Mas sobre métodos numéricos

El método de Verlet tiene problemas de estabilidad cuando no se conserva la energia. No es muy usado en la práctica. Una alternativa es el:

Método de Euler: Para ecuaciones de primer orden

$$\dot{x}(t) = f(x, t)$$

$$x(t_0) = x_0$$

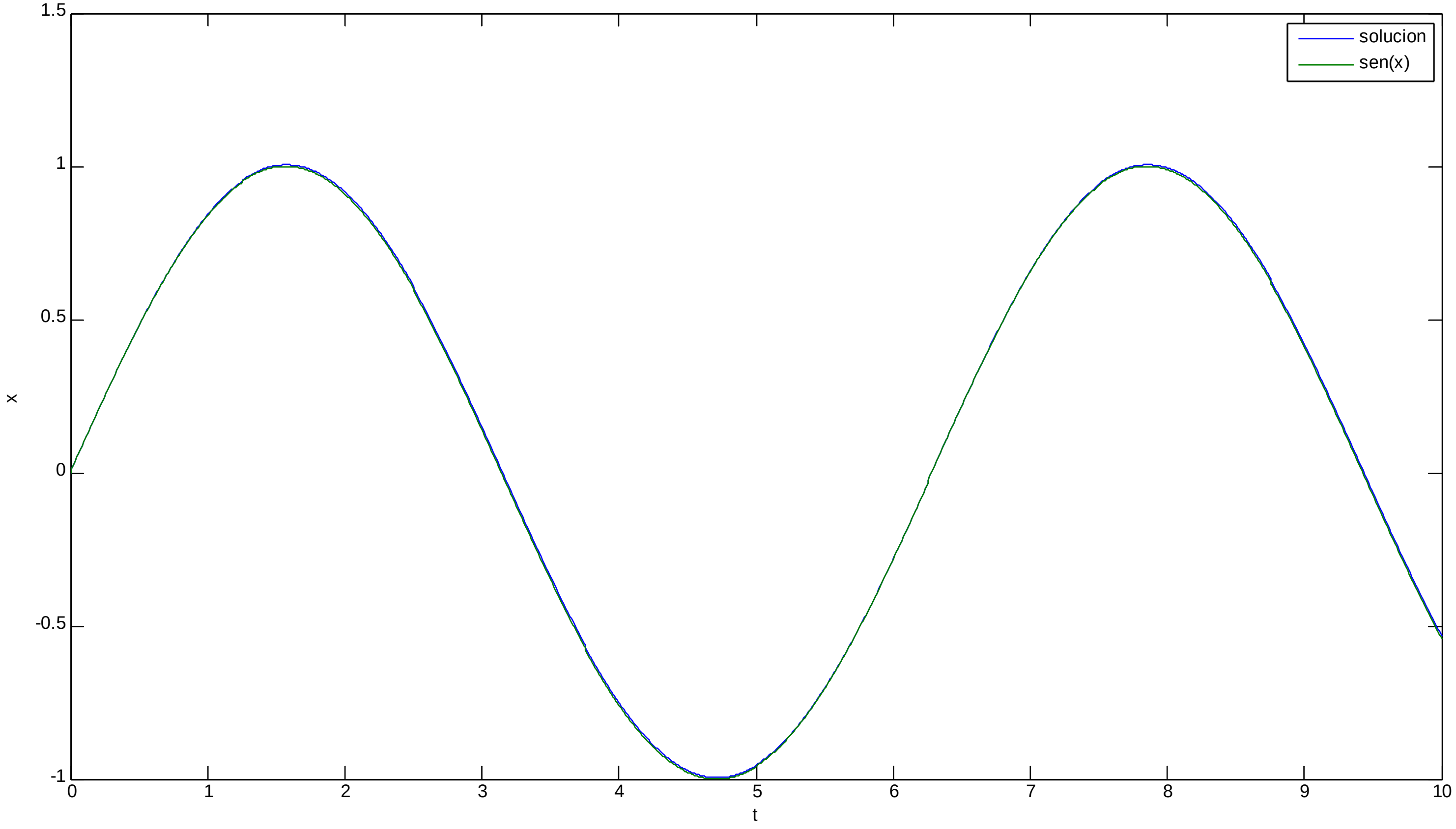
$$x_n = x_{n-1} + \Delta t \cdot \dot{x}_{n-1}$$

Ejemplo : $\dot{x}(t) = \cos(t), \qquad x(0) = 0$

$\Delta t = 0.01$

- Código Matlab

```
>> dt=0.01;
>> t=0:dt:10;
>> x=zeros(1,length(t));
>> for i=2:length(t)
x(i)=x(i-1)+dt*cos(t(i-1));
end
>> plot(t,x,t,sin(t))
```



Solución numérica a la ecuación y solución exacta.

Ecuación de segundo orden

$$\ddot{x} = f(\dot{x}, x, t)$$

Método de Euler

$$y = \dot{x}$$

$$\dot{y} = f(y, x, t)$$

$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \begin{pmatrix} y \\ f(y, x, t) \end{pmatrix}$$

$$x(t_0) = x_0, \qquad y(t_0) = v_0$$

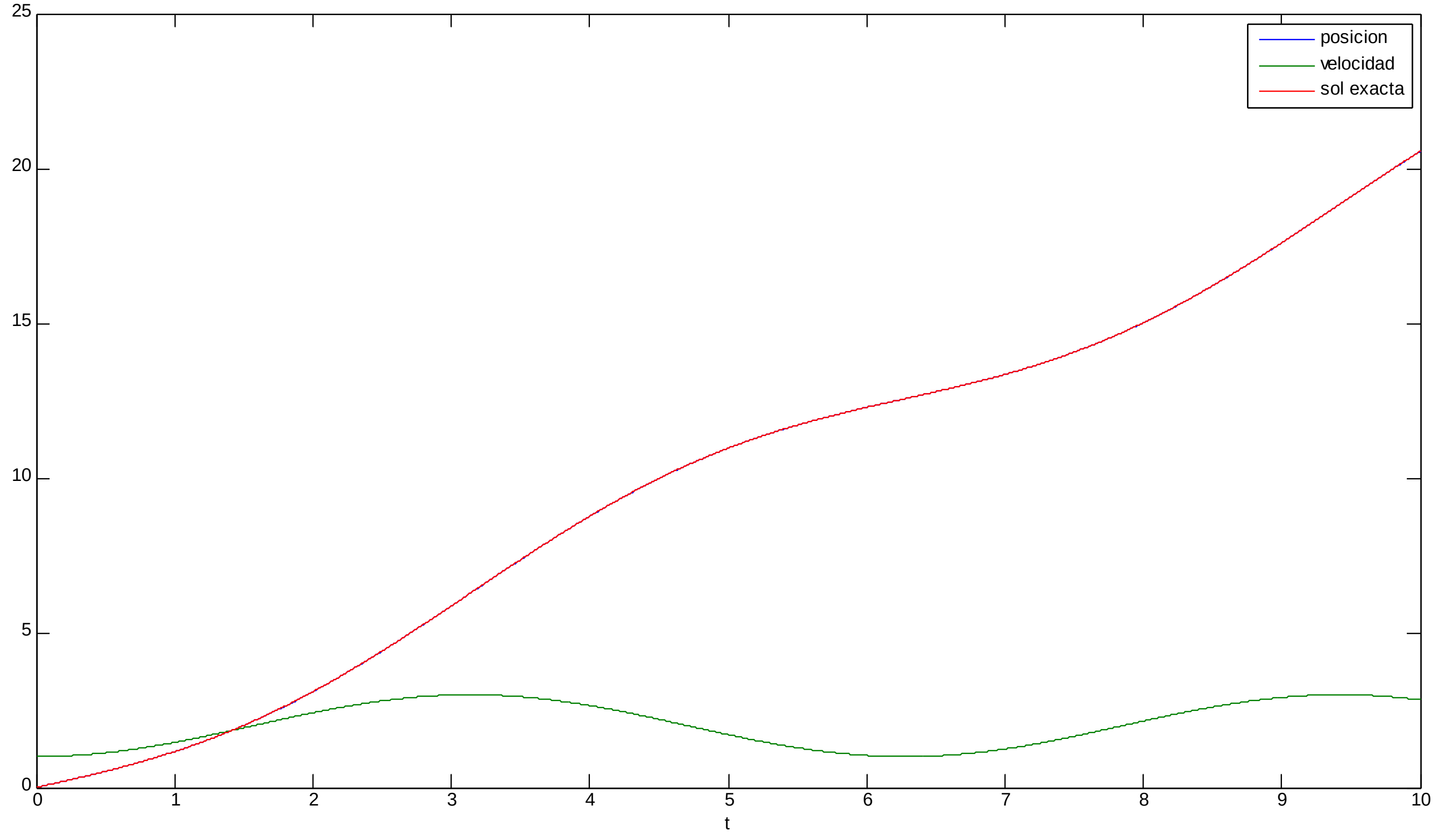
$$\begin{pmatrix} \boldsymbol{x_n} \\ \boldsymbol{y_n} \end{pmatrix} = \begin{pmatrix} \boldsymbol{x_{n-1}} \\ \boldsymbol{y_{n-1}} \end{pmatrix} + \Delta \boldsymbol{t} \cdot \begin{pmatrix} y_{n-1} \\ f(y_{n-1}, x_{n-1}, t_{n-1}) \end{pmatrix}$$

Ejemplo $\quad \boldsymbol{\ddot{x} = sen(t), \quad x(0) = 0, \quad \dot{x}(0) = 1}$

Solución exacta: x(t) = 2t – sen(t)

Código Matlab

```
>> dt=0.01;
>> t=0:dt:10;
>> x=zeros(1,length(t));
>> v=zeros(1,length(t));
>> v(1)=1;
>> for i=2:length(t)
x(i)=x(i-1)+dt*v(i-1);
v(i)=v(i-1)+dt*sin(t(i-1));
end
>> plot(t,x,t,2*t-sin(t),t,v)
```

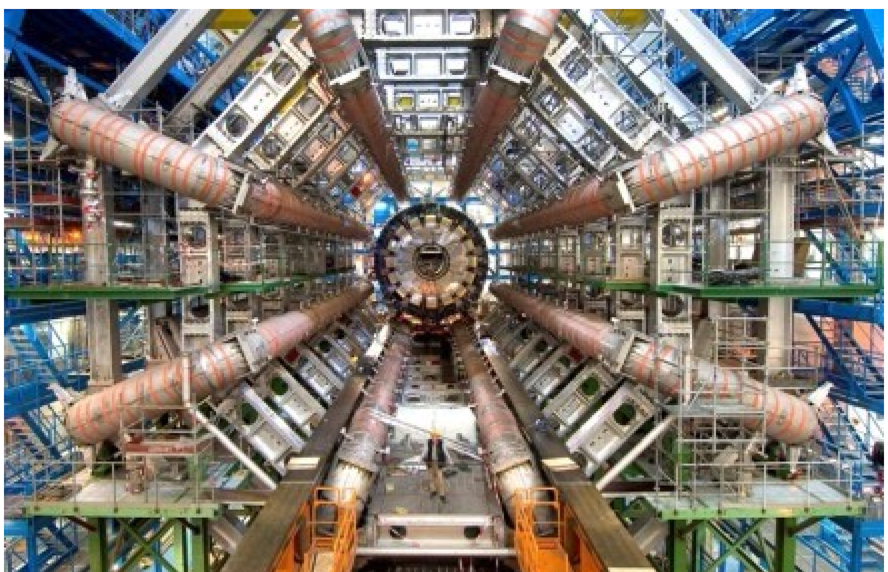


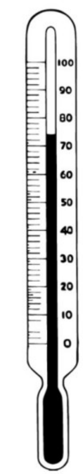
Solución obtenida por método de Euler.
Se obtiene al mismo tiempo la derivada.

- El método de Euler no es muy preciso para tiempos largos. Usualmente se usa un algoritmo más perfeccionado basado en el de Euler llamado Runge-Kutta, que es el que viene implementado en Matlab para resolver EDOs.

Métodos experimentales

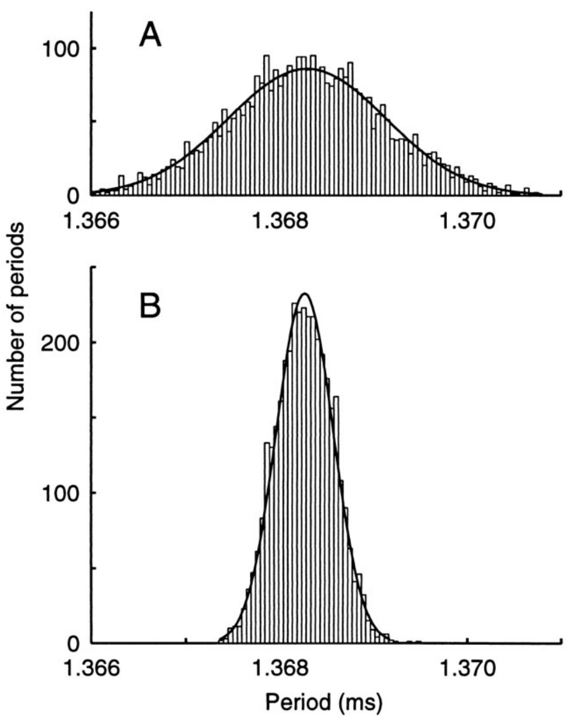
- Queremos medir cantidades: distancias, tiempos, velocidades, aceleraciones, fuerzas, voltajes, corrientes, temperatura, campos, presiones, etc.



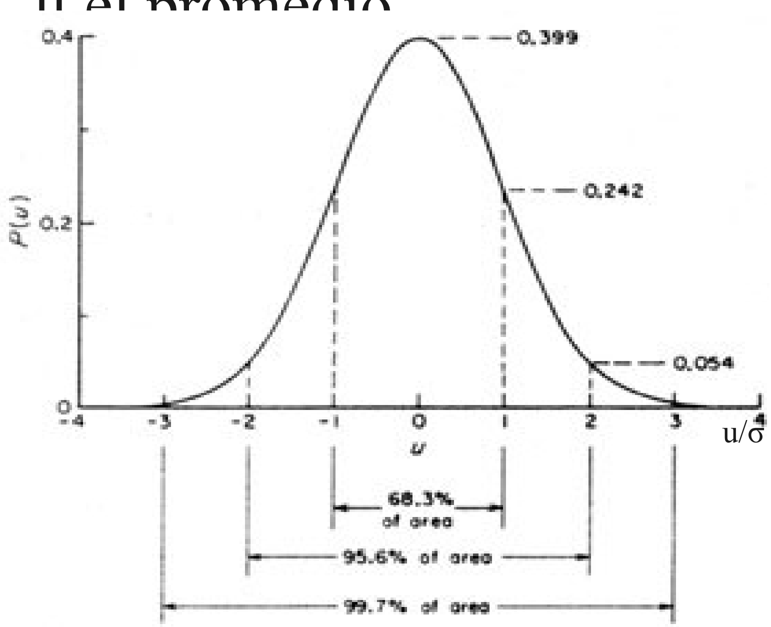
Estadística básica

- Al medir una cantidad repetidas veces, el histograma de frecuencias tiene una distribución normal

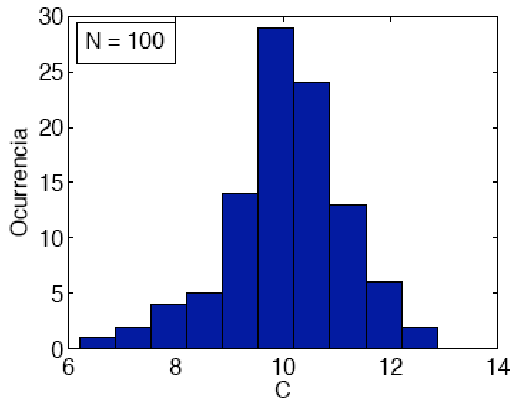


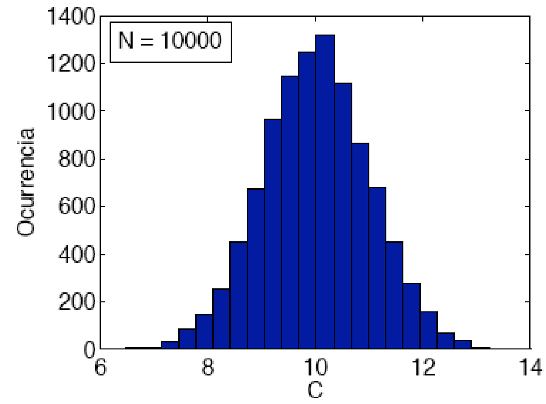
$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right),$$

Donde σ es la desviación estándar
Y μ el promedio



- Para una muestra de datos discretos





Se tiene

$$\mu = \langle C \rangle = \frac{1}{N}\sum_{i=1}^{N} C_i,$$

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(C_i - \langle C \rangle\right)^2}$$

- Cuando el número de datos medidos es pequeño, la curva obtenida no se asemeja mucho a una gausiana, y la desviación estándar calculada con la fórmula anterior subestima la desviación estándar real. En ese caso se debe usar la *desviación estándar de muestra*:

$$\sigma = \sqrt{\frac{1}{N-1}\sum_{i=1}^{M}\left(C_i - \langle C \rangle\right)^2}$$

- En Matlab, si se tiene un vector de datos, ej: x=[1 1 2 1.5 2]
- std(x)=0.5
- Calcula la desviación estándar de muestra, a diferencia del comando:
- std(x,1)=0.4472
- Que calcula la desviación estándar por la definición usual.

Tipos de errores

- Errores sistemátic

Falla en la medición

http://www.space.com/news/orbiter_error_990930.html

Mars climate orbiter

- Errores aleatorios

Debidos a la naturaleza del sistema de medicón, usualmente inevitables.

- Usualmente, si medimos varias veces una cantidad experimental, asociamos su error aleatorio con la desviación estándar de muestra.

Tratamiento matemático de errores

Error absoluto

$$C = \langle C \rangle \pm \Delta C, \qquad \text{con } \Delta C = \sigma$$

Error relativo

$$\epsilon_c = \frac{\Delta C}{\langle C \rangle}.$$

Suma $\quad a = \langle a \rangle \pm \Delta a \text{ y } b = \langle b \rangle \pm \Delta b$

$$c = a + b$$

$$c = \langle c \rangle \pm \Delta c = (\langle a \rangle + \langle b \rangle) \pm \sqrt{(\Delta a)^2 + (\Delta b)^2},$$

- **Resta:** $c = a - b$

$$c = \langle c \rangle \pm \Delta c = (\langle a \rangle - \langle b \rangle) \pm \sqrt{(\Delta a)^2 + (\Delta b)^2},$$

- **Multiplicación:** $c = a \cdot b$

$$c = \langle c \rangle \pm \Delta c = (\langle a \rangle \cdot \langle b \rangle) \pm (\langle a \rangle \cdot \langle b \rangle)\sqrt{\left(\frac{\Delta a}{\langle a \rangle}\right)^2 + \left(\frac{\Delta b}{\langle b \rangle}\right)^2},$$

- **División:** $c = a/b$

$$c = \langle c \rangle \pm \Delta c = \frac{\langle a \rangle}{\langle b \rangle} \pm \frac{\langle a \rangle}{\langle b \rangle}\sqrt{\left(\frac{\Delta a}{\langle a \rangle}\right)^2 + \left(\frac{\Delta b}{\langle b \rangle}\right)^2},$$

- **Función** $f(x)$**:**

$$f(a) = f(\langle a \rangle \pm \Delta a) = f(\langle a \rangle) \pm \left(\frac{df}{dx}\right)_{x=\langle a \rangle} \cdot \Delta a,$$

Distribución de Poisson

- Aplica a eventos discretos que ocurren en un cierto intervalo de tiempo. Por ejemplo: número de llamados telefónicos en un día, número de autos que pasan por una calle, número de decaimientos radiactivos, número de fotones captados por un fotodiodo, etc.
- Si el número de cuentas es N, luego:

$$\sigma \approx \sqrt{N}$$

Cifras significativas

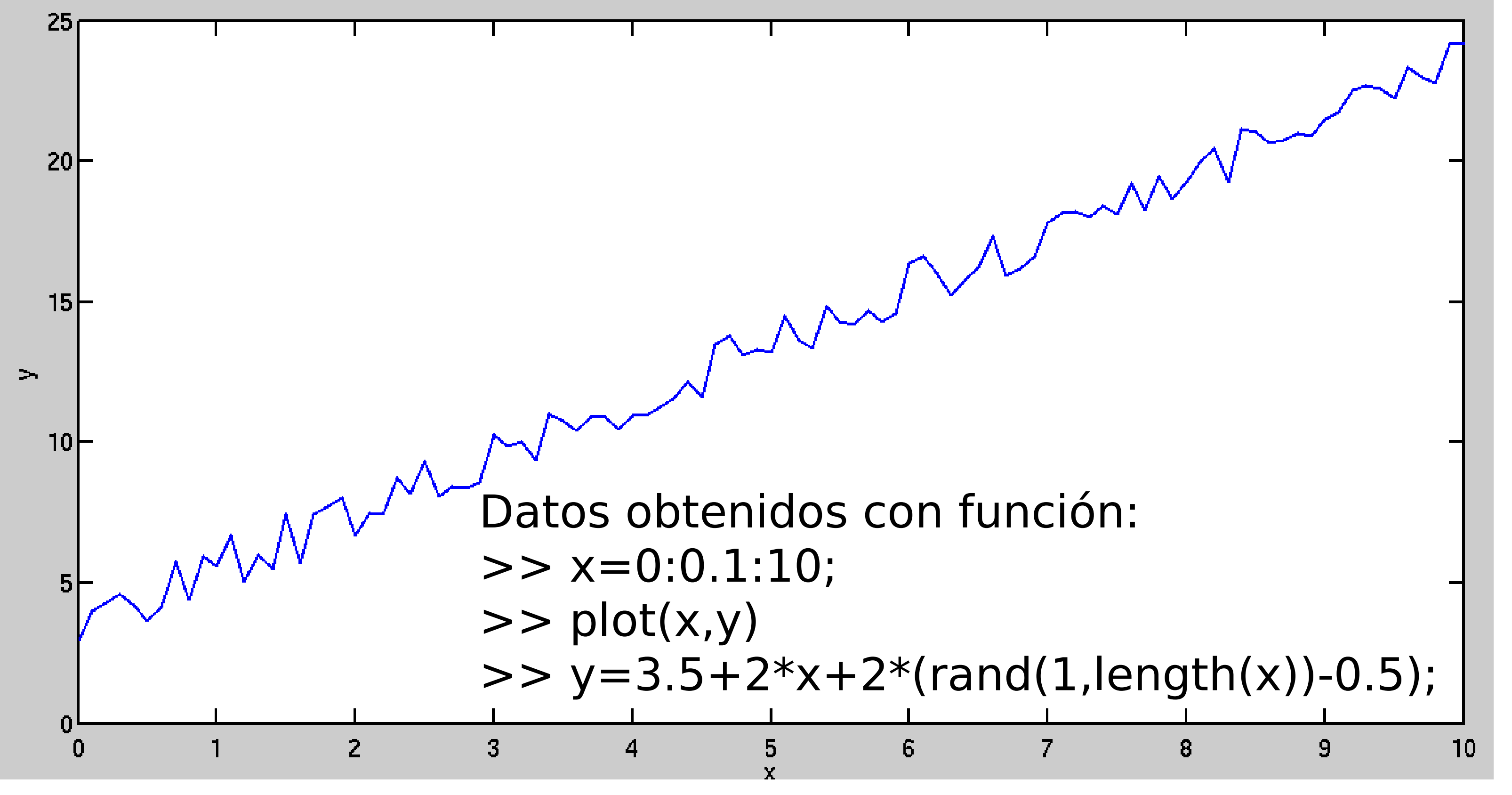
- $T=1{,}345434 \pm 0.003$

sin sentido

Precisión: máximo error esperado de una medición.

Resolución: mínima diferencia de valores medibles.

- Regresión lineal: a veces tenemos datos experimentales en los cuales nos interesa ajustar una dependencia lineal.



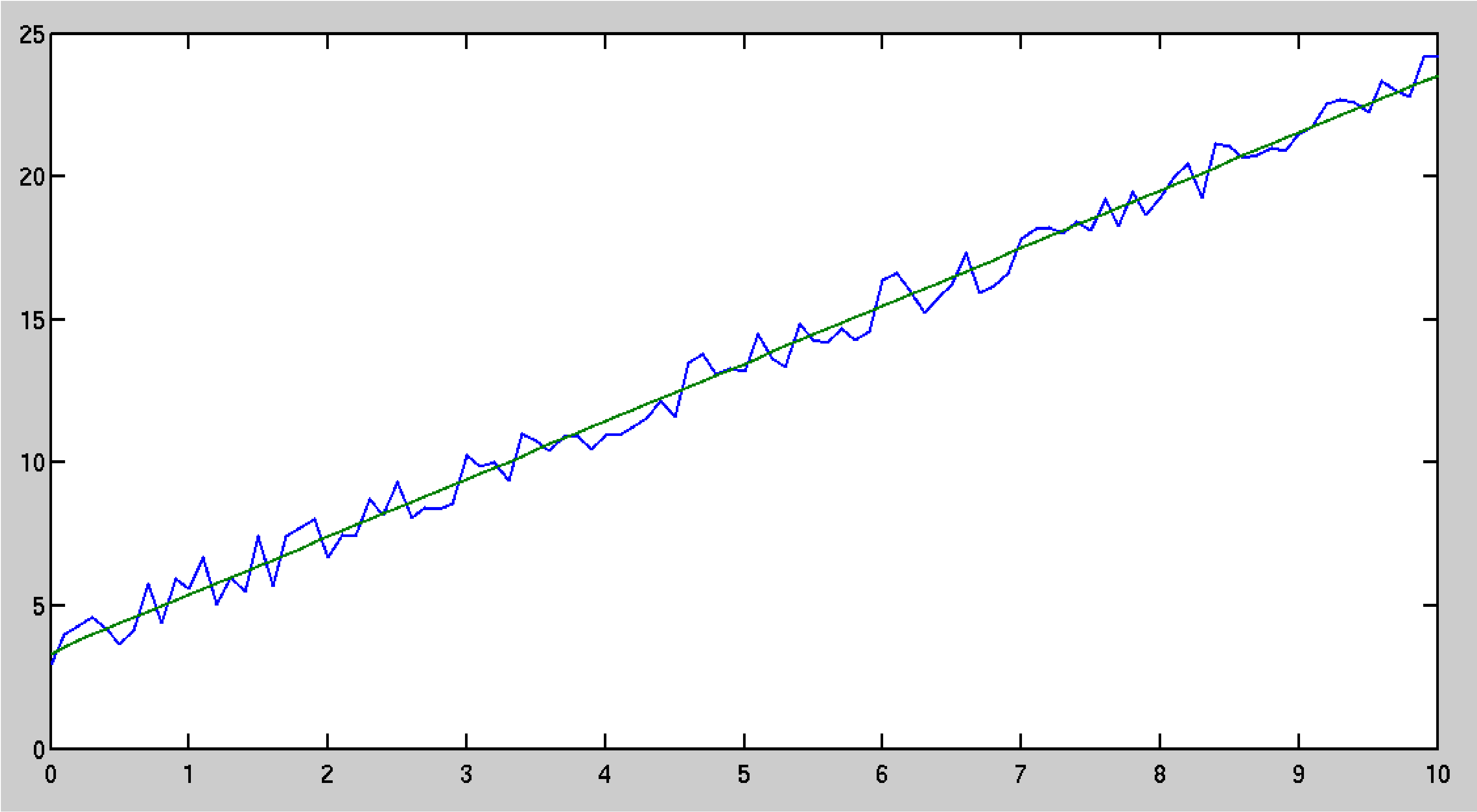
- Si se tiene una conjunto de datos $\{x_1, x_2, x_3, \ldots\}$, $\{y_1, y_2, y_3, \ldots\}$ y se usa el modelo: $y=\alpha+\beta x$
- Entonces se puede demostrar que el mejor ajuste de los parámetros α y β está dado por:

$$\hat{\beta} = \frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sum_{i=1}^{n}(x_i-\bar{x})^2} = \frac{\sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} x_i \sum_{j=1}^{n} y_j/n}{\sum_{i=1}^{n}(x_i^2) - (\sum_{i=1}^{n} x_i)^2/n}$$

$$= \frac{\overline{xy}-\bar{x}\bar{y}}{\overline{x^2}-\bar{x}^2} = \frac{\mathrm{Cov}[x,y]}{\mathrm{Var}[x]} = r_{xy}\frac{s_y}{s_x},$$

$$\hat{\alpha} = \bar{y} - \hat{\beta}\,\bar{x},$$

- En matlab, tenido los datos en los arreglos x e y, se puede hacer usando el comando p=polyfit(x,y,1), el cual entrega el valor de los coeficientes. Se pueden evaluar usando la función polyval(p,x).



Datos experimentales (azul) y ajuste. Valor de parámetros
α=3.3742 y β=2.0181.