Mètodes Numèrics
Pràctica 2: Zeros de funcions

Albert Rodas Barceló i Marc Seguí Coll
Universitat Autònoma de Barcelona

Març 2019

1 Exercici 1

1.1 Mètode de Cardano

A partir del mètode de Cardano, donada una equació de tercer grau de la forma $x^3 + ax + b = 0$ amb $a, b \in \mathbb{R}$, si $\Delta = b^2 + \frac{4}{27}a^3 > 0$ llavors l'única solució real α de l'equació és

$$\alpha = \sqrt[3]{\frac{-b + \sqrt{\Delta}}{2}} + \sqrt[3]{\frac{-b - \sqrt{\Delta}}{2}} \tag{1}$$

Notem que el càlcul d'aquesta arrel pot sofrir cancelacions quan $\Delta \approx b^2$ independentment del signe de b. Si $b = 0$ el resultat és trivial.

Considerant l'equació $x^3 - x - 40 = 0$ al programa `p1a.c` es comprova aquest error. Els valors obtinguts per a l'arrel real α són, en presició simple, $\alpha_{\mathtt{s}} = 3.5173848$ i, en precisió doble, $\alpha_{\mathtt{d}} = 3.517393514052852$. Avaluant dits valors a la funció $f(x) = x^3 - x - 40$ s'obté

$$f(\alpha_{\mathtt{s}}) = -3.1661987 \cdot 10^{-4} \quad \text{i} \quad f(\alpha_{\mathtt{d}}) = 1.236344360222574 \cdot 10^{-12}$$

És clar que hi ha hagut algun error en el càlcul ja que cap de les arrels trobades en avaluar-les en f donen 0. Com $b = -40$ i $a = -$ llavors $\Delta = 1600 - \frac{4}{27} \approx 1600 = b^2$ i per tant al segon sumand de la fórmula es produeix error de cancelació en les dues representacions (tant simple com doble).

1.2 Mètode de Newton

Aplicant el mètode de Newton per a determinar una aproximació d'α, la fórmula per als successius iterats és

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} = x_n - \frac{x_n^3 - x_n - 40}{3x_n^2 - 1} \tag{2}$$

Fent $x_0 = 2$, el nombre màxim de decimals correctes que es poden assolir en *float* és 7 (6 xifres correctees a la part decimal). Amb els resultats de `p1b.c`, es pot observar que $\alpha_{\mathtt{s}} = x_5 = 3.5173936$ on el darrer decimal és incorrecte com veurem en *double*. A partir de la sisena iteració del mètode, la màquina tracta $\frac{f(x_n)}{f'(x_n)} = 0$ per a $n > 5$ i els iterats no milloren l'aproximació.

En precisió doble s'obtenen 8 decimals correctes amb 6 iteracions $\alpha_{\mathtt{8d}} = x_6 = 3.51739351$ i 15 decimals correctes amb 7, donant $\alpha_{\mathtt{15d}} = x_7 = 3.517393514052818$. Avaluant tots aquests resultats per f queda:

$$f(\alpha_{\mathtt{s}}) = 3.8146973 \cdot 10^{-6} \quad \text{i} \quad f(\alpha_{\mathtt{8d}}) = 1.84741111 \cdot 10^{-13} \quad f(\alpha_{\mathtt{15d}}) = -2.131628207280301 \cdot 10^{-14}$$

Podem afirmar aquests resultats ja que en fer $f(x_8)$ l'ordinador retorna el valor idènticament 0 i els 15 decimals d'x_7 coincideixen amb els d'x_8.

1.3 Comparació dels mètodes de bisecció, secant i Newton

Considerarem a partir d'ara l'equació $x^3-x-400=0$ i $g(x)=x^3-x-400$. Calculant $g(2)\cdot g(8)$ a `p1c.c` es determina que $g(2)\cdot g(8)=-40976<0$ i pel teorema de Bolzano existeix $\beta\in[2,8]$ tal que $g(\beta)=0$. Calculant el valor de β amb el mètode de Cardano (1) es determina que $\beta_{\mathtt{s}}=7.4131327$ i $\beta_{\mathtt{d}}=7.413302725859884$. Avaluant en g es té que $g(\beta_{\mathtt{s}})=0.045070298$ i $g(\beta)=3.254285729781259\cdot10^{-10}$. L'error produït per cancel·lacions en *float* és de l'ordre de 10^{-4}. Si calculem el segon sumand de la fórmula en *float* obtenim 0.045070298 i en *double* 0.04524029710266585. Notem que aquí és on està el problema ja que la quarta xifra decimal ha canviat, per tant es perd precisió a causa de la contaminació de l'error de cancel·lació.

Els resultats següents s'han realitzat en precisió doble per raons òbvies. Mitjançant els programes `p1c1.c`, `p1c2.c` i `p1c3.c` s'ha determinat el valor de β mitjançant els mètodes de bisecció, secant i Newton respectivament. Respecte els dos primers mètodes, aquests s'apliquen a l'interval $[2,8]$ i al darrer la llavor inicial és $x_0=2$. Els valors obtinguts es representen a la següent taula:

Mètode	Iteracions	Ordre	Aproximació de β	$g(\beta)$
Bisecció (`p1c1.c`)	53	1	7.413302725857897	$-1.13686837721616\cdot10^{-13}$
Secant (`p1c2.c`)	8	≈1.6	7.413302725857899	$1.70530256582424\cdot10^{-13}$
Newton (`p1c3.c`)	11	2	7.413302725857898	0

El darrer 0 significa 0 en *double*, és a dir, realment és el valor és 0 ± 10^{-15} deguda la precisió limitada en la representació. Amb aquests resultats, podem concloure que el mètode de Newton és el millor dels tres. Notem que en aquest cas requerix més iteracions que el mètode de la secant, cosa que es deu al fet que la llavor inicial és $x_0=2$. Amb la llavor inicial $x_0=7$ s'aconsegueix el mateix resultat amb només 5 iteracions. Una bona estratègia seria realitzar una o dues aplicacions del mètode de la bisecció o el de la secant (determinant així una bona llavor x_0) i posteriorment iterar amb Newton a partir de la llavor obtinguda, ja que la convergència d'aquest mètode és més ràpida.

2 Exercici 2

En aquest exercici estudiem la mateixa equació $g(x)=0$ que en l'apartat 1.3 amb el mètode següent

$$\begin{aligned} x_{k+1} &= x_k + b_k g(x_k) \\ b_{k+1} &= b_k\cdot(2-g'(x_{k+1})b_k) \end{aligned} \tag{3}$$

agafant $x_0=6$ i $b_0=\frac{1}{g'(x_0)}$. Treballant amb precisió doble obtenim en 8 iteracions un valor de $x_8=7.413302725857898$ tal que $g(x_8)=0\pm10^{-15}$. A més, iterant el mètode una vegada més obtenim $x_9=x_8$ (on la igualtat s'entèn en representiació en punt flotant *double*) de manera que tenim 16 xifres decimals correctes.

Per calcular l'ordre de convergència cal trobar un valor n tal que el límit del quocient entre diferències d'iterats consecutius elevant el denominador a n tendeixi a un número real diferent de 0 i ∞ (ens referim al límit de la successió formada pels termes de l'equació 4). En el nostre cas, com que disposem de finits iterats, el que es fa és calcular uns quants termes de la successió i veure per quin ordre veiem que els valors k_j són més estables.

$$k_j = \frac{|x_{j+1} - x_j|}{|x_j - x_{j-1}|^n} \tag{4}$$

On n és l'ordre de convergència i k_j és el valor que pel valor d' n adequat es mantindrà aproximadament constant per a $j \in \{1, 2 \ldots, N\}$ on N és el nombre d'iteracions menys un. El nombre n no té per que ser natural, podria tractar-se inclús d'un irracional. Si k_j creix vol dir que hem provat per un ordre de convergència massa gran i si per contra k_j va tendint a 0 vol dir que hem provat per un ordre de convergència massa petit. A continuació es mostren els resultats obtinguts (`p2.c`):

Valor de j	n=1	n=2	n=3
2	0.72188	3.94297	21.53678
3	0.32842	2.48497	18.80240
4	0.08573	1.97526	45.50772
5	0.00769	2.06823	555.77350
6	$7.00793 \cdot 10^{-5}$	2.44673	85424.65534
7	$4.42494 \cdot 10^{-7}$	220.45219	109830136302.9294

En calcular els valors de k_j per a $n = 2, 3, 4$ i per a $j = 2, 3, \ldots, 7$ (ja que disposavem de 8 iterats) veiem com els valors de k_j més estables són per $n = 2$, concloent així que el mètode té ordre de convergència 2.

3 Exercici 3

En aquest apartat aplicarem el mètode de *Halley* per tal de trobar una de les arrels de $g(x)$ i comprovarem que aquest mètode té ordre de convergència cúbic, és a dir 3. El mètode de *Halley* és un mètode iteratiu definit per

$$x_{k+1} = x_k - \frac{2g(x_k)g'(x_k)}{2(g'(x_k)^2 - g(x_k)g''(x_k))} \tag{5}$$

Treballant en precisió doble i amb $x_0 = 6$, igual que a l'exercici anterior, en 3 iteracions obtenim els mateixos resultats, és a dir, $x_3 = x_4 = 7.413302725857898$ i també $g(x_3) = 0 \pm 10^{-15}$.

De nou, per calcular l'ordre de convergència utilitzem la fórmula 4, però ara, com que volem veure que el mètode té ordre 3, farem servir $n = 2, 3, 4$. Fem servir també el valor x_0 per tal d'obtenir més valors de k_j ja que sinó només tindríem un valor i no podríem estimar de cap manera el seu comportament. A la taula següent es presenten els resultats, provinents del programa `p3.c`.

Valor de j	n=2	n=3	n=4
1	0.02462	0.01801	0.01317
2	0.00057	0.01243	0.27015

Veiem, doncs, com per $n = 2$ el valor k_j decreix, per $n = 4$, creix i per $n = 3$ manté un valor estable. Per tant, podem concloure que l'ordre de convergència és cúbic. En comparar amb el nombre d'iterats en l'exercici 1 on s'utilitza el mètode de Newton i amb l'exercici 2 on s'utilitza un mètode també quadràtic, observem com el fet de ser cúbic optimitza notablement el nombre d'iteracions (mantenint la llavor constant).

4 Exercici 4

Si es defineix la iteració $p_k = \frac{2a_k^2}{s_k}$ on

$$a_k = \frac{a_{k-1} + b_{k-1}}{2}, \quad b_k = \sqrt{a_{k-1}b_{k-1}}, \quad c_k = a_k^2 - b_k^2 \quad \text{i} \quad s_k = s_{k-1} - 2^k c_k$$

amb condicions inicials $a_0 = 1$, $b_0 = \frac{1}{\sqrt{2}}$ i $s_0 = \frac{1}{2}$ per a $k = 0, 1, \ldots$ analíticament aquesta convergeix a π. Emprant el mètode per a determinar l'ordre de convergència de l'exercici 2 apartat (a), mitjançant el programa `p4.c` es pot concloure que l'ordre és 2 ja que els quocients corresponents (denominador al quadrat) són del mateix ordre (10^{-2}). Al mateix programa es determinen les iteracions (dutes a terme per ordinador) necessàries per que la successió comenci a divergir (aquest fet s'explica posteriorment). Les iteracions necessàries per que l'error absolut comenci a crèixer són 4 per al cas en precisió simple i 5 en doble. Aquests resultats es poden observar a `p4.c`. En aquest mateix programa es pot veure que l'algoritme aproxima π en 6 xifres decimals (3.1415977, la part decimal és correcta en 5 xifres) en *float* i en 13 xifres decimals (3.141592653589831) en *double*.

El fet que a partir d'un cert índex k_0 el terme p_k s'allunyi de π és degut que per la precisió limitada, s'arriba que $a_k = a_{k_0}$, $b_k = b_{k_0}$ i $c_k = c_{k_0}$ per a tot $k > k_0$ i l'únic terme que varia a partir de k_0 és s_k degut al factor 2^k que no es manté constant en les iteracions. D'aquesta manera $p_k = \frac{C}{s_k}$ on $C = 2a_{k_0}^2$ si $k > k_0$. Observem que (s_k) és una successió decreixent cap a $-\infty$, cosa que provoca la desviació de p_k. Dit fet queda retratat a la primera part de `p4.c`. Per a major claredat s'adjunta el següent *plot*:

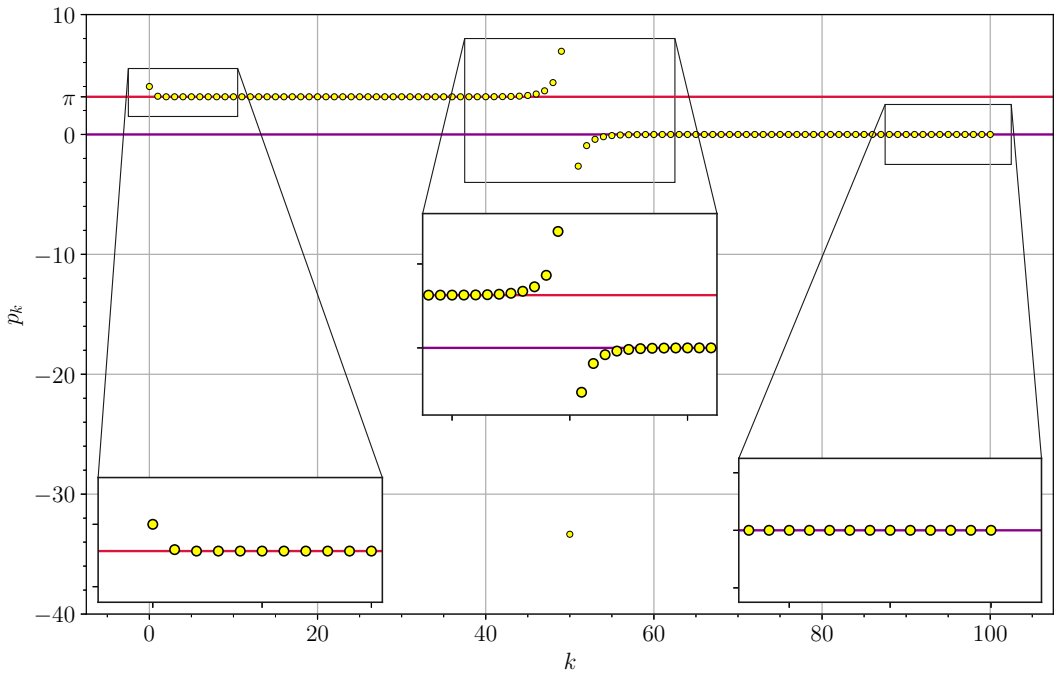

Figura 1: Il·lustració dels valors de la successió p_k en funció de k, calculada amb `C`. Es pot observar que els primers termes s'acosten a π, mentre que al voltant de la iteració número 50 pren valors llunyans i finalment convergeix cap a 0.