- Las soluciones a los problemas en color rojo se deberán entregar individualmente en la Hoja de Respuestas (doc o pdf con código, resultados y gráficas pedidas) en la tarea correspondiente.
- En algunos ejercicios se incluye en azul una posible solución.

## Representación en coma flotante. Error relativo

**Ejercicio 0**: Se recomienda hacer los ejercicios resueltos en diapositivas del Tema 1.

**Ejercicio 1.** Se quiere estudiar el comportamiento de las siguientes expresiones (matemáticamente equivalentes) para valores de x pequeños:

$$f(x) = \sqrt{1+x^2} - 1, \quad g(x) = \frac{x^2}{\sqrt{1+x^2}+1},$$

¿Cuál de estas expresiones proporciona los mejores resultados numéricos? ¿Para qué valores de x los resultados de ambas funciones son distintos?. Para ello:

- Crear el vector x=10^-n, con n=[0 1 2 … 10].
- Evaluar la función f en esos puntos.
- Evaluar la función g en esos puntos.
- Evaluar el error absoluto y a partir de él, el error relativo.
- Representar el error relativo (*rojo) respecto de x, en ejes logarítmicos (loglog).
- Calcular el número de cifras decimales significativas y representar gráficamente repecto de x, en una escala semilogarítmica (semilogx).

(Solución en diapositivas Tema 1)

**Ejercicio 2.** Parámetros doble precisión. Vamos a calcular el “overflow”, “underflow” y eps de la máquina (Matlab los denota por inf, 0 y eps, respectivamente): Para ello vamos a calcular los números enteros k_over, k_under y k que verifican:

$$2^{k\_over} < \inf, \quad 2^{k\_over+1} = \inf$$
$$2^{-k\_under} > 0, \quad 2^{-(k\_under+1)} = 0$$
$$1+2^{-k} > 1, \quad 1+2^{-(k+1)} = 1$$

- A partir de x=1, escribir el código para calcular k_over el nº de veces que podemos multiplicar x por 2 antes de tener “overflow” ¿Cuál es el valor de k_over? ¿Qué significado tiene en la representación en coma flotante utilizada?
- A partir de x=1, escribir el código para calcular k_under el nº de veces que podemos dividir x por 2 antes de tener un 0 (“underflow”) ¿Cuál es el valor de k_under? ¿Coincide con el valor de k_over? ¿Por qué?
- Escribir código que calcule el menor entero k tal que $1+2^{-k} = 1$ ¿Qué significado tiene en la representación de coma flotante utilizada? ¿Cómo se denomina $2^{-k}$ ?

Sabiendo que la máquina utiliza representación binaria en coma flotante, deducir a partir de los enteros calculados anteriormente: ¿Cuántos bits utiliza esta representación para el exponente y la mantisa?

(Solución en diapositivas Tema 1).

**Ejercicio 3.** Matlab dispone de la función log1p(x) (denotaremos por vex) que teóricamente calcula log(1+x) (denotaremos por vaprox). Se van a comparar los resultados de ambas funciones para valores de x próximos a 0.

a) Evaluar ambas funciones para valores de x=10^-1, 10^-1,…,10^-16.
b) Calcular el error relativo entre ambas expresiones y el nº de cifras decimales significativas.
c) Dibujar en una gráfica los valores de x en el eje horizontal y el nº de cifras significativas en el vertical, utilizando una escala adecuada.
d) Si a=10^-8, evaluar $\sum_{n=1}^{100} \frac{(-1)^{n+1} a}{n}$ y comparar el valor obtenido con log1p(a) y con log(1+a). Hacerlo volcando los resultados con la función fprintf usando notación científica con 10 decimales (%10e).

(Examen Julio 2016. Soluciones en Moodle**)**

**Ejercicio 4.** Consideramos las dos siguientes expresiones que proporcionan resultados similares para valores de h cercanos a cero:

$$\cosh(1) \cong \frac{\cosh(1+h) - 2\cosh(1) + \cosh(1-h)}{h^2}$$

Se pide:

- Construir un vector n con valores 1, 2, 3,…, 8.
- A partir de n construir un vector h con valores 0.1, 0.01, 0.001,….., 0.00000001.
- Construir un vector con los resultados de evaluar la expresión de la derecha para el vector h.
- Calcular el error relativo de los resultados de la expresión de la derecha con respecto a la de la izquierda.
- Dibujar, en la escala adecuada, la relación entre n y el error relativo (los valores de n deben ir en el eje horizontal y los del error relativo en el vertical).
- Calcular el número de cifras decimales correctas entre ambas expresiones.
- Dibujar, en la escala adecuada, la relación entre h y las cifras (los valores de h deben ir en el eje horizontal y los de las cifras en el vertical).
- Para que valor de h se consiguen 8 cifras correctas.

(Examen Julio 2017. Soluciones en Moodle**)**

**Ejercicio 5**. a) A partir de la serie $\frac{1}{e} = 1 - \frac{1}{1!} + \frac{1}{2!} - \frac{1}{3!} + \frac{1}{4!} - \;....\pm \frac{1}{n!} \mathrm{m} ....$

calcular el valor aproximado de 1/e (exp(-1) usando 11 sumandos de la serie, calcular el error relativo y el nº de cifras decimales significativas.

b) Crear la función **[Valor,Erel,Ncif]=inversoe(n)** que recibe como argumento de entrada un número entero n y devuelve el valor que se obtiene al sumar los n+1 primeros términos de la serie anterior, el error relativo y el nº de cifras decimales significativas :

$$\frac{1}{e} = 1 - \frac{1}{1!} + \frac{1}{2!} - \frac{1}{3!} + \frac{1}{4!} - \;....\pm \frac{1}{n!} \mathrm{m} ....$$

c) Aplicar la función creada para los valores n=5, 10, 15, ,… hasta que el error relativo sea <=1e-15. Indicar el número de cifras significativas de precisión que se obtienen para cada n.

**Ejercicio 6.** Se considera el desarrollo en serie

$$\log(1+x) \approx \sum_{k=1}^{n} \frac{(-1)^{(k+1)} x^k}{k} = x - \frac{x^2}{2} + \frac{x^3}{3} - \ldots \qquad (-1<x<1)$$

1. El objetivo es calcular valores aproximados de **log(1.5)** considerando distinto números de sumandos de la serie, analizar el error que se comete en cada caso, y su visualización. Paras ello, si **x=0.5**:

- Calcular el valor exacto v_ex=log(1.5) (comando log de Matlab). Construir un vector (que llamaremos términos) que contenga los 30 primeros términos de la serie (forma general $\frac{(-1)^{(k+1)} x^k}{k}$**).**
- Crear un bucle para construir un vector v_aprox, cuya componente v_aprox(n) contenga el valor aproximado obtenido al sumar los n primeros términos de la serie con n=1, 2, 3,.., 30.
- Construir un vector e_rel que contenga los errores relativos de estimar log(1+0.5) con las aproximaciones del apartado anterior (que dependían del nº de sumandos empleados en cada aproximación). Construir un vector n_dec_sig que contenga el número de cifras decimales significativas que se obtienen con las aproximaciones del apartado anterior. ¿Cuántas cifras decimales significativas se obtienen con 30 sumandos?
- Representar gráficamente los vectores e_rel y n_dec_sig creados en el apartado anterior respecto del nº de sumandos empleado en la aproximación, utilizando en cada caso una escala adecuada que permita visualizar correctamente los resultados. Comentar dichas gráficas ¿Cuántos sumandos (aproximadamente) son necesarios para calcular log(1+0.5) con 5 cifras decimales significativas?

2. El objetivo es aproximar la función **log(1+x)** por el valor dado por la serie considerando 20 sumandos $fap(x) = \sum_{k=1}^{20} \frac{(-1)^{(k+1)} x^k}{k}$ para cada punto x del vector **xx=0:0.1:0.9**. Para ello:

- Construir un vector fap que lleve el valor aproximado fap(x) para cada punto x del vector xx. Pintar la función exacta (f en rojo) y la función aproximada (fap en verde) respecto de xx.
- Representar gráficamente (usando una escala adecuada) el error relativo que se comete al aproximar la función f(x) por fapr(x) para los x del vector xx ¿Para qué valores de x, próximos a 0 o próximos a 1, la fórmula aproximada anterior proporciona mayor precisión?

**Ejercicio 7.** Se considera la siguiente expresión matemática del seno hiperbólico $\sinh(x) = \frac{e^x - e^{-x}}{2}$

***Cálculo de valores exactos y aproximados***. Vamos a utilizar el comando sinh de Matlab y la expresión anterior para calcular la función seno hiperbólico para valores cada vez menores de x. Estudiaremos el comportamiento de dicha expresión y el error relativo para valores de x pequeños.

- Partimos de un vector x de valores cada vez mas pequeños: 1, 0.1, 0.01, 0.001,… etc. Para ello generar un vector n desde 0 hasta 20 y calcular x=10.^-n.
- Evaluar el valor 'exacto' de la función seno hiperbólico de los valores del vector x usando la función *sinh(x)* de Matlab. Llamarlo *v_exact*.

- Evaluar la expresión $\frac{e^x - e^{-x}}{2}$ en el vector x. Llamarlo *v_aprox*.
- Pintar en el mismo objeto gráfico sinh(x) (descriptor 'bo') y $\frac{e^x - e^{-x}}{2}$ ('r*') respecto de x en escala adecuada (loglog) ¿Qué se observa?

***Estudio el error***:

- Calcular el error absoluto (denominar eabs=abs(v_exact-v_aprox)) al vector), el error relativo (denominamos erel al vector obtenido) y el nº de cifras decimales significativas (vector ncif) a partir de v_exact y v_aprox. Comprobad (comando *whos)* que todas estas variables son vectores de tamaño 1x21.
- En una ventana gráfica dividida en dos subventanas (subplot) incluir las gráficas en escalas adecuadas del: error relativo <u>en función de x</u> (*loglog*) con descriptor 'bo-' (azul con los puntos como círculos (o) y las líneas que los unen (-)) y número de cifras significativas <u>en función de x</u> (semilogx) con descriptor 'r*'. Comentar las gráficas.
  Nota: Fijarse en la diferencia de las gráficas plot(ncif)=plot(1:21, ncif) (en el eje de abcisas están los índices del vector), plot(n,ncif) (en el eje abcisas está el vector n, que indica -potencias de los elementos de x) y semilogx(x,ncif) (en el eje de abcisas está el vector x con valores decrecientes que llegan a ser muy pequeños).
- Imprimir (comando fprintf) los resultados en una tabla que incluya en la 1ª columna el valor de x (%2d), en la 2ª columna el correspondiente valor del error relativo (%0.2e) y en la 3ª columna el correspondiente nº de cifras significativas (%2d). Para ello, crear primero una matriz a cuyas filas sean n, x,erel y ncif y usar fprintf('n=%2d error rel: %0.2e nº cifras: %2d \n', a).
- (Opcional) Se puede probar que una cota del error relativo de la expresión es $\frac{eps}{\sinh(x)}$ donde eps es el épsilon de 1 (comando *eps*). Evaluar esta cota en la variable *cota*. Repetir la gráfica del error relativo, añadiendo la gráfica de la cota, usando el descriptor 'ro-' (rojo) ¿Se verifica que la cota es mayor que el error relativo?