

## Part 1

- En primer lloc veurem com és un programa senzill en llenguatge C, com s'inclouen les llibreries, com es declaren variables, i com donar per pantalla els resultats de forma ordenada.

Aquest programa és un exemple de propagació dels errors: Una de les variables que calculem hauria de valer sempre 1, però, degut als arrodoniments, numèricament no és així.

1 Donats  $a$  i  $b$ , considerem el següent esquema iteratiu:

$$\begin{aligned} x_0 & \text{ fixat} \\ y_0 &= b\sqrt{1 - (x_0/a)^2} \\ t_0 &= (x_0/a)^2 + (y_0/b)^2 = 1 \\ \left\{ \begin{array}{l} x_{n+1} &= 2x_n y_n / b \\ y_{n+1} &= b((x_n/a)^2 - (y_n/b)^2) \\ t_{n+1} &= (x_{n+1}/a)^2 + (y_{n+1}/b)^2 \end{array} \right. \quad n = 0, 1, \dots \end{aligned}$$

Es verifica  $t_{n+1} = ((x_n/a)^2 + (y_n/b)^2)^2 = t_n^2 = 1$  (per inducció, partint de  $t_0 = 1$ ).

Comencem per calcular la successió per uns valors de  $a, b, x_0$  fixats en el programa. A continuació teniu el programa per  $a = 1, b = 1$  i  $x_0 = 0.3$ .

```
/* Calcul d'una successio en precisió simple
 * fixats a, b, x0
 */
#include <stdio.h>
#include <math.h>
int main(void) {
    int n;
    float a, b, x0, y0, x1, y1, t, aux0, aux1;

    a = 1.f;
    b = 1.f;
    x0 = 0.3f;
    aux0 = x0 / a;
    y0 = b * sqrt(1.f - aux0*aux0);
    aux1 = y0 / b;
    printf("%3s_ %18s\n", "#n", "t");
    t = aux0*aux0 + aux1*aux1;
    printf("%3d_ %18.6e\n", 0, t);

    for (n = 1; n <= 30; n++) {
        aux0 = x0 / a;
        aux1 = y0 / b;
        x1 = 2.f*x0*y0 / b;
        y1 = b*( aux0*aux0 - aux1*aux1 );

        aux0 = x1 / a;
        aux1 = y1 / b;
        t = aux0*aux0 + aux1*aux1;
        printf("%3d_ %18.6e\n", n, t);
        x0 = x1;
        y0 = y1;
    }
}
```

```

    }
    return 0;
}

```

Comentari: hem escrit `#include <math.h>`, per tal de tenir la informació necessària de la funció arrel quadrada, `sqrt`.

- El programa anterior calcula una successió per a uns valors fixats de les variables  $a$ ,  $b$  i  $x_0$ , si volem calcular-ho per a diferents valors d'aquestes variables caldria modificar el programa cada cop. Vegem com fer el programa de manera que en cada execució donem valor a aquestes variables.

**2** Modifiquem el programa anterior de forma que els valors de  $a$ ,  $b$  i  $x_0$  s'hagin de llegir.

```

/* Calcul d'una successio en precisió simple
 * es llegeixen a, b, x0 */
#include <stdio.h>
#include <math.h>
void exit(int);
int main(void) {
    int n, k;
    float a, b, x0, y0, x1, y1, t, aux0, aux1;
    k = scanf("%f%f%f",&a,&b,&x0);
    aux0 = x0 / a;
    if (fabs(aux0) > 1.){
        printf("No es pot fer l'arrel quadrada\n");
        exit(1);
    }
    y0 = b*sqrt(1.f - aux0*aux0);
    aux1 = y0 / b;
    printf("%3s%18s\n", "#n", "t");
    t = aux0*aux0 + aux1*aux1;
    printf("%3d%18.6e\n", 0, t);
    for (n = 1; n <= 30; n++) {
        aux0 = x0 / a;
        aux1 = y0 / b;
        x1 = 2.f*x0*y0 / b;
        y1 = aux0*aux0 - aux1*aux1;
        aux0 = x1 / a;
        aux1 = y1 / b;
        t = aux0*aux0 + aux1*aux1;
        printf("%3d%18.6e\n", n, t);
        x0 = x1;
        y0 = y1;
    }
    return 0;
}

```

Executeu-lo ara pels valors

a	b	$x_0$
1	1	0.3
1	1.0001	0.3
1	0.9999	0.3
1	1	0.3001
1	1	0.2999

- Si volem fer la gràfica del valor de  $t$  en funció de  $n$ , necessitem guardar en un fitxer els valors que obtenim per pantalla, per això redireccionem la sortida:

`./nom.exe > nomfitxer.res`

Torna a executar el programa redireccionant la sortida als fitxers que apareixen a la taula.

a	b	$x_0$	nom fitxer
1	1	0.3	f1103.res
1	1.0001	0.3	f11+03.res
1	0.9999	0.3	f11-03.res
1	1	0.3001	f1103+.res
1	1	0.2999	f1103-.res

Feu la gràfica dels diferents casos, mitjançant el programa `gnuplot`.

En primer lloc dibuixeu les gràfiques pels rangs de  $t$  següents:  $[0.5, 2.5]$  i  $[0.99, 1.01]$ . Comenteu els resultats.

Ara feu el dibuix usant escala logarítmica:

```
plot 'f1103.res' u 1:(log10($2)) w l, ...
```

- Hem calculat la successió de termes usant precisió simple (les variables eren de tipus `float`). Què passa si usem precisió doble? (les variables han de ser de tipus `double`) Compara les gràfiques obtingudes en precisió simple i doble.

**3** Modifiqueu el programa anterior per treballar en doble precisió i calculant 60 termes de la successió. Executeu-lo ara pels valors de la taula, redireccionant la sortida. Feu la gràfica mitjançant el programa `gnuplot`.

- Exercici d'autoavaluació:

**4** Definim l'èpsilon de la màquina com el nombre positiu  $\epsilon$  tal que:

$$\forall x \in [0, \epsilon) \quad fl(1+x) = 1 \quad \text{i} \quad \forall x \geq \epsilon \quad fl(1+x) > 1,$$

on  $fl(y)$  és la representació en punt flotant de  $y$ .

Feu un programa que calculi l'epsilon de la màquina pels tipus `float` i `double`.

Quina conclusió en traieu?