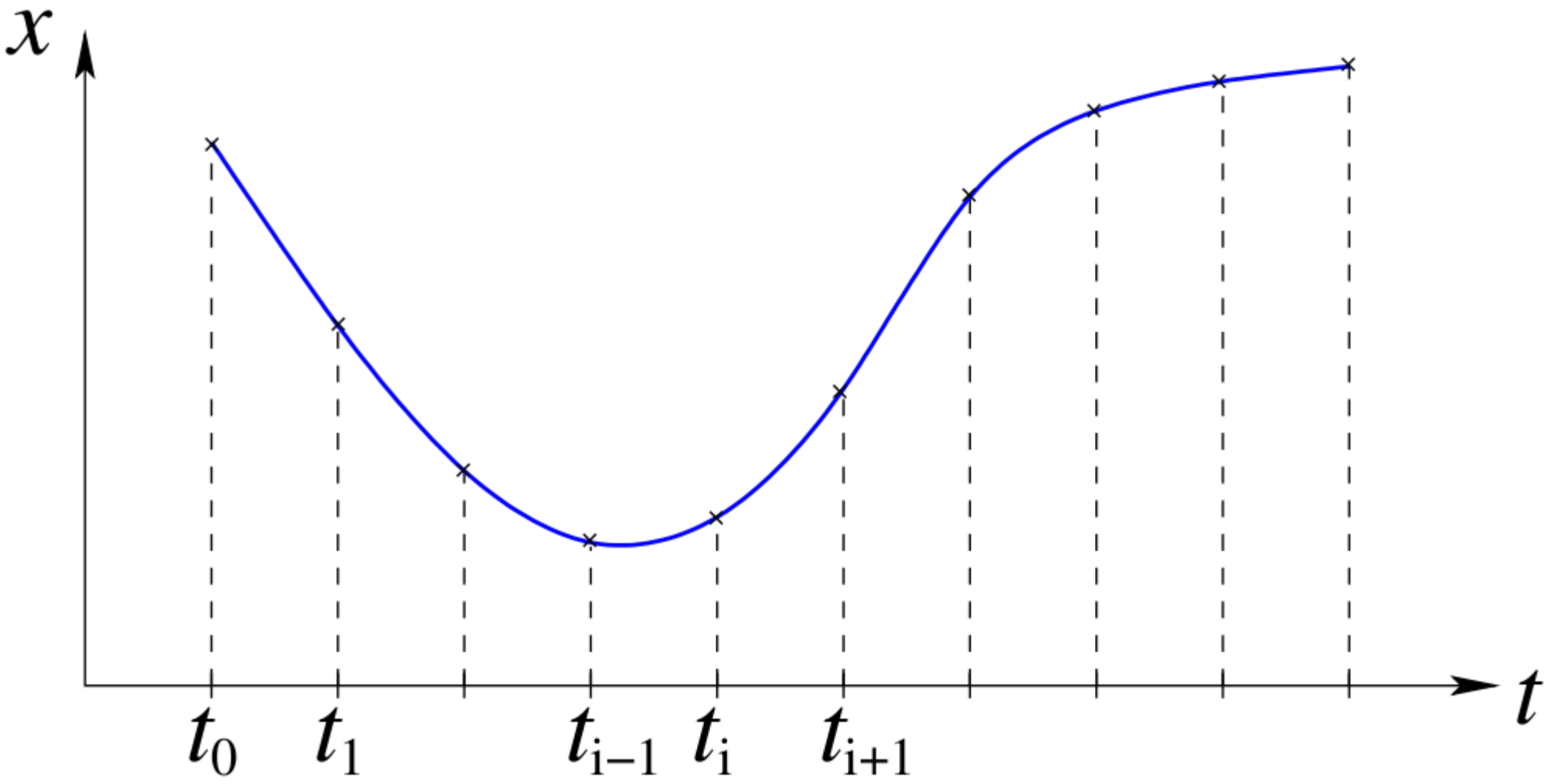
```
>> m = 80  % Se asignan valores sin unidades
>> L=10
>> k=10
>> g=9.8
>> % Se define la ecuacion que se quiere resolver
>> ec=@(h) -m*g*h + k*(sqrt(h^2+L^2)-L)^2
>>
>> % Se resuelve usando la funcion fzero(ecuacion,adivinanza)
>> hsol= fzero(ec,1.0) % Se resuelve para h
```

Para definir una función en matlab se usa la sintaxis:

```
funcion = @(variable) definicion
```

Ecuación de Newton

- La ecuación de Newton es:
- $$m\ddot{x} = F(x, \dot{x})$$
- Esta ecuación en la mayoría de los casos no tiene solución analítica. Luego deseamos obtener una solución numérica. Un método numerico es el de Verlet.
- Primero necesitamos discretizar el problema.



Se divide el intervalo de tiempo usado en partes iguales separadas por un tiempo Δt. Luego:

$$x_i = x(t_i)$$

- Las derivadas pueden ser aproximadas por:
-
$$\dot{x}(t_i) \approx \frac{x(t_i + \Delta t) - x(t_i)}{\Delta t}$$
$$\approx \frac{x(t_{i+1}) - x(t_i)}{\Delta t}$$
$$\approx \frac{x_{i+1} - x_i}{\Delta t}$$
- En donde se calculó la velocidad 'hacia adelante'

- También se puede calcular 'hacia atras':

$$
\begin{aligned}
\dot{x}(t_i) &\approx \frac{x(t_i - \Delta t) - x(t_i)}{-\Delta t} \\
&\approx \frac{x(t_i) - x(t_i - \Delta t)}{\Delta t} \\
&\approx \frac{x(t_i) - x(t_{i-1})}{\Delta t} \\
&\approx \frac{x_i - x_{i-1}}{\Delta t}
\end{aligned}
$$

- La aceleracion luego será:
-
$$\ddot{x}(t) = \frac{\frac{x(t+\Delta t)-x(t)}{\Delta t} - \frac{x(t)-x(t-\Delta t)}{\Delta t}}{\Delta t}$$
-
$$= \frac{x(t+\Delta t) - 2x(t) + x(t-\Delta t)}{\Delta t^2}$$
-
- O lo mismo:

$$\ddot{x}(t_i) = \frac{x_{i+1} - 2x_i + x_{i-1}}{\Delta t^2}$$

- Ahora, la ecuación de Newton será:
- $$m(x_{i+1} - 2x_i + x_{i-1})/\Delta t^2 = F(x_i, \dot{x}_i)$$
- Luego:
- $$x_{i+1} = 2x_i - x_{i-1} + F(x_i, \dot{x}_i)\Delta t^2/m$$
- Si la fuerza depende de la velocidad, debe usarse la velocidad retrógrada.

- Luego, el algoritmo de Verlet consiste en lo siguente:
-
  - Dados: x_0, v_0
  - Se calcula: $x_1 = x_0 + v_0 \Delta t$
  - Se itera desde $i = 1$ hasta el tiempo final:

$$x_{i+1} = 2x_i - x_{i-1} + F\left(x_i, \frac{x_i - x_{i-1}}{\Delta t}\right) \Delta t^2 / m$$

- Ejemplo: Sea F(x)=-kx, o sea, la fuerza elástica. Dadas las condiciones iniciales:
- $m = 1\text{kg},\ k = 0{,}5\text{N/m},\ x_0 = 5{,}0\text{m}$ y $v_0 = 2{,}0\text{m/s}$
- El algoritmo en matlab será:

```
>> m=1.0 % Se dan los valores de m y k
>> k=0.5
>> Tfin = 20.0 % Se resuelve hasta 20 segundos
>> dt = 0.1  % Se da el paso de tiempo Deltat
>> % Se dan las condiciones iniciales
>> x0=5.0
>> v0=2.0
```

```
>> % Se define el arreglo de tiempos
>> t=0:dt:Tfin;
>> % Se define el arreglo de posiciones
>> x=zeros(1,length(t));
>> % Se ponen las condiciones iniciales
>> x(1) = x0;
>> x(2) = x0+v0*dt;
>> % Se itera usando el algoritmo de Verlet
>> for i=2:length(x)-1
>>   x(i+1) = 2*x(i)-x(i-1) - k*x(i)*dt^2/m;
>> end
>> plot(t,x)
```