Relazioni di Laboratorio di Fisica Computazionale

Carlo Sana 18 maggio 2012

1 Introduzione

2 Metodi di integrazione

L'obiettivo di questa parte è di stimare il valore dell'integrale definito di una funzione di una variabile:

$$I = \int_{x_{min}}^{x_{max}} f(x)\,dx$$

Le prime routine che sono state scritte sono un'implementazione delle formule di Newton-Cotes (al primo e secondo ordine) e sul metodo delle quadrature gaussiane.
Per aumentare la precisione del calcolo, il dominio di integrazione viene suddiviso in sottointervalli. La larghezza di ogni intervallo è uniforme ed è possibile scegliere il numero di intervalli in cui si vuole dividere il dominio di integrazione prima di chiamare le funzioni. Per ottenere la stima dell'integrale è necessario sommare le stime degli integrali ottenute per i sottointervalli.
Nel nostro caso n sarà il numero di sottointervalli. Definiamo così in modo naturale una partizione dell'insieme di integrazione:

$$h = \frac{x_{max} - x_{min}}{n} \qquad \Longrightarrow \qquad a_i = x_{min} + ih \qquad \text{per} \quad i = 0, 1, ..., n$$

La funzione che si occupa di questo compito è la seguente:

```
double partition (double min, double max , int n, int methodNumber , double (*f) (double) ){
        double h  = 0;
        int i = 0;
        double Sum = 0;
        double (*method) (double ,double , double (*) (double));
        switch(methodNumber){
                case 1:
                        method = trapezio;
                        break;
                case 2 :
                        method = Simpson;
                        break;
                case 3 :
                        method = gaussianQuad;
                        break;
                default:
                        printf("Bad integration method!");
                        exit(EXIT_FAILURE);
                        break;
        }
        h = (max-min)/n;
        for (i = 0; i< n; i++){
                Sum += method( min + i*h, min + (1+i)*h, f);
                }
        return Sum;
}
```

E' necessario passare all'argomento della funzione il metodo di integrazione desiderato, attraverso un intero:

1 viene utilizzato il metodo di Newton-Cotes al primo ordine.

2 il metodo è Newton-Cotes al secondo ordine.

3 vengono utilizzate le quadrature gaussiane, con polinomio di Legendre di quinto grado.

Le funzioni che implementano i tre metodi di integrazione devono avere gli stessi parametri di input. Nel caso gli estremi di integrazione sono scambiati, ossia $x_min > x_max$, l'integrazione avviene comunque correttamente, visto che in questo caso h sarà negativo.

Newton-Cotes

Le formule di Newton-Cotes si ottengono interpolando la funzione integranda con polinomi di Lagrange. Il polinomio di Lagrange j-esimo di grado n è definito come:

$$l_j^n(x) \;=\; \prod_{i=0, i\neq j}^{n} \frac{x - x_i}{x_j - x_i}$$

Come si può vedere è un polinomio di grado n, definito in base alla partizione scelta per l'intervallo, con la proprietà:

$$l_j^n(x_j) = \delta_{ij}$$

E' ora immediato costruire un polinomio $P(x)$ tale che $P(x_i) = f(x_i) \qquad \forall 0 < i < n$. Questo polinomio è il seguente:

$$P(x) = \sum_{i=0}^{n} f(x_i) l_i^n(x)$$

La stima dell'integrale diventa così:

$$I \;=\; \int_{x_{min}}^{x_{max}} P(x)\,dx \;\;=\; \int_{x_{min}}^{x_{max}} \sum_{i=0}^{n} f(x_i) l_i^n(x) \;=\; \sum_{i=0}^{n} f(x_i) \int_{x_{min}}^{x_{max}} l_i^n(x)\,dx$$

Si dimostra inoltre che, con un cambio di variabile:

$$\omega_j = \int_{x_{min}}^{x_{max}} l_i^n(x)\,dx = \int_0^n \prod_{i=0\ i\neq j}^{n} \frac{z - i}{j - i}\,dz$$

Questa è ovviamente una stima dell'integrale e si dimostra che l'errore, utilizzando $n+1$ punti è uguale a :

$$E_n = \frac{1}{(n+1)!} \int_{x_{min}}^{x_{max}} f^{n+1}(\xi) \prod_{i=0}^{n} \frac{x - x_i}{} dx$$

dove ξ è un punto interno all'intervallo. L'errore è facilmente sovrastimabile, valutando il massimo della derivata $n+1$-esima all'interno dell'intervallo. L'errore sulla stima dell'integrale, però, non viene calcolato dalla routine d'integrazione. Ciò deriva dal fatto che esso è stimabile analiticamente, essendo necessario solo calcolare derivate e valutarne il massimo nell'intervallo.

Newton-Cotes:1°ordine

Questo metodo consiste nell'approssimare la funzione fra due punti a_i e a_{i+1} con un segmento. L'area si ottiene calcolando l'area del trapezio sotteso da questo segmento, oppure applicando le formule di Newton-Cotes, ponendo $n = 1$:

$$\omega_0 = \frac{1}{2} \qquad \omega_1 = \frac{1}{2}$$

La routine che implementa la formula è la seguente:

```
double trapezio ( double min , double max, double (*f) (double) ){
        return ((max-min)*( f(min) + f(max) )/2.0 ) ;
        }
```

dove f è il puntatore a funzione della funzione integranda.

Newton-Cotes:2°ordine

In questo caso, l'approssimazione viene fatta con polinomi di grado 2, ossia parabole. I pesi ω_i valgono:

$$\omega_0 = \frac{1}{6} \qquad \omega_1 = \frac{2}{3} \qquad \omega_2 = \frac{1}{6}$$

```
double Simpson ( double min , double max, double (*f) (double) ){
                return ((max-min)*( f(min) + 4.0*f(min + (max-min)/2) + f(max))/6.0);
        }
```

come nel caso precedente, f è il puntatore alla funzione da integrare.

Quadrature gaussiane

Nel caso delle quadrature gaussiane, i punti della partizione non vengono più scelti cieca in modo da essere equidistanti, ma vengono scelti in maniera più opportuna: saranno scelti in modo da essere gli zeri del polinomio ortogonale scelto.
E' fondamentale l'uso di polinomi ortogonali in un certo intervallo $[a, b]$ con il peso $\omega(x)$:

$$\int_a^b \omega(x)\ P_n x\ P_m x\ =\ \delta_{m,n}$$

Si può dimostrare che la stima dell'integrale è:

$$I\ =\ \int_a^b f(x)\ =\ \sum_{i=0}^{n} \omega_i f(x_i)$$

dove ω_i, nel caso dei polinomi di Legendre, sono gli stessi pesi definiti per i polinomi di Lagrange costruiti sull'insieme degli zeri del polinomio di Legendre considerato. Considerando che un polinomio ortogonale di grado n in $[a, b]$ ha n zeri in $[a, b]$, anche i polinomi di lagrange saranno di grado n. Inoltre x_i sono gli zeri del polinomio considerato.

```
double gaussianQuad ( double min , double max , double (*f) (double)){
        /*Viene usato un polinomio di grado 5 */
        double zero[5] = {0 ,
                        sqrt(245.0 - 14.0*sqrt(70.0))/21.0,
                        -sqrt(245.0 - 14.0*sqrt(70.0))/21.0,
                        sqrt(245.0 + 14.0*sqrt(70.0))/21.0,
                        -sqrt(245.0 + 14.0*sqrt(70.0))/21.0};
```

```
        double weight[5] = {(double)128.0/225.0,
                        (double)1/900.0*(322.0 + 13*sqrt(70.0)),
                        (double)1/900.0*(322.0 + 13*sqrt(70.0)),
                        (double)1/900.0*(322.0 - 13*sqrt(70.0)),
                        (double)1/900.0*(322.0 - 13*sqrt(70.0))} ;
        double integral =  0;
        /* Porto [min,max] in [-1,1] */
        int i = 0;
        for (i = 0 ; i< 5 ; i++){
                integral += weight[i]*f((max-min)/2.0*zero[i]+(max+min)/2.0);
        }
        return integral*(max-min)/2.0 ;
}
```

Gli zeri del polinomio sono stati inseriti manualmente, lasciando però al calcolatore il compito di calcolarne il valore in virgole mobile, per aumentarne la precisione. All'interno dell'algoritmo, è necessario effettuare un cambio di variabile in modo da mappare l'intervallo di integrazione nell'intervallo $[-1, 1]$, nel quale il polinomio considerato (Legendre) è ortogonale. Tale cambio di variabile è:

$$x' = 2\frac{x - x_{min}}{x_{max} - x_{min}} - 1 \qquad dx' = 2\frac{dx}{x_{max} - x_{min}}$$

Utilizzo

Il programma "integral" si occupa di utilizzare le librerie appena discusse per stimare il valore dell'integrale di una funzione, confrontando i tre metodi discussi sopra. Per testare le routine ho utilizzato come funzioni da integrare:

$$integrandLog = log(1 + x) \quad \Rightarrow \quad primitiveLog = -x + log(1 + x) + xlog(1 + x)$$

$$integrandPoly = x^9 - x^7 + 3 \quad \Rightarrow \quad primitivePoly = \frac{x^{10}}{10} - \frac{x^8}{8} + 3x$$

le quali sono facilmente integrabili. In questo modo ho potuto confrontare i tre metodi di integrazione con il valore vero dell'integrale, valutandone lo scostamento dal valore vero.
Riporto in una tabella una serie di risultati ottenuti variando il numero di intervalli, ma mantenendo costanti gli estremi di integrazione. In questo caso l'intervallo di integrazione è stato $[1, 2]$. La prima tabella si riferisce alla funzione integranda di tipo logaritmico, la seconda a quella di tipo polinomiale.

Numero intervalli	Newton-Cotes I	Newton-Cotes II	Quadrature gaussiane
10	1.388645e-04	6.101824e-09	2.220446e-16
20	3.472070e-05	3.816787e-10	2.220446e-16
40	8.680460e-06	2.386025e-11	2.220446e-16
80	2.170133e-06	1.491696e-12	2.220446e-16
160	5.425343e-07	9.336976e-14	5.551115e-16
320	1.356337e-07	6.217249e-15	1.221245e-15
640	3.390842e-08	2.220446e-16	-3.330669e-16
1280	8.477104e-09	1.443290e-15	1.332268e-15
2560	2.119278e-09	-6.661338e-16	-6.661338e-16
5120	5.298200e-10	-2.664535e-15	-2.775558e-15
10240	1.324560e-10	-1.443290e-15	-1.443290e-15
20480	3.310918e-11	-2.220446e-15	-2.220446e-15
40960	8.281043e-12	-2.220446e-15	-2.109424e-15
81920	2.071121e-12	3.108624e-15	2.997602e-15

Numero intervalli	Newton-Cotes I	Newton-Cotes II	Quadrature gaussiane
10	-1.541035e+00	-9.908609e-04	0.000000e+00
20	-3.860018e-01	-6.203492e-05	0.000000e+00
40	-9.654698e-02	-3.878841e-06	1.421085e-14
80	-2.413966e-02	-2.424535e-07	1.421085e-14
160	-6.035096e-03	-1.515373e-08	2.842171e-14
320	-1.508785e-03	-9.471250e-10	0.000000e+00
640	-3.771970e-04	-5.921663e-11	2.842171e-14
1280	-9.429930e-05	-3.652190e-12	-2.842171e-14
2560	-2.357483e-05	-1.847411e-13	2.842171e-14
5120	-5.893707e-06	-4.263256e-14	-2.842171e-14
10240	-1.473427e-06	1.705303e-13	1.705303e-13
20480	-3.683565e-07	-4.263256e-14	-4.263256e-14
40960	-9.208941e-08	3.552714e-13	3.552714e-13
81920	-2.302234e-08	-4.263256e-14	-2.842171e-14

2.1 Buffon

2.2 Campionamento d'importanza

3 Integrali di cammino nell'oscillatore armonico

4 Runge-Kutta IV

L'implementazione di questo metodo di risoluzione delle equazioni differenziali ordinarie è valida per sistemi di equazioni differenziali di ordine 2: ossia riconducibili a un sistema di 2 equazioni differenziali.
Nel nostro caso è stata risolta un'equazione newtoniana, ossia della forma

$$\ddot{x} \;=\; f(x,t) \Longleftrightarrow \begin{cases} \dot{x_1} &= x_2 \\ \dot{x_2} &= f(x,t) \end{cases}$$

Il sistema risolto in questo caso è un pendolo smorzato con forzante esterna sinusoidale, ossia risolvente la seguente equazione differenziale:

$$\ddot{\theta} = f(\theta,\dot{\theta},t) \;-\frac{g}{R}\,\theta - b\,\dot{\theta} + Q\cos(\omega t)$$

L'algoritmo è implementato, sostanzialmente in questa funzione. Essendo un'implementazione in dimensione due, riceve come argomento 2 vettori in cui saranno salvati gli incrementi, le due variabili spaziali, e le due funzioni date dal sistema di equazioni differenziali. Nel nostro caso avremo:

$$\begin{cases} x_1 = \theta \\ x_2 = \dot{\theta} \end{cases} \quad\Longrightarrow\quad f_1 = x_2 \qquad f_2 = f(x_1,x_2,t)$$

Nella funzione è utilizzata la variabile di preprocessore H che rappresenta il passo dell'incremento infinitesimo nel tempo.

```
void kn_fill (double *k1_vec, double *k2_vec,double x1,double x2,double t,
                double (*f1) (double ,double,double ),
                double (*f2) (double, double,double)){
  int i = 0;
  k1_vec[0]= H*f1(x1,x2,t);
  k2_vec[0]= H*f2(x1,x2,t);
  for(i = 0; i< 2 ; i++){
```

```
    k1_vec[i+1]= H*f1(x1+k1_vec[i]/2,x2,t+H/2 );
    k2_vec[i+1]= H*f2(x1,x2+k2_vec[i]/2,t+H/2 );
  }
  k1_vec[3]= H * f1(x1+k1_vec[2],x2,t+H);
  k2_vec[3]= H * f2(x1,x2+k1_vec[2],t+H);

}
```

Un'ultima funzione, infine, valuta l'incremento effettivo della variabile, a seconda dei valori calcolati dalla funzione riportata sopra e immagazzinati in uno dei due vettori $k1$ o $k2$.
I pesi sono definiti dal metodo Runge-Kutta IV.

```
double increment_k ( double * k ){
  return ( k[0]/3+k[1]/6+k[2]/6+k[3]/3);
}
```