

# Càlcul numèric - Treball pràctic 2

Victor Ruiz, Louis Clergue

December 23, 2016

## 1 Projectió en un espai de funcions

a) Plantejament d'un algorisme per calcular la projecció.

Sabem que la funció  $s$  de l'espai  $S$  que minimitza la norma de la diferència amb  $f$ , si la norma està associada a un producte escalar, és la projecció ortogonal de  $f$  sobre l'espai  $S$ .

Per això podem plantejar les equacions normals de projecció normal i triar una base per tal que la matriu del sistema lineal sigui més senzilla a fi de trobar la solució.

Si expressem la nostra solució com:

$$s^* = \sum_{i=1}^{n+1} s_i \cdot \psi_i$$

Podem trobar els coeficients  $s_i$  resolent el sistema d'equacions normals:

$$\sum_{j=1}^{n+1} \langle \psi_i, \psi_j \rangle s_j = \langle f, \psi_i \rangle \quad \forall i = 1 \div n+1$$

La matriu del sistema lineal és matriu de producte escalar, per tant simètrica i definida positiva, llavors el sistema té solució. Seguirem els següents passos per resoldre aquest problema :

1. Trobar una base de l'espai  $S$
2. Calcular la matriu  $A$  de productes escalars de la base
3. Calcular el terme independent  $b$  de producte de la base amb  $f$
4. Resoldre el sistema lineal resultat per trobar els coeficients de la solució

b) Projectió d'una funció en l'espai de splines lineals  $C^0$  donats uns punts base

Usarem de base dels splines lineals  $C^0$  la definida per imposar:

$$\psi_i(x_j) = \delta_{ij} \quad \forall i, j \in 1 \div n+1$$

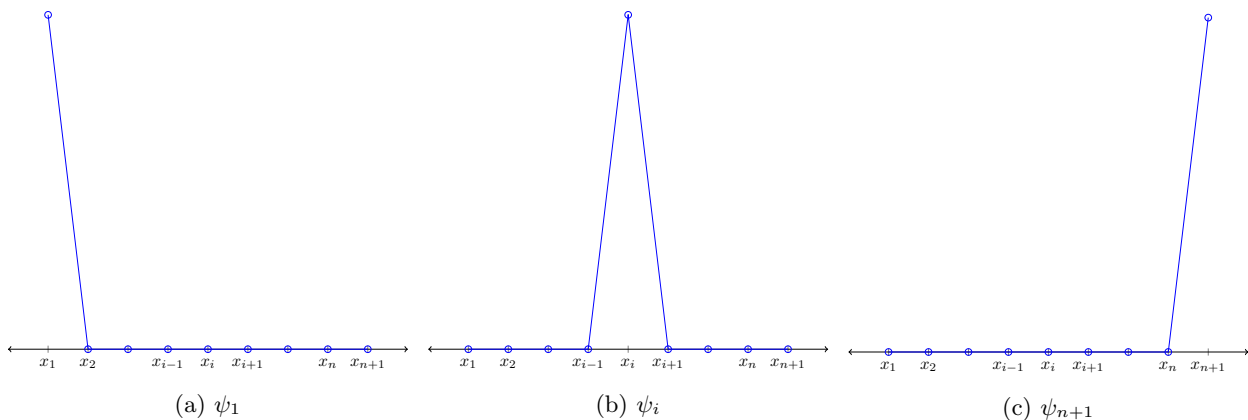


Figure 1: Base dels splines lineals

La matriu d'aquesta base (dimensió  $n+1 \times n+1$ ) és tridiagonal degut al suport local de la base, llavors podrem resoldre el sistema amb cost lineal.

$$A = \begin{pmatrix} a_{11} & a_{12} & & & & \\ a_{21} & a_{22} & a_{23} & & & \\ & a_{32} & \ddots & \ddots & & \\ & & \ddots & \ddots & a_{n,n+1} & \\ & & & a_{n+1,n} & a_{n+1,n+1} & \end{pmatrix} \quad a_{ij} = \langle \psi_i, \psi_j \rangle$$

Per a la implementació del mètode, veure el fitxer *problema1.m* amb valor de *splines* = 0; i les funcions associades *leastSquareSplineLinealC0.m* i *basisSplineC0.m*

c) Figura de  $f(x) = \sin(x) + \frac{x}{4}$  i de la seva projecció.

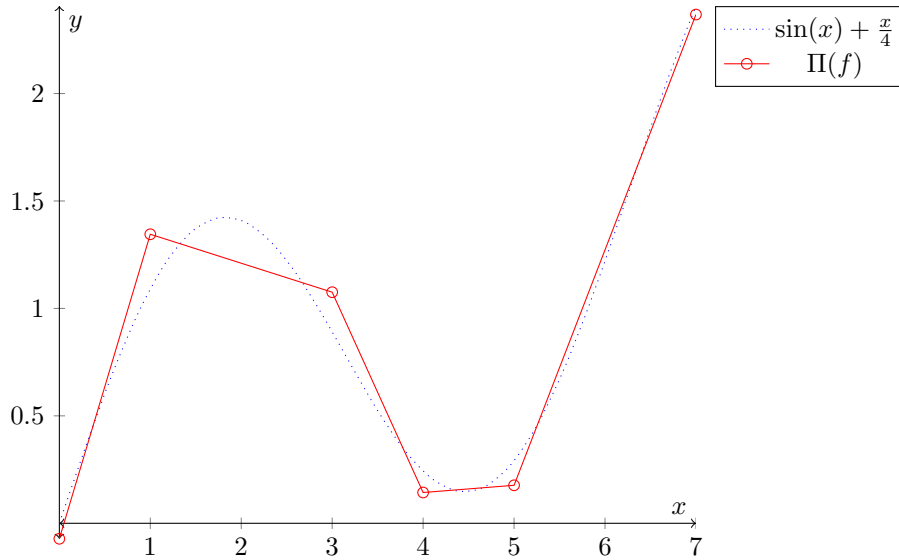


Figure 2:  $f$  i la seva projecció sobre splines lineals

L'aproximant no coincideix en general amb el valor de la funció als punts base, això és degut que estem minimitzant el valor d'una norma i no té perquè interpolar. De fet el nostre spline serà més "proper" a la funció original al no tenir restriccions de passar per certs punts.

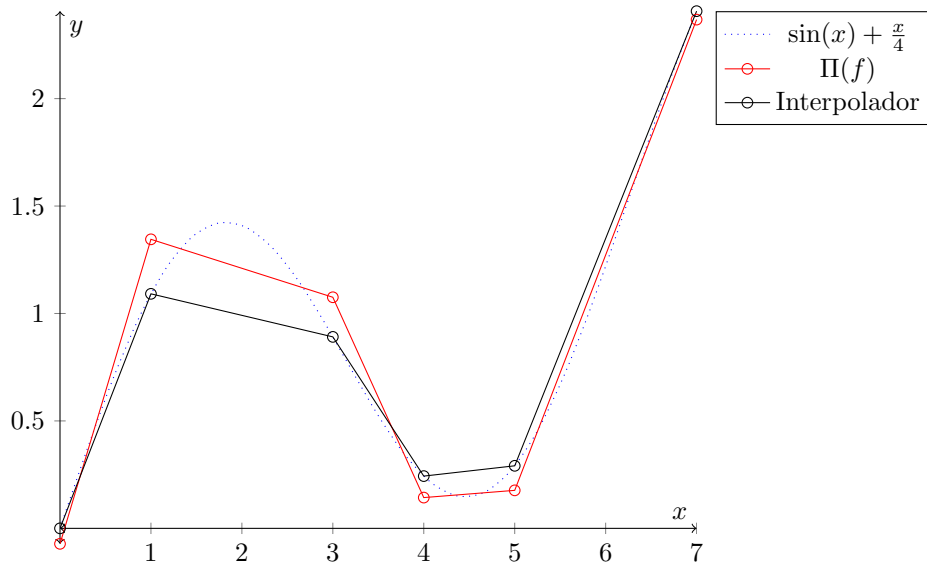


Figure 3: Comparació amb el spline lineal interpolador

El spline lineal interpolador està determinat pels punts base i segons com els hem escollit pot donar una aproximació més o menys bona però, si la funció oscil·la o és poc lineal, serà una solució poc fina. El mètode de mínims quadrats trobarà una solució molt més propera i és nota en la primera curvatura del

sinus on el spline allunya els punts de la corba per poder tenir menys diferència en tot l'interval. Tot i això són prou similars al tenir formes semblants.

d) Apartats b), c) per splines  $C^1$  cúbics.

Els splines cúbics  $C^1$  sobre  $n + 1$  punts base tenen dimensió  $2(n + 1)$  i queden determinats pel valor de l'spline i la seva derivada en els extrems. Escollirem una base amb suport local per tenir una matriu el màxim de diagonal possible:

$$\left. \begin{array}{l} \psi_i(x_j) = \delta_{ij} \\ \psi'_i(x_j) = 0 \end{array} \right\} \quad \forall i, j \in 1 \div n + 1 \quad \left. \begin{array}{l} \hat{\psi}_i(x_j) = 0 \\ \hat{\psi}'_i(x_j) = \delta_{ij} \end{array} \right\} \quad \forall i, j \in 1 \div n + 1$$

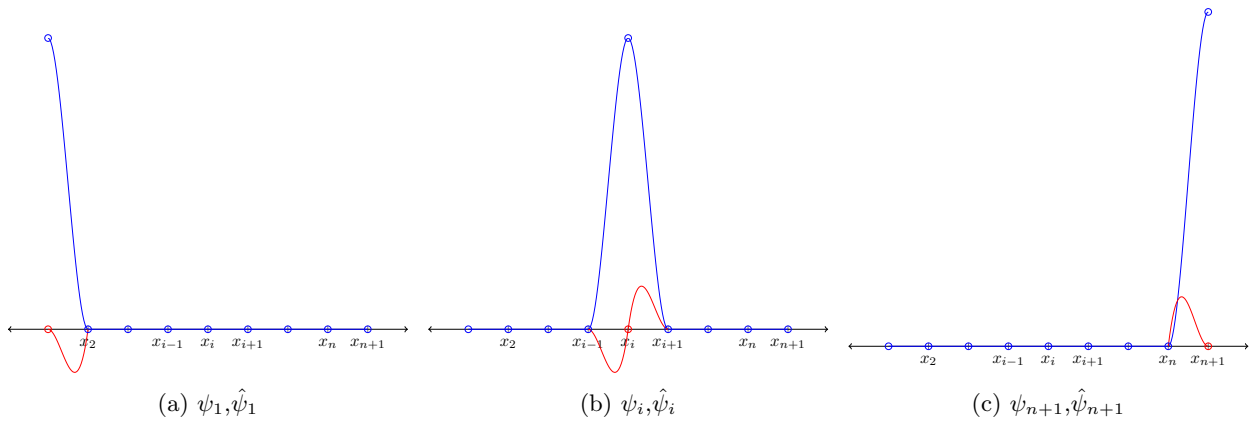


Figure 4: Base dels splines cúbics  $C^1$

La matriu corresponent al producte escalar d'aquesta base (dimensió  $2(n + 1) \times 2(n + 1)$ ) és pentadiagonal si ordenem la base de la següent manera:  $\psi_1, \hat{\psi}_1, \psi_2, \hat{\psi}_2, \dots, \psi_{n+1}, \hat{\psi}_{n+1}$ .

Existeix un algorisme per resoldre un sistema pentadiagonal de manera més eficient però per no haver de deduir-lo ni implementar-lo hem usat l'operador \ d'Octave per resoldre el sistema.

La figura obtinguda és la següent:

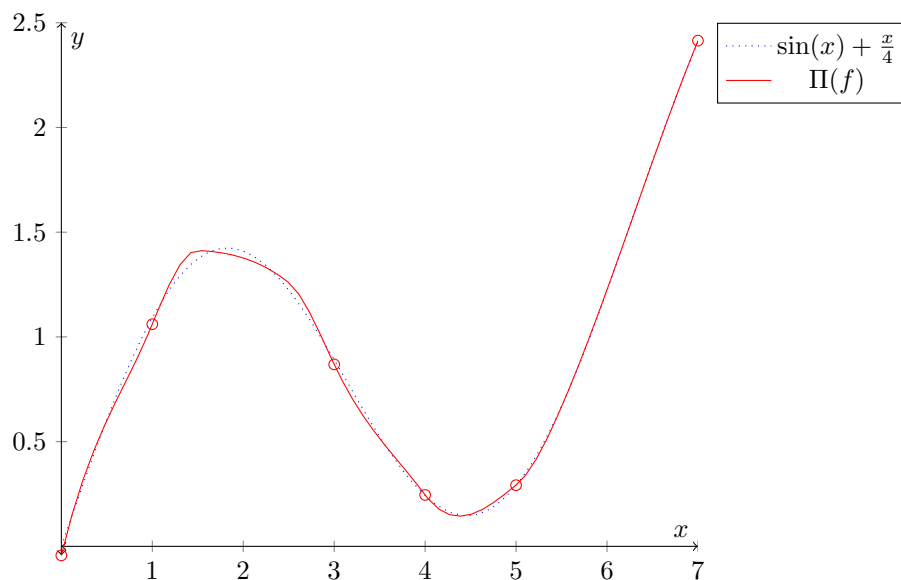


Figure 5:  $f$  i la seva projecció sobre splines cúbics

Al igual que pels splines lineals el valor de l'aproximant als punts base no és igual que el del polinomi, tot i que ara és molt més proper però amb diferències de l'ordre de  $10^{-2}$ .

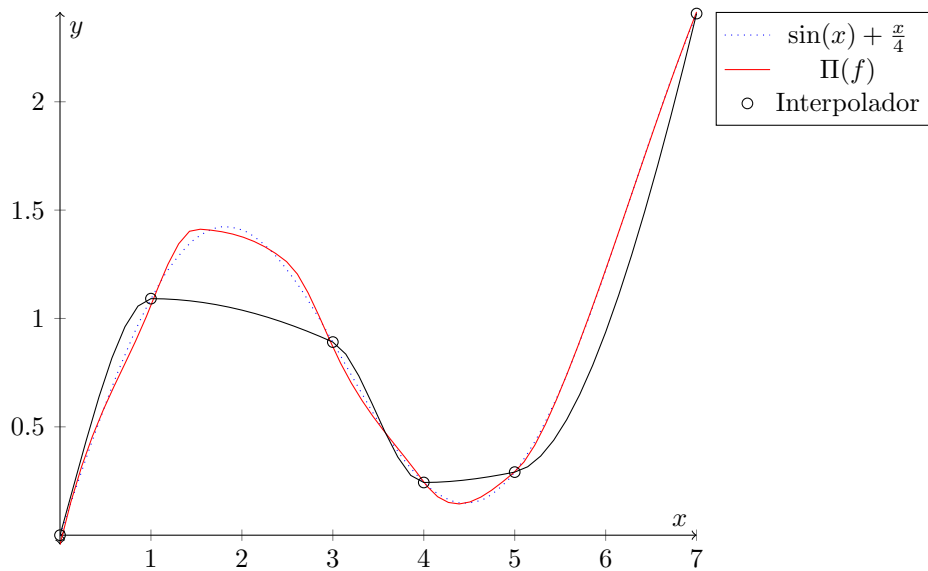


Figure 6: Comparació amb el spline cúbic interpolador

Podem observar un comportament similar que pels splines lineals, el spline cúbic  $C^1$  interpolador no aproxima correctament zones de la funció amb canvis més o menys pronunciats com el nostre sinus. Amb el polinomi de mínim quadrats tenim en consideració tota la funció i podem compensar aquests desperfectes.

Cal remarcar que el spline aproximant per mínims quadrats és millor que un spline interpolador però per el seu cost computacional és molt més elevat i cal poder avaluar la funció  $f$  en suficientment punts com per integrar-la de forma prou exacte. Així que splines interpoladors són una bona primera aproximació per veure la forma general però per detalls més fins caldrà algun mètode similar a l'implementat aquí amb splines per mínims quadrats.

## 2 Quadratures

Càlcul de la integral  $I$  utilitzant una quadratura de Simpson composta i una quadratura de Simpson adaptativa.

$$I = \int_0^3 \sin(e^x) dx$$

### 2.1 Quadratures compostes

- a) Aproximació de la integral mitjançant les fórmules compostes del trapezi i de Simpson.

Observem a les gràfiques de l'error que el mètode de Simpson, quan  $m$  augmenta, és més eficient respecte el número d'evaluacions de la funció que el mètode del trapezi, encara que al principi els 2 mètodes són similars. Els mètodes es comporten com s'esperava ja que els dos convergeixen quan  $m$  augmenta i el mètode del trapezi és una mica més ineficient que el de Simpson i, per tant, també tenen la convergència esperada.

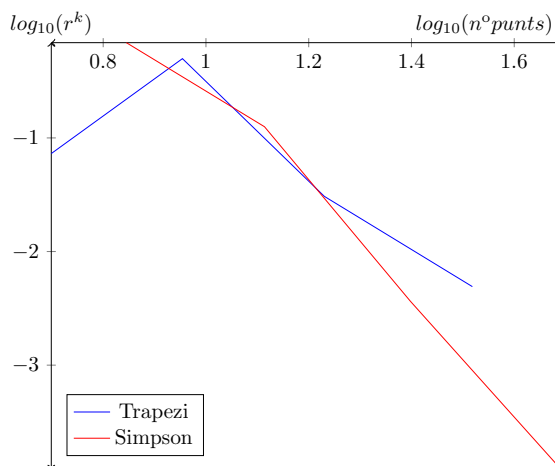


Figure 7: Convergència de les quadratures del trapezi i de Simpson

- b) Deducció del número d'interval·ls per aconseguir 6 xifres significatives correctes amb cadascuna de les dues quadratures i comprovació.

Calcularem la constant  $C$  de convergència de les dues quadratures usant la fórmula de l'error per  $m = 32$  (calculat a l'anterior apartat), i a partir d'aquesta deduirem el número de subinterval·ls necessaris per a cada quadratura usant també les fórmules de l'error per a cada quadratura (buscant un error menor que  $0.5 \cdot 10^{-6}$ ). Les  $m$  aconseguides són:

$$m_{Trap} = 1003; m_{Simp} = 71$$

Calculem l'error per a cada quadratura:

$$er_{Trap} = 4.5181 \cdot 10^{-6}; er_{Simp} = 4.3556 \cdot 10^{-6}$$

En efecte, s'aconsegueixen 6 xifres significatives correctes ja que l'error aconseguit suficientment petit (menor que la tolerància desitjada), i, per tant, s'aconsegueix la precisió requerida.

### 2.2 Quadratura de Simpson adaptativa

- c) Justificació de l'algorisme recursiu.

Segui la divisió en subinterval·ls resultants  $[x_i, x_{i+1}]_{i=0 \dots n-1}$  obtinguts amb el mètode en aplicar-ho en  $[a, b]$ , amb error estimat a cada interval  $r_i$ . Segui  $r$  l'error estimat total. Utilitzant que  $r_i < \epsilon(x_{i+1} - x_i)$  i  $x_0 = a$ ,  $x_n = b$ , tenim que:

$$r = \sum_{i=0}^{n-1} r_i < \sum_{i=0}^{n-1} \epsilon(x_{i+1} - x_i) = \epsilon(b - a)$$

El sumatori es una suma telescòpica on els termes es van anul·lant excepte  $x_0$  i  $x_n$ , els quals són  $a$  i  $b$ . Per tant, en aquest cas particular,  $r < 3\epsilon$ .

d) Implementació de l'algorisme.

Per a calcular un valor de referència  $I$  de la integral, fem servir la funció *quad* d'Octave:

$I \approx 0.60612$  ;  $res = 0.60612$

Observem que les primeres 5 xifres decimals són correctes, encara que l'error estimat (demanat) és de  $10^{-4}$ , el mètode es comporta encara millor del que s'esperava.

e) Representació gràfica de la divisió de l'interval en subintervalls.

Clarament s'observa com la divisió de l'interval és més fina a prop del punt  $b = 3$ , on la funció varia més, i menys fina a prop del punt  $a = 0$ , on la funció és més constant. En casos com aquest, on el comportament de la funció varia, és molt útil aquest tipus de quadratura, ja que s'aprofita per fer menys avaluacions de la funció on aquesta és menys variant, i a més, per tal que l'error no augmenti massa, s'ajusta més la integral on la funció és més variant.

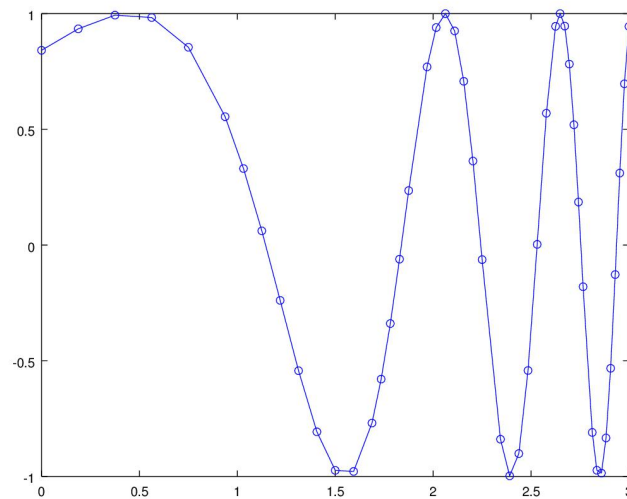


Figure 8: Punts de la quadratura de Simpson adaptativa amb error menor a  $10^{-4}$

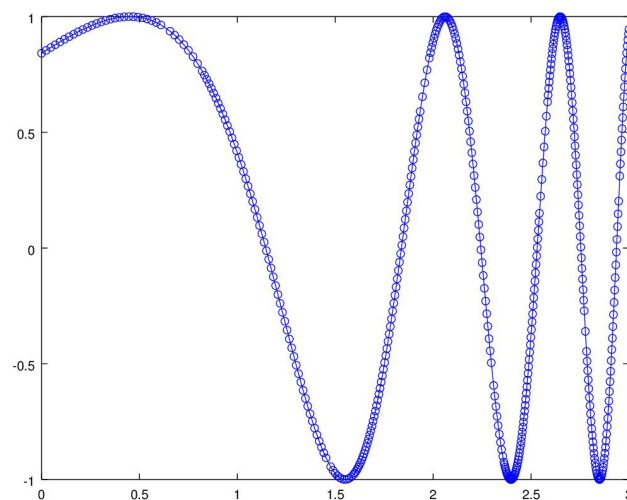


Figure 9: Punts de la quadratura de Simpson adaptativa amb error menor a  $10^{-8}$

f) Comparació amb la quadratura composta de Simpson equiespaiada.

Calcularem el número de subintervalls  $m$  necessaris amb la quadratura de Simpson augmentant aquest número fins que l'error sigui menor que la tolerància desitjada (en aquest cas,  $10^{-4}$  i  $10^{-8}$ ) ja que volem una deducció de  $m$  prou bona per comparar amb la quadratura adaptativa. Per no ralentitzar molt el

programa, en el cas de  $10^{-4}$  augmentarem  $m$  de 2 en 2, i per a  $10^{-8}$  l'augmentarem de 4 en 4 (no hi haurà quasi diferència en el resultat, ja que les xifres són centenes). Utilitzarem com a valor de referència de la integral el donat per la funció *quad* d'Octave.

Quadratura de Simpson adaptativa:

$$a_{10^{-4}} = 47; a_{10^{-8}} = 434$$

Quadratura de Simpson composta:

$$c_{10^{-4}} = 36; c_{10^{-8}} = 328$$

Observem que la quadratura de Simpson composta necessita una mica menys de subinterval·ls per aconseguir la tolerància desitjada que la quadratura adaptativa. Tot i això, la quadratura adaptativa té l'avantatge que, donada una tolerància, s'aconsegueix, sense haver de deduir res, el número de subinterval·ls necessaris i no s'ha d'iterar sobre  $m$  fins a aconseguir aquesta tolerància desitjada, cosa que la fa molt útil la quadratura adaptativa.

### 3 Aproximació d'una circumferència

- a) Plantejament de la estratègia per determinar la circumferència que millor aproxima les mesures.

Una circumferència està totalment definida per seu centre  $C = (c_x, c_y)$  i el seu radi  $r$ . Donat un punt qualsevol  $z = (x, y)$  definim l'error que hi ha al intentar aproximar el punt  $z$  per la circumferència  $S = (C, r)$  per la distància entre la circumferència i el punt; és a dir:

$$err_S(z) = |||x - c|| - r| = |||(x - c_x, y - c_y)|| - r| = |\sqrt{(x - c_x)^2 + (y - c_y)^2} - r|$$

Si tenim un conjunt de punts  $Z = \{z_i\}_{i=1}^n$  s'estén fàcilment aquesta definició a una mesura de l'error al aproximar el conjunt  $Z$  per la circumferència  $S$ .

$$err_S(Z) = \sqrt{\sum_{z \in Z} err_s(z)^2}$$

Llavors la circumferència que millor els aproximació és la que minimitza aquest error:

$$S_{\text{optima}} = \underset{S}{\operatorname{argmin}} \ err_S(Z)$$

Desenvolupant i aplicant que  $err_S$  sempre és positiu i que l'arrel és creixent, l'objectiu és minimitzar la següent funció:

$$f(x) = \sum_i (\sqrt{(x_i - c_x)^2 + (y_i - c_y)^2} - r)^2$$

Derivant i igualant a 0 arribem al següent sistema no-lineal:

$$2 \frac{\sum_i (\sqrt{(x_i - c_x)^2 + (y_i - c_y)^2} - r)}{\sum_i (\sqrt{(x_i - c_x)^2 + (y_i - c_y)^2})} \cdot \begin{pmatrix} x_i - c_x \\ y_i - c_y \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad (1)$$

$$-2 \sum_i (\sqrt{(x_i - c_x)^2 + (y_i - c_y)^2} - r) = 0 \quad (2)$$

De la equació 2 podem expressar el radi en funció del centre i només hem de resoldre un sistema no-lineal 2x2.

- b) Aplicació de la estratègia i resolució del problema.

Usarem d'aproximació inicial de la solució el punt mig del conjunt de punts (el radi ja queda determinat).

Un primer intent va consistir en usar derivació numèrica per tenir el sistema i fer una segona aproximació per la derivada, ràpidament vam veure que no donava resultats ja que divergia. Després de derivar implícitament la funció vam provar el mètode de Newton amb Jacobiana numèrica obtenint una convergència raonable.

Executant el programa "problema3.m" obtenim els següents resultats pel mètode de Newton:

```

IT 1 > sol = [9.9638;20.375]    r = 1.775118e-01
IT 2 > sol = [10.04;20.29]     r = 5.055009e-03
IT 3 > sol = [10.04;20.29]     r = 3.894542e-06
IT 4 > sol = [10.04;20.29]     r = 2.583156e-12
IT 5 > sol = [10.04;20.29]     r = 7.846804e-17
Solution [10.0401318287962;20.28973263147664]
radius = 50.0840433756909

```

Llavors la solució (amb 6 xifres significatives) pel conjunt de punts donats és la circumferència de centre  $C = (10.0401, 20.2897)$  i radi  $r = 50.0840$ . Amb un valor de l'error  $err_S(Z) = 1.9182488$

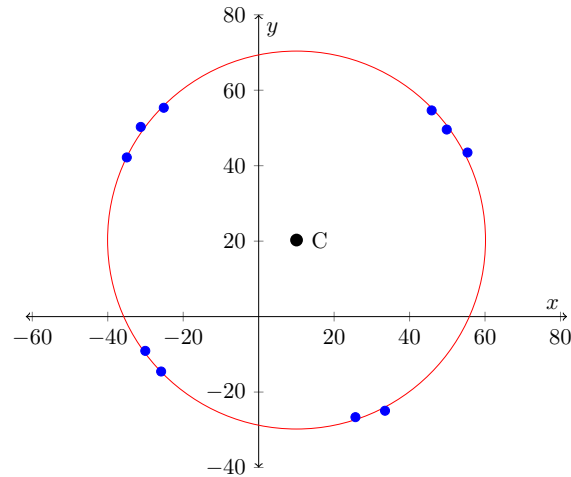


Figure 10: Circumferència solució i conjunt de punts

Com poden veure a la figura, la columnata era realment circular ja que els punts donats s'aproximen molt bé amb la circumferència que hem trobat.

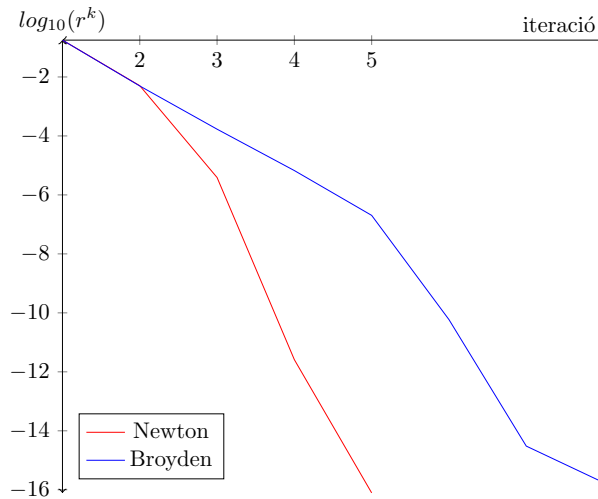


Figure 11: Convergència del mètode de Newton i Broyden

Amb Newton-Raphson, 5 iteracions són suficients per arribar a precisió màquina i la convergència és quadràtica com era d'esperar. També hem executat amb Broyden per comprovar resultats i al principi és lineal però millora al apropar-se.



## 4 Aproximació d'una corba (fet amb Matlab)

Donada una parametrització d'una corba regular, s'ha de trobar una distribució equiespaiada de punts sobre la corba  $\gamma$ , usant el paràmetre arc:

$$s(t) = \int_a^t \|\gamma'(x)\| dx$$

- a) Implementació d'una funció per calcular  $s(t)$  amb una quadratura de Simpson composta i deducció del número d'interval·ls necessaris  $m$  per tal que l'error estimat sigui menor que  $10^{-4}$ .

Deduirem la  $m$  augmentant la  $m$ , aproximant l'error relatiu  $E_r$  amb la següent expressió, fins que  $E_r$  sigui menor que  $10^{-4}$ :

$$E_r = \frac{|s_m - s_{100}|}{s_{100}}$$

On  $s_m$  és l'aproximació de la integral amb la quadratura composta de Simpson amb  $m$  interval·ls. S'utilitza com a referència el valor de la integral per  $m = 100$ .

El valor aconseguit de  $m$  és 21. El valor resultant de  $s(b)$  és 21.2271.

- b) Proposem un mètode per calcular l'antimatge  $t$  corresponent a un valor donat  $s$ .

Si tenim  $s$ , només hem de trobar un  $t$  tal que  $s(t) = s$ ; és a dir, trobar un zero de la funció  $f(t) = s - s(t)$ . Per trobar aquest zero, hem escollit el mètode de Newton amb derivada numèrica per reduir el número d'iteracions, encara que el mètode de la secant també hagués estat una bona opció.

Per calcular la derivada de la funció  $f(t)$ , com és la integral d'una funció (la norma de la derivada de  $\gamma$ ), evaluem directament la funció integrant en el punt.

- c) Implementació del mètode de l'anterior apartat. Aplicació per a  $s = (s(a) + s(b))/2$ .

$s = 10.6134$  i el valor trobat de  $t$  és 0.7988.

- d) Càlcul d'una distribució equiespaiada de 35 punts sobre la corba.

	t	x	y	z
$p_1$	0.0500	-4.2500	-0.0001	0.0001
$p_2$	0.3900	-4.0496	-0.2950	0.4985
$p_3$	0.4650	-3.5383	-0.5233	0.7461

La distància entre  $p_1$  i  $p_2$  és la mateixa que la distància entre  $p_2$  i  $p_3$ ;  $d = 0.6243$ . Aixó comprova que hem calculat correctament els  $t_i$ , ja que la distribució dels punts  $p_i$  era una distribució equiespaiada i, per tant, les distàncies entre ells han de ser la mateixa.