

Relazioni di Laboratorio di Fisica Computazionale

Carlo Sana 16 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:

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

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. 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:

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

Come si può vedere è un polinomio di grado n con la proprietà:

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

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

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

La stima dell'integrale diventa così:

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

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 :

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

dove ξ è un punto interno all'intervallo. L'errore è facilmente sovrestimabile, valutando il massimo della derivata $n+1$ -esima all'interno dell'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 $n = 1$:

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

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:

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

```
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);
}
```

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.

E' fondamentale l'uso di polinomi ortogonali in un certo intervallo $[a, b]$ con il peso $\omega(x)$:

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

Inoltre, un teorema afferma che un polinomio ortogonale di grado n in $[a, b]$ ha n zeri in $[a, b]$.

```
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 ;
}
```

3 Integrali di cammino nell'oscillatore armonico