Computación Numérica

Tema 1: Introducción al Cálculo Numérico. Teoría de Errores.

Introducción al Cálculo Numérico. Teoría de Errores

- El cálculo numérico. Objetivos.
- Tratamiento numérico de un problema.
- Clasificación de los métodos numéricos.
- Complejidad computacional.
- Tipos de error; errores absoluto y relativo; dígitos significativos.
- Representación de números reales en el ordenador.
- Operaciones en el ordenador: errores de redondeo.
- Sugerencias para minimizar los errores.

El Cálculo Numérico. Objetivos

- Elaborar métodos para hallar, **eficientemente**, soluciones **aproximadas** de problemas expresados matemáticamente.
- Eficiencia: hacer buen uso de unos recursos limitados:
  - Tiempo
  - Espacio
  - Potencia de cálculo requerida
- Soluciones aproximadas, pero con garantía de precisión.

Tratamiento Numérico de un Problema

- Identificación del problema y definición de objetivos
- Descripción matemática
- Análisis numérico
- Programación
- Verificación
- Producción
- Interpretación

Tratamiento Numérico de un Problema

- Identificación del problema
  - Cálculo de la velocidad que alcanza un paracaidista al saltar desde una determinada altura transcurrido un tiempo dado.
  - Simplificaciones y condiciones:
    - La velocidad inicial es nula
    - Despreciamos velocidad horizontal
    - Consideramos paracaidista masa puntual
    - La resistencia al aire es proporcional a la velocidad con una constante 'c'
    - La atracción de la gravedad es constante

  - Datos de entrada: masa (m), coef. de fricción (c), constante gravitatoria (g), tiempo (t)
  - Datos de salida: velocidad en instante t, v(t)

Tratamiento Numérico de un Problema

- Descripción matemática

$F=m*a$

$F=F_g+F_r$

$F_g=m*g$

$F_r=-c*v$

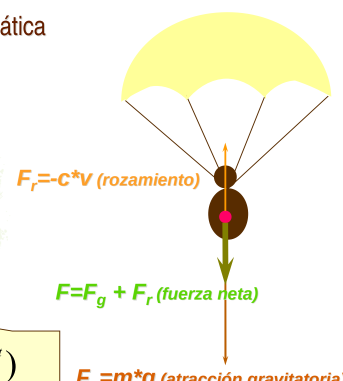
$m*a=m*g-c*v$

$a=dv/dt$

$m*dv/dt=m*g-c*v$

$dv/dt=g-c*v/m$

$$v(t)=\frac{g\cdot m}{c}\left(1-e^{-\frac{c}{m}\cdot t}\right)$$



Tratamiento Numérico de un Problema

- Análisis numérico dv/dt=g-c*v/m

$$\frac{dv}{dt} \approx \frac{\Delta v}{\Delta t} = \frac{v(t_{i+1}) - v(t_i)}{t_{i+1} - t_i}$$

$$\frac{v(t_{i+1}) - v(t_i)}{t_{i+1} - t_i} = g - \frac{c}{m} v(t_i)$$

$$v(t_{i+1}) = v(t_i) + \left(g - \frac{c}{m} v(t_i)\right)(t_{i+1} - t_i)$$

Sólo aparecen operaciones elementales

Tratamiento Numérico de un Problema

- Programación

```
function v = paracaid(t,iter,c,m)

g=980; dt=t/iter; v=0;
disp('tiempo(s)|velocidad(cm/s)');
for i = dt : dt : t
  v = v + ( g - (c/m) * v ) * dt;
  disp( [ i , v ] );
end
```


Tratamiento Numérico de un Problema

- Verificación
- Producción

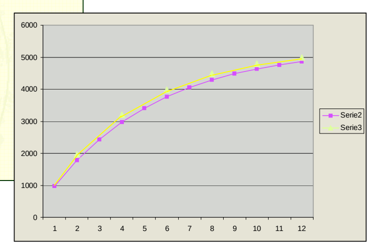
¿Son correctos los resultados?

- *En el instante inicial la velocidad inicial es cero.*
- *La velocidad aumenta con el tiempo*
- *El aumento de velocidad decrece con el tiempo (rozamiento).*

```
» paracaid(12,6,12500,68100);
tiempo(s)|velocidad(cm/s)
 --------|--------------
     2.0 |    1960.0
     4.0 |    3200.5
     6.0 |    3985.6
     8.0 |    4482.4
    10.0 |    4796.9
    12.0 |    4995.9
»
```


Tratamiento Numérico de un Problema

```
» paracaid(12,12,12500,68100)
tiempo(s)|velocidad(cm/s)
 --------|--------------
     1.0 |     980.0
     2.0 |    1780.1
     3.0 |    2433.4
     4.0 |    2966.7
     5.0 |    3402.2
     6.0 |    3757.7
     7.0 |    4047.9
     8.0 |    4284.9
     9.0 |    4478.4
    10.0 |    4636.4
    11.0 |    4765.4
    12.0 |    4870.7
```



Tratamiento Numérico de un Problema

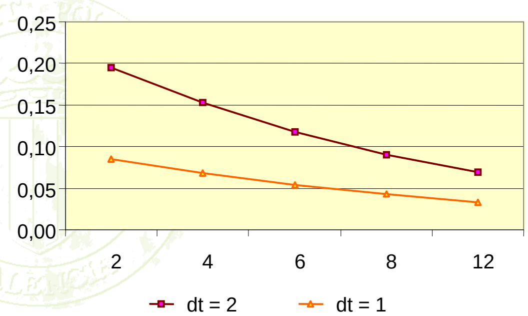
- Interpretación

$$v(t) = \frac{g \cdot m}{c}\left(1 - e^{-\frac{c}{m} \cdot t}\right)$$

```
» paraca(12,6,12500,68100);
tiempo(s)|velocidad(cm/s)[2]|velocidad(cm/s)[1]|    v(t)
 --------|------------------|------------------|---------
     2.0 |      1960.0      |       1780.1     |   1640.5
     4.0 |      3200.5      |       2966.7     |   2776.9
     6.0 |      3985.6      |       3757.7     |   3564.2
     8.0 |      4482.4      |       4284.9     |   4109.5
    10.0 |      4796.9      |       4636.4     |   4487.3
    12.0 |      4995.9      |       4870.7     |   4749.0
     inf |      5339.0      |       5339.0     |   5339.0
»
```


Tratamiento Numérico de un Problema

- Interpretación (error relativo)



Tratamiento Numérico de un Problema

- Identificación del problema y definición de objetivos
- Descripción matemática
- Análisis numérico
- Programación
- Verificación
- Producción
- Interpretación

En cualquier momento puede ser necesario revisar las decisiones tomadas en los pasos anteriores...

Clasificación de los Métodos Numéricos

- **Métodos directos**
  Solución exacta (salvo errores de redondeo) tras un número finito (y conocido a priori) de pasos
  - ejemplos: $ax^2+bx+c=0$, regla de Cramer
- Métodos iterativos
- Métodos basados en la discretización del continuo

Clasificación de los Métodos Numéricos

- Métodos directos ...
- **Métodos iterativos**
  Métodos que van construyendo una sucesión de soluciones aproximadas a partir de una solución inicial, esperando que esa sucesión converja a la verdadera solución.
  - requiere una aproximación inicial
  - puede no converger
  - ejemplo: búsqueda de raíces por bisección
- Métodos basados en la discretización del continuo

Clasificación de los Métodos Numéricos

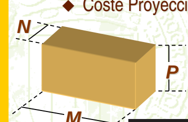
- Métodos directos ...
- Métodos iterativos ...
- **Métodos basados en la discretización del continuo**
  Aproximación de un problema continuo (infinitos elementos) mediante un equivalente discreto (número finito y limitado de elementos)
  - ejemplo:
    - Modelos de elementos finitos

Complejidad Computacional

- Un algoritmo está compuesto por un número bien definido de reglas (instrucciones) que resuelven un problema en un número finito de pasos.
- Existen dos parámetros que evalúan el rendimiento:
  TIEMPO y ESPACIO.
- **Complejidad temporal**:
  - Determina el tiempo necesario para obtener la solución, o el coste de la máquina necesaria para producir las soluciones a tiempo.
- **Complejidad espacial**:
  - Determina la cantidad de memoria necesaria para la ejecución del problema.

Complejidad Computacional: Ejemplo

- Ejemplo práctico: productividad en aplicaciones médicas (TAC, RM, ecografía...)
- Complejidad temporal
  - Coste Proyección * N * M * P
- Complejidad espacial
  - Volumen: N * M * P
  - Proyección: max(N, M, P) ^{2}



Complejidad Temporal

Complejidad temporal; dos formas de calcularla:

- Análisis "a priori":
  - Proporciona una cota del coste temporal
  - Se puede estimar con rapidez (a veces)
  - Independiente de la máquina
- Análisis "a posteriori":
  - Puede dar un resultado exacto
  - Requiere la implementación del algoritmo sobre la máquina destino
  - Dependiente de la máquina

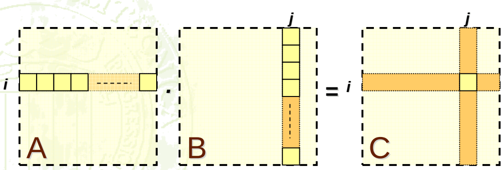
Complejidad Temporal: flop

Complejidad temporal; unidades de medida:

- Una buena forma de expresar el coste computacional de un proceso es indicar el número de operaciones básicas (sumas, productos...) que requiere. Cada una de estas operaciones cuesta un **flop** (contracción de "*floating point operation*").
- Esta medida depende del ***tamaño del problema***, y se suele expresar como una función de este.

Complejidad Temporal: Ejemplo

- Cálculo del coste *a priori* del producto de dos matrices cuadradas de NxN.



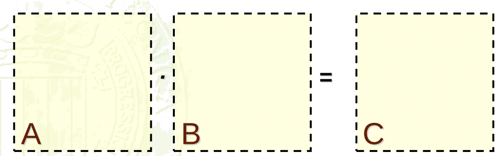
c_{ij} = sum(k=1:N, a_{ik}* b_{kj})

2*N flops por elemento

$2*N^3$ en total

Complejidad Espacial: Ejemplo

- Cálculo del coste *a priori* del producto de dos matrices cuadradas de NxN.



$3*N^2$ en total

Complejidad Espacial: Ejemplo

- Almacenamiento CSR

$$A = \begin{pmatrix} 1 & 0 & 0 & 2 & 3 \\ 0 & 4 & 0 & 0 & 0 \\ 0 & 0 & 5 & 0 & 0 \\ 2 & 0 & 0 & 6 & 7 \\ 3 & 0 & 0 & 7 & 8 \end{pmatrix}$$

- Coste de almacenamiento denso: 5x5x8=200 bytes.
- A=[1, 2, 3, 4, 5, 2, 6, 7, 3, 7, 8] Elementos no nulos
- JA=[1, 4, 5, 2, 3, 1, 4, 5, 1, 4, 5] Columna en que aparece
- IA=[1, 4, 5, 6, 9, 12] Posición del primer elemento no nulo de la fila
- IA(i+1)-IA(i) = Nº elementos no nulos en la fila i
- Coste de almacenam. disperso: 11x8+11+6=105 bytes.

Coste Computacional

- Para el cálculo del coste computacional, se pueden utilizar las siguientes expresiones:

$$\sum_{i=1}^{n} 1 = n \qquad \sum_{j=i}^{n} 1 \approx (n-i)$$

$$\sum_{i=1}^{n} i \approx \frac{n^2}{2} \qquad \sum_{i=1}^{n} i^2 \approx \frac{n^3}{3}$$

Coste Computacional

```
for i=1:n
   x = x + v(i);
end
```

$$\sum_{i=1}^{n} 1 = n$$

```
for i=1:m
   for j=1:n
      x = x + A(i,j);
   end
end
```

$$\sum_{i=1}^{m} \sum_{j=1}^{n} 1 = \sum_{i=1}^{m} n = m \cdot n$$

Coste Computacional

```
for i=1:n
   for j=i:n
      x = x + A(i,j);
   end
end
```

$$\sum_{i=1}^{n} \sum_{j=i}^{n} 1 \approx \sum_{i=1}^{n} (n-i) = n^2 - \sum_{i=1}^{n} i \approx$$

$$\approx n^2 - \frac{n^2}{2} \approx \frac{n^2}{2}$$

Coste Computacional

$$\sum_{i=1}^{n} \sum_{j=i}^{n} \sum_{k=i}^{n} 1 \approx \sum_{i=1}^{n} \sum_{j=i}^{n} (n-i) \approx$$

$$\approx \sum_{i=1}^{n} (n-i)(n-i) = \sum_{i=1}^{n} \left(n^2 - 2ni + i^2\right) =$$

$$= \sum_{i=1}^{n} n^2 - 2n \sum_{i=1}^{n} i + \sum_{i=1}^{n} i^2 \approx$$

$$\approx n^3 - \frac{2n^3}{2} + \frac{n^3}{3} =$$

$$= \frac{n^3}{3}$$

```
for i=1:n
   for j=i:n
      for k=i:n
         x = x + A(i,k);
      end
   end
end
```


Errores Numéricos

- **Los errores causados por problemas numéricos han producido catástrofes que han implicado cuantiosas pérdidas humanas y económicas.**

- **Sonda espacial Mars C. O.**
  - 23 de Septiembre de 1999. La sonda espacial Mars Climate Observer se pierde en la órbita marciana.
  - La causa principal fue el "olvido" de traducir unidades inglesas (millas) en unidades métricas en uno de los módulos del sistema de navegación.
  - La sonda entró en la atmósfera marciana 49 segundos antes de lo previsto con una velocidad muy superior a la prevista, con lo que se desintegró.

Errores Numéricos

- **Fallo en el misil Patriot**
  - Informe del gobierno americano GAO/IMTEC-92-26.
  - 25 de Febrero de 1991, Guerra del Golfo: Un misil Patriot falla al intentar interceptar un misil Scud. Este misil cae en el campamento americano matando 28 soldados.
  - El contador de tiempo monitorizaba la posición cada 1/10 seg (1/10 en Binario es un número periódico):
    - 0.0001100110011001100110011001100110011...
  - Pero el misil utilizaba registros de 24 bits:
    - 0.00011001100110011001100
  - Generaba un error de 0.000000095 segundos por cada décima.
  - Autonomía del misil Patriot: 100 horas. Error total: 0.34 seg.
  - Velocidad del misil: 1676 m/s. Error de posición: ~500 m.

Errores Numéricos

- **Explosión del Ariane 5**

  - 4 de Junio de 1996, un cohete Ariane 5 explotó 40 segundos después del lanzamiento. Las pérdidas económicas fueron de 500M€.
  - El sistema de control de la posición horizontal en el lanzamiento actuaba sobre los motores para evitar que se inclinara.
  - La posición era obtenida como un real de 64 bits, pero se truncaba a un entero con signo [-32,768 , 32,767].
  - Cuando la velocidad horizontal superó esta cantidad, el sistema produjo un error de ejecución.
  - Curiosamente, una vez superados los primeros segundos, este sistema dejaba de ser importante.

Tipos de Errores

- **Errores inherentes**
  - provienen de los datos de entrada
  - imprecisión medidas
  - imposibilidad representar números irracionales
- **Errores por truncamiento**
  - desarrollos en serie...
- **Errores de redondeo**
  - se generan durante el proceso
  - pueden acumularse hasta degradar completamente el resultado

Error Absoluto y Relativo

Independientemente de su origen (inherente, truncamiento o redondeo), la magnitud de un error se puede expresar como:

- **Error absoluto** o si no conocemos x
  $e_x = x - x^*$ $|x - x^*| < c_x$ (es una cota)
- **Error relativo** o si no conocemos x
  $E_x = e_x / x = (x - x^*)/x$ $E_x \approx e_x / x^*$
  si tampoco conocemos e_x
  $|E_x| \approx |e_x| / |x^*| < c_x / |x^*| = C_x$ (es una cota)

Error Absoluto y Relativo

Ejemplo 1:
$x=0{,}5$ y $x^*=0{,}4$
$e_x=0{,}1$
$E_x=0{,}1/0{,}5=0{,}2=20\%=\underline{2*10^{-1}}$

Ejemplo 2:
$x=6583$ y $x^*=6500$
$e_x=83$
$E_x=83/6583=0{,}012=1{,}26\%=\underline{1{,}2*10^{-2}}$

Conclusión:
El error relativo da una mejor estimación del error que el error absoluto. No depende de la unidad de medida.

Error Absoluto y Relativo

Ejemplo 3: $x^2 + 111{,}11\,x + 1{,}2121 = 0$

Real

- $b^2 = 12345{,}432$
- $4ac = 4{,}8484$
- $b^2\text{-}4ac = 12340{,}584$
- $\text{sqrt}(b^2\text{-}4ac) = 111{,}08823$
- $-b+\text{sqrt}(b^2\text{-}4ac) = -0{,}02182$
- $x_1 = -0{,}01091$

Aprox (5 cifras)

- $b^2 = 12345$
- $4ac = 4{,}8484$
- $b^2\text{-}4ac = 12340$
- $\text{sqrt}(b^2\text{-}4ac) = 111{,}09$
- $-b+\text{sqrt}(b^2\text{-}4ac) = -0{,}02000$
- $x_1 = -0{,}01000$

$|e_{x1}| = 0.00091;$ $|E_{x1}| = 0.0834 = 8{,}34\% = 8{,}34*10^{-2}$

Dígitos Significativos

- Se dice que un número **x*** aproxima al número **x** con **k** dígitos significativos si

$$|E_x| = \left|\frac{x - x^*}{x}\right| < 10^{-k}$$

Representación Normalizada

- La **representación normalizada** es la utilizada para representar los números en coma flotante dentro de un computador.
- La representación normalizada utiliza la notación de **mantisa y exponente**, con un determinado número de cifras para la mantisa (**t**).
- Primera cifra de la mantisa representa **la primera cifra significativa** y se sitúa **"delante" del punto decimal**.

$x=0.00234543456 \rightarrow x^*=2.345*10^{-3}$
$x=23454.3456 \rightarrow x^*=2.345*10^{+4}$
(t=4)

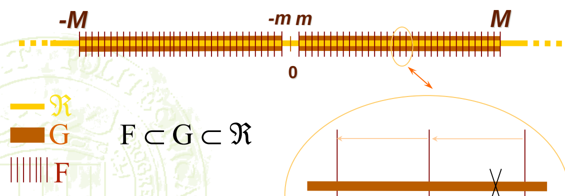
Representación Normalizada

- Llamamos **conjunto F** al conjunto de los números representables en notación normalizada (*números de máquina*).
- Este conjunto es un **subconjunto de** $\Re$ que se define mediante cuatro enteros (**B**, **t**, **L**, **U**) de la siguiente forma:

  $F=\{f\in\Re \;\; f=\pm d_1.d_2,\ldots,d_t\cdot B^e \quad 0\leq d_i<B,\; d_1\neq 0,\; L\leq e\leq U\}$

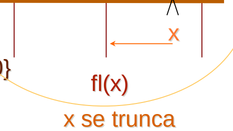
  - **B** es la base
  - **t** es el número de cifras de la mantisa (la **precisión**)
  - **[L,U]** es el intervalo permitido para el exponente
- El número cero no está en F: $0 \notin F$.
- Llamaremos m y M al número más pequeño y más grande representables (en valor absoluto): $m \leq |f| \leq M \;\; \forall f \in F$.
- Números representables: $G = \{x\in\Re \;/\; m \leq |x| \leq M\} \cup \{0\}$

Representación Normalizada



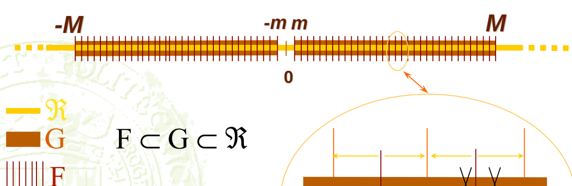
El operador **fl** permite representar un número de **G** dentro del conjunto **F** $\cup$ {0}

fl:G $\rightarrow$ ***F*** $\cup$ ***{0}***



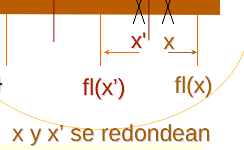
fl(x)= $c \in F \cup \{0\}$ más cercano con $|c| \leq |x|$ (**truncamiento**)
$c \in F \cup \{0\}$ más cercano (**redondeo**)

Representación Normalizada



El operador **fl** permite representar un número de **G** dentro del conjunto **F** $\cup$ {0}

fl:G $\rightarrow$ ***F*** $\cup$ ***{0}***



fl(x)= $c \in F \cup \{0\}$ más cercano con $|c| \leq |x|$ (**truncamiento**)
$c \in F \cup \{0\}$ más cercano (**redondeo**)

Representación Normalizada

$$d_1.d_2 \ldots d_t\, d_{t+1} \ldots d_k \cdot B^e$$

Regla para **truncamiento** a **t** dígitos:

$$d_1.d_2 \ldots d_t \cdot B^e$$

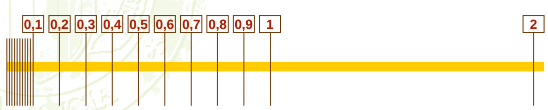
Regla para **redondeo** a **t** dígitos:

- si $d_{t+1} \geq B/2$, $d_1.d_2 \ldots d_t{+}1 \cdot B^e$
- si $d_{t+1} < B/2$, $d_1.d_2 \ldots d_t \cdot B^e$

Representación Normalizada

El espaciado se hace "más compacto" a medida que se reduce el exponente...

Ejemplo (muy forzado) B=10, t=1:



Representación Normalizada: Épsilon

Épsilon de la máquina

- Es la distancia entre el número uno y el siguiente número mayor que uno en formato normalizado.
- Es dependiente de la máquina (depende del formato normalizado utilizado: B, t, L, U).
- Ejemplo: B=2, t=8
  - El número uno sería: $1.0000000 \cdot 2^0$
  - El siguiente número: $1.0000001 \cdot 2^0$
  - Épsilon: $0.0000001 \cdot 2^0 = 2^{-7}$
- Suele usarse como una cota del error relativo al representar un número en un ordenador.

Operaciones Aritméticas en Ordenador

Pasos para realizar una operación elemental cualquiera # (+, -, ×, /) en el ordenador con operandos x e y:

1. Normalizar los operandos (obtener fl(x), fl(y))
2. Operar con los operandos, utilizando registros extendidos (cálculo de fl(x) # fl(y))
3. Normalizar el resultado de la operación: fl(fl(x) # fl(y))

Operaciones Aritméticas en Ordenador

Ej.: Sumar x=62.527, y=3715.26 (B=10, t=4 y redondeo).

- Aritmética real:
  $x + y = 62.527 + 3715.26 = 3777.787$
- En ordenador:
  1. Normalizar los operandos:
     $x^* = fl(62.527) = 6.253 \cdot 10^1$ $y^* = fl(3715.26) = 3.715 \cdot 10^3$
  2. Operar (usando registros extendidos): $x^* + y^* = 3.77753 \cdot 10^3$
  3. Normalizar resultado: $fl(fl(x)+fl(y)) = 3.778 \cdot 10^3$
- error absoluto = (x+y) - fl(fl(x) + fl(y)) = -0.213
- error relativo = (x+y) - fl(fl(x) + fl(y)) / (x+y) = $-5.64e \cdot 10^{-5}$

Cancelación de Dígitos Significativos

Ej.: Restar x=0.54617, y=0.54601 (B=10, t=4 y redondeo).

- Aritmética real:
  $x - y = 0.54617 - 0.54601 = 0.00016$
- En ordenador:
  1. Normalizar los operandos:
     $x^*=fl(0.54617)= 5.462 \cdot 10^{-1}$ $y^*=fl(0.54601)= 5.460 \cdot 10^{-1}$
  2. Operar (usando registros extendidos): $x^* - y^* = 2 \cdot 10^{-4}$
  3. Normalizar resultado: $fl(fl(x)+fl(y)) = 2.000 \cdot 10^{-4}$
- error absoluto = (x-y) - fl(fl(x) - fl(y)) = $-4 \cdot 10^{-5}$
- error relativo = (x-y) - fl(fl(x) - fl(y)) / (x-y) = -0.25 = **-25%**

Gran error relativo debido a **cancelación de dígitos significativos**

Amplificación del Error

- Si el resultado se usa como divisor de números más grandes, el error se amplifica:

$$\frac{1}{x-y} = \frac{1}{0.00016} = 6250$$

$$\frac{1}{fl(fl(x)-fl(y))} = \frac{1}{0.0002} = 5000$$

Sugerencias para Minimizar los Errores

- evitar restas de valores próximos (o sumas con signo)
- evitar divisiones por números muy pequeños
- agrupar sumandos por signo del error de forma que se compensen
- agrupar las sumas por orden de magnitud (ej: sumar por orden de menor a mayor magnitud)

Ejemplo: Sumar los números: 32.45, 0.0043, 0.003, 0.003, utilizando 4 dígitos decimales (t=4) (32.4603)

de izquierda a derecha: →	de derecha a izquierda: ←
32.45 + 0.0043 + 0.003 + 0.003 = 32.45	32.45 + 0.0043 + 0.003 + 0.003 = 32.46

Autocontrol

1.- Dado el fragmento de código siguiente, ¿cuál es el coste computacional?

```
para i:=1 hasta n
   para j:=i hasta n
         para k:=j hasta n
              A(i,j)=A(i,j) + alpha × A(i,k)
         finpara
   finpara
finpara.
```

a) El número de sumas es del orden de $n^3/6$ y el orden de productos es de n^2.

b) El número de productos es de orden $n^3/6$ y el orden de sumas es de n^2.

c) El número de Flops es del orden de $n^3/3$.

d) El número de Flops es del orden de n^2.

Autocontrol

2.- Supongamos que tenemos la siguiente función $f(x)=1-\cos x$. ¿Cuál de las siguientes afirmaciones es correcta ?

a) Si queremos evaluar la función en un punto cercano a cero, no habrá problemas de cancelación de términos significativos.

b) Debemos utilizar la función $g(x)=\frac{sen^2 x}{1-\cos x}$, que es una función equivalente a $f(x)$ y no presenta problemas en puntos cercanos a cero.

c) Si queremos evaluar la función en puntos cercanos a $(2k+1)\pi$, donde k es un número entero deberemos utilizar la función $g(x)=\frac{sen^2 x}{1+\cos x}$.

d) Si queremos evaluar la función $f(x)$ en puntos cercanos a $2k\pi$, donde k es un número entero, podemos utilizar la función $g(x)=\frac{sen^2 x}{1+\cos x}$, que es una función equivalente a $f(x)$.

Autocontrol

3.- Los errores de redondeo, son debidos:

a) A la falta de precisión en los aparatos de medida.

b) A que la evaluación de muchas funciones requieren la suma de infinitos términos, por lo que hay que truncar el sumatorio.

c) A que los ordenadores trabajan en aritmética finita.

d) Ninguna de las anteriores.

Autocontrol

4.- El error relativo:

a) Es un error que se debe tener en cuenta, pero interesa utilizar el error absoluto.

b) Da una mejor estimación del error producido que el error absoluto.

c) Se define como el error absoluto partido por el error inherente.

d) No da información del número de dígitos exactos de la solución.

Autocontrol

5.- Los métodos directos de resolución de sistemas lineales...

a) Permiten aproximar la solución tanto como queramos.

b) No resuelven un sistema en un número finito de pasos.

c) Siempre tienen un error grande.

d) Ninguna de las anteriores.

Autocontrol

6.- Un error inherente es:

a) Aquel que se produce al realizar operaciones en coma flotante.

b) Aquel que se produce en los datos de entrada.

c) Aquel que se produce al truncar una función que necesita de infinitos términos para calcularla.

d) Ninguna de las anteriores.

Autocontrol

7.- El error de cancelación de cifras significativas se produce:

a) Cuando dividimos un número por otro muy pequeño.

b) Cuando restamos dos números de la misma magnitud y del mismo signo.

c) Cuando realizamos el producto de dos números muy grandes.

d) a y c son correctas.

Autocontrol

8.- Si queremos calcular 1-*sen x*, para valores de *x* cercanos a $\pi/2$, ¿cuál de las siguientes afirmaciones es cierta?

a) No se puede calcular pues 1-*sen x*, no es una función real.

b) Es mejor calcular $\frac{cos^2 x}{1+sen\, x}$ para evitar la cancelación de dígitos significativos.

c) Se puede calcular siempre pues no se produce cancelación de dígitos significativos.

d) Es mejor calcular 1+*sen x* y restarle $\pi/2$ al resultado.

Autocontrol

9.- ¿Cuál de las siguientes afirmaciones es la correcta?

a) Los métodos iterativos obtienen soluciones exactas tras un número finito de operaciones elementales.

b) Los métodos basados en la discretización del continuo son métodos directos inexactos.

c) Los métodos directos van obteniendo una sucesión de soluciones aproximadas.

d) Los métodos iterativos, si convergen, obtienen soluciones aproximadas.

Autocontrol

10.- Dados $x = 0.4523\ 10^4$, e $y = 0.2115\ 10^{-2}$ en un ordenador decimal con una mantisa normalizada de cuatro dígitos, al calcular $x + y$ en coma flotante:

a) No se produce ningún error.

b) El error absoluto cometido es $0.4523\ 10^4$.

c) El error absoluto cometido es $0.2115\ 10^{-2}$.

d) El error relativo no se puede calcular pues no conocemos el verdadero valor.