Mètodes Numèrics
Pràctica 2: Zeros de funcions

Raquel Garcia, Arnau Mas

11 de Març 2018

Problema 1

Considerem l'equació polinòmica

$$x^3 = x + 40. \tag{1}$$

UtIlitzant les fórmules de Cardano trobem l'arrel α que ve donada per:

$$\alpha = \left(20 + \frac{1}{9}\sqrt{32397}\right)^{1/3} + \left(20 - \frac{1}{9}\sqrt{32397}\right)^{1/3} \tag{2}$$

Tot i que ens dóna l'arrel exacta, aquesta expressió no és bona des del punt de vista numèric, ja que al segon terme hi ha una resta que produeix cancelació. El programa `prob1a.c` avalua aquesta expressió en doble i en simple precisió. En precisió doble obtenim $\alpha = 3.517\,393\,514\,052\,852$ i en precisió simple $\alpha = 3.517\,384\,77$, mentre que el resultat exacte amb 15 decimals és $\alpha = 3.517\,393\,514\,052\,818$. Per tant en precisió doble, l'error relatiu que s'ha produït ha sigut $\varepsilon_d = 9.67 \times 10^{-15}$, i en simple $\varepsilon_s = 2.51 \times 10^{-6}$. Ara aplicarem la fórmula de propagació de l'error relatiu al segon terme de (2) (sense l'arrel cúbica) per a provar d'estimar aquest error:

$$\varepsilon_r\left(20 - \frac{1}{9}\sqrt{32397}\right) = \frac{20 + \frac{1}{9}\sqrt{32397}}{20 - \frac{1}{9}\sqrt{32397}}\epsilon_r \tag{3}$$

On ϵ_r és l'error relatiu que suposarem prové només de l'expressió en punt flotant i que és el mateix per als dos sumands, de l'ordre de 10^{-17} en precisió doble, i de 10^{-10} en precisió simple. D'aquesta manera obtenim una estimació de $\varepsilon_r(\alpha) \sim 10^{-13}$ en precisió doble i $\varepsilon_r(\alpha) \sim 10^{-6}$ en precisió simple, on també hem suposat que la resta d'operacions que es realitzen en l'avaluació d'α no modifiquen de manera significativa l'ordre d'aquests errors relatius.

A continuació utilitzarem el mètode de Newton per a resoldre (1), amb $f(x) = x^3 - x - 40$, i com a punt inicial $x_0 = 2$. El programa `prob1b_do.c` executa el mètode en precisió doble, i `prob1b_fl.c` l'executa en precisió simple. En doble obtenim $\alpha = 3.517\,393\,514\,052\,818$ després de 7 iteracions, i en simple obtenim $\alpha = 3.517\,393\,59$ després de 5 iteracions. Si considerem el mètode de Newton com un mètode del punt fix amb funció d'iteració

$$g(x) = x - \frac{x^3 - x - 40}{6x^2 - 1} \tag{4}$$

tenim que $|g'(2)| \approx 3.37 > 1$, per tant per a aquest x_0 no tenim clar si convergirà, ni podem fer una estimació a priori del nombre d'iteracions necessàries. El valor que obtenim després de fer una iteració del mètode de Newton és $x_1 \approx 5.091$, per a aquest valor $|g'(x_1)| \approx 0.45$, que és molt millor que x_0 i ens permet estimar a priori que el nombre d'iteracions necessàries serà com a màxim de

$$n \sim \left(\frac{\log(\varepsilon(1 - 0.45)/|5.091 - 2|)}{\log(0.45)} \right) + 1, \tag{5}$$

on ε és la fita de l'error que volem aconseguir. Per a tenir 15 decimals correctes, és a dir $\varepsilon < 10^{-15}$ a priori necessitem $n \sim 47$, i per a tenir 8 decimals, $\varepsilon < 10^{-8}$ a priori necessitem $n \sim 27$ iteracions. Aquestes fites són tan elevades degut a que el punt $x_0 = 2$ és un punt molt dolent on començar.

Considerem ara l'equació polinòmica

$$x^3 = x + 400. \tag{6}$$

Si escrivim aquesta equació com a $p(x) = x^3 - x - 400 = 0$, veiem que només hi ha un canvi de signe en els seus coeficients, i per tant per la regla dels signes de Descartes tenim que aquest polinomi té una única arrel positiva. Com que $p(2) = -394 < 0$, $p(8) = 104 > 0$ i $p(x)$ és continu, pel teorema de Bolzano tenim que aquesta arrel β es troba a l'interval $[2, 8]$. La fórmula de Cardano que obtenim per a aquesta arrel és

$$\beta = \left(200 + \frac{1}{9}\sqrt{3239997}\right)^{1/3} \left(200 - \frac{1}{9}\sqrt{3239997}\right)^{1/3}. \tag{7}$$

El programa `prob1c.c` avalua β en precisió doble. Obtenim $\beta = 7.413\,302\,725\,859\,884$, però el valor veritable amb 15 xifres decimals és $\beta = 7.413\,302\,725\,857\,898$, per tant es produeix un error absolut $\varepsilon_a(\beta) = 1.986 \times 10^{-12}$. Aquest error prové fonamentalment de la cancelació que es produeix al segon terme de (7).

Fent servir el programa `prob1c123.c` obtenim 15 decimals correctes de β, treballant amb precisió doble, aplicant el mètode de la bisecció i el mètode de la secant partint de l'interval $[2, 8]$, i el mètode de Newton amb $x_0 = 2$.

Amb el mètode de la bisecció hem necessitat 50 iteracions per a trobar β, amb el de la secant hem necessitat 8 iteracions, mentre que amb el de Newton n'hem necessitat 10. Observem que el mètode de la bisecció és molt lent comparat amb els altres dos, però el mètode de Newton i el de la secant tampoc han sigut especialment ràpids degut a que el punt $x_0 = 2$ es troba prou allunyat de l'arrel.

Problema 2

El programa `prob2a.c` conté codi que executa la iteració descrita al problema 2. Si comencem la iteració a $x_0 = 7.5$ aleshores obtenim l'arrel x^* amb 15 xifres decimals correctes després de 4 iteracions. Obtenim $x^* = 7.413\,302\,725\,857\,898$, que és coherent amb els resultats del problema anterior.

Per estudiar l'ordre de convergència aproximarem $|x_k - x^*|$ per $e_k = |x_k - x_{k+1}|$. El codi de `prob2a.c` també realitza el càlcul de e_k/e_{k-1}, e_k/e_{k-1}^2 i e_k/e_{k-1}^3. Com que després de 4 iteracions ja hem obtingut x^* amb més precisió que la que permet el format `double` e_5 és 0. Per tant el quocient e_6/e_5 dóna `nan` com a resultat. Tot i això, només amb 4 iteracions ja podem dir que l'ordre de convergència és quadràtic.

Taula 1: Anàlisi de l'ordre convergència amb pivot inicial 7.5

k	$\frac{e_k}{e_{k-1}}$	$\frac{e_k}{(e_{k-1})^2}$	$\frac{e_k}{(e_{k-1})^3}$
1	0.0114	0.1336	1.5993
2	0.0235	23.9687	24 428
3	0.0054	23.3059	10^6
4	10^{-7}	23.0688	10^9

Efectivament, tal i com mostra la taula 1, l'únic quocient que es manté estable és $\frac{e_k}{(e_{k-1})^2}$. A més també observem que $\frac{e_k}{(e_{k-1})^3} \to \infty$ i $\frac{e_k}{e_{k-1}} \to 0$ la qual cosa ens porta a concloure que l'ordre de convergència és exactament quadràtic.

Problema 3

Considerem l'equació $f(x) = 0$, amb $f(x)$ continuament derivable. Si x^* és una arrel simple, de manera que compleix $f(x^*) = 0$ i $f'(x) \neq 0$ en un entorn de x^*, aleshores podem utilitzar el mètode de Halley que consisteix en la iteració

$$x_{k+1} = x_k - \frac{2f(x_k)f'(x_k)}{2(f'(x_k))^2 - f(x_k)f''(x_k)} \tag{8}$$

per a aproximar x^*.

Al programa `prob3.c` hem utilitzat aquest mètode per a calcular l'arrel de $f(x) = x^3 - x - 400$ amb 15 decimals correctes. Hem obtingut $x^* = 7.413\,302\,725\,857\,898$ en 5 iteracions, partint de $x_0 = 2$.

Per a comprovar que aquest mètode té ordre de convergència 3 considerem $e_k = |x_k - x_{k-1}|$ i estudiem els quocients $\frac{e_k}{(e_{k-1})^3}$. Els resultats es mostren a la taula 2.

Taula 2: Anàlisi de l'ordre de convergència del mètode de Halley

k	x_k	$\frac{e_k}{(e_{k-1})^3}$
1	3.744	—
2	6.305	0.4828
3	7.392	0.0647
4	7.413	0.0163
5	7.413	0.0124

Veiem que $e_k \sim (e_{k-1})^3$, de manera que el mètode de Halley té ordre de convergència cúbic.

Problema 4

El programa `prob4.c` conté codi que executa la iteració proposada. Com que sabem a quin valor ha de convergir la iteració podem analitzar directament l'error absolut $e_k = |p_k - \pi|$. Aquest decreix fins a la quarta iteració, però es compleix $e_5 > e_4$. A partir d'aquí perdem la convergència ja que l'error absolut comença a creixer. Si analitzem l'algoritme sembla que l'error ha d'aparèixer a c_k o bé a s_k, ja que per calcular-les s'ha de fer una resta, la qual cosa pot introduir errors de cancel·lació. Quan observem el valor de c_k i s_k a cada iteració veiem que a partir de la quarta iteració c_k és 0 i que, per tant, s_k es manté constant. Això té sentit ja que de fet $c_k \to 0$.

La desigualtat aritmètica-geomètrica ens dóna $a_k > b_k$. Per tant $a_{k+1} = \frac{a_k+b_k}{2} < a_k$ i tenim que la successió a_k és decreixent i fitada inferiorment —per b_0, per exemple— i per tant convergent. Similarment, $b_{k+1} = \sqrt{b_k a_k} > b_k$ ja que $a_k > b_k$. Per tant, com que b_k és una successió creixent i fitada superiorment—per exemple per a_0— aleshores és convergent. Posem $a_k \to \alpha$ i $b_k \to \beta$. Aleshores $\alpha \geq \beta$. I si $\alpha > \beta$ aleshores $\alpha - \beta > 0$. En particular, existeix $K \in \mathbb{N}$ tal que $a_K - \alpha < \frac{\alpha-\beta}{2}$ i tal que $\beta - b_K < \frac{\alpha-\beta}{2}$. Per tant $a_K + b_K < \alpha + \beta$ i $a_{K+1} = \frac{a_K+b_K}{2} < \frac{\alpha+\beta}{2} < \alpha$, que és una contradicció ja que sabem que a_k decreix cap a α. Per tant $\alpha = \beta$, que implica $c_k \to 0$. En particular, a partir d'una iteració N tindrem que $c_k < \epsilon$ per $k \geq N$ amb ϵ l'èpsilon màquina. I per tant $\mathsf{fl}(s_{k+1}) = \mathsf{fl}(s_k)$ a partir d'aquesta iteració. Això és precisament el que observem a partir de la quarta iteració.

Pel que fa a l'ordre de convergència, si apliquem el mateix mètode que al problema 2, amb l'avantatge de què coneixem el límit de la iteració i per tant $e_k = |p_k - \pi|$ trobem que només el quocient $e_k / (e_{k+1})^2$ es manté estable, tal i com es comprova a la taula 3 i concloem que la convergència és quadràtica.

Taula 3: Anàlisi de l'ordre convergència

k	$\frac{e_k}{e_{k-1}}$	$\frac{e_k}{(e_{k-1})^2}$	$\frac{e_k}{(e_{k-1})^3}$
1	0.0537	0.0625	0.0729
2	0.0019	0.0413	0.8957
3	10^{-6}	0.0398	454

Problema 5

El nostre objectiu és obtenir una aproximació de l'arrel quadrada d'un nombre utilitzant l'expressió

$$\sqrt{1+x} = f(x)\sqrt{1+g(x)}$$

on g és un infinitèsim d'ordre més petit que x per a x tendint a 0. Si triem $f(x)$ com una aproximació de $\sqrt{1+x}$ aleshores es pot calcular $g(x)$ com

$$g(x) = \frac{1+x}{f(x)^2} - 1. \tag{9}$$

Triarem $f(x)$ com una funció racional $p(x)/q(x)$ de manera que p i q tenen el mateix grau i el seu desenvolupament de Taylor coincideix amb el de $\sqrt{1+x}$ fins a un cert grau.

Si volem que p i q siguin lineals, aleshores el desenvolupament de $p(x) - q(x)\sqrt{1+x}$ ha de tenir els tres primers termes nuls, i es diu que f és una aproximació de Padé de la funció $\sqrt{1+x}$. Posem $p(x) = a_1x + a_0$ i $q(x) = b_1x + b_0$ de manera que s'ha de complir

$$a_1x + a_0 - \left(1 + \frac{1}{2}x - \frac{1}{8}x^2\right)(b_1x + b_0) = 0, \tag{10}$$

i trobem

$$f_1(x) = \frac{3x+4}{x+4}.$$

Per a aquesta $f_1(x)$ tenim que $g_1(x)$ és:

$$g_1(x) = \frac{x^3}{9x^2+24x+16} = \frac{x^3}{(3x+4)^2}. \tag{11}$$

Considerem ara la successió $a_0 = x$, $a_{n+1} = g(a_n)$ i $b_n = f(a_n)$. Volem comprovar que

$$\sqrt{1+x} = \left(\prod_{j=0}^{k} b_j\right)\sqrt{1+a_{k+1}}. \tag{12}$$

Ho veurem per inducció sobre k. Per a $k = 0$ tenim

$$\sqrt{1+x} = f(x)\sqrt{1+g(x)}$$

que és cert ja que és l'hipòtesi inicial. Per tot $k \in \mathbb{N}$ tenim que

$$a_{k+1} = \frac{1+a_k}{b_{k+1}^2} - 1 \tag{13}$$

a partir de 9. Aleshores, fent servir la hipòtesi d'inducció tenim

$$\left(\prod_{j=0}^{k+1} b_j\right)\sqrt{1+a_{k+1}} = \left(\prod_{j=0}^{k+1} b_j\right)\sqrt{\frac{1+a_k}{b_k^2}} = \left(\prod_{j=0}^{k} b_j\right)\sqrt{1+a_k} = \sqrt{1+x},$$

que conclou la prova.

A partir de la forma explícita per a g_1 que hem trobat a 11, havent fet la tria de f_1 com a quocient de polinomis de grau 1 és clar que $g_1(x) = O(x)$ ja que $g_1(x)/x \to \frac{1}{9}$ quan $x \to 0$. A més tenim

$$g_3'(x) = \frac{3x^2(3x+4)^2 - 6x^3(3x+4)}{(3x+4)^4} = \frac{3x^3 + 12x^2}{(3x+4)^3} < \frac{1}{9}$$

per $x > 0$. Per tant, pel teorema del valor mitjà, per $x > 0$ g_3 és Lipschitz amb constant $\frac{1}{9}$ i per tant contractiva. A més, com que $g_3(0) = 0$, $|g_3(x)| < |x|$ per $x > 0$.

Si volem que $f(x)$ sigui quocient de polinomis de grau 3, podem fer una cosa semblant a (10), ja que sempre podrem aconseguir equacions lineals suficients. Nosaltres en aquest cas hem utilitzat SageMath per a trobar l'aproximació de Padé amb polinomis de tercer grau de $\sqrt{1+x}$, i hem obtingut

$$f_3(x) = \frac{7x^3 + 56x^2 + 112x + 64}{x^3 + 24x^2 + 80x + 64}.$$

Per a aquesta $f_3(x)$ tenim que $g_3(x)$ és:

$$g_3(x) = \frac{x^7}{49x^6 + 784x^5 + 4704x^4 + 13440x^3 + 19712x^2 + 14336x + 4096}. \tag{14}$$

A continuació volem comprovar la desigualtat

$$\left| \sqrt{1+x} - \prod_{j=0}^{k} b_j \right| \leq \frac{a_{k+1}}{2}\sqrt{1+x}. \tag{15}$$

Observem que aïllant $\prod b_j$ de (12) i substituïnt-lo en (15), obtenim que la desigualtat que volem provar és equivalent a

$$\left| 1 - \frac{1}{\sqrt{1+a_{k+1}}} \right| \leq \frac{a_{k+1}}{2}. \tag{16}$$

Podem eliminar el valor absolut ja que $\sqrt{1+a_k} > 1$ i per tant $1/\sqrt{1+a_{k+1}} < 1$. Per tant la desigualtat és equivalent a

$$1 - \frac{a_{k+1}}{2} \leq \frac{1}{\sqrt{1+a_{k+1}}}. \tag{17}$$

Com que el desenvolupament de Taylor de $1/\sqrt{1+x}$ fins a grau 2 al voltant de 0 és

$$\frac{1}{\sqrt{1+x}} = 1 - \frac{x}{2} + \frac{3}{8}x^2 + O(x^3). \tag{18}$$

Per tant, el residu de Lagrange és $\frac{3}{8(1+c)^{5/2}}x^2$ amb $c \in [0, a_{k+1}]$, i per tant sempre positiu.

Per a $f_3(x)$, el programa `probextrag.c` calcula els valors que pren la successió $a_{n+1} = g(a_n)$ amb $a_0 = 1$, d'aquesta manera obtenim que $a_3 = 4.537\,551 \times 10^{-261}$, i per tant observem que es verificarà la desigualtat

$$|\sqrt{2} - b_0 b_1 b_2| \leq 4.537551 \times 10^{-261}\frac{\sqrt{2}}{2} \tag{19}$$

La conclusió que podem prendre d'aquest fet, és que $\prod_{j=0}^{k} b_j$ amb $b_n = f(a_n)$ és una molt bona aproximació de l'arrel quadrada d'un nombre a_0+1, ja que per ser $g(x)$ contractiva i tenir com a únic punt fix $g(0) = 0$, la successió a_n tendeix a 0 per a $n \to \infty$, i per tant $\prod_{j=0}^{k} b_j \to \sqrt{1+a_0}$. De fet veiem que per a valors petits de a_0 aquesta convergència és molt ràpida. Amb aquestes succesions podriem dissenyar un programa per a calcular arrels quadrades de nombres majors que 1 ($a_0 \geq 0$): només cal tenir guardades les funcions $f(x)$ i $g(x)$ i executar les succesions anteriors on només intervenen productes, sumes i quocients, que són operacions segures des del punt de vista numèric.

Al programa `probextrah.c` hem programat aquestes succesions, utilitzant f com a quocient de polinomis d'ordre 3 i també d'ordre 1. Per exemple al calcular $\sqrt{2}$ hem obtingut $\sqrt{2} = 1.414\,214\,553\,395\,256$ en 2 iteracions amb f_3. I fent servir f_1 obtenim $\sqrt{2} = 1.414\,213\,562\,373\,095$ en 3 iteracions. Observem que l'error és major utilitzant polinomis de grau 3! En teoria $f_3(x)$ hauria de ser molt més efectiva, ja que l'error mesurat amb $g_3(x)$ tendeix a zero molt més ràpidament. Però de fet, aquest és el problema. Si ens fixem en $f_3(x)$, veiem que per a x molt petita ens donarà $64/64 = 1$ com a resultat, ja que si operem amb precisió doble, $g(a_k)$ serà menor que l'èpsilon màquina molt depressa. Així veiem que amb $f_3(x)$ tenim $a_1 = 1.890\,824 \times 10^{-5}$ i $a_2 = 2.109\,448 \times 10^{-37}$, i només és útil a_1 ja que a_2 ja és menor que l'èpsilon màquina. En canvi, amb $f_1(x)$ tenim $a_1 = 2.040\,816 \times 10^{-2}$, $a_2 = 5.153\,446 \times 10^{-7}$ i $a_3 = 8.554\,074 \times 10^{-21}$. Ara a_3 és menor que l'èpsilon màquina i per tant podem fer una iteracio més que amb grau 3 i amb una fita per l'error inferior.