Integración numérica

Lección 18

Dr. Pablo Alvarado Moya

CE3102 Análisis Numérico para Ingeniería
Área de Ingeniería en Computadores
Tecnológico de Costa Rica

I Semestre 2018

Contenido

Fórmulas de Newton-Cotes de integración abierta

- Las fórmulas de integración abierta tienen límites que se extienden más allá del intervalo de datos
- El tamaño de paso será aquí:

$$h = \frac{b-a}{n}$$

  con n el número de **segmentos**
- Si p es el número de puntos usados, $n = p + 1$
- Los puntos están en $x_i = a + ih$, $i = 1, \ldots, n$

Fórmulas de Newton-Cotes de integración abierta

Se resumen en la tabla las fórmulas de Newton-Cotes abiertas

Segm.	Pts.	Fórmula		Error
2	1	$(b-a)f(x_1)$	Método del punto medio	$(1/3)h^3 f''(\xi)$
3	2	$(b-a)\dfrac{f(x_1)+f(x_2)}{2}$		$(3/4)h^3 f''(\xi)$
4	3	$(b-a)\dfrac{2f(x_1)-f(x_2)+2f(x_3)}{3}$		$(14/45)h^5 f^{(4)}(\xi)$
5	4	$(b-a)\dfrac{11f(x_1)+f(x_2)+f(x_3)+11f(x_4)}{24}$		$(95/144)h^5 f^{(4)}(\xi)$
6	5	$(b-a)\dfrac{11f(x_1)-14f(x_2)+26f(x_3)-14f(x_4)+11f(x_5)}{20}$		$(41/140)h^7 f^{(6)}(\xi)$

Integración de ecuaciones

Integración de ecuaciones

- Hasta ahora lo métodos presentados pueden utilizar puntos aislados conocidos a priori (por ejemplo, provenientes de mediciones)
- Si se tiene la función $f(x)$, entonces es posible evaluarla para cuanto valor de x sea necesario
- Los métodos a evaluar a continuación utilizan la función directamente para evaluar los valores que sean necesarios para reducir el error de la integral progresivamente.
- Transformación de algoritmos de Newton-Cotes para evaluar funciones es directa (de argumento se usa puntero a función, o funtor en vez de pasar vector de valores f)

Integración de Romberg

- Método eficiente para obtener integrales numéricas de funciones.
- Se basa en aplicaciones sucesivas de la regla del trapecio.
- El principio matemático utilizado se denomina **extrapolación de Richardson**

Extrapolación de Richardson (1)

- La estimación del valor exacto I de la integral, el valor $I(h)$ aproximado con regla del trapecio de aplicación múltiple y el error $E(h)$ se relacionan con

$$I = I(h) + E(h)$$

- Si n es el número de segmentos, entonces $h = (b - a)/n$
- Dos estimaciones con pasos h_1 y h_2 deben cumplir

$$I(h_1) + E(h_1) = I(h_2) + E(h_2)$$

- El error para la regla del trapecio de aplicación múltiple es

$$E \approx -\frac{b-a}{12} h^2 \bar{f}''$$

Extrapolación de Richardson (2)

- Si se asume $\bar{f}''$ constante para todo h se obtiene:

$$\frac{E(h_1)}{E(h_2)} \approx \frac{h_1^2}{h_2^2} \qquad \Rightarrow \qquad E(h_1) \approx E(h_2)\left(\frac{h_1}{h_2}\right)^2$$

- Retomando

$$\begin{aligned} I(h_1) + E(h_1) &= I(h_2) + E(h_2) \\ I(h_1) + E(h_2)\left(\frac{h_1}{h_2}\right)^2 &= I(h_2) + E(h_2) \end{aligned}$$

de donde se despeja

$$E(h_2) = \frac{I(h_1) - I(h_2)}{1 - (h_1/h_2)^2}$$

Extrapolación de Richardson (3)

con $I = I(h_2) + E(h_2)$ se obtiene entonces una mejor estimación de la integral

$$I \approx I(h_2) + [I(h_2) - I(h_1)] \frac{1}{(h_1/h_2)^2 - 1}$$

- Para el caso particular $h_2 = h_1/2$ la ecuación anterior es

$$\begin{aligned} I &\approx I(h_2) + \frac{I(h_2) - I(h_1)}{2^2 - 1} \\ &= \frac{4}{3} I(h_2) - \frac{1}{3} I(h_1) \end{aligned}$$

Ejemplo: Extrapolación de Richardson (1)

Ejemplo

Asúmase que conoce las aproximaciones de la integral de $f(x)$ utilizando la regla del trapecio dadas por:

Segm.	h	$I(h)$	ϵ_t %
1	0,8	0,1728	89,5
2	0,4	1,0688	34,9
4	0,2	1,4848	9,5

Calcule una mejor aproximación de la integral. Se sabe que el valor verdadero de la integral es $I = 1,640533$

Ejemplo: Extrapolación de Richardson (2)

Solución: Con la ecuación obtenida anteriormente, y los primeros dos resultados se calcula

$$I \approx \frac{4}{3}(1,0688) - \frac{1}{3}(0,1728) = 1,367467$$

El nuevo error es $\epsilon_t = 16,6\,\%$.
Si se combinan ahora los últimos dos resultados se obtiene:

$$I \approx \frac{4}{3}(1,4848) - \frac{1}{3}(1,0688) = 1,623467$$

para un error $\epsilon_t = 1,0\,\%$

Error de extrapolación de Richardson (1)

- La regla del trapecio tiene un error $\mathcal{O}\left(h^2\right)$
- Se puede demostrar que con la estimación

$$I \approx I(h_2) + [I(h_2) - I(h_1)]\frac{1}{(h_1/h_2)^2 - 1}$$

  se obtiene un error $\mathcal{O}\left(h^4\right)$
- Si se combinan ahora dos estimaciones calculadas con error $\mathcal{O}\left(h^4\right)$ se reduce el error con $\mathcal{O}\left(h^6\right)$; y si se combinan dos de estos resultados se obtiene un error de $\mathcal{O}\left(h^8\right)$.

Error de extrapolación de Richardson (2)

- Con la división sucesiva a la mitad del tamaño la ecuación para una exactitud $\mathcal{O}\left(h^6\right)$ es

$$I \approx \frac{16}{15} I_m - \frac{1}{15} I_l$$

con I_m la estimación más exacta e I_l la estimación menos exacta.

- Para exactitud $\mathcal{O}\left(h^8\right)$ la ecuación es

$$I \approx \frac{64}{63} I_m - \frac{1}{63} I_l$$

- Observe que en los tres casos la suma de los coeficientes es 1

Algoritmo de integración de Romberg

- Las fórmulas anteriores se pueden expresar de forma general como
$$I_{j,k} \approx \frac{4^{k-1} I_{j+1,k-1} - I_{j,k-1}}{4^{k-1} - 1}$$
- $I_{j+1,k-1}$ e $I_{j,k-1}$ son las integrales más y menos exactas, respectivamente
- El subíndice k indica el nivel de integración, con $k = 1$ equivalente a la regla del trapecio, $k = 2$ corresponde a $\mathcal{O}\left(h^4\right)$, etc.
- El subíndice j distingue de estimación más exacta $(j + 1)$ y menos exacta (j)
- La aplicación sistemática de la ecuación anterior se denomina **integración de Romberg**

Ejemplo: Integración de Romberg (1)

Ejemplo

Encuentre la sucesión de estimaciones de la integral de Romberg

Ejemplo: Integración de Romberg (2)

Solución:

Con $h_i = h_{i-1}/2$

$$
\begin{array}{cccccccc}
 & \mathcal{O}\left(h^2\right) & & \mathcal{O}\left(h^4\right) & & \mathcal{O}\left(h^6\right) & & \mathcal{O}\left(h^8\right) \\
h_1 & 0{,}172800 & \rightarrow & 1{,}367467 & \rightarrow & 1{,}640533 & \rightarrow & 1{,}640533 \\
 & & \nearrow & & \nearrow & & \nearrow & \\
h_2 & 0{,}068800 & \rightarrow & 1{,}623467 & \rightarrow & 1{,}640533 & & \\
 & & \nearrow & & \nearrow & & & \\
h_3 & 1{,}484800 & \rightarrow & 1{,}639467 & & & & \\
 & & \nearrow & & & & & \\
h_4 & 1{,}600800 & & & & & &
\end{array}
$$

Criterio de terminación

- Como criterio de terminación se utiliza un umbral con el error relativo

$$|\epsilon_a| = \left| \frac{I_{1,k} - I_{2,k-1}}{I_{1,k}} \right|$$

Número de evaluaciones

- La regla de Simpson $1/3$ de aplicación múltiple requeriría 256 segmentos para alcanzar la precisión dada en el ejemplo.
- Dicha regla está limitada por error de redondeo, por lo que más segmentos no mejoran la estimación
- El algoritmo de Romberg requiere la estimación para 1, 2, 4 y 8 segmentos, esto es: solo 15 evaluaciones, y alcanza mejor precisión.
- Este es el método de preferencia si se tienen funciones suaves (derivables) sin singularidades.

Código de integración de Romberg

```
// F:     tipo de functor
// n:     número de segmentos
// [a,b]: intervalo de integración
template<class F, typename T>
T trapezoid(F f, int n, T a, T b) {
  T h = (b-a)/n;
  T x=a;
  T sum = f(x);
  for (int i=1;i<n;++i) {
    x += h;
    sum += 2*f(x);
  }
  sum += f(b);
  return (b-a)*sum/(2*n);
}
```

Código de integración de Romberg

```
template<class F, typename T>
T romberg(F f, T a, T b,
          int maxIt = 10, T es=std::limits<T>::epsilon()) {
  matrix<T> I(maxIt,maxIt);
  int n=1;
  I[0][0] = trapezoid(f,n,a,b);
  T ea=std::limits<T>::max();
  int i;
  for (i=1;i<maxIt && ea>es;++i) {
    n *= 2;
    I[i][0] = trapezoid(f,n,a,b);
    T fk=T(1);
    for (int k=1;k<=i;++k) {
      j = i - k;
      fk*=4; // esto es pow(4,k);
      I[j][k] = (fk*I[j+1][k-1] - I[j][k-1])/(fk-T(1));
    }
    ea = abs((I[0][i]-I[1][i-1])/I[0][i]); // Corregir!
  }
  return I[0][i-1];
}
```

Cuadratura adaptativa

- Aunque la integración de Romberg es más eficiente que la aplicación múltiple de la regla de Simpson $1/3$, ambas usan puntos equiespaciados
- Se ignora que las funciones tienen regiones con más variabilidad que otras.
- La **cuadratura adaptativa** ajusta el tamaño de paso tal que se usen pasos pequeños en intervalos de variación rápida, y pasos mayores si la función cambia lentamente.
- Se utiliza como base la regla de Simpson $1/3$ en una estructura similar al algoritmo de Romberg
- Recursivamente se parte el problema en dos subintervalos si el error supera un umbral

Algoritmo de cuadratura adaptativa (1)

```
template<class F,typename T>
T quadadapt(F f,T a,T b,T tol=std::limits<T>::epsilon()) {
  T c = (a+b)/2;
  T fa = f(a);
  T fc = f(c);
  T fb = f(b);
  return qstep(a,b,tol,fa,fc,fb);
}
```

Algoritmo de cuadratura adaptativa (2)

```
template<class F,typename T>
T qstep(F f,T a, T b, T tol, T fa, T fc, T fb) {
  T h1 = b-a;
  T h2 = h1/2;
  c = (a+b)/2;
  fd = f( (a+c)/2 );
  fe = f( (c+b)/2 );
  T I1 = h1/6 * (fa+4*fc+fb); // Regla de Simpson 1/3
  T I2 = h2/6 * (fa+4*fd+2*fc+4*fe+fb); // 2x Regla 1/3
  T I();
  if (abs(I2-I1)<=tol) {
    I = I2 + (I2-I1)/15; // Mejoramiento como Richardson
  } else {
    Ia = qstep(a,c,tol,fa,fd,fc);
    Ib = qstep(c,b,tol,fc,fe,fb);
    I = Ia+Ib;
  }
  return I;
}
```

Resumen

1. Fórmulas de integración abierta

2. Integración de ecuaciones
    - Extrapolación de Richardson
    - Integración de Romberg
    - Cuadratura adaptativa

Este documento ha sido elaborado con software libre incluyendo LaTeX, Beamer, GNUPlot, GNU/Octave, XFig, Inkscape, LTI-Lib-2, GNU-Make y Subversion en GNU/Linux