Kode Makalah M-6

PENYELESAIAN INTEGRASI NUMERIK DENGAN MATLAB

Ratna Widyati

Jurusan Matematika, FMIPA

Universitas Negeri Jakarta

ABSTRAK

Metode numerik dapat digunakan untuk menghampiri integrasi yang dapat menyelesaikan masalah dengan cepat tanpa harus menghitung secara manual. Salah satu metode yang digunakan adalah metode *Newton Cotes* yaitu metode yang umum untuk menurunkan kaidah integrasi numerik. Polinom interpolasi menjadi dasar metode *Newton Cotes*, yaitu menghampiri fungsi f(x) dengan polinom interpolasi $p_n(x)$. Dengan bantuan perangkat lunak Matlab maka hasil yang diperoleh dapat lebih cepat, tepat dan akurat dibanding cara manual biasa.

Kata kunci : *Newton-Cotes*, integrasi numerik, polinom interpolasi, Matlab

PENDAHULUAN

Integral mempunyai banyak terapan dalam bidang sains dan rekayasa. Dalam praktek rekayasa seringkali fungsi yang dintegrasikan adalah fungsi empirik yang diberikan dalam bentuk tabel atau *integrand*-nya, tidak dalam bentuk fungsi elementer (seperti sinh x, fungsi Gamma, dan sebagainya) atau fungsi eksplisit f yang terlalu rumit untuk dintegralkan. Oleh sebab itu metode numerik dapat digunakan untuk menghampiri integrasi. Dalam penyelesaian masalah integrasi numerik, penggunaan software atau perangkat lunak sangatlah membantu karena dapat mengetahui penyelesaiannya secara cepat sehingga dapat menganalisis persoalan dengan tepat dan akurat. Perangkat lunak yang digunakan untuk penyelesaian masalah adalah Matlab. Menurut Higham (2004), Matlab memiliki sistem yang interaktif untuk perhitungan secara numerik. Sedangkan menurut Webb Peter, dkk (1999) bahasa Matlab untuk numerik telah dikembangkan agar digunakan sesuai dengan aturan sains dan rekayasa serta bahasa *script* seperti Visual Basic dan Perl membantu dalam aplikasi non numerik.

Rumusan Masalah

Permasalahan integrasi numerik adalah menghitung secara numerik integral tentu yaitu

$$I = \int_a^b f(x)\,dx$$

yang dalam hal ini a dan b adalah batas-batas integrasi, f adalah fungsi yang dapat diberikan secara eksplisit dalam bentuk persamaan ataupun secara empirik dalam bentuk tabel nilai. Kemudian masalah ini akan diselesaikan dengan menggunakan Matlab 7 *Release* 14 sebagai perangkat lunaknya.

Tujuan

Penelitian ini bertujuan untuk mempermudah dan mempercepat perhitungan integrasi numerik dengan menggunakan perangkat lunak Matlab dibandingkan dengan cara manual biasa. Selain itu, hasil yang diperoleh dari suatu masalah diharapkan dapat lebih tepat dan akurat.

Pembatasan Masalah

Dalam menurunkan rumus integrasi numerik terdapat tiga pendekatan, yaitu berdasarkan tafsiran gometri integral tentu, polinom interpolasi dan kuadratur Gauss. Yang akan dibahas pada penelitian kali ini adalah berdasarkan pendekatan kedua, yaitu polinom interpolasi dengan rumus integrasi numerik yang diturunkan digolongkan ke dalam metode *Newton Cotes*.

METODE PENELITIAN

Metode *Newton Cotes* adalah metode yang umum untuk menurunkan kaidah integrasi numerik. Menurut Munir (2003), polinom interpolasi menjadi dasar metode *Newton Cotes* yaitu menghampiri fungsi $f(x)$ dengan polinom interpolasi $p_n(x)$

$$I = \int_a^b f(x)\,dx = \int_a^b p_n(x)\,dx$$

yang dalam hal ini,

$$p_n(x) = a_0 + a_1x + a_2x^2 + \dots + a_{n-1x}^{n-1} + a_nx^n$$

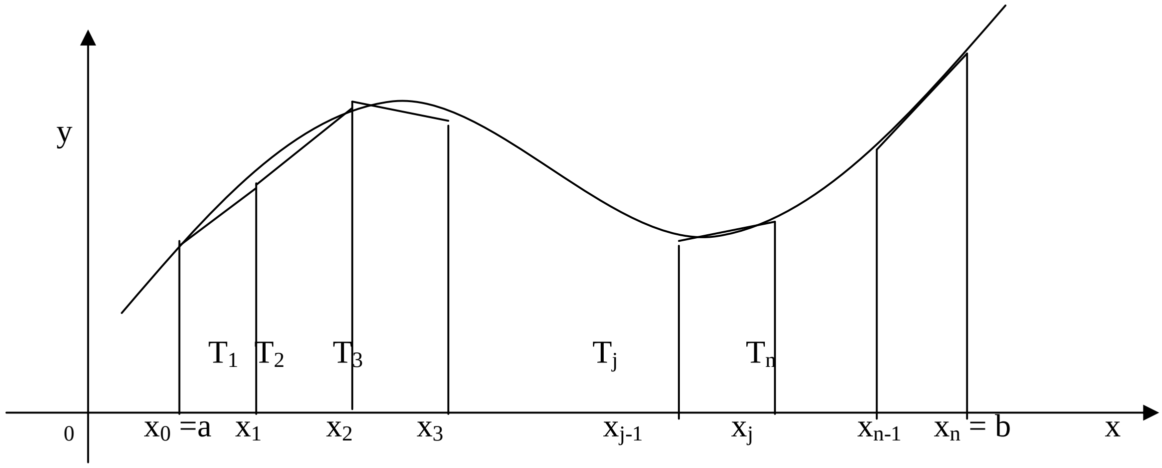
Polinom interpolasi digunakan karena suku-suku polinom mudah diintegralkan dengan rumus integral yang sudah baku yaitu

$$\int ax^n \, dx = \frac{a}{n+1} x^{n+1} + C$$

Dari beberapa kaidah integrasi numerik yang diturunkan dari Metode *Newton Cotes*, dua diantaranya yang akan diselesaikan dengan Matlab adalah Kaidah Trapesium (*Trapezoidal Rule)* dan Kaidah Simpson 1/3 *(Simpson 1/3 Rule*).

- Kaidah Trapesium

  Interval [a,b] dipartisi menjadi n interval yang panjangnya sama, (b-a)/n = h.



$$I(f) = \int_a^b f(x)dx = \int_{x_0}^{x_1} f(x)dx + \mathbf{L} + \int_{x_{j-1}}^{x_j} f(x)dx + \mathbf{L} + \int_{x_{n-1}}^{x_n} f(x)dx$$

Setiap integral bagian nilainya dihampiri oleh luas trapesium sehingga diperoleh:

$$\int_{x_{j-1}}^{x_j} f(x)dx = \{ f(x_{j-1}) + f(x_j)\} \frac{h}{2} = \frac{h}{2} f(x_{j-1}) + \frac{h}{2} f(x_j)$$

T_j =

$I(f) = T_1 + T_2 + \ldots + T_j + \ldots + T_n$

$= \{ \frac{h}{2} f(x_0) + \frac{h}{2} f(x_1)\} + \{ \frac{h}{2} f(x_1) + \frac{h}{2} f(x_2)\} + \ldots + \{ \frac{h}{2} f(x_{j-1}) +$

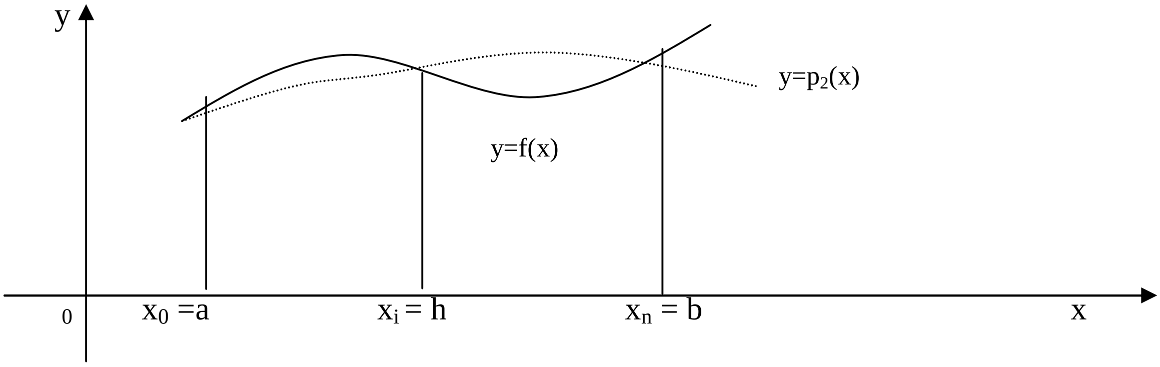
$\frac{h}{2} f(x_j)\} + \ldots + \{ \frac{h}{2} f(x_{n-1}) + \frac{h}{2} f(x_n)\}$

$= \frac{h}{2}(f(x_0) + f(x_1)) + \frac{h}{2}(f(x_1) + f(x_2)) + \ldots + \frac{h}{2}(f(x_{n-1}) + f(x_n))$

$= \frac{h}{2} \{f(x_0) + f(x_n) + 2 \sum_{i=1}^{n-1} f(x_i))$

- Kaidah Simpson 1/3

Interval [a,b] dipartisi menjadi n interval yang panjangnya sama, (b-a)/n = h.



Pada aturan Simpson, misalkan fungsi *f(x)* dihampiri dengan polinom interpolasi derajat 2 yang grafiknya berbentuk parabola. Luas daerah yang dihitung sebagai hampiran nilai integrasi adalah daerah dibawah parabola. Untuk itu dibutuhkan 3 buah titik data, misalkan (0,*f(0)*), (*h*,*f(h)*), dan (*2h*,*f(2h)*). Polinom interpolasi derajat 2 yang melalui ketiga titik tersebut adalah :

$$P_2(x) = f(x_0) + \frac{x}{h}\Delta f(x_0) + \frac{x(x-h)}{2!h^2}\Delta^2 f_0$$

Integrasikan $p_2(x)$ di dalam interval [0,2h] sehingga diperoleh :

$$I \approx \int_0^{2h} f(x)\,dx \approx \int_0^{2h} P_1\,dx$$

$$\approx \frac{h}{3}(f_0 + 4f_1 + f2)$$

Misalkan kurva sepanjang interval integrasi [a,b] dibagi menjadi n+1 buah titik diskrit x_0, x_1, …, x_n dengan n genap dan setiap tiga buah titik dikurva dihampiri dengan parabola (polinom interpolasi derajat 2) maka akan dipunyai n/2 buah potongan parabola. Bila masing-masing polinom derajat dua tersebut diintegralkan didalam subinterval integrasinya, maka jumlah seluruh integral tersebut membentuk kaidah Simpson 1/3 gabungan :

$$I_{tot} = \int_a^b f(x)\,dx = \int_{x_0}^{x_2} f(x)\,dx + \int_{x_2}^{x_4} f(x)\,dx + \ldots + \int_{x_{n-2}}^{x_n} f(x)\,dx$$

$$\approx \frac{h}{3}(f_0 + 4f_1 + f_2) + \frac{h}{3}(f_2 + 4f_3 + f_4) + \ldots + \frac{h}{3}(f_{n-2} + 4f_{n-1} + f_n)$$

$$\approx \frac{h}{3}(f_0 + 4f_1 + 2f_2 + 4f_3 + 2f_4 + \ldots + 2f_{n-2} + 4f_{n-1} + f_n)$$

$$\approx \frac{h}{3}\left(f_0 + 4\sum_{i=1,3,5}^{n-1} f_i + 2\sum_{i=2,4,6}^{n-2} f_i + f_n \right)$$

HASIL PENELITIAN DAN PEMBAHASAN

Teknik Integrasi numeris menghitung fungsi *f(x)* dengan fungsi lain *g(x)*, dimana *g(x)* dipilih sehingga dengan mudah dapat menghitung bidang dibawah *g(x)*. Menurut Etter, D.M, dkk (2002), makin baik perhitungan *g(x)* ke *f(x)*, makin baik pula perhitungan dari integral *f(x).* Sedangkan menurut Susila, I. Nyoman (1993), dua dari teknik integrasi yang paling umum yaitu menghitung *f(x)* dengan sekumpulan fungsi linear potongan atau dengan sekumpulan fungsi parabolis potongan. Bila menghitung fungsi dengan fungsi linear potongan kemudian dapat menghitung bidang dari trapesium yang menyusun bidang di bawah fungsi linear potongan maka teknik ini disebut Kaidah Trapesium. Sedangkan bila menghitung fungsi dengan fungsi kuadratis potongan, kemudian dapat menghitung dan menambahkan bidang dari komponen ini disebut dengan Kaidah Simpson.

A. Kaidah Simpson 1/3

Hampiran nilai integrasi yang lebih baik dapat ditingkatkan dengan menggunakan polinom interpolasi berderajat lebih tinggi. Misalkan fungsi *f(x)* dihampiri dengan polinom interpolasi derajat 2 yang grafiknya berbentuk parabola. Luas daerah yang dihitung sebagai hampiran nilai integrasi adalah daerah dibawah parabola. Untuk itu dibutuhkan 3 buah titik data, misalkan (0,*f(0)*), (*h*,*f(h)*), dan (*2h*,*f(2h)*). Dalam kaidah ini, sebutan 1/3 muncul karena didalam rumus penyelesaiannya terdapat faktor 1/3 (sekaligus untuk membedakan kaidah Simpson yang lain yaitu 3/8). Menurut Liendfield, G. dan Penny, John (2000), dalam penggunaan kaidah Simpson 1/3 mensyaratkan jumlah ***subinterval*** (n) harus genap, hal ini berbeda dengan kaidah trapesium yang tidak mempunyai persyaratan mengenai jumlah subinterval.

Dimisalkan fungsi yang akan diselesaikan secara Matlab dengan menggunakan kaidah Simpson 1/3 adalah :

$$I = \int_0^1 e^{x^2}\, dx$$

maka penyelesaian untuk fungsi diatas dengan menggunakan Matlab adalah sebagai berikut :

```
%Kaidah Simpson 1/3 menggunakan vektor
function q=simp1(func,a,b,m)
h=(b-a)/m;
x=[a:h:b]; y=feval(func,x);
v=2*ones(m+1,1);
v2=2*ones(m/2,1);
v(2:2:m)=v(2:2:m)+v2;
v(1)=1; v(m+1)=1;
q=y*v;
q=q*h/3;
fv=exp(x.^2);
```

Hasil yang diperoleh adalah sebagai berikut :

```
>> n=4; i=1;
>> disp(‘ n    Nilai Integral’);
>> while n<128
simpval=simp1('f402',0,1,n);
fprintf('%3.0f%14.9f\n',n,simpval);
n=2*n; i=i+1;
end
 n    Nilai Integral
  4   1.463710760
  8   1.462723415
 16   1.462656321
 32   1.462652033
 64   1.462651764
>>
```

B. Kaidah Trapesium

Dengan menggunakan fungsi yang sama, yaitu :

$$I = \int_0^1 e^{x^2}\, dx$$

maka penyelesaian untuk fungsi diatas dengan kaidah trapesium dan diselesaikan dengan menggunakan Matlab adalah sebagai berikut :

```
function fv=f402(x)
  fv=exp(x.^2);
```

Setelah dijalankan, maka hasil yang diperoleh adalah :

```
>> n=4; i=1;
>> disp(‘ n    Nilai Integral’);
>> while n<128
h=1/n; x=0:h:1; f=f402(x);
trapval=h*trapz(f);
fprintf('%3.0f%14.9f\n',n,trapval);
n=2*n; i=i+1;
end
 n    Nilai Integral
  4   1.490678862
  8   1.469712276
 16   1.464420310
 32   1.463094103
 64   1.462762349
>>
```

KESIMPULAN

Penyelesaian integral secara numerik dengan alat bantu perangkat lunak Matlab diharapkan dapat memberikan solusi yang lebih cepat, tepat dan mudah. Dengan menggunakan metode Newton Cotes, yaitu kaidah Simpson 1/3 dan Trapesium maka dapat dilihat perbedaan ketelitian dari masing-masing kaidah. Kaidah Simpson 1/3 mempunyai ketelitian yang lebih tinggi dibanding dengan kaidah Trapesium karena mempunyai polinomial berderajat lebih tinggi.

DAFTAR PUSTAKA

Etter,D.M. Kuncicky, D.C., with Hull, Doug., 2002, *Pengantar Matlab 6*, Indeks, Jakarta.

Higham, Nicholas, J., 2004, **Matlab: A Tool for Teaching and Research**, Department of Mathematics, University of Manchester.

Liendfield, George, dan Penny, John, 2000, *Numerical Methods using Matlab*, Prentice Hall, New Jersey.

Munir, Rinaldi, 2003, *Metode Numerik*, Informatika, Bandung.

Susila, I. Nyoman, 1993, *Dasar-Dasar Metode Numerik*, DEPDIKBUD, Jakarta.

Webb, Peter and Wilson, Gregory V., 1999, **Matlab as a Scripting Language**, Dr. Dobb’s Journal.