Mètodes numèrics

Isaac Pizarro Contreras (1636421) — 2n de grau en Matemàtiques
Gina Gardeñes Farré (1638998) — Curs 2023 - 2024

PRÀCTICA 1

1. Introducció

En aquesta primera pràctica de l'assignatura de Mètodes Numèrics, s'explora la estabilitat numèrica de diferents mètodes de càlcul per a funcions trigonomètriques, en particular el sinus i cosinus de múltiples d'un angle. L'objectiu és analitzar com les variacions en el càlcul afecten la precisió i l'eficiència, utilitzant quatre algoritmes implementats en C. El primer es basa en crides directes a funcions estàndard, mentre que els altres utilitzen recurrències trigonomètriques per a optimitzar el rendiment i reduir l'error d'arrodoniment. Mitjançant experiments, es mesura la precisió i eficiència de cada mètode, proporcionant una comprensió detallada de les compensacions entre precisió, estabilitat numèrica i rendiment. Aquesta pràctica destaca la importància de la selecció metòdica en el càlcul de funcions en general i, en aquest cas particular, de funcions trigonomètriques, demostrant l'aplicabilitat dels conceptes teòrics de mètodes numèrics en la resolució efectiva de problemes.

2. Implementació del sowftare

En aquest apartat es comentaran decisions rellevants per al millor funcionament del codi però no es farà una descripció intensiva de tota la programació realitzada ja que aquesta ja ve descrita en el manual del software d'aquesta mateixa memòria i els comentaris dels fitxers de c.

- En la funció trigrec1 es guarda el valor de la segona component del vector c en una variable c_1 perquè al bucle for s'utilitza en cada iteració el valor de c_1 i per tal de que el compilador no hagi de moure's de la posició j-èssima del vector c fins a la posició 2 del mateix a cada iteració s'utilitza el valor guardat a la variable c_1. És una mesura cautelar ja que presuposem que el compilador tindria formes eficients de solucionar aquest problema.
- A la funció trigrec2 apliquem la mateixa solució que abans però afegint el mateix raonament sobre la variable s_1 que guarda el respectiu valor de s.
- A la funció trigrec3 per calcular la diferència inicial per al sinus en comptes de computar la operació

  $$(-1)^{\frac{x}{\pi}}$$

  comprovem si la part entera de x/pi és parell o no, si és parell aleshores el resultat es 1 i si és senar es -1.
- En les utilitats trigrec_escr i trigrec_temps_err utilitzem malloc per reservar memòria per les matrius c i s. En aquest cas la particularitat és que s'ha triat la estructura de matriu per c i s de forma que cada fila correspon als resultats d'un algorisme.

3. Manual del sowftare

3.1 La funció trigrec0

La funció `trigrec0()` calcula el cosinus i sinus dels múltiples d'un angle.

Prototipus:

```
void trigrec0 (int k, double x, double c[], double s[]);
```

Descripció:

- Comença inicialitzant `c[0]` a 1 i `s[0]` a 0, corresponents als valors de $\cos(0)$ i $\sin(0)$.
- Entra en un bucle que recorre des de $j = 1$ fins a $j = k$.
- En cada iteració del bucle, calcula $\cos(j \cdot x)$ i $\sin(j \cdot x)$.
- Emmagatzema aquests valors en les posicions corresponents dels vectors `c` i `s`.

Paràmetres:

- `k`: Número màxim de múltiples a calcular.
- `x`: Angle original en radians.
- `c`: Vector on es guardaran els valors de cosinus calculats.
- `s`: Vector on es guardaran els valors de sinus calculats.

3.2 La funció trigrec1

La funció `trigrec1()` calcula de manera eficient el cosinus i sinus de múltiples d'un angle mitjançant recurrències trigonomètriques.

Prototipus:

```
void trigrec1 (int k, double x, double c[], double s[]);
```

Descripció:

- Inicialitza `c[0]` a 1 i `s[0]` a 0, corresponents als valors de $\cos(0)$ i $\sin(0)$. També inicialitza `c[1]` a $\cos(x)$ i `s[1]` a $\sin(x)$, preparant així els dos primers elements per a les recurrències.
- Entra en un bucle que recorre des de $j = 1$ fins a $j < k$.
- Utilitza una relació de recurrència dins del bucle per calcular de manera eficient el cosinus i sinus de múltiples de x, evitant així el càlcul directe amb funcions trigonomètriques. Aquest mètode millora l'eficiència i potencialment la precisió dels càlculs.
- Emmagatzema els resultats calculats en les posicions corresponents dels vectors `c` i `s`.

Paràmetres:

- `k`: Número màxim de múltiples a calcular.
- `x`: Angle original en radians.
- `c`: Vector on es guardaran els valors de cosinus calculats.
- `s`: Vector on es guardaran els valors de sinus calculats.

3.3 La funció trigrec2

La funció `trigrec2()` calcula de manera més eficient el cosinus i sinus de múltiples d'un angle utilitzant fórmules trigonomètriques.

Prototipus:

```
void trigrec2 (int k, double x, double c[], double s[]);
```

Descripció:

- Inicialitza els primers elements de `c` i `s` amb els valors de $\cos(0)$, $\sin(0)$, $\cos(x)$ i $\sin(x)$, respectivament, preparant els quatre primers elements per a les recurrències.
- Utilitza un bucle que recorre els múltiples de x, des de $j = 1$ en endavant, fins a l'últim múltiple especificat.
- Dins d'aquest bucle, aplica una relació de recurrència basada en les identitats trigonomètriques per a angles sumats per calcular el cosinus i sinus de múltiples de x. Aquesta aproximació evita els càlculs directes amb funcions trigonomètriques.
- A través d'aquesta metodologia, optimitza el càlcul i redueix l'error numèric. Les fórmules s'utilitzen de manera que cada nou valor calculat depèn dels valors anteriors, la qual cosa millora significativament l'eficiència comparada amb mètodes que requereixen càlculs directes repetits.

Paràmetres:

- `k`: Número màxim de múltiples a calcular.
- `x`: Angle original en radians.
- `c`: Vector on es guardaran els valors de cosinus calculats.
- `s`: Vector on es guardaran els valors de sinus calculats.

3.4 La funció trigrec3

La funció `trigrec3()` calcula el cosinus i el sinus de múltiples d'un angle utilitzant recurrències millorades considerant les diferències entre cosinus i sinus consecutius.

Prototipus:

```
void trigrec3 (int k, double x, double c[], double s[]);
```

Descripció:

- Empreu una tècnica de recurrència per calcular de manera més eficient i precisa el cosinus i sinus de múltiples d'un angle x. Aquesta metodologia redueix l'error numèric i millora l'estabilitat dels càlculs en comparació amb els mètodes anteriors.
- Inicialitza el primer terme de `c` i `s` com a $\cos(0) = 1$ i $\sin(0) = 0$, respectivament, establint així les bases per als càlculs subsequents.
- Calcula els increments d_c i d_s utilitzant el sinus de $x/2$. Aquests increments es fan servir per calcular de manera recorrent els valors de `c[j]` i `s[j]` per a j des de 1 fins a k.
- Ajusta la recurrència en cada pas basant-se en els valors calculats anteriorment. S'utilitza un factor de correcció t per a aquest propòsit, garantint així que els canvis següents siguin coherents i precisos.

Paràmetres:

- `k`: Número màxim de múltiples a calcular.
- `x`: Angle original en radians.
- `c`: Vector on es guardaran els valors de cosinus calculats.
- `s`: Vector on es guardaran els valors de sinus calculats.

Validació del software

3.5. Utilitat trigrec_escr

Comanda per generar l'executable: `make trigrec_escr`

1	1	1	1	0	0	0	0
0.9999243559536896	0.9999243559536896	0.9999243559536896	0.9999243559536896	0.01229968985784608	0.01229968985784608	0.01229968985784608	0.01229968985784608
0.9996974352588016	0.9996974352588017	0.9996974352588016	0.9996974352588016	0.02459751891907374	0.02459751891907374	0.02459751891907374	0.02459751891907374
0.9993192722457354	0.9993192722457357	0.9993192722457354	0.9993192722457354	0.03689162666858092	0.03689162666858092	0.03689162666858092	0.03689162666858092
0.9987899241260516	0.9987899241260523	0.9987899241260517	0.9987899241260517	0.04918015315425573	0.04918015315425572	0.04918015315425572	0.04918015315425572
0.9981094709838179	0.9981094709838189	0.99810947098383	0.99810947098383	0.06146123926836503	0.06146123926836502	0.06146123926836503	0.06146123926836502

Exemple d'ús: ./trigrec_escr 5 0.0123

3.6 Utilitat trigrec_temps_err

Comanda per generar l'executable: `make trigrec_temps_err`

Exemple d'ús: ./trigrec_temps_err 10000000 0.0123

```
trigrec0 ha trigat 0.219614 segons
trigrec1 ha trigat 0.04472099999999998 segons
trigrec2 ha trigat 0.04958800000000002 segons
trigrec3 ha trigat 0.04183799999999999 segons

errc1 = 2.493230234594523e-08    errc2 = 3.066316089928023e-10    errc3 = 1.524187026591406e-11
errs1 = 2.49511272962244e-08     errs2 = 3.066330522827343e-10    errs3 = 1.509501031116134e-11
```

4. Part experimental

Exercici 4 Explica en termes d'anàlisi els errors obtinguts en calcular les dades experimentals de la taula:

	k	x
1	100	0.0123
2	1000	0.00123
3	10000	0.000123
4	100000	0.0000123
5	1000000	0.00000123
6	10000000	0.000000123

Taula 1: Taula enumerada dels valors de k i x pels quals calculem l'error.

Procedirem doncs a enregistrar en cada taula particular els valors obtinguts pels algorismes de les dades corresponents de k i x. Com que el primer algorisme calcula $\cos(jx)$ i $\sin(jx)$ per $j = 1, ..., k$ de manera exacta a través de la pròpia funció de la computadora, és evident doncs que aquest no propaga cap error i que el càlcul és exacte per a tot valor de x i per tot j, per tant, podem anar directament a estudiar els algorismes 1, 2 i 3.

trigrec1	Error $\cos(jx)$	Error $\sin(jx)$
1	2.25264251696444e-13	1.271205363195804e-13
2	2.062616744069601e-11	1.03283769740796e-11
3	2.751270100631587e-09	1.239185198897985e-09
4	3.516659667912592e-07	1.614744689071301e-07
5	4.204362881488066e-05	1.933210492643944e-05
6	0.001008500573494902	0.0005532091341097023

Taula 2: Taula d'errors de l'algorisme trigrec1 .

En aquesta primera part experimental és compararà els resultats obtinguts per la màquina amb els estipulats pel document de la pràctica. Els donats pel documents, ens expliquen el següent:

Cas 1: trigrec1

Donat un valor x, la recurrència implementada en el codi aproxima el càlcul exacte de la funció $\cos(x)$, que denotarem com $c = \cos(x)$ i $\pm\sqrt{1-c^2} = \sin(x)$ (per algun valor de x) a una constant $\tilde{c}$, de manera que:

$$e_r(\tilde{c}, c) = \frac{\tilde{c} - c}{c}$$

és l'error relatiu e fitat per l'epsilon de la màquina, $|e| \leq \epsilon$. Recordem que l'epsilon de la màquina és :

$$\epsilon_m = \text{mín}\,\{\,\epsilon \,|\, 1 + \epsilon \neq 1\,\}$$

Donat aquest aclariment, i considerant que la recurrència evalua les funcions:

$$\varphi_1(c) = \cos(k \arccos c), \quad \varphi_2(c) = \sin(k \arccos c)$$

on $x = \arccos c$. Llavors ens trobem amb que l'error absolut de les corresponents funcions, partint de la fòrmula de propagació dels errors donada a teoria de mètodes numèrics, és respectivament, pel valor deles funcions aproximades $\tilde{\varphi_1}$ i $\tilde{\varphi_2}$:

$$e\frac{\cos x}{\sin x}k\sin(kx), \quad -e\frac{\cos x}{\sin x}k\cos(kj)$$

Com que calcular l'error relatiu, és a dir, e, del cosinus i el sinus per cada x i cada k seria una tasca poc precisa i comparar els mateixos errors substituïts directament de la fórmula no permetria el nivell d'anàlisi desitjat, la idea a la que sotmetrem l'estudi dels errors és basa en el càlcul de la raó d'aquest, i en funció de e, fent que així, l'error relatiu desapareguí i puguem comparar-los entre si i amb la màquina.

Aleshores, partint del coneixement que el primer algorisme és numèricament inestable, podem esperar que la raó sigui tan gran com ens imaginem. Com podem veure:

Per x=0.0123 i k=100:

$$|Err(\tilde{\varphi_1}(0{,}0123, 100)| = |e\frac{\cos(0{,}0123)}{\sin(0{,}0123)}100\sin(100 * 0{,}0123)| \approx 7662{,}124161e \tag{1}$$

$$|Err(\tilde{\varphi_2}(0{,}0123, 100)| = |-e\frac{\cos(0{,}0123)}{\sin(0{,}0123)}100\cos(100 * 0{,}0123)| \approx 2717{,}242856e$$

I per x=$1.23{\cdot}10^{-7}$ i k=10^7,

$$|Err(\tilde{\varphi_1}(1{,}23 * 10^{-7}, 10^7)| = |e\frac{\cos(1{,}23 * 10^{-7})}{\sin(1{,}23 * 10^{-7})}10^7\sin(10^7 * 1{,}23 * 10^{-7})| \approx 7{,}662511 * 10^{13}e$$

$$|Err(\tilde{\varphi_2}(1{,}23 * 10^{-7}, 10^7))| = |e\frac{\cos(1{,}23 * 10^{-7})}{\sin(1{,}23 * 10^{-7})}10^7\cos(10^7 * 1{,}23 * 10^{-7})| \approx 2{,}711738 * 10^{13}e$$

Per tant:

$$\frac{|Err(\tilde{\varphi_1}(x_6, k_6)|}{|Err(\tilde{\varphi_2}(x_1, k_1))} \approx 1{,}00005 * 10^{10} \approx 10^{10}$$

$$\frac{|Err(\tilde{\varphi_2}(x_6, k_6))|}{|Err(\tilde{\varphi_2}(x_1, k_1))} \approx 0{,}99799 * 10^{10} \approx 10^{10}$$

Tal i com hem predit, els resultats creixen ràpidament a mesura que la x és fa petita i la k és fa gran, ja sabíem doncs, que l'algorisme era inestable a nivell numèric. Ara però, comparem-los amb els resultats que ens dona l'ordinador. Volem veure si la propagació creix amb el mateix nivell. De la taula que hem enregistrar anteriorment, calculem la raó amb els mateixos valors de k i x que hem fet servir en les fórmules dels errors. Tenim doncs:

Pel que fa a la funció $\cos(jx)$:

$$\frac{e(\cos(x_6 j_6))}{e(\sin(x_1 j_1))} \approx 0{,}44770 * 10^{10} \approx 10^{10}$$

Pel que fa a la funcio $\sin(jx)$:

$$\frac{e(\cos(x_6 j_6))}{e(\sin(x_1 j_1))} \approx 0{,}419451 * 10^{10} \approx 10^{10}$$

Així doncs, veiem que que efectivament, Err(φ_1) Err(φ_2 representen les fites superiors dels errors generts de l'algorisme amb —e— fitat per l'epsilon de la màquina. Així:

$$|e(\cos(x,j))| \leq Err(\tilde{\varphi_1}), \quad |e(\sin(x,j))| \leq Err(\tilde{\varphi_2})$$

A més, al comparar d'aquesta manera els resultats dels errors, podem veure que ens justifiquen que efectivament la propagació dels errors es descontrola a mesura que els valors de k creixen i els de x decreixen. Per tant, l'algorisme 1 no ens interessa doncs no ens dona la precisió numèrica que busquem.

El mateix estudi el fem amb l'algorisme 2 i 3 per tal de veure quines conclusions en podem extreure:

Cas 2: trigrec2

A l'igual que abans, la base del segon algorisme el determina una sèrie de recurrències sortides de diferents igualtats trigonomèriques que relacionen els valors de les funcions en funció de les anteriors. Així doncs, pel que fa a l'algorisme dos, i amb un procediment numèric similar al d'abans, les fites dels erros venen donades per les següents expressions:

$$e_c k\cos(kx)\cos((k-1)x) + e_s k\sin(kx)(-\sin((k-1)x))$$

$$e_c k\cos(kx)\sin((k-1)x) + e_s k\sin(kx)\cos((k-1)x)$$

On e_c i e_s representen els errors relatius de l'aproximació de les funcions $\cos(x)$ i $\sin(x)$ respectivament. Aquest errors venen fitats per l'epsilon màquina, de manera que $|e_c|, |e_s| \leq \epsilon$. Per tant, com abans, tenim per una banda els valors d'aquestes expressions en funció de x_1 ik_1i x_6 i k_6 respecte la taula de valors. Així,

- Per a x=0.0123 i k=100:

$$|Err(\tilde{\varphi_1}(0{,}0123, 100))| =$$
$$= |e_c 100\cos(100 * 0{,}0123)\cos((99x) + e_s 100\sin(100 * 0{,}0123)(-\sin(99 * 0{,}0123))| \approx$$
$$\approx |31,36174279e_c - 88,43434e_s|$$

$$|Err(\tilde{\varphi_2}(0{,}0123, 100))| =$$
$$= |e_c 100\cos(100 * 0{,}0123)\sin((99 * 0{,}0123) + e_s 100\sin(100 * 0{,}0123)(\cos((99 * 0{,}0123))| \approx$$
$$\approx 31,36175e_c + 32,59172e_s$$

- Per a x=$1.23{\cdot}10^{-7}$ i k=10^7:

$$|Err(\tilde{\varphi_1}(1{,}23 * 10^{-7}, 10^7))| =$$
$$= |e_c 10^7\cos(10^7 * 1,23^{-7})\cos((10^7 - 1) * 1{,}23 * 10^{-7}) - e_s 10^7\sin(10^7 * 1{,}23 * 10^{-7})\sin((10^7 - 1) * 1{,}23 * 10^{-7})| \approx$$
$$\approx |1117148,97e_c - 8882851,03000e_s|$$

$$|Err(\tilde{\varphi_1}(1{,}23 * 10^{-7}, 10^7))| =$$
$$= |e_c 100\cos((10^7 - 1) * 1{,}23 * 10^{-7}) + e_s 10^7\sin(10^7 * 1{,}23 * 10^{-7})\cos((10^7 - 1) * 1{,}23 * 10^{-7})| \approx$$
$$\approx 3150153,013e_c + 3150154,243e_s$$

Obtenim doncs, pel que fa als errors repsectius de les funcions:

$$\frac{|Err(\tilde{\varphi_1}(x_6, k_6)|}{|Err(\tilde{\varphi_1}(x_1, k_1|)} = \frac{|1117148,97e_c - 8882851,03000e_s|}{|31,36174279e_c - 88,43434e_s|} \leq \frac{|1117148,97|\epsilon + |8882851,03000|\epsilon}{|31,36174279|\epsilon + |88,43433|\epsilon} =$$
$$= \frac{1117148,97 + 8882851,03000}{31,36174279 + 88,43434} \approx 0{,}83475 * 10^5 \approx 10^5$$

$$\frac{|Err(\tilde{\varphi_2}(x_6, k_6)|}{|Err(\tilde{\varphi_2}(x_1, k_1)|} = \frac{3150153,013e_c + 3150154,243e_s}{31,36175e_c + 32,59172e_s} \leq \frac{3150153,013\epsilon + 3150154,243\epsilon}{31,36175\epsilon + 32,59172\epsilon} =$$
$$= \frac{3150153,013 + 3150154,243}{31,36175 + 32,59172} \approx 0{,}98513 * 10^5 \approx 10^5$$

trigrec2	Error $\cos(jx)$	Error $\sin(jx)$
1	1.332267629550188e-15	2.886579864025407e-15
2	1.265654248072678e-14	2.575717417130363e-14
3	1.674216321134736e-13	3.373867771835851e-13
4	2.086331107875594e-12	4.328315483803635e-12
5	2.503708351753176e-11	5.182421158878014e-11
6	6.240574723648251e-11	1.321913689622534e-10

Taula 3: Taula d'errors de l'algorisme trigrec2 .

Aquí ja podem observar que, en comparació amb els errors del primer algorisme, la raó de proporció entre el creixemet de k i el decreixement de x en el càlcul de les funcions sinus i cosinus no és tan desproporcionat però que tot i així les fites no es controlen com ens agradaria. Segueix sent doncs, un algorsime numèricament inestable i per tant no ens interessa. Per l'altra banda, tenim els errors generats per l'ordenador:

I considerant els mateixos valors que portem estudiats, tenim:

$$\frac{e(\cos(x_6k_6))}{e(\cos(x_1k_1)} = \frac{6{,}240574723648251e-11}{1{,}332267629550188e-15} = 0,46841 * 10^5 \approx 10^5$$

$$\frac{e(\sin(x_6k_6)}{e(\sin(x_1k_6))} = \frac{1{,}321913689622534e-10}{2{,}886579864025407e-15} = 0{,}46809 * 10^5 \approx 10^5$$

Veiem igual que amb les expressions anteriors, que el creixement dels errors es descontrola amb el nivell de propagació prou paregut. A més, veiem que això implica, com amb l'algorisme 1 que efectivament:

$$|e(\cos(x_ik_i))| \leq Err(\tilde{\varphi_1}(x_i,k_i)), \quad |e(\sin(x_ik_i))| \leq Err(\tilde{\varphi_2}(x_i,k_i))$$

$\forall$ i $\in$ {1,2,3,4,5,6} i doncs, les expressions que trobem en el guió de la pràctica serveixen per fitar l'error absolut dels algorismes.

Cas 3: trigrec3

L'algorisme 3 se'ns presenta considerant les diferències entre cosinus i sinus consecutius de manera que les expressions donades donen lloc al codi més estable dels tres. Tindrem doncs, que, evaluant i aproximant les funcions:

$$\phi_1(s) := \cos(2k \arcsin(s)) \approx \tilde{\phi_1}$$
$$\phi_2(c) := \sin(2k \arcsin(s)) \approx \tilde{\phi_2}$$

on s representa $s = \sin(x/2)$ i que s'itereren a partir de la seva aproximació , fitada per l'error relatiu $|e| \leq \epsilon$. Com fins ara, de les expressions obtingudes a partir de la fórmula de la propagació dels errors, veiem que les respectives fites corresponen a:

$$Err(\tilde{\phi_1}) = -2k \tan(\frac{x}{2}) \sin(kx)e, \qquad Err(\tilde{\phi_2}) = 2k \tan(\frac{x}{2}) \cos(kx)e$$

Per tant, com hem fet fins ara, considerarem els primers valors de x i k i els últims, així, calcularem els corresponents erros de les funcions i les compararem per provar que, efectivament, sí son numèricament estables. Aleshores:

- Per x=0.0123 i k=100:

$$|Err(\tilde{\phi_1}(x_1,k_1))| = |-2 * 100 \tan(\frac{0{,}0123}{2}) \sin(100 * 0{,}0123)e| \approx |-1,15928e|$$

$$|Err(\tilde{\phi_2}(x_1,k_1)| = |2 * 100 \tan(\frac{0,0123}{2}) \cos(100 * 0{,}0123)e| \approx |0{,}41112e|$$

- Per x= 1,23*10^{-7} i k=10^7:

$$|Err(\tilde{\phi_1}(x_6,k_6))| = |-2 * 10^7 \tan(\frac{1,23 * 10^{-7}}{2}) \sin(10^7 * 0{,}0123)e| \approx |-1,15926e|$$

$$|Err(\tilde{\phi_2}(x_6,k_6)| = |2 * 10^7 \tan(\frac{1,23 * 10^{-7}}{2}) \cos(1,23 * 10^{-7})e| \approx |1,23000e|$$

Així, si comparem els resultats, la raó que obtenim en cada cas és :

- Per la funció cosinus:
$$\frac{Err(\tilde{\phi_1}(x_6,k_6)}{Err(\tilde{\phi_1}(x_1,k_1)} = \frac{|-1,15926e|}{|-1,15928e|} = \frac{1,15926}{1,15928} \approx 1$$

- Per la funció sinus:
$$\frac{Err(\tilde{\phi_2}(x_6,k_6)}{Err(\tilde{\phi_2}(x_1,k_1)} = \frac{|1,23000e|}{|0{,}41112e|} = \frac{1,23000}{0{,}41112} \approx 2{,}99183 \approx 3$$

Al veure els resultats d'aquest últim algorisme, veiem que ara la propagació dels errors s'estabilitza ràpidament i no augmenta de manera desproporcionada a mesura que els valros de x i k canvien, dit d'una altra manera, sempre que $k|x| \approx 1$ és mantenen en l'ordre de l'epsilon màquina i per tant, la recurrència que planteja l'algorisme 3 és numèricament estable i en conseqüència, és òptima per treballar. Si ho comparem ara amb els errors obtinguts per l'ordinador, aleshores:

trigrec3	Error $\cos(jx)$	Error $\sin(jx)$
1	3.33066907387547e-16	2.220446049250313e-16
2	1.332267629550188e-15	7.77156117237609e-16
3	4.107825191113079e-15	5.662137425588298e-15
4	7.66053886991358e-15	9.325873406851315e-15
5	2.386979502944087e-14	4.696243394164412e-14
6	1.18793863488917e-13	7.582823248189819e-14

Taula 4: Taula d'errors de l'algorisme trigrec3 .

I per tant, al comparar els errors, considerem:

- De la funció cosinus:
$$\frac{e(\cos(k_6x_6))}{e(\cos(k_1x_1))} = \frac{1{,}18793863488917e-13}{3{,}33066907387547e-16} = 0,35667 * 10^3 \approx 10^3$$

- De la funció sinus:
$$\frac{e(\sin(k_6x_6))}{e(\sin(k_1x_1))} = \frac{7{,}582823248189819e-14}{2{,}220446049250313e-16} = 0,34149 * 10^3 \approx 10^3$$

Aquí realment la proporció en comparació amb les fórmules dels errors no són del tot similar, però tot i així ens mostren aquesta diferència en l'establitat numèrica que té l'algorisme 3 respecte els altres dos, que és el que realment volem concloure estudiant les raons, per tant, tot i que desconeixem la justificació radere aquests valors, els resultats adquirits concorden amb l'anàlisi final, tal i com volíem veure i extreure.

Exercici 5 Feu experiments per tal d'estimar el nombre d'operacions per segon que fan els algorismes 2 i 3. A partir d'això, estimeu a quantes operacions aritmètiques equival (en termes de temps d'execució) avaluar les funcions sin i cos a l'algorisme 0 (algorisme 2.1 respecte el document de la pràctica) .

Per començar, calculem el nombre d'operacions aritmètiques que s'efectuen en cadascun dels algorismes.

- **Nombre d'operacions de l'algorisme 2:**
$$n1 = 6(k-1)$$
Doncs tenim dues sumes i quatre productes per a cada j, en una recurrència que s'inicialitza en $k = 1$ i arriba fins a $j = k - 1$.

- **Nombre d'operacions de l'algorisme 3:**
$$n2 = 6k + 6$$
En aquest cas per a cada $j = 1, ..., k$ hi ha dues multiplicacions i quatre sumes. Veiem però, que per k prou gran, les operacions aritmètiques necessàries per calcular les δ_1^c i δ_1^s són pràcticament negligibles doncs són independents de l'escalar k i per tant, ens son irrellevants per al compteig final. Així doncs, per a l'algorisme 4 treballarem amb un total de $6k$ operacions.

Per començar amb aquest experiment, el primer que farem serà executar 100 cops l'utilitat trigrec_temps_err per a que els algorismes 2 i 3 calculin 100 cops cada una de les recurrències fixant x = 0.0123 i k = 10000000. Un cop tinguen enregistrats tots els temps corresponents, farem una estimació dels segons que tarda a partir de tots els resultats guardats.

$$t = \frac{\sum_{i=1}^{m=100}(t_i)}{100}$$

on t_i representa el temps d'ejecució de cada execució. Treballarem amb l'aproximació de t, $\tilde{t}$, assumint que el resultat serà un nombre llarg.

Busquem ara, el nombre d'operacions per segon dels corresponents algorismes 2 i 3:

$$N_1 = \frac{n_1}{t_1} = \frac{6(k-1)}{\frac{\sum_{i=1}^{m}(t_{1\,i})}{m}}, \quad N_2 = \frac{n_2}{t_2} = \frac{6k}{\frac{\sum_{i=1}^{m}(t_{2\,i})}{m}}, \quad N = \frac{N_1 + N_2}{2}$$

Un cop obtinguts aquest resultats, farem el mateix per l'algorisme 0: una estimació del temps d'execució dels resultats pels mateixos x i k emprats anteriorment. Un cop tinguem l'estimació desitjada, que denotarem per $\tilde{t_0}$, el nombre d'operacions que fa l'algorisme 0 serà:

$$N_0 = N\tilde{t_0}$$

- **Resultats de l'experiment:**
  Nombre aproximat d'operacions per segon de l'algorisme 2:

  $$N_1 = \frac{59999994}{0{,}0496} = 1209677298$$

  Nombre aproximat d'operacions per segon de l'algorisme 3:

  $$N_2 = \frac{60000000}{0{,}0418} = 1435406698$$

  Nombre aproximat d'operacions de l'algorisme 0:

  $$N_0 = \frac{1209677298 + 1435406698}{2} 0{,}217 = 286991613$$

  Notem que la quantitat d'operacions que realitza el algorisme 0 és aproximadament 5 vegades més que la quantitat d'operacions que realitzen els algorismes 2 i 3. Aquest fet es reflexa en el temps d'execució dels diferents algorismes, ja que l'algorisme 0 és clarament molt més lent que els 2 i 3.