Liet. matem. rink., T. 41, spec. nr., 184–190

Kompiuterinės algebros ir skaitinių metodų sąsaja

Aleksas DOMARKAS (VU, LKA), Rimantas-Jonas RAKAUSKAS (LKA),
Silvijus CICĖNAS(VU)
el. paštas: aleksas@ieva.mif.vu.lt, rimantas.rakauskas@tmk.lka.lt

1. Įvadas

Darbe nagrinėjamas vienas potencialo teorijos uždavinys, kurio sprendimui panaudojame MAPLE ir MATLAB sistemas. Tuo yra parodoma kompiuterinės algebros ir skaitinių metodų sąsaja. Ši sąsaja gali būti pritaikyta ir kitų uždavinių sprendimui.

Erdvėje išdėstytų N sferų ($N \geqslant 2$) potencialas u Dekarto koordinatėse apytiksliai yra išreiškiamas formule

$$u = \sum_{k=1}^{N} \sum_{l=0}^{lmax} \sum_{m=-l}^{l} \frac{b_{k,l,m}\mathrm{CSH}(l, m, r_k)}{|r_k|^{(2l+1)}}, \tag{1}$$

čia $lmax$ yra skleidinio eilė, CSH yra kompleksinės sferinės funkcijos (Complex Solid Harmonic), $r_k = (x - x_k, y - y_k, z - z_k)$, x_k, y_k, z_k – k-osios sferos centro koordinatės, koeficientai $b_{k,l,m}$ yra randami iš lygčių sistemos

$$\sum_{\substack{j=1\\ j\neq i}}^{N} \sum_{l_1=-l}^{l} \sum_{m_1=-l}^{l} b_{j,l_1,m_1}\mathrm{c}(l_1, m_1, l, m, i, j) r_{0i}^{l} (\varepsilon_i - \varepsilon)\, l$$
$$+ \frac{b_{i,l,m}(\varepsilon_i l + \varepsilon(l+1))}{r_{0i}^{l}} = -A_{i,l,m} l r_{0i}^{l} (\varepsilon_i - \varepsilon), \tag{2}$$

$i = 1 \dots N, l = 1 \dots lmax, m = -l \dots l$, koeficientai $\mathrm{c}(l_1, m_1, l, m, i, j)$ yra išreiškiami per sferines funkcijas ir Klebšo-Gordano koeficientus, r_{0i} – sferų spinduliai, $\varepsilon_i, \varepsilon$ – sferų ir aplinkos dielektrinės konstantos, koeficientai charakterizuoja išorinį lauką (žr. [1], [2]). Ši sistema tiesinė ir iš viso turi $N(lmax + 1)^2$ lygčių. Šiame darbe pasitelkiant MAPLE ir MATLAB sistemas randama analizinė sprendinio išraiška.

Lygčių sistema (tiksliau koeficientų matrica) yra formuojama kompiuterinės algebros MAPLE 6 sistemoje. Kompleksines sferines funkcijas mes naudojame iš paketo *orbitals*, kuris yra MAPLE R4 ir MAPLE R5 Schare bibliotekose. Klebšo-Gordano koeficientams apskaičiuoti yra sudaryta procedūra $cg(j1, m1, j2, m2, j3, m3)$. Analogiška procedūra yra FORTRAN kalboje ir Mathematica sistemoje. Bendru atveju gaunama tiesinių lygčių sistema su kompleksiniais koeficientais. Sistemos sprendimui mes naudojame MATLAB.

Gautą sprendinį mes perkeliame iš MATLAB į MAPLE ir konstruojame analizinę potencialo išraišką. Rastas potencialas tiriamas analitiškai ir grafiškai. Kaip pavyzdį mes demonstruojame penkių vienetinių sferų uždavinį.

Naudojant 2xPIII 800 MHz 1Gb RAM kompiuterį šis uždavinys yra sėkmingai sprendžiamas, kai gaunamos sistemos lygčių skaičius yra mažesnis negu 1500. Didesnę spendimo laiko dalį užima lygčių sistemos formavimas. Šį laiką gerokai galėtų sutrumpinti išankstinis Klebšo-Gordano koeficientų apskaičiavimas ir jų kompaktiškas saugojimas kompiuterio atmintyje. Jei sferos yra išsidėsčiusios simetriškai, tai pasinaudojant šia simetrija, sistemos eilę galima sumažinti.

2. Lygčių sistemos formavimas

Toliau yra pateikiama MAPLE 6 programa. Tarpiniai rezultatai yra neišvedami.

```
> restart;
> lmax:=3:
> with(share):readshare(orbitals,science): with(orbitals):
```

Aplinkos ε:

```
> epsilon:=1:
```

Įvedame sferų skaičių, centrų koordinates, spindulius ir jų ε:

```
> N:=5:
> S1:=[2,2,0],1,20:
> S2:=[2,-2,0],1,20:
> S3:=[-2,2,0],1,20:
> S4:=[-2,-2,0],1,20:
> S5:=[0,0,4],1,20:
> sk:=N*(lmax+1)^2:
> AA:=(i,j,k)->if i=1 and j=0 then 1 else 0 fi:
> for k to N do r||k:=S||k[2];epsilon||k:=S||k[3]; od:
> for i to N do for j  to N do
> rv||i||j:=S||j[1]-S||i[1];r||i||j:=linalg[norm](%,2);od;od:
```

Klebšo-Gordano koeficientų skaičiavimo procedūra:

```
> cg:=proc(j1,m1,j2,m2,j3,m3)
> local fa, NUMAX;
> fa:=x->if x<0 then 0 else 1/x! fi;
> NUMAX:=min(j1+j2-j3,j1-m1,j2+m2)+1;
> if abs(j1-j2)>j3
> or j3>j1+j2
> or m3<>m1+m2
```

```
> or 2*max(j1,j2,j3)-j1-j2-j3>0
> or  (j1<abs(m1) or j2<abs(m2) or j3<abs(m3))
> then 0 else
> (-1)^(j1-j2+m3)*sqrt((2*j3+1)*(j1+j2-j3)!*
> (j1-j2+j3)!*(-j1+j2+j3)!/(j1+j2+j3+1)!*
> (j1+m1)!*(j1-m1)!*(j2+m2)!*(j2-m2)!*(j3+m3)!*(j3-m3)!)*
> sum('(-1)^(z+j1-j2-m3)*(fa(z)*fa(j1+j2-j3-z)*fa(j1-m1-z)*
> fa(j2+m2-z)*fa(j3-j2+m1+z)*fa(j3-j1-m2+z))','z'=0..NUMAX);
> fi;
> RETURN(%);
> end proc:

> eqn:=(i,j)->if j=i then B||i[l,m]*
> (epsilon||i*l+epsilon*(l+1))/(r||i)^l
> else Sum(Sum('B||j[l1,m1]*c(l1,m1,l,m,i,j)*
> (r||i)^l*(epsilon||i-epsilon)
> *l','m1'=-'l1'..'l1'),'l1'=0..'lmax') fi:
> EQs:=seq(sum('eqn(i,j)','j'=1..N)=
> -A(l,m,i)*l*r||i^l*(epsilon||i-epsilon),i=1..N):
> eqs:=[seq(seq(seq(combine(G||k(l,m)),m=-l..l),l=0..lmax),
> k=1..N)]:
> for k from 1 to N do G||k:=unapply(value(EQs[k]),l,m) od:
> A:=AA:
> c:=(l2,m2,l1,m1,i,j)->
> (-1)^(l2+m1-m2)*f2(2*l2+2*l1-1)/f2(2*l2-1)/
> f2(2*l1+1)/r||i||j^(l1+l2+1)*Y(i,j,l1+l2,m2-m1)*
> ((2*l1+2*l2+1)*(2*l2+1)/(2*l1+1))^(1/2)*
> CG(l1+l2,0,l2,0,l1,0)*CG(l1+l2,m1-m2,l2,m2,l1,m1):
> Y:=(i,j,l,m)->expand(subs(r=r||i||j,x=rv||i||j[1],
> y=rv||i||j[2],z=rv||i||j[3],
> 2*sqrt(Pi)*ComplexSolidHarmonic(l,m,x,y,z)/r^l)):
> f2:=n->if type(n,even) then 2^(n/2)*(n/2)!
> elif n=-1 then 1 else n!/f2(n-1) fi:
> CG:=cg:
> ks:=[seq(seq(seq(B||k[l,m],m=-l..l),l=0..lmax),k=1..N)]:
> C:=linalg[genmatrix]([combine(eqs[1])],ks,b||1):
> for k from 2 to sk do
> C:=linalg[stackmatrix](C,linalg[genmatrix]
> ([combine(expand(eqs[k]))],ks,b||k)):
> print(k):
> od:
> C:=evalm(map(evalf,C)):
> B:=[seq(b||k[1],k=1..sk)]:
```

```
> rA:=linalg[matrix](sk,sk,(i,j)->evalf(Re(C[i,j]))):
> iA:=linalg[matrix](sk,sk,(i,j)->evalf(Im(C[i,j]))):
> writedata(ia,iA,float):
> writedata(ra,rA,float):
> writedata(b,B,float):
> fclose(ia); fclose(ra); fclose(b);
```

3. Lygčių sistemos sprendimas (Matlab)

Toliau MATLAB sistemoje vykdoma programa „sist-spr.m“

```
% programa \kk sist-spr.m\dk :
 format long
 B=load('b'); R=load('ra'); M=load('ia');
 S=inv(R+i*M)*B; RE=real(S); IM=imag(S);
 save 'rspr' RE -ascii -double
 save 'ispr' IM -ascii -double;
```

4. Sprendinio formavimas

```
> readdata(rspr):
> readdata(ispr):
> linalg[vector]([seq(%%[k]+I*%[k],k=1..sk)]):
> sol:=linalg[geneqns](linalg[diag](1$sk),
> [seq(seq(seq(B||k[l,m],m=-l..l),l=0..lmax),k=1..N)],%):
> assign(%):
> sum(sum(b[l,m]*CSH(l,m,x,y,z)/r^(2*l+1),m=-l..l),l=0..lmax):
> CSH:=ComplexSolidHarmonic:
> u:=map(normal,%%):
> [seq(subs(seq(seq(b[l,m]=B||k[l,m],m=-l..l),l=0..lmax),u),
> k=1..N)]:
> sum('subs(r=sqrt(x^2+y^2+z^2),x=x-S||k[1][1],
> y=y-S||k[1][2],z=z-S||k[1][3],%[k])','k'=1..N):
> if has(%,I)then evalc(%);subs(I=0,%);map(combine,%,power):fi:
> map(combine,%,power):
```

5. Sprendinys

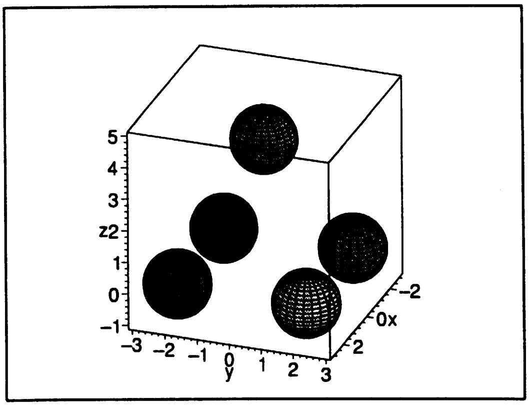
Sprendinys (tiksliau – sprendinio realioji dalis). Sprendinio mes čia neparodome, nes jis užimtų keletą puslapių.

```
> U:=unapply(%,x,y,z):
>
```

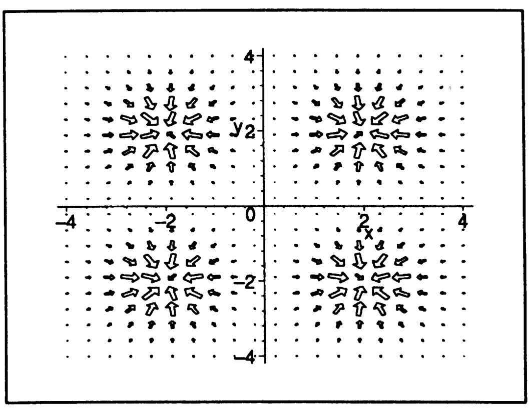
6. Sprendinio grafinis tyrimas

Mes pateikiame tik kelias spendinio grafinio išvedimo galimybes. Yra fiksuojamas vienas arba du kintamieji ir pagal likusius kintamuosius brėžiamas grafikas.

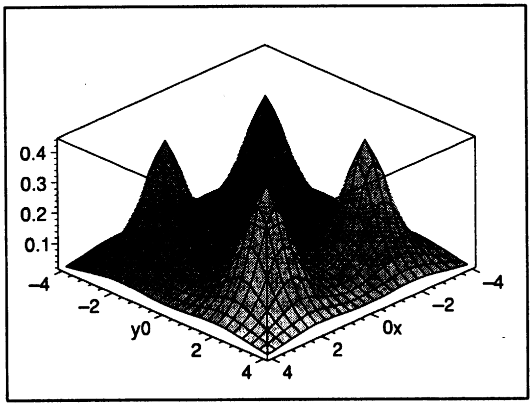
```
> with(plottools,sphere):with(plots,display,sphereplot):
> for k to N do c||k:=sphere(S||k[1],S||k[2] ): od:
> plots[display]({c||(1..N)},scaling=constrained,axes=boxed,
> labels=[x,y,z],style=patch,orientation=[20,65]);
```



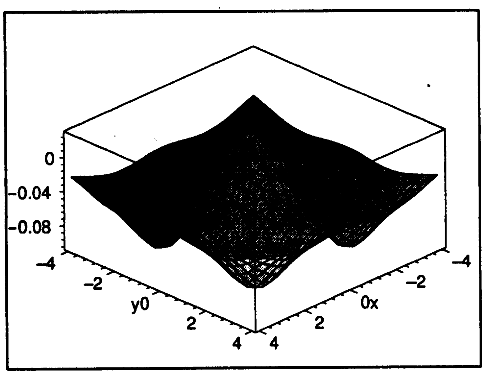

```
> plots[gradplot](U(x,y,-1),x=-4..4,y=-4..4,arrows=THICK);
```



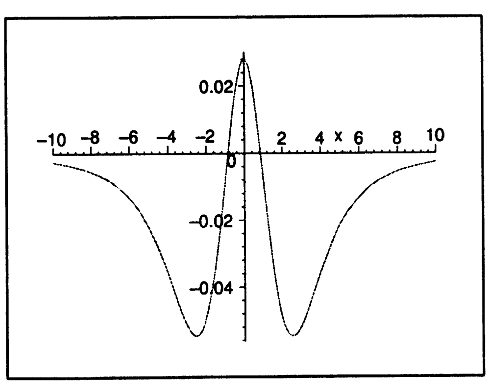

```
> plot3d(U(x,y,-1),x=-4..4,y=-4..4,style=PATCH,axes=boxed);
```



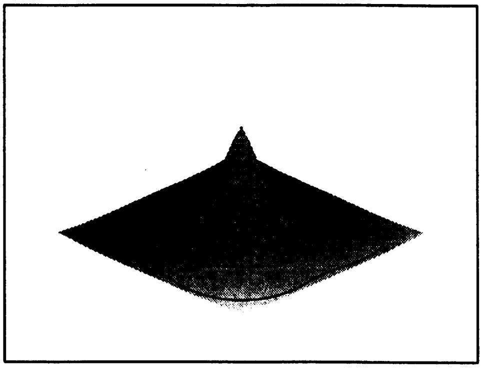

```
> plot3d(U(x,y,2),x=-4..4,y=-4..4,style=PATCH,axes=boxed);
```




```
> plot(U(x,0,2),x=-10..10);
```




```
> plot3d(U(x,y,3),x=-4..4,y=-4..4,style=PATCHCONTOUR);
```



Literatūra

[1] B.S. Aleksandrov, A.B. Bolotin, N.P. Pošiūnaitė, R.J. Rakauskas, V.K. Šugurov, *Mnogocentrovyje Integrali*, Vilnius, VU (1974).
[2] L.D. Landau, E.M. Lifšic, Kvantovaja mechanika, *Teoretičeskaja fizika*, t. III, M., Nauka (1974).

The conjunction between computer algebra and numerical methods

A. Domarkas, R.J. Rakauskas, S. Cicėnas

The report presents the efficiency of using in common Maple and Matlab packages for finding solution (1), which describes the influence of n spherical particles to the external electrostatic field by means of general potential theory.