Mètodes numèrics

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

PRÀCTICA 3

1. Introducció

En aquesta tercera pràctica de l'assignatura de Mètodes Numèrics, ens enfoquem en la implementació i validació de rutines d'interpolació polinomial utilitzant el mètode de Neville. Aquest treball s'emmarca en la necessitat del Departament de Dinàmica Planetària de l'Agència Espacial de Bellaterra (AEB) de disposar d'eines per avaluar polinomis interpoladors i les seves derivades a partir de taules astronòmiques.

El mètode de Neville és una tècnica numèrica recurrent que permet calcular el valor del polinomi interpolador i la seva derivada en un punt determinat. A partir d'un conjunt de punts de suport (x_i, y_i), aquest mètode construeix una taula que facilita l'avaluació del polinomi interpolador mitjançant una sèrie de càlculs iteratius. Aquesta pràctica inclou el desenvolupament de la funció `intnev_avald()` i dues utilitats pràctiques: `intaimg` i `intext`.

La primera utilitat, `intaimg`, busca el n-img–èssim canvi de signe en una sèrie de dades i utilitza la interpolació inversa per trobar el valor de x tal que $f(x) \approx vimg$. La segona utilitat, `intext`, està dissenyada per trobar els extrems en una sèrie de dades mitjançant el mètode de la bisecció aplicat a la derivada del polinomi interpolador.

2. Implementació del software

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 `intnev_avald`, per tal d'evitar el problema d'ús de valors actualitzats en càlculs de les derivades, s'utilitza una variable auxiliar `aux_nev` per emmagatzemar el valor anterior de `nev[i]` abans de ser actualitzat. Això garanteix que es fa servir el valor correcte en el càlcul de `dnev[i]`:
- S'ha creat la utilitat `valida_intnev` per provar la funció `intnev_avald` de forma àgil i flexible. Aquesta utilitat prova `intnev_avald` amb tres funcions però, modificant el `return` de les funcions, es pot provar amb la funció que es vulgui.
- En les utilitats `intaimg` i `intext`, s'utilitza la funció `memmove` per moure els valors dels vectors `xi` i `yi` una posició cap a l'esquerra. Aquesta operació és necessària per tal de mantenir els últims `n` punts de suport actualitzats a mesura que es llegeixen nous punts de dades des del fitxer d'entrada. `memmove` és preferible a `memcpy` en aquest cas perquè `memmove` gestiona correctament els solapaments de memòria. Quan es mouen blocs de memòria que es poden solapar, `memmove` assegura que les dades es copien correctament, evitant la corrupció de dades que podria ocórrer amb `memcpy`. A més, l'ús de `memmove` permet que el codi sigui més eficient i fàcil de llegir, ja que mou tots els elements d'una sola vegada en lloc de requerir un bucle addicional per fer-ho manualment. D'aquesta manera, es manté la simplicitat i l'eficiència del codi alhora que s'assegura la integritat de les dades.

- En la utilitat `intext`, fem ús d'una estructura `struct_intext` per encapsular els paràmetres necessaris per a la funció `intnev_bis`. Aquesta estructura conté els vectors de punts de suport `x_sup` i `y_sup`, així com el nombre de punts de suport `n`. L'ús d'aquesta estructura permet passar tots els paràmetres necessaris a la funció `intnev_bis` com un sol argument, simplificant així la gestió dels paràmetres.
- En la utilitat `intext` la funció `intnev_bis` s'utilitza per calcular la derivada del polinomi interpolador en un punt `x`. Aquesta és la funció de la qual `biseccio` ha de trobar un zero donat un interval `[a, b]` inicial ja que, l'objectiu, és trobar els extrems de les dades que entren per standard input.

3. Manual del software

3.1. La funció intnev_avald

Aquesta funció avalua el polinomi interpolador i la seva derivada usant el mètode de Neville.

Prototipus

```
void intnev_avald(double x, int n, double xi[], double yi[], double *p, double *dp);
```

Paràmetres

- **x**: Punt on es vol avaluar el polinomi i la seva derivada.
- **n**: Nombre de punts de suport.
- **xi**: Array de longitud n que conté els valors x dels punts de suport.
- **yi**: Array de longitud n que conté els valors y dels punts de suport.
- **p**: Punter on es guardarà el valor del polinomi interpolador en x (pot ser NULL).
- **dp**: Punter on es guardarà el valor de la derivada del polinomi interpolador en x (pot ser NULL).

3.2. La utilitat valida_intnev

Aquest programa prova la funció `intnev_avald` per avaluar un polinomi interpolador i la seva derivada utilitzant el mètode de Neville. Es generen tres funcions diferents (lineal, quadràtica, i trigonomètrica), es calculen els valors de suport `xi` i `yi`, i després es crida la funció `intnev_avald` per avaluar el polinomi interpolador i la seva derivada en un punt donat.

Les funcions de suport generen valors `xi` a intervals regulars i calculen els valors corresponents `yi` per a cada funció de prova. A continuació, es crida `intnev_avald` per calcular el polinomi interpolador i la seva derivada en un punt específic i es mostren els resultats.

Prototipus de les funcions utilitzades

```
void genera_valors(double (*func)(double), double xi[], double yi[], int n, double pas);
void print_p_suport(double *xi, double *yi, int n);
double lineal(double x);
double quadratica(double x);
double trigonometrica(double x);
```

Paràmetres de `genera_valors`

- **func**: Funció a avaluar.
- **xi**: Array per als valors de suport `xi`.
- **yi**: Array per als valors de suport `yi`.
- **n**: Nombre de punts de suport.
- **pas**: Pas entre els punts `xi`.

Execució
Per compilar aquest programa utilitzar la comanda: `make valida_intnev`
Per executar aquest programa utilitzar la comanda: `./valida_intnev n x pas ixrr`

- **n**: Nombre de punts de suport `xi` i `yi`.
- **x**: Punt on avaluarem el polinomi i la seva derivada.
- **pas**: Pas entre els punts `xi`.
- **ixrr**: 1 si es vol que la utilitat escrigui els punts de suport generats per cada funció; 0 en cas contrari.

3.3. La utilitat intaimg

Aquesta utilitat llegeix un fitxer d'entrada amb dues columnes (`xi`, `yi`) i busca el `nimg`-èssim canvi de signe de `yi - vimg`. A continuació, troba el valor de `x` tal que `f(x) = vimg` mitjançant interpolació inversa amb un polinomi interpolador de grau `n`. També calcula i mostra el valor del polinomi interpolador en `x`.

Execució
Per compilar aquest programa, utilitzar la comanda: `make intaimg`
Per executar aquest programa utilitzar la comanda: `./intaimg n nimg vimg`

- **n**: Nombre de punts de suport `xi` i `yi`.
- **nimg**: Índex del canvi de signe desitjat.
- **vimg**: Valor de la imatge per a la interpolació inversa.

3.4. La utilitat intext

Aquesta utilitat llegeix un fitxer d'entrada amb dues columnes (`xi`, `yi`) i busca el `next`-èssim extrem de la funció. Utilitza el mètode de la bisecció per trobar els zeros de la derivada del polinomi interpolador.

Prototipus de `intnev_bis`

```
double intnev_bis(double x, void *prm);
```

Paràmetres

- **x**: Punt on es vol calcular la derivada.
- **prm**: Estructura que encapsula els punts de suport (`x_sup`, `y_sup`) i el nombre de punts (`n`). `intext`.

Execució
Per compilar aquest programa, utilitzar la comanda: `make intext`
Per executar aquest programa utilitzar la comanda: `./intext n next tol ixrr`

- **n**: Nombre de punts de suport `xi` i `yi`.
- **next**: Índex de l'extrem desitjat.
- **tol**: Tolerància per al mètode de la bisecció.
- **ixrr**: 1 si es vol imprimir les iteracions; 0 en cas contrari.

4. Validació del software i part experimental

4.1. Validació intnev

```
Amb ixrr = 1 i un pas petit entre x
./valida_intnev 10 2.5 1 1

Funció lineal: 2x + 3
xi = [0,1,2,3,4,5,6,7,8,9]
yi = [3,5,7,9,11,13,15,17,19,21]

x = 2.5: p = 8, dp = 2 dif = 0

Funció quadràtica: x² - 4x + 6
xi = [0,1,2,3,4,5,6,7,8,9]
yi = [6,3,2,3,6,11,18,27,38,51]

x = 2.5: p = 2.25, dp = 1 dif = 0

Funció trigonomètrica: sin(x)
xi = [0,1,2,3,4,5,6,7,8,9]
yi = [0, 0.8414709848078965, 0.9092974268256817, 0.1411200080598672,
-0.7568024953079282, -0.9589242746631385, -0.2794154981989259,
0.6569865987187891, 0.9893582466233818, 0.4121184852417566]

x = 2.5: p = 0.5988337422282185, dp = -0.8014289648616557 dif = 0.0003615981242619526
```

```
Amb ixrr = 0 i un pas més gran entre x
./valida_intnev 10 2.5 3 0

Funció lineal: 2x + 3
x = 2.5: p = 8, dp = 2 dif = 0

Funció quadràtica: x² - 4x + 6
x = 2.5: p = 2.25, dp = 1 dif = 0

Funció trigonomètrica: sin(x)
x = 2.5: p = 2.180075087732769, dp = -4.701846040749075 dif = 1.581602943628812
```

4.2. Validació intaimg

Primer 0 de horizons.txt:

```
./intaimg 10 1 0 < horizons.txt
S'ha trobat un canvi de signe entre yi[5] = -0.051060 i yi[6] = 0.344000

Interpolació inversa: f(x)  0, x = 110.1292050169844
Interpolació: x = 110.1292050169844, y = 8.78270702664045e-10
```

Segon 0 de horizons.txt:

```
./intaimg 10 2 0 < horizons.txt
S'ha trobat un canvi de signe entre yi[5] = -0.051060 i yi[6] = 0.344000

S'ha trobat un canvi de signe entre yi[5] = 0.206460 i yi[6] = -0.182620

Interpolació inversa: f(x)  0, x = 296.5307219537426
Interpolació: x = 296.5307219537426, y = -1.966667685309768e-09
```

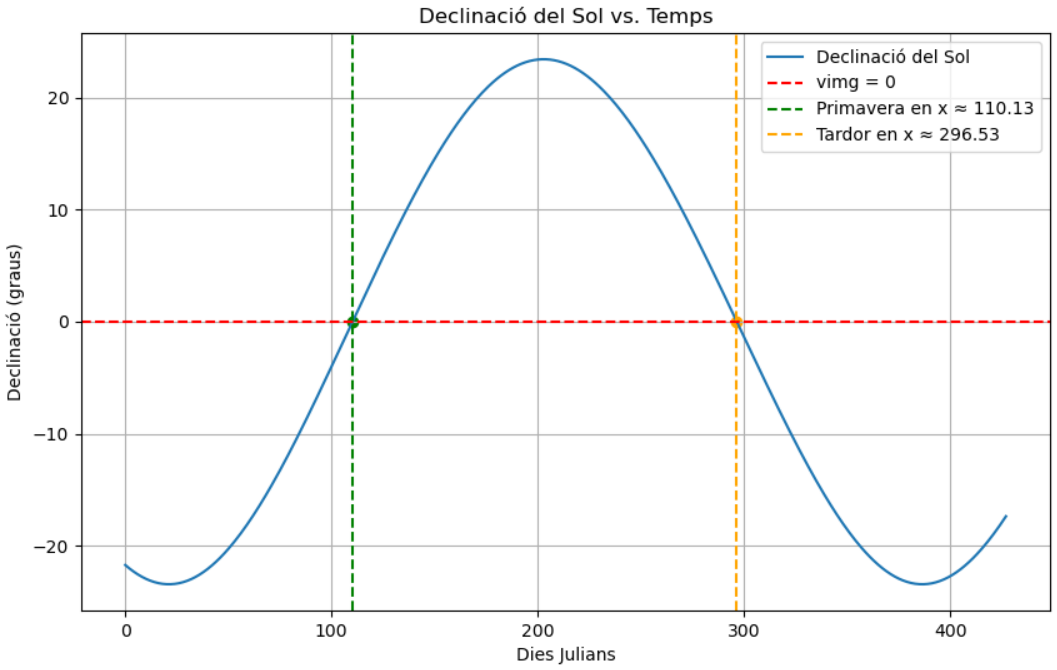

Figura 1: Equinoccis de primavera i tardor.

4.3. Validació intext

Primer extrem de horizons.txt:

```
./intext 10 1 1e-8 0 < horizons.txt

Canvi de signe amb dp1 = -0.001085, dp2 = 0.006762, x[5] = 21.000000, x[6] = 22.000000

Valor interpolat de x per l'extrem  21.13814683258533: -3.654108127193467e-11
```

Segon extrem de horizons.txt:

```
./intext 10 2 1e-8 0 < horizons.txt

Canvi de signe amb dp1 = 0.005947, dp2 = -0.000922, x[5] = 202.000000, x[6] = 203.000000

Valor interpolat de x per l'extrem  202.8655062168837: -6.602824953461095e-11
```

Tercer extrem de horizons.txt:

```
./intext 10 3 1e-8 0 < horizons.txt

Canvi de signe amb dp1 = -0.003097, dp2 = 0.004744, x[5] = 386.000000, x[6] = 387.000000

Valor interpolat de x per l'extrem  386.3956086188555: 6.910241268087702e-11
```

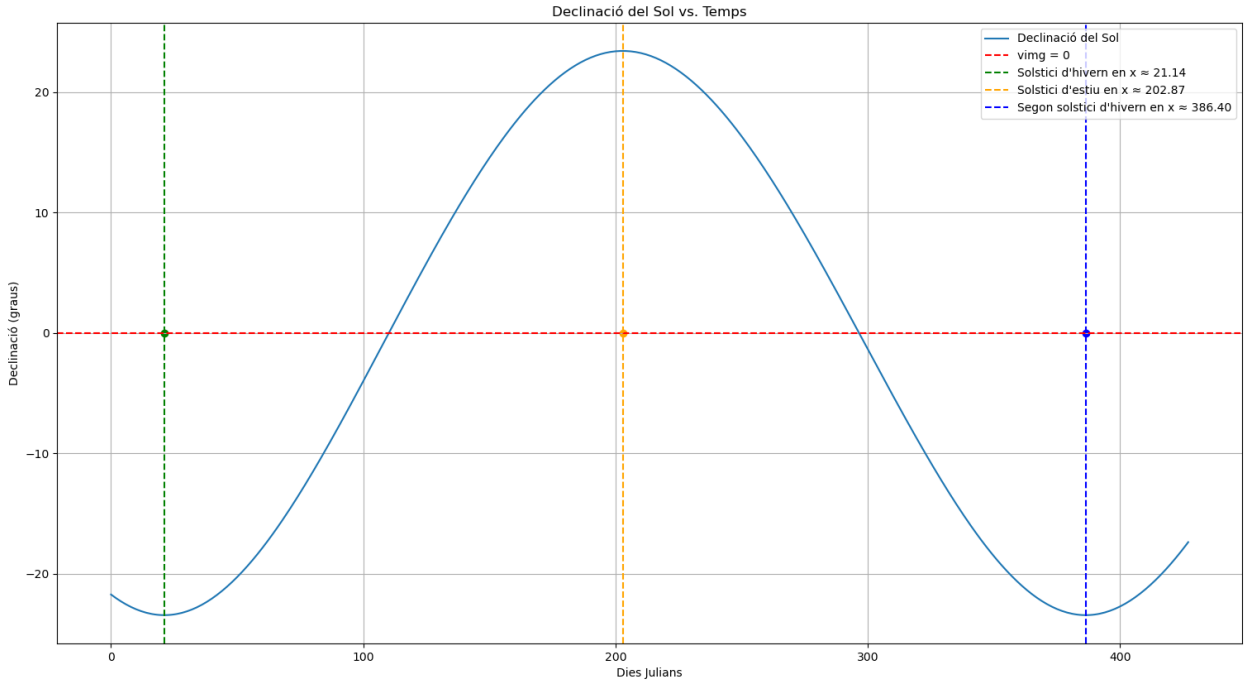

Figura 2: Solsticis d'hivern i estiu.