MÈTODES NUMÈRICS

Informe pràctica 1A - QP1718

Grau en Estadística. UB-UPC

Laura Julià Melis
NIUB: 16810883
6 d'abril de 2017

Índex

1.1 Algoritmes

Considereu el següent algoritme per calcular el nombre π : ***"Genereu n parelles de nombres al·leatoris $\left\{ (x_k, y_k) \right\}_{k=1 \div n}$ de l'interval [0, 1]. Compteu el nombre m dels que es troben dins del primer quadrant del cercle unitat. Resulta que π és el límit de la successió $\pi_n = \frac{4m}{n}$."***

1. **Construïu un programa en `Matlab` per calcular el termes la successió π_n.**

Arxiu MATLAB: Exercici1_Apartat1.m

2. **Feu un joc de proves per a valors de n = 5ᵏ , per exemple 1 ≤ k ≤ 15. El resultat ha d'ésser una taula de la forma:**

S'han realitzat proves de 5^k per a 1 ≤ k ≤ 10:

n	***Valor π_n***	***Error absolut***	***Error relatiu***
5	2.4	0.741592653589793	0.236056273158902
25	3.2	0.0584073464102071	0.0185916357881302
125	3.264	0.122407346410207	0.0389634685038927
625	3.1232	0.0183926535897929	0.00585456347078487
3125	3.15392	0.0123273464102067	0.00392391623278106
15625	3.143424	0.00183134641020688	0.000582935667396046
78125	3.143168	0.00157534641020707	0.000501448336533055
390625	3.137792	0.00380065358979298	0.00120978561159102
1953125	3.140204544	0.00138810958979318	0.000441849005537697
9765625	3.1426387968	0.00104614321020691	0.000332997726172906

Arxiu MATLAB: Exercici1_Apartat2.m

3. **A partir dels valors de la taula, l'exactitud creix o decreix en funció de n? Quants decimals iguals obteniu? Quantes xifres significatives obteniu? Els resultats del teu càlcul es corresponen amb el concepte *límit d'una successió*? Raona totes les teves respostes**

Amb els valors de la taula es pot observar com, a mesura que la n augmenta, la aproximació de π_n a π és més precisa: la exactitud va creixent en funció de n. Com veiem en la segona columna, el valor estimat comença essent 2,4 (valor molt allunyat del vertader valor de π) i cada cop s'hi va apropant més fins que s'obté el valor 3.1426 (valor més aprop de π=3.1415926…).

S'obtenen 2 decimals iguals i t = 3 xifres significatives correctes, ja que t és el nombre natural més gran tal que:

$$\frac{|x - \tilde{x}|}{|x|} < 0.5 \cdot 10^{-t} \longrightarrow 0.00033 < 0.5 \cdot 10^{-3}.$$

Els resultats del càlcul sí es corresponen amb el concepte de límit d'una succesió. Per explicar-ho, primer cal definir aquest concepte. El límit d'una successió és la quantitat que posa límit a la successió de valors, és a dir, el valor al que tendeixen els termes de la successió quan n pren valors molt grans (tendeix a infinit).

Ara, podem veure que les parelles de nombres aleatoris $\left\{ (x_k, y_k) \right\}_{k=1 \div n}$ són una successió numèrica, el terme general de la qual té limit π (o sigui que tendeix a π) quan n tendeix a ∞. I ho podem confirmar perquè, com ja hem afirmat abans, com major és n en la nostra taula, més s'apropa el valor π_n a π.

1.2 Errors de cancel·lació

Es demana:

1. **Cerca documentació sobre l'ús de la regla de Horner per avaluar polinomis. Escriu un breu resum del que has entès (màxim 1/2 full). Dóna les teves fonts bibliogràfiques.**

La regla de Horner, anomenada així en honor al matemátic anglés Willian George Horner (1789-1837), és una algoritme que ens permet avaluar eficientment polinomis d'una manera monomial[1]. L'objectiu d'aquesta regla és trobar solucions aproximades al problema, a través d'una seqüència d'operacions algebraiques.

El mètode ens diu que donat el polonomi:

$$p(x) = a_0 + a_1x + a_2x^2 + a_3x^3 + \ldots + a_nx^n \qquad \text{on } a_0, \ldots, a_n \text{ són nombre reals,}$$

si es vol avaluar el polinomi en un valor donat x, que anomanem x_0, cal calcular:

$d_0 = a_0$
$d_k = a_k + d_{k-1}x_0$ per a tot $k = 1,...,n-1$
$d_n = P(x_0)$.

Així doncs, d_n és el valor de $P(x_0)$, el valor que s'estava buscant. Observar que amb aquest algoritme s'aconsegueix avaluar un polinomi amb només n sumes i n multiplicacions, el nombre mínim possible. Per aquest motiu es pot afirmar que la regla de Horner és òptima (quan x és una matriu no és òptima).

Bibliografía:

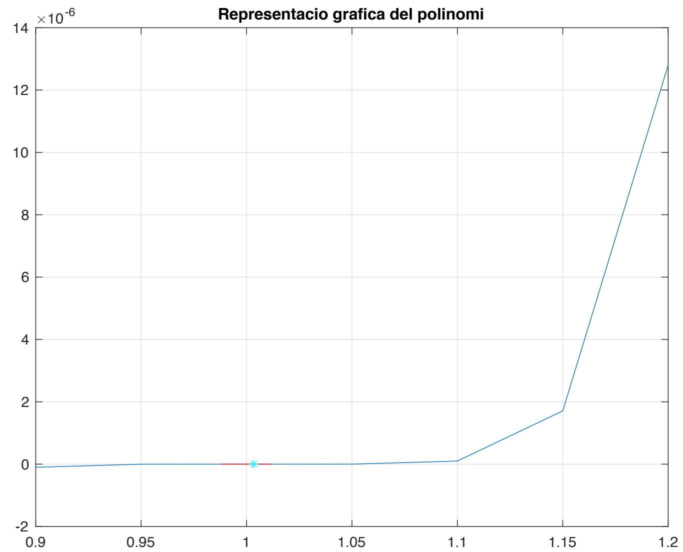
- https://es.wikipedia.org/wiki/Algoritmo_de_Horner
- http://www.ehu.eus/juancarlos.gorostizaga/mn11b/temas/horner.pdf

2. **Escriure una funció de `Matlab` que avalui el polinomi**

$$p(x) = x^7 - 7x^6 + 21x^5 - 35x^4 + 35x^3 - 21x^2 + 7x - 1$$

per a valors equiespaiats a l'interval [0.988, 1.012], prenent Δx = 0.00005. Representa gràficament el polinomi.

x_n	$f(x_n)$
0.988	-3.5527136788005e-14
0.98805	-4.44089209850063e-14
0.9881	-4.17443857259059e-14
0.98815	-3.19744231092045e-14
0.9882	-2.22044604925031e-14
…	…
1.003	0
1.00305	-5.32907051820075e-15
1.0031	5.32907051820075e-15
1.00315	-3.5527136788005e-15
1.0032	-7.105427357601e-15
1.00325	-1.77635683940025e-15
…	…
1.01185	3.81916720471054e-14
1.0119	2.39808173319034e-14
1.01195	2.93098878501041e-14
1.012	4.61852778244065e-14



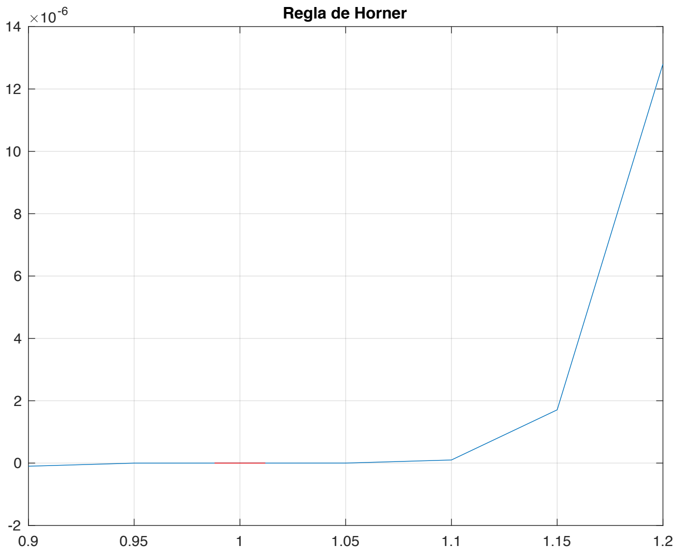
En la gràfica s'observa com la funció talla amb l'eix d'abscices en el punt 1.003 (*`fzero(f,0)= 1.00334874707265`*, representat en color cian), punt que es troba entre l'interval donat *[0.988, 1.012]*. En la taula d'avaluacions del polinomi es confirma aquest fet. Cal mencionar que, numèricament, s'han obtingut més valors x_n que han fet 0 la funció; això sigui segurament a causa de la precisió de `MATLAB`.

Arxiu MATLAB: Exercici2_Apartat2.m

[1] Un monomi és un polinomi d'un únic terme.

3. Escriure una funció de `Matlab` que avalui el polinomi *p(x)* fent ús de la regla de Horner. Per a valors equiespaiats a l'interval [0.988, 1.012], prenent Δx = 0.00005 representa gràficament els valors obtinguts.

n	x_n	$f(x_n)$
1	0.988	3.68594044175552e-14
2	0.98805	3.36397576461422e-14
3	0.9881	3.41948691584548e-14
4	0.98815	3.25295346215171e-14
5	0.9882	3.05311331771918e-14
6	0.98825	3.39728245535298e-14
…	…	…
478	1.01185	-3.37507799486048e-14
479	1.0119	-3.30846461338297e-14
480	1.01195	-3.37507799486048e-14
481	1.012	-3.5527136788005e-14

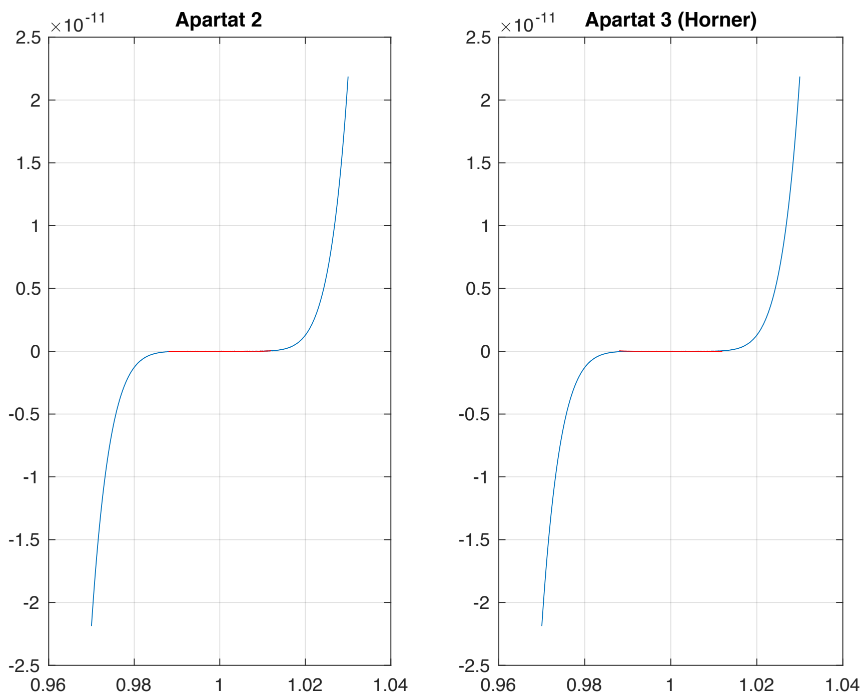


<u>Arxiu MATLAB</u>: Exercici2_Apartat3_Principal.m i Exercici2_Apartat3_Funció.m

4. Compareu les gràfiques obtingudes en els dos apartats anteriors amb la gràfica del polinomi $(x - 1)^7$ en el mateix domini. Quines semblances i quines diferències observeu? Raona totes les teves respostes.

<u>Interpretació de les gràfiques:</u> en color blau, la representació gràfica del polinomi de 0.9 a 1.2. Les franges en color vermell representen els valors obtinguts en les avaluacions del polinomi, de 0.988 a 1.012.

S'han realitzat les dues gràfiques dels apartats anteriors però només des de 0.97 a 1.03 per poder veure-ho amb més claredat. S'observa com ambdues es veuen quasi bé iguals. Però, si veiem els valors numèrics de les iteracions dels algoritmes, els valors del polinomi en cada x_n són diferents.



<u>Arxiu MATLAB</u>: Exercici2_Apartat4.m

1.3 Error de truncament

El nombre π és la suma de la sèrie:

$$\pi = \sum_{n=0}^{\infty} 16^{-n} \left(\frac{4}{8n+1} - \frac{2}{8n+4} - \frac{1}{8n+5} - \frac{1}{8n+6} \right). \tag{1.1}$$

En aquest cas es pot calcular una aproximació de π sumant fins al terme N-èssim, per a un n prou gran.

Es demana:

1. **Escriure una *function* `MATLAB` per a calcular les sumes parcials finites**

$$S_N = \sum_{n=0}^{N} 16^{-n} \left(\frac{4}{8n+1} - \frac{2}{8n+4} - \frac{1}{8n+5} - \frac{1}{8n+6} \right). \tag{1.2}$$

Arxiu MATLAB: Exercici3_Apartat1_Principal.m i Exercici3_Apartat1_Funcio.m

2. **Per a quin valor de *N* s'obté el valor de π amb la precisió de `MATLAB`.**

Volem saber quantes iteracions són necessàries per tal d'obtenir una aproximació del valor pi amb la funció creada en l'apartat 1 que sigui igual de precisa que el valor pi de MATLAB (3.14159265358979).

A través del codi de MATLAB afirmem que a partir d'una N=10 obtendrem el valor de pi amb la precisió de MATLAB.

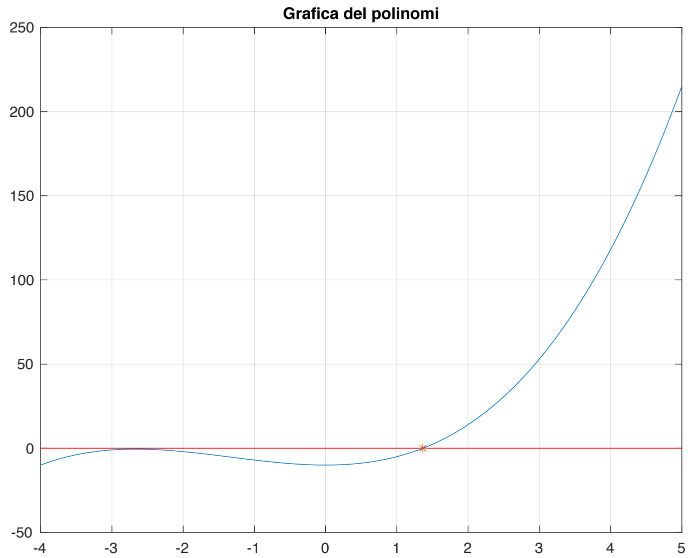
Arxiu MATLAB: Exercici3_Apartat2.m

1.4 Solucions d'equacions no lineals

Calcular valors aproximats de l'arrel real de l'equació $x^3 + 4x^2 - 10 = 0$.

Es demana:

1. **Digueu quantes arrels té $f(x) = 0$ per $f(x) = x^3 + 4x^2 - 10$ i justifiqueu-ho. Utilitzeu el teorema de Bolzano per a determinar intervals que separin les arrels.**



El polinomi $x^3 + 4x^2 - 10 = 0$ té una única arrel, la funció f(x) només talla un cop amb l'eix d'absices. En el gràfic s'observa que això ocorre en x=1.3652.

El teorema de Bolzano diu:

Sigui la funció f(x) contínua definida en un interval [a,b], si es compleix que f(a)·f(b) < 0, llavors existeix un punt c que perant a l'interval tal que f(c)=0.

Així, com que $f(x) = x^3 + 4x^2 - 10$ és contínua en $[1, 2]$ i f(1)·f(2) < 0 (ja que f(1)=-5 i f(2)=14) podem confirmar que existeix un punt c (c=1.3652) tal que f(c)=0 (és arrel), que pertany a $[1, 2]$.

<u>*Arxiu MATLAB*</u>: Exercici4_Apartat1.m

2. **Calculeu la arrel real com a (mínim 6 decimals correctes) per cadascun dels següents mètodes:**

   (a) Mètode de la bisecció. Presenteu els resultats en una taula.

<u>Taula de resultats:</u>

n	***a_n***	***b_n***	***x_n***	***$f(x_n)$***	***$(b_n - a_n)/2$***
1.000000000000000	0	0	1.500000000000000	2.375000000000000	1.000000000000000
2.000000000000000	1.000000000000000	2.000000000000000	0	-10.000000000000000	0
3.000000000000000	1.000000000000000	1.500000000000000	1.250000000000000	-1.796875000000000	0.500000000000000
4.000000000000000	0	1.500000000000000	0.750000000000000	-7.328125000000000	1.500000000000000
5.000000000000000	1.250000000000000	1.500000000000000	1.375000000000000	0.162109375000000	0.250000000000000
6.000000000000000	0.750000000000000	1.500000000000000	1.125000000000000	-3.513671875000000	0.750000000000000
7.000000000000000	0.750000000000000	1.375000000000000	1.062500000000000	-4.284912109375000	0.625000000000000
8.000000000000000	1.125000000000000	1.375000000000000	1.250000000000000	-1.796875000000000	0.250000000000000
9.000000000000000	1.062500000000000	1.375000000000000	1.218750000000000	-2.248321533203125	0.312500000000000
10.000000000000000	1.250000000000000	1.375000000000000	1.312500000000000	-0.848388671875000	0.125000000000000
11.000000000000000	1.218750000000000	1.375000000000000	1.296875000000000	-1.091266632080078	0.156250000000000
12.000000000000000	1.312500000000000	1.375000000000000	1.343750000000000	-0.350982666015625	0.062500000000000
13.000000000000000	1.296875000000000	1.375000000000000	1.335937500000000	-0.476797580718994	0.078125000000000
14.000000000000000	1.343750000000000	1.375000000000000	1.359375000000000	-0.096408843994141	0.031250000000000
15.000000000000000	1.335937500000000	1.375000000000000	1.355468750000000	-0.160421192646027	0.039062500000000
16.000000000000000	1.359375000000000	1.375000000000000	1.367187500000000	0.032355785369873	0.015625000000000
17.000000000000000	1.355468750000000	1.375000000000000	1.365234375000000	0.000072024762630	0.019531250000000
18.000000000000000	1.355468750000000	1.367187500000000	1.361328125000000	-0.064310245215893	0.011718750000000
19.000000000000000	1.355468750000000	1.365234375000000	1.360351562500000	-0.080367251299322	0.009765625000000

20.000000000000000	1.361328125000000	1.365234375000000	1.363281250000000	-0.032149970531464	0.003906250000000
21.000000000000000	1.360351562500000	1.365234375000000	1.362792968750000	-0.040195823763497	0.004882812500000
22.000000000000000	1.363281250000000	1.365234375000000	1.364257812500000	-0.016046690754592	0.001953125000000
23.000000000000000	1.362792968750000	1.365234375000000	1.364013671875000	-0.020073957581189	0.002441406250000
24.000000000000000	1.364257812500000	1.365234375000000	1.364746093750000	-0.007989262812771	0.000976562500000
25.000000000000000	1.364013671875000	1.365234375000000	1.364624023437500	-0.010003981611590	0.001220703125000
26.000000000000000	1.364746093750000	1.365234375000000	1.364990234375000	-0.003959101522923	0.000488281250000
27.000000000000000	1.364624023437500	1.365234375000000	1.364929199218750	-0.004966732310322	0.000610351562500
28.000000000000000	1.364990234375000	1.365234375000000	1.365112304687500	-0.001943659010067	0.000244140625000
29.000000000000000	1.364929199218750	1.365234375000000	1.365081787109375	-0.002447542255965	0.000305175781250
30.000000000000000	1.365112304687500	1.365234375000000	1.365173339843750	-0.000935847281880	0.000122070312500
31.000000000000000	1.365081787109375	1.365234375000000	1.365158081054688	-0.001187805868529	0.000152587890625
32.000000000000000	1.365173339843750	1.365234375000000	1.365203857421875	-0.000431918799251	0.000061035156250
33.000000000000000	1.365158081054688	1.365234375000000	1.365196228027344	-0.000557902333581	0.000076293945312
34.000000000000000	1.365203857421875	1.365234375000000	1.365219116210938	-0.000179948903227	0.000030517578125
35.000000000000000	1.365196228027344	1.365234375000000	1.365215301513672	-0.000242941730654	0.000038146972656
36.000000000000000	1.365219116210938	1.365234375000000	1.365226745605469	-0.000053962541529	0.000015258789062
37.000000000000000	1.365215301513672	1.365234375000000	1.365224838256836	-0.000085459220308	0.000019073486328
38.000000000000000	1.365226745605469	1.365234375000000	1.365230560302734	0.000009030992743	0.000007629394531
39.000000000000000	1.365224838256836	1.365234375000000	1.365229606628418	-0.000006717412914	0.000009536743164
40.000000000000000	1.365224838256836	1.365230560302734	1.365227699279785	-0.000038214180050	0.000005722045898
41.000000000000000	1.365229606628418	1.365230560302734	1.365230083465576	0.000001156788073	0.000000953674316
42.000000000000000	1.365227699279785	1.365230560302734	1.365229129791260	-0.000014591610221	0.000002861022949
43.000000000000000	1.365227699279785	1.365230083465576	1.365228891372681	-0.000018528707493	0.000002384185791
44.000000000000000	1.365229129791260	1.365230083465576	1.365229606628418	-0.000006717412914	0.000000953674316
45.000000000000000	1.365228891372681	1.365230083465576	1.365229487419128	-0.000008685962586	0.000001192092896
46.000000000000000	1.365229606628418	1.365230083465576	1.365229845046997	-0.000002780312879	0.000000476837158

L'arrel real és aproximadament 1.365230, l'interval inicial és $[1, 2]$ i el criteri d'aturada: $\eta = 0.5 \cdot 10^{-6}$.

Arxiu MATLAB: Exercici4_Apartat2a.m

(b) Mètode de la secant. Presenteu els resultats en una taula.

Taula de resultats:

n	***a_n***	***b_n***	***f(x_n)***
1	1	2	14
2	2	1.26315789473684	-1.60227438402099
3	1.26315789473684	1.33882783882784	-0.430364748004529
4	1.33882783882784	1.36661639471935	0.0229094307759521
5	1.36661639471935	1.36521190263186	-0.000299067919327101
6	1.36521190263186	1.36523000111086	-2.03168273316123e-07
7	1.36523000111086	1.36523001341421	1.80477854883065e-12

L'arrel real és aproximadament 1.365230, l'interval inicial és $[1, 2]$ i el criteri d'aturada: $\eta = 0.5 \cdot 10^{-6}$.

Arxiu MATLAB: Exercici4_Apartat2b.m

(c) **Mètode de Newton. Presenteu els resultats en una taula.**

Taula de resultats:

n	x_n	$f(x_n)$	$x_n - x_{n-1}$
1.000000000000000	1.000000000000000	-5.000000000000000	1.000000000000000
2.000000000000000	1.454545454545455	1.540195341848236	0.454545454545455
3.000000000000000	1.368900401069519	0.060719688639942	-0.085645053475936
4.000000000000000	1.365236600202116	0.000108770610424	-0.003663800867403
5.000000000000000	1.365230013435367	0.000000000351239	-0.000006586766749

L'arrel real és aproximadament 1.365230, el punt inicial és 1 i el criteri d'aturada: $\eta = 0.5 \cdot 10^{-6}$

Arxiu MATLAB: Exercici4_Apartat2c.m

3. Considereu els mètodes iteratius següents:

i) $x_{n+1} = x_n - x_n^3 - 4x_n^2 + 10,$

ii) $x_{n+1} = \dfrac{1}{2}\sqrt{10 - x_n^3},$

iii) $x_{n+1} = x_n - \left(\dfrac{x_n^3 + 4x_n^2 - 10}{3x_n^2 + 8x_n}\right).$

(a) Per cada un dels mètodes, *i), ii),* i *iii)*, demostreu la seva convergència/divergència del mètode a l'arrel positiva de l'equació $x^3 + 4x^2 - 10 = 0$. Busqueu un interval que asseguri la convergència del metode analitzat. ("a priori")

En l'apartat 1 ja hem pogut confirmar que l'equació $x^3 + 4x^2 - 10 = 0$ té una solució a l'interval [1,2].

i) Prenem $x_0 = 2$. Obtenem la següent taula de resultats:

x_n	$\| x_n - x_{n-1} \|$	$\| f(x_n) \|$
2	0	14
-12	14	1162
1150	1162	1526164990
-1526163840	1526164990	3.55470428040497e+27
3.55470428040497e+27	3.55470428040497e+27	4.49169678727241e+82
-4.49169678727241e+82	4.49169678727241e+82	9.06215099894595e+247
9.06215099894595e+247	9.06215099894595e+247	Inf
-Inf	Inf	NaN
NaN	NaN	NaN
NaN	NaN	NaN

Si observem x_n, no veiem repetició de nombres o de decimals. Mirant $| f(x_n) |$, podem confirmar que el mètode iteratiu és divergent.

ii) Prenem $x_0 = 2$. Obtenem la següent taula de resultats:

x_n	$\| x_n - x_{n-1} \|$	$\| f(x_n) \|$
2	0	14
0.707106781186548	1.29289321881345	7.64644660940673
1.5529364611444	0.845829679957849	3.39152627446434
1.25049193669355	0.302444524450842	1.78964780418111
1.41814739529461	0.167655458601051	0.896663977315489
1.33677823158967	0.081369163704937	0.463305166943456
1.37942101339216	0.0426427818024957	0.235974850103023
1.35786914673768	0.0215518666544865	0.121114684223397
1.36897307888722	0.0111039321495436	0.0619242113494121
1.36330709595476	0.0056659829324599	0.0317239757891628

Si observem x_n, veiem que hi ha molts decimals iguals. Mirant $| f(x_n) |$, es demostra la convergència del mètode iteratiu.

iii) Prenem $x_0 = 2$. Obtenem la següent taula de resultats:

x_n	$\| x_n - x_{n-1} \|$	$\| f(x_n) \|$
2	0	14
1.5	0.5	2.375
1.37333333333333	0.126666666666667	0.134345481481482
1.36526201487463	0.00807131845870668	0.000528461179515105
1.36523001391615	3.20009584799941e-05	8.29054869200263e-09
1.3652300134141	5.02049735118248e-10	0
1.3652300134141	0	0
1.3652300134141	0	0
1.3652300134141	0	0
1.3652300134141	0	0

Si observem x_n, veiem que hi ha molts decimals iguals, sembla que convergeix i ho fa més ràpidament que en ii). Mirant $| f(x_n) |$, es confirma que el mètode iteratiu és convergent.

Arxiu MATLAB: Exercici4_Apartat3a.m

(b) Per cada mètode convergent, obteniu el punt fix amb el punt inicial del mètode de Newton. Doneu els punts inicials i el criteri d'aturada (fins a 6 decimals correctes).

ii) El punt fix obtingut en 10 iteracions és *x1 = 1.36487821719368*, el punt inicial és $x_0 = 1$ i el criteri d'aturada és $\eta = 0.5 \cdot 10^{-6}$.

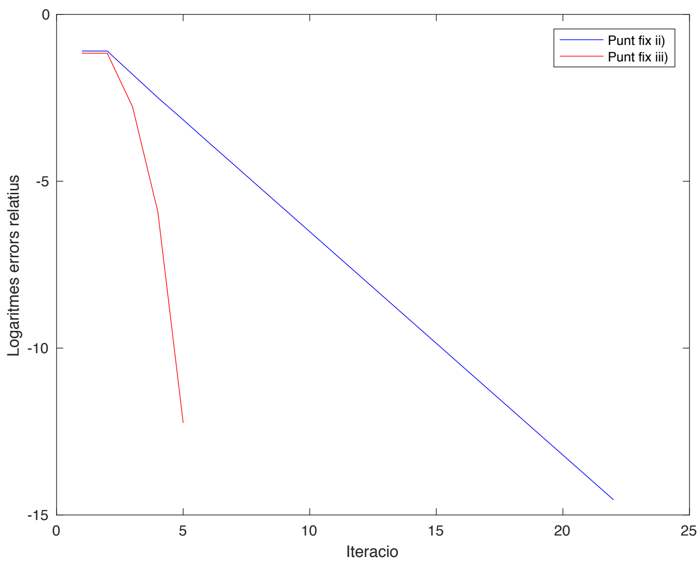
iii) El punt fix obtingut en 10 iteracions és *x2 = 1.36523001343537*, el punt inicial és $x_0 = 1$ i el criteri d'aturada és $\eta = 0.5 \cdot 10^{-6}$.

Arxiu MATLAB: Exercici4_Apartat3b_Principal.m i Exercici4_Apartat3b_Funcio.m

4. Representeu en un gràfic els logaritmes dels valors absoluts dels errors relatius aproximats:

$$r^{n+1} = \frac{x^{n+1} - x^n}{x^{n+1}}$$

per als mètodes convergents. Cada mètode un color diferent. A partir dels valors de les taules i les gràfiques dels errors, quin seria el millor procediment per obtenir cada una de les solucions de l'equació $x^3 + 4x^2 - 10 = 0$. Raona les teves respostes.



El millor procediment per obtenir la solució de l'equació donada a l'enunciat és el segon (*iii*) ja que és el que disminueix més ràpidament (observar línia vermella) i el que menys iteracions necessita per arribar a tenir 6 decimals correctes. Fixar-se que en l'apartat 1 hem trobat que l'arrel és x=1.3652 i en l'obtenció del punt fix de l'apartat 3, el mètode ii) no ha arribat al valor 1.3652 en les 10 iteracions, és a dir, que en necessita fer més. En canvi, el mètode iii) si que ha arribat al valor de l'arrel en només 10 iteracions.

Així doncs, el mètode iii) és el que convergeix més ràpidament i el que es considera millor.

Arxiu MATLAB: Exercici4_Apartat4.m

Annex

1.1 Algoritmes

Exercici1_Apartat1.m

```
function [xN] = Exercici1_Apartat1 (n)
k=10;
for i=1:k
    n(i)=5^i;
    x=rand(n(i),1);
    y=rand(n(i),1);
    z = x.^2+y.^2;
    v = (z <= 1);
    m=sum(v);
    pi_n(i) =4* m/n(i);
    error_absolut(i)=abs(pi-pi_n(i));
    error_relatiu(i)=error_absolut(i)/abs(pi);
end
  xN= pi_n(k)
  taula_resultats= [n',pi_n',error_absolut',error_relatiu']
end
```

Exercici1_Apartat2.m

```
clc
clear all
format long g
n= 5;
pi=Exercici1_Apartat1 (n);
pi
```

1.2 Errors de cancel·lació

Exercici2_Apartat2.m

```
% Grafica
clear all
format long g
f=@(x)x.^7-7.*x.^6+21.*x.^5-35.*x.^4+35.*x.^3-21.*x.^2+7.*x-1
x=0.9:0.05:1.2;
y=0.988:0.00005:1.012;
z=zeros(size(x));
x0=fzero(f,0);

plot(x,  f(x),y,  f(y),  'red',x0,  0,  'c*'),  grid,  title('Representacio
grafica del polinomi')

% Avaluacio del polinomi
a=0.988:0.00005:1.012;
[a;f(a)]'
```

Exercici2_Apartat3_Principal.m

```
clear all
format long g

% Algoritme de Horner
f=@(x)x.^7-7.*x.^6+21.*x.^5-35.*x.^4+35.*x.^3-21.*x.^2+7.*x-1;
x=0.988:0.00005:1.012;
a=[1 -7 21 -35 35 -21 7 -1];
n= 7;

d= Exercici2_Apartat3_Funcio(f,n,a,x)

n=1:1:length(d);
taula_resultats=[n;x;d]'

% Grafica
y=0.9:0.05:1.2;
plot(y,f(y),x,d,'red'), grid, title('Regla de Horner')
```

Exercici2_Apartat3_Funció.m

```
function[d]=Exercici2_Apartat3_Funcio(f,n,a,x)
%   f: polinomi a evaluar
%   n: nombre de graus del polinomi
%   a: coeficients del polinomi
%   x: valor en el que es vol evaluar el polinomi
d=a(n+1);
      for i= 1:n
          d=a(n+1-i)+d.*x;
      end
end
```

Exercici2_Apartat4.m

```
clear all
format long g

% Grafica apartat 2 apropada
f=@(x)x.^7-7.*x.^6+21.*x.^5-35.*x.^4+35.*x.^3-21.*x.^2+7.*x-1
x1=0.97:0.00005:1.03;
y1=0.988:0.00005:1.012;
subplot(1,2,1), plot(x1, f(x1),y1, f(y1), 'red'), grid, title('Apartat
2')

% Grafica apartat 3 apropada (Horner)
x2=0.988:0.00005:1.012;
a=[1 -7 21 -35 35 -21 7 -1];
n= 7;
d= Exercici2_Apartat3_Funcio(f,n,a,x2)
y2=0.97:0.00005:1.03;
subplot(1,2,2),  plot(y2,f(y2),x2,d,'red'),  grid,  title('Apartat  3
(Horner)’)
```

1.3 Error de truncament

Exercici3_Apartat1_Principal.m

```
clear all
format long g
N= input('valor N:')

resultat_pi=Exercici3_Apartat1_Funcio(N);
resultat_pi
```

Exercici3_Apartat1_Funcio.m

```
function [pi] = Exercici3_Apartat1_Funcio (N)
      pi=0;
      for n=0:N
            pi= 16^(-n)*(4/(8*n+1)-2/(8*n+4)-1/(8*n+5)-1/(8*n+6)) + pi;
      end
      pi;
end
```

Exercici3_Apartat2.m

```
clear all
format long g
N=20;
i=1;
while i <= N
    resultat_pi = Exercici3_Apartat1_Funcio(i);
      if resultat_pi == pi
        break
      end
    i=i+1;
end
i
```

1.4 Solucions d'equacions no lineals

Exercici4_Apartat1.m

```
f = @(x)x.^3 + 4.*x.^2 - 10;
x=-4:0.05:5;
z=zeros(size(x));
x0=fzero(f,0);
plot(x, f(x), x, z, 'r',x0,0, '*'),grid,title('Grafica del polinomi')
```

Exercici4_Apartat2a.m

```
clear all
format long
f = @(x)x.^3 + 4.*x.^2 - 10;
a(1)=1;
b(1)=2;
l(1)=b(1)-a(1);
m(1)=(a(1)+b(1))./2;
eps=0.5*(10.^-6);
```

```
k=1;
while abs(b-a)>eps

    if f(a(k))*f(m(k))<0
        a(k+1)=a(k);
        b(k+1)=m(k);
        k=k+1;
    else
        a(k+1)=m(k);
        b(k+1)=b(k);
        k=k+1;
    end

    m(k+1)=(a(k)+b(k))./2;
    r(k-1)=(m(k)-m(k-1))./m(k);
    l(k+1)=b(k)-a(k);
end

a=[0,a];
b=[0,b];
n=1:1:length(a);
taula_resultats=[n; a; b; m; f(m); l]'
```

Exercici4_Apartat2b.m

```
clear all
format long g
f=@(x)x.^3 + 4*x.^2 - 10;
a(1)=1
b(1)=2
tolx(1)=abs(b(1)-a(1));
tolf(1)=max(abs(f(a(1))),abs(f(b(1))));
eps=0.5*(10.^-6);
k=1
while (tolx(k)>eps && tolf(k)>eps)

   c = a(k);
   a(k+1) = b(k);
   b(k+1) = b(k) + (b(k) - c)/(f(c)/f(b(k))-1);
   tolx(k+1)= abs(b(k)-a(k));
   tolf(k+1)= abs(f(b(k)));
   k=k+1;

end


n=1:1:length(a);
taula_resultats=[n;a;b;f(b)]'
```

Exercici4_Apartat2c.m

```
f=@(x)x.^3 + 4*x.^2 - 10;
g=@(x) 3.*x.^2+ 8.*x;
xv(1)=1;
```

```
l(1)=xv(1);
eps=0.5*(10.^-6);
tolf= abs(f(xv(1)));
tolx = 1;
k=1

while (tolx>eps && tolf>eps)

   xN=xv(k)-(f(xv(k))/g(xv(k)))
   tolf= abs(f(xN));
   tolx= abs(xN-xv(k));
   xv(k+1)=xN;
   l(k+1)=xv(k+1)-xv(k);
   k= k + 1
end

n=1:1:length(xv);
taula_resultats=[n;xv;f(xv);l]'
```

Exercici4_Apartat3a.m

```
%% Metode iteratiu i).
f = @(x)x.^3 + 4.*x.^2 - 10;
x(1)=2; % Arrel positiva de l'equaci? donada
for i=2:10
    x(i)=x(i-1)-x(i-1).^3-4.*x(i-1).^2+10;
    tolx(i)=abs(x(i)-x(i-1));
end

% Estudi convergencia:
tolf1=abs(f(x))';  % Divergent
sol1=[x',tolx',tolf1]


%% Metode iteratiu ii).
f = @(x)x.^3 + 4.*x.^2 - 10;
y(1)=2;
for i=2:10
    y(i)=0.5*sqrt(10 - y(i-1).^3);
    toly(i)=abs(y(i)-y(i-1));
end

% Estudi convergencia:
tolf2=abs(f(y))';  % Convergent
sol2=[y',toly',tolf2]


%% Metode iteratiu iii).
f = @(x)x.^3 + 4.*x.^2 - 10;
z(1)=2;
for i=2:10
        z(i)=z(i-1)-((z(i-1).^3  +  4.*z(i-1).^2  -  10)/(3.*z(i-1).^2  +
8.*z(i-1)));
    tolz(i)=abs(z(i)-z(i-1));
end
```

```
% Estudi convergencia:
tolf3=abs(f(z))';  % Convergent
sol3=[z',tolz',tolf3]
```

Exercici4_Apartat3b_Principal.m

```
clc
clear all
format long g

%% Metode iteratiu convergent ii).
f = @(x)x.^3 + 4.*x.^2 - 10;
g=@(x)0.5*sqrt(10 - x.^3);
x0=1;
tol=0.5*(10.^-6);
N=10;

x1 = Exercici4_Apartat3b_Funcio ( f, g, x0, tol, N)

%% Metode iteratiu convergent iii).
f = @(x)x.^3 + 4.*x.^2 - 10;
g=@(x) x-((x.^3 + 4.*x.^2 - 10)/(3.*x.^2 + 8.*x));
x0=1;
tol=0.5*(10.^-6);
N=10;

x2 = Exercici4_Apartat3b_Funcio ( f, g, x0, tol, N)
```

Exercici4_Apartat3b_Funcio.m

```
function [ xN ] = Exercici4_Apartat3b_Funcio ( f, g, x0, tol, N)
%Metode  iteratiu del punt fix
%    g: funcio, x= g(x)
%    x=: punt inicial
%    tol: cota error
%    N= cota iteracions
k=0;
xV=x0;
tolx=1;
tolf=abs(f(x0));
while (k<N && tolx>tol && tolf>tol)

    xN=g(xV);
    k=k+1;
    tolf=abs(f(xN));
    tolx=abs(xN-x0);
    xV=xN;

end
end
```

Exercici4_Apartat4.m

```
clc
clear all
format long g

%% Metode iteratiu convergent ii).
g=@(x)0.5*sqrt(10 - x.^3);
x(1)=1;
x(2)=g(x(1));
eps=0.5*(10.^-6);
dif(1)=x(2)-x(1);
r(1)=(x(2)-x(1))./x(2);

m=2
while (abs(x(m)-x(m-1)) > eps) && (abs(g(x(m))) > eps)
    dif(m)=x(m)-x(m-1);
    r(m)=(x(m)-x(m-1))./x(m);
   x(m+1)=g(x(m));
   m=m+1;
end

l1=log(r);

%% Metode iteratiu convergent iii).
g=@(y) y-((y.^3 + 4.*y.^2 - 10)/(3.*y.^2 + 8.*y));
y(1)=1;
y(2)=g(y(1));
dif2(1)=y(2)-y(1);
r2(1)=(y(2)-y(1))./y(2);
n=2;

while (abs(y(n)-y(n-1)) > eps) && (abs(g(y(n))) > eps)
    dif2(n)=y(n)-y(n-1);
    r2(n)=(y(n)-y(n-1))./y(n);
   y(n+1)=g(y(n));
   n=n+1;
end

l2=log(r2);

%% Gràfic dels logaritmes dels valors absoluts dels errors relatius
aproximats
plot(1:(m-1),l1,'blue', 1:(n-1),l2,'red')
xlabel('Iteracio')
ylabel('Logaritmes errors relatius')
legend('Punt fix ii)','Punt fix iii)')
```