

Počítačová podpora aplikácií numerickej kvadratúry

Alena Vagaská

*Technická univerzita v Košiciach, Fakulta výrobných technológií, Katedra matematiky,
informatiky a kybernetiky
e-mail: vagaska.alena@fvt.sk*

Abstrakt

In this paper will be described some possibilities of using the program system Matlab to applications of numeric quadrature.

1. Úvod

Na technických univerzitách by malo byť samozrejmosťou, že učiteľ matematiky má záujem prehliadnúť aplikačný charakter matematických predmetov, napr. zavádzaním technických aplikácií matematiky do výučby. Taktiež by malo byť samozrejmé, že naučí študentov zvládnuť zvyšujúce sa nároky na rýchlosť a presnosť matematických výpočtov v takýchto aplikáciách pomocou využitia tzv. CAS systémov (Matlab, Maple, ...). Pozrime sa, aké možnosti nám ponúka Matlab v aplikáciách numerickej kvadratúry pri riešení aplikačných diferenciálnych rovníc.

2. Aplikácie numerickej kvadratúry

Ukážme si, ako môžeme pomocou Matlabu numericke vypočítať určitý integrál, ku ktorému sa dostaneme po aplikácii teórie diferenciálnych rovníc pri riešení nasledovného problému.

Každý tepelný proces, teda aj ohrev vody, môžeme popísať diferenciálnou rovnicou. Ak by sme mali určiť čas t , za ktorý elektrickou špirálou ohrejeme 1 kg vody z izbovej teploty $T = 20^\circ\text{C}$ na teplotu varu $T = 100^\circ\text{C}$, ak vieme, že elektrické napätie je 120 V, odpor špirály $14,4\Omega$ a je známe, že 1 kg vody ochladne zo 40°C na 30°C za 10 minút; tak počas riešenia sa dopracujeme k diferenciálnej rovnici ohrevu vody [1], ktorá za daných podmienok vyzerá nasledovne:

$$\frac{dT}{dt} = 0,24 - \frac{\ln 2}{600} \cdot T + \frac{\ln 2}{30}. \quad (1)$$

Separáciou premenných v rovnici (1) a aplikáciou určitého integrálu nájdeme hľadaný čas t :

$$t = 600 \int_{20}^{100} \frac{dT}{144 + 20 \ln 2 - T \ln 2} = -\frac{600}{\ln 2} [\ln|144 + 20 \ln 2 - T \ln 2|]_{20}^{100} \approx 421 \text{ s}. \quad (2)$$

Hľadaný čas t ohrevu vody na požadovanú teplotu je teda $t \approx 421 \text{ s} = 7 \text{ min } 1 \text{ s}$.

Je zrejmé, že určitý integrál zo vzťahu (2) sa dá vypočítať aj analyticky, jeho presná hodnota je $\left(\frac{-600}{\ln 2}\right) \cdot (\ln|144 + 20 \ln 2 - 100 \ln 2| - \ln 144) = 420,9201059$. Vďaka tomu pri numerickej aproximácii integrálu (2) budeme vedieť určiť, akej skutočnej chyby sa pri použití jednotlivých metód dopustíme.

Pri numerickej kvadratúre integrálu (2) použijeme známe Newton-Cotesove vzorce, ktorých podstatou je aproximácia určitého integrálu $\int_a^b f(x)dx$ funkcie f na intervale $[a,b]$ na základe hodnôt funkcie f v konečnom počte bodov x_i intervalu $[a,b]$. Interval $[a,b]$ sa rozdelí sieťou bodov na jednotlivé podintervaly, na ktorých sa funkcia f aproximuje interpolačným polynómom P^i . V závislosti od stupňa s interpolačného polynómu P^i získavame odpovedajúce elementárne N-C vzorce: obdĺžnikový, lichobežníkový, Simpsonov, Booleov a Milneho vzorec (písané postupne pre $s=0,1,2,3,4$, kde $s=m-1$, m je počet uzlov interpolácie). Pre elementárne N-C vzorce môžeme písať všeobecný kvadrატúrny vzorec $I=(b-a) \cdot \mathbf{w} \cdot \mathbf{f}$ v ktorom $\mathbf{w}$ je riadkový vektor váh a $\mathbf{f}$ je stĺpcový vektor funkčných hodnôt (f -hodnôt) v uzloch interpolácie, t.j. $\mathbf{f}=[f_1, f_2, \dots, f_m]^T$. Pre riadkový vektor váh platí: ak $m=2$, $\mathbf{w}=\left[\frac{1}{2}, \frac{1}{2}\right]$; ak $m=3$, $\mathbf{w}=\left[\frac{1}{6}, \frac{4}{6}, \frac{1}{6}\right]$; ak $m=4$ $\mathbf{w}=\left[\frac{1}{8}, \frac{3}{8}, \frac{3}{8}, \frac{1}{8}\right]$, ak $m=5$, $\mathbf{w}=\left[\frac{7}{90}, \frac{32}{90}, \frac{12}{90}, \frac{32}{90}, \frac{7}{90}\right]$. Aplikovaním elementárnych N-C vzorcov na jednotlivé podintervaly získame zložené N-C vzorce[2].

3. Matlab a M-funkcie v numerickom integrovaní

Pri aproximácii určitého integrálu zo vzťahu (2) nám Matlab ponúka viaceré možnosti. Buď editujeme príkazy (podľa jednotlivých N-C vzorcov) rovno do promptov, čo je zdĺhavejšie, alebo môžeme použiť M-funkcie. Uvedme si najprv ukážku numerickej integrácie v Matlabe bez použitia M-funkcií. Zvoľme pre náš prípad numerickú integráciu pomocou lichobežníkovej a Simpsonovej metódy s vopred daným počtom podintervalov $n=8$. Využívajúc lichobežníkový (3) a Simpsonov (4) zložený vzorec

$$\int_a^b f(T) dT = \frac{h}{2} \cdot (T_0 + 2(T_1 + \dots + T_{n-1}) + T_n), \quad (3)$$

$$\int_a^b f(T) dT = \frac{h}{3} \cdot (T_0 + 4T_1 + 2T_2 + 4T_3 + 2T_4 + \dots + 4T_{n-1} + T_n), \quad (4)$$

budeme do jednotlivých promptov editovať nasledujúce príkazy, vďaka ktorým po odklepnutí cez Enter získavame aproximáciu určitého integrálu.

```
>> a=20;b=100; h=(b-a)/8; x=linspace(a, b, 9)
x =
    20    30    40    50    60    70    80    90   100
>> fx=600*((144 + 20*log(2) - x.*log(2)).^(-1))
fx =
    4.1667    4.3774    4.6105    4.8699    5.1602    5.4873    5.8587    6.2841    6.7760
>> Lich=h*( (fx(1)+fx(9))/2 + sum(fx(2:8)) )
Lich =
    421.1948
>> Simp=(h/3)*(fx(1)+4*fx(2)+2*fx(3)+4*fx(4)+2*fx(5)+4*fx(6)+2*fx(7)+4*fx(8)+fx(9))
Simp =
    420.9210
```

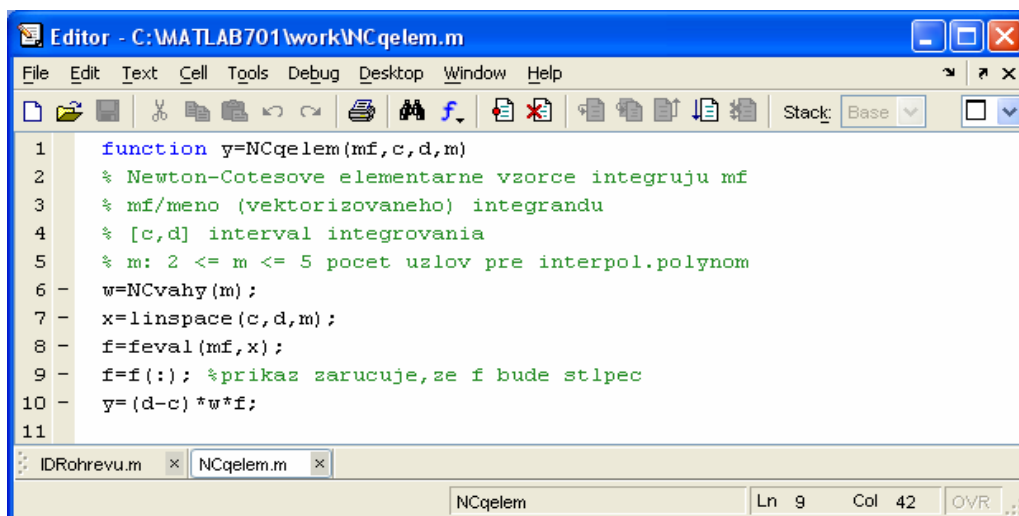
Využitím zložených N-C vzorcov už pri Simpsonovej metóde dostávame celkom presný výsledok. Ak by sme sa rozhodli zvyšovať presnosť aproximácie zvyšovaním počtu

podintervalov, či využiť Booleov alebo Milneho vzorec, museli by sme vždy nanovo editovať zmenené zložené N-C vzorce do príkazových riadkov, čo by bolo pomerne pracné. Tomu možno predísť využitím M-funkcií, ktoré umožňujú modifikovať kolekciu príkazov (meniť N-C vzorce či počet podintervalov) bez pracovného vpisovania zmien do promptov.

M-funkcia je (po obsahovej stránke) postupnosť príkazov zapísaná do súboru pod nejakým menom. Súbor musí mať vždy rovnaké meno ako funkcia. M-funkciu napíšeme v editore Matlabu, ktorý vyvoláme cez Menu: File/New/M-file. Prvý riadok M-funkcie musí začínať kľúčovým slovom `function`, za ním nasleduje vektor výstupných argumentov, meno funkcie a v okrúhlych zátvorkách mená vstupných argumentov. Do druhého a ďalších riadkov M-funkcie je vhodné písať komentár. Takéto riadky musia začínať symbolom „%“, vtedy ich interpreter Matlabu ignoruje. Vďaka komentáru bude pre nás M-funkcia zrozumiteľná aj po dlhšom čase a taktiež si pripomenieme význam a poradie vstupných a výstupných argumentov, čo je nevyhnutné pre správne volanie funkcie. Ukážme si príklad prvej M-funkcie nazvanej „NCvahy“, ktorá nám poskytuje vektory váh, potrebné v druhej M-funkcii nazvanej „NCqelem“ (viď obr.1).

```
function w=NCvahy(m)
%Vahy pre Newton-Cotesove formule (uzavreteho typu)
% m: 2 <= m <= 5 pocet uzlov pre interpolacny polynom
if m==2
    w=[1 1]/2;           %lichobeznikovy vzorec
elseif m==3
    w=[1 4 1]/6;        %Simpsonov vzorec
elseif m==4
    w=[1 3 3 1]/8       %Booleov vzorec
else
    w=[7 32 12 32 7]/90; %Milneho vzorec
end
```

Názov „NCqelem“ nie je náhodný. Písmeno „q“ je z anglického slova „quadrature“, „elem“ znamená „elementárne“. Funkcia „NCqelem“ totiž realizuje kvadratúru Newton-Cotesovými kvadratúrnymi elementárnymi vzorcami podľa vzťahu $I = (d - c) \cdot \mathbf{w} \cdot \mathbf{f}$, ak interval integrovania je $[c, d]$, $\mathbf{w}$ je riadkový vektor váh a $\mathbf{f}$ je stĺpcový vektor funkčných hodnôt funkcie v uzloch interpolácie.



Obr.1 Zápis M-funkcie *NCqelem* v editore Matlabu

Ak chceme pomocou M-funkcie „NCqelem“ aproximovať určitý integrál (2), ku ktorému sme dospeli riešením diferenciálnej rovnice ohrevu, tak je vhodné funkciu vystupujúcu ako integrand v (2) zapísať ako M-funkciu napr. s názvom „IDRohrevu“.

```
function y=IDRohrevu(x)
% num.kvadratura integralu z DR ohrevu
y=600*((144+20*log(2)-x.*log(2)).^(-1))
```

Podľa komentára funkcie „NCqelem“ je zrejmé, že „IDRohrevu“ bude prvým vstupným argumentom M-funkcie „NCqelem“. Nasledujúcou ukážkou použitia M-funkcie „NCqelem“ ilustrujeme, že pri kvadratúre elementárnymi vzorcami zvyšovaním počtu uzlov interpolácie m v intervale $[c, d] = [20, 100]$ sa zvyšuje presnosť integrovania. Je nám jasné, že prvý výsledok sme získali pomocou elementárneho lichobežníkového vzorca (2 uzly interpolácie), druhý elementárnym Simpsonovým vzorcom (3 uzly), tretí Booleovým vzorcom (4 uzly) a štvrtý výsledok pomocou elementárneho Milneho vzorca (5 uzlov).

```
>> format long
>> for i=2:5, aprox(i)=NCqelem('IDRohrevu',20,100,i); end, aprox(2:5)
ans =
```

1.0e+002 *

4.37705412497180

4.21113536472869

4.21008046716879

4.20922133759089

Vidíme, že zvyšovanie počtu uzlov interpolácie (a tým aj stupňa interpolačného polynómu) viedlo k zvýšeniu presnosti elementárneho N-C vzorca. No pre funkcie, o ktorých je známe, ich interpolačné polynómy s rovnomerne rozloženými uzlami neaproximujú dobre (napr. Rungeho funkcia) to neplatí. Preto je výhodnejšie najprv rozdeliť interval $[c, d]$ na podintervaly a až na nich uplatniť elementárne N-C kvadráturálne vzorce. Vtedy hovoríme o kvadratúre pomocou zložených N-C vzorcov. To, že týmto spôsobom získame presnejšie výsledky už pri nižšom stupni interpolačného polynómu, nám dokazujú výsledky, ktoré sme získali pomocou zloženého lichobežníkového a Simpsonovho vzorca priamym editovaním príkazov do promptov.

Ukážme si, že aj pri tzv. zloženej kvadratúre môžeme výhodne využiť M-funkcie, vďaka čomu sa rýchlo dopracujeme k výsledkom pri ľubovoľnej modifikácii vstupných argumentov. Vytvoríme si M-funkciu s názvom „NCqzloz“ podľa danej predlohy:

```
function y=NCqzloz(maf,a,b,m,n)
% Newton-Cotesova kvadratura na [a,b]
% m - pocet uzlov elem.vzorca na podintervale
% n - pocet podintervalov na [a,b]
%
w=NCvahy(m); %vektor vah
x=linspace(a,b,n*(m-1)+1); %vektor uzlov na [a,b]
fx=feval(maf,x);
fx=fx(:); %fx bude stlpcovy vektor f-hodnot
y=0;
index1=1;
index2=m;
for i=1:n
    y=y + w*fx(index1:index2);
    index1=index2;
    index2=index2 + m-1;
end
y=y*(b-a)/n;
```

Aproximujme určitý integrál z (2) pomocou funkcie „NCqzloz“. Ak sa rozhodneme, že počet podintervalov bude napríklad $n = 8$ a na každom z nich použijeme Booleov vzorec, t.j. $m = 4$, jednoduchým volaním funkcie rýchle získame výsledok – bez pracného vypisovania zložených vzorcov do príkazových riadkov.

```
>> NCqzloz('IDRohrevu', 20, 100, 4, 8)
ans =
    4.209201316352159e+002
```

4. Záver

Matlab nám pri numerickej kvadratúre ponúka ešte aj M-funkciu „quadl“ (máme ju k dispozícii napr. vo verzii MATLAB 7), ktorej podstatou sú moderné metódy numerického integrovania používajúce adaptívne algoritmy. Ide o to, že M-funkcia používa adaptívny NC vzorec, čo znamená, že algoritmus optimálne určuje polohu bodov, v ktorých sa vyčísľujú hodnoty funkcie (nejde teda o rovnomerné rozloženie uzlov interpolácie). Napr. pre náš prípad by volanie funkcie vyzeralo takto:

```
>> [nI,k]=quadl('IDRohrevu', 20, 100, 5e-8)
nI =
    4.209201059354103e+002
```

Volanie funkcie aj so štvrtým vstupným argumentom umožňuje stanoviť chybu výpočtu. Na základe skúsenosti s výučbou cvičení z predmetu Numerická matematika a štatistika môžeme vysloviť presvedčenie, že vhodnou kombináciou klasických metód vyučovania s novými metódami, využívajúcimi softvérovu podporu, docielime skvalitnenie a zefektívnenie edukačného procesu.

Literatúra

- [1] Пономарев, К. К.: Составление дифференциальных уравнений. Минск: Вышэйшая школа, Минск, 1973
- [2] Volauf, P.: Numerické a štatistické výpočty v Matlabe. Bratislava: STU, 2005, ISBN 80-227-2259-6