UNIVERSIDAD DE MURCIA

Departamento de Matemáticas
Cálculo Numérico en una Variable
Curso 2022–23

Prof. S. Sánchez-Pedreño

Práctica 1

26-27 de septiembre de 2022

Es importante haber leído con atención (y practicado) el documento `CN1V_OctaveIntroduccion.pdf` incluido en los recursos del Aula Virtual antes de abordar la realización de esta práctica. Recuerda que el código de las prácticas debes guardarlo en la carpeta `/CN1V_2022_23/CodigoPracticas`. Dentro de ésta debes colocar la carpeta `/Biblioteca` y crear la carpeta `Practica1` (dentro de `/CodigoPracticas`, al mismo nivel que `/Biblioteca`). Cada uno de los tres ejercicios de esta práctica dará lugar a un archivo: `Ejercicio1.m`, etc., todos ellos debes colocarlos en la carpeta `Practica1`.

1. En la carpeta `/Biblioteca` se ha incluido la función `redondeo.m`. Dado un valor numérico `x`, de cualquier tipo (incluso matricial), y un entero `numdig`, `redondeo(x,numdig)` devuelve el valor de `x` redondeado al número de cifras decimales indicado por la variable `numdig`. Estudia el código de esta función teniendo en cuenta para ello que:

   a) `mat2str(x,n)` convierte el valor de `x` en una cadena de caracteres, redondeando `x` a `n` dígitos de precisión.

   b) `eval(cad)` ejecuta la cadena `cad` como si fuera código de Octave. El resultado en la función `redondeo.m` es un valor numérico (o matricial) en coma flotante.

   c) El argumento `''local''` en la función `output_precision` permite que el efecto de esta función deje de estar activo al terminar la ejecución de la misma o de otro script que pueda utilizarla.

   Crea un archivo script en el que se calculen los resultados de las operaciones que siguen con cuatro cifras decimales (redondeo el que utiliza Octave por defecto):

   a) $0.6688 \oplus 0.3334$;

   b) $1000 \ominus 0.05001$;

   c) $2.000 \otimes 0.6667$;

   d) $25.00 \oslash 16.00$.

   Calcula en cada caso los errores absolutos y relativos cometidos.

   A la vista de los resultados (o de otros intentos que puedas hacer, por ejemplo, directamente en la ventana de comandos), ¿puedes deducir qué tipo de redondeo utiliza Octave?

2. Consideremos la ecuación de segundo grado

$$1.002\,x^2 - 11.01\,x + 0.01265 = 0$$

   Trabajando con la función `redondeo`, calcula las soluciones de la ecuación realizando todos los cálculos con cuatro cifras significativas de las dos formas detalladas a continuación. Para una ecuación $ax^2 + bx + c = 0$ las raíces se pueden calcular:

   a) mediante las expresiones bien conocidas $\frac{-b \pm \sqrt{b^2-4ac}}{2a}$, y

   b) calculando una raíz x_1 de forma que se evite la pérdida de cifras significativas y teniendo en cuenta entonces que las raíces verifican $x_1 x_2 = \frac{c}{a}$.

Para hacernos una idea de la precisión de ambos cálculos vamos a reconstruir el polinomio original utilizando las dos funciones siguientes.

- La función `poly[x1,x2,...,xn]` de OCTAVE devuelve un vector compuesto por los coeficientes del polinomio cuyas raíces son `x1,x2,...,xn`; los coeficientes están ordenados de forma decreciente, es decir, desde el coeficiente del término de mayor grado, hasta el término independiente.
- La función `polyout(v,'x')` de OCTAVE, devuelve una cadena igual a la expresión del polinomio en la variable `x` que tiene los coeficientes incluidos en el vector `v`.

Realiza los mismos cálculos con 2 y 8 cifras significativas.

3. De acuerdo con el ejercicio 1.27 (con las precauciones sugeridas por el apartado c) del mismo ejercicio) parece que será más preciso realizar una suma de términos positivos de forma que el tamaño de los sumandos sea creciente. Vamos a realizar la experiencia con una suma bien conocida:

$$\sum_{k=1}^{\infty} \frac{1}{k^2} = \frac{\pi^2}{6}$$

El objetivo es calcular las sumas

$$\sum_{k=1}^{n} \frac{1}{k^2}$$

para distintos valores de n en orden creciente y decreciente y comparar los resultados.

Al final del enunciado de este ejercicio se incluyen detalles sobre las órdenes de OCTAVE necesarias para desarrollar el ejercicio.

a) Calcula, **en precisión simple**, la suma en orden creciente y decreciente de sumandos para n igual a 50, 100, 500, 1000, 10000, 100000. Haz una tabla con los valores de n, los dos valores de la suma y los respectivos errores relativos (usando el valor exacto $\pi^2/6$).

b) Haz lo mismo en precisión doble.

c) Repite el resultado utilizando la función `redondeo` con cinco cifras decimales (evita en principio el caso $n = 100000$ en este apartado, porque es muy lento).

d) Una vez hayas depurado todos los apartados anteriores introduce los comandos `tic` y `toc` que permiten controlar el tiempo de ejecución: `tic` inicia el cronómetro, `toc` lo detiene. Puedes asignar el valor de `toc` a una variable (por ejemplo, `t=toc`), y así mostrar el tiempo de ejecución de un proceso en el formato que desees.

Observaciones.

a) Para realizar un cálculo en precisión simple basta escribir `single(calculo)`. Incluso puedes escribir `single(a)` siendo `a` un valor fijo o una variable numérica. Para realizar los cálculos en precisión doble no hay que tomar ninguna precaución, OCTAVE los hace con esta precisión por defecto.

b) Los bucles en OCTAVE los puedes escribir de varias formas

```
for k=1:n       for k=n:-2:1    for k=[n1,n2,...,np]
...             ...             ...
endfor          endfor          endfor
```

c) Una idea para obtener tablas en la salida de OCTAVE es utilizar especificaciones de formato en la orden de escritura. Por ejemplo:

`printf(form,x1,x2,x3)`

escribirá las variables `x1`, `x2` y `x3` con el formato indicado por la especificación `form`, que es una cadena o una variable de tipo cadena que debemos haber definido previamente. Por ejemplo, podríamos definir (o utilizar la cadena siguiente directamente en `printf`):

```
form='%12.7e,%16.8f,%6u';
```

lo que significa que: la primera variable se imprimirá ocupando el ancho de 12 caracteres con 7 cifras decimales en notación exponencial, la segunda ocupará el ancho de 16 caracteres con 8 decimales y la tercera será un entero sin signo (de «**u**nsigned») ocupando el ancho de 6 caracteres.

Departamento de Matemáticas. Campus de Espinardo. 30100 Espinardo Murcia
e-mail: pedreno@um.es