PROGRAMACIÓ CIENTÍFICA. SEMESTRE DE TARDOR. PRÀCTIQUES DE LABORATORI D'ORDINADORS

LLISTA 1. REPÀS D'ELEMENTS DE PROGRAMACIÓ

1 Per a cada valor x guardat en una variable de tipus `double`, sigui $fl(x)$ el mateix valor x, però guardat en `float`. Llavors la diferència $e_a(x) = fl(x) - x$ pot ser considerada com *l'error absolut en l'emmagatzematge del valor* `float` x. Anàlogament, $e_r(x) = e_a(x)/x$ seria *l'error relatiu.*

Genereu taules d'errors absoluts i relatius quan x varia en intervals del tipus $[2^i, 2^{i+1}]$ (per exemple, feu que x prengui 1001 valors equidistant en un d'aquests intervals).

Feu-ho per a diversos valors de i (per exemple, -2, -1, 0, 1, 2).

Dibuixeu les taules d'errors absoluts i d'errors relatius usant `gnuplot`. Mireu d'entendre els comportaments que s'observen.

2 Estimeu el valor de π pel *mètode de Montecarlo.* Cal generar molts punts aleatoris (x, y) dins del quadrat $Q = [0, 1] \times [0, 1]$, i cal comptar quants d'ells són dins del cercle unitat (o sigui, verifiquen $x^2 + y^2 < 1$). Teòricament, la proporció ha de ser $\pi/4$.

Podeu trobar, experimentalment, alguna relació entre la precisió obtinguda i el nombre de punts generats?

3 Programeu el càlcul de `sin`(x), per a qualsevol valor de x, amb una precisió determinada `prec` (per exemple 10^{-12}), usant la *sèrie de Taylor* en el 0:

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots \ .$$

Com que la sèrie és *convergent i alternada*, l'error que es comet quan se sumen uns quants termes és menor que el valor absolut del primer terme no sumat. Cal aprofitar això per anar acumulant termes de la sèrie mentre calgui per tal d'obtenir la precisió desitjada. Però no s'han de sumar més termes que els estrictament necessaris.

Abans de passar a acumular termes, useu la periodicitat i les simetries de la gràfica de la funció `sin`(x) per a reduir el valor de x a l'interval $[0, \pi/2]$. D'altra banda, procureu estalviar operacions: calculeu cada nou terme de la sèrie a partir del terme anterior ja conegut.

El programa ha de llegir els valors de x, de *prec* i el nombre màxim de termes que se sumaran *maxterm*, per tal de preveure el cas que, per algun motiu, no es pot assolir la precisió desitjada.

Compareu el valor calculat amb el de la funció `sin`(x) de C, per tal de comprovar que el programa funciona bé.

Nota. Podeu fer programes semblants per a altres funcions, per exemple `cos`(x).

4 Programeu el *mètode de Newton-Raphson* per a calcular arrels reals d'un polinomi real $p(x)$. Cal llegir el grau (fitat) i els coeficients del polinomi (en la base natural); cal cercar un interval que contingui totes les arrels reals (consulteu com trobar una cota de les arrels reals en funció dels coeficients); cal programar la detecció de canvis de signe del polinomi quan es va variant x amb un pas h fixat (per exemple, 10^{-2}); i, per a cada canvi de signe trobat, cal programar el mètode iteratiu $x_{i+1} = x_i - \frac{p(x_i)}{p'(x_i)}$ $\forall i \geq 0$. Es pot prendre, com a x_0, el punt mig entre les dues abscisses que limiten el canvi de signe.

Cal llegir també la precisió en què es volen trobar les arrels, i cal limitar el nombre màxim d'iteracions de Newton-Raphson permeses. O sigui, hi ha la possibilitat que no hi hagi convergència.

Mireu d'estructurar bé el programa: useu una funció `main` i diverses funcions d'usuari: una per a avaluar el polinomi, una altra per a avaluar la derivada, ...

5 Programeu el *mètode dels trapezis* per a calcular integrals definides de funcions contínues en intervals compactes:

$$\int_a^b f(x)dx \approx \frac{h}{2}\left(f(a) + 2\sum_{i=1}^{n-1} f(x_i) + f(b)\right) ,$$

on $n \in N$ és un valor gran, $h = (b-a)/n$ és el pas entre els punts on s'avalua la funció, i $x_i = a + ih$ $\forall i = 1, 2, \ldots, n-1$ són punts equidistants.

S'han de llegir els valors a, b i n. Feu una funció d'usuari en C per a avaluar l'integrand f.

Aplicació: $a = 0$, $b = 1$, $f(x) = ln(x)\, ln(1-x)$, diverses n. El resultat exacte és $2 - \frac{\pi^2}{6}$.

6 Feu un programa que llegeixi un valor n (enter positiu), llegeixi una permutació P del conjunt $\{0, 1, 2, \ldots, n-1\}$, comprovi que és una permutació correcta, i trobi i escrigui tots els *cicles* que conté la permutació P. Penseu bé un algorisme abans de començar a programar (o busqueu-ne algun).

Per exemple, si $n = 8$ i $P = \{2, 4, 0, 1, 7, 3, 6, 5\}$ llavors hi ha 3 cicles: $(2, 0)$, $(4, 7, 5, 3, 1)$ i (6).

Feu una variació en la qual només es llegeixi n, i la permutació es generi aleatòriament.

7 Programeu el producte de dues matrius $n \times n$: $C = A\,B$. Les fórmules són:

$$c_{ij} = \sum_{k=0}^{n-1} a_{ik} b_{kj} \; , \quad \forall j = 0, 1, \ldots, n-1 \; , \quad \forall i = 0, 1, \ldots, n-1 \; .$$

Llegiu només la dimensió n (fitat per $N = 590$), i genereu aleatòriaments els coeficients de les matrius A i B a l'interval $[-1, +1]$.

Cal fer 3 bucles repetitius "aniuats"(amb índexs i, j, k). Feu-ho de 6 maneres diferents (les 6 possibles permutacions de l'ordre dels bucles), i compareu els 6 temps d'execució.

Comproveu que si augmenteu N, no es pot executar el programa.