



2004-2006 m. Bendrojo programavimo dokumento 2 prioriteto „Žmogiškųjų išteklių plėtra“ 4 priemonė „Mokymosi visą gyvenimą sąlygų plėtra“

Projekto sutarties numeris: ESF/2004/2.4.0-K01-160/SUT-261

Projekto pavadinimas: **Inovatyvūs mokymosi metodai ir naujausios technologijos gamtos mokslų bakalauro rengimui**

FIZ 313 KOMPIUTERINĖ FIZIKA

Laboratorinis darbas

MSIMSL PAPROGRAMIŲ BIBLIOTEKŲ TAIKYMAS FIZIKOS UŽDAVINIŲ SPRENDIMUI

1. Darbo tikslas:

1.1. Naudojant MSIMSL Fortan bibliotekos paprogrames apskaičiuoti srovių stiprius elektros grandinėje..

2. Darbo uždavimai:

1. Susipažinti su Fortan paprogramių naudojimu.
2. Naudojant Kirchhofo taisykles sudaryti tiesinių lygčių sistemą duotai elektros grandinei.
3. Naudojant pateiktą pavyzdį parašyti programą, kuri išspręstų tiesinių lygčių sistemą naudodama MSIMSL paprogramę LSLRG.

3. Bendroji teorija

FORTRAN bibliotekų naudojimas

Microsoft IMSL biblioteką sudaro dvi skirtingos, tačiau tarpusavyje susietos dalys:

- MATH/LIBRARY – taikomios matematikos ir specialiųjų funkcijų biblioteka;
- STAT/LIBRARY – statistiniams skaičiavimams skirtų procedūrų biblioteka.

Daugelis procedūrų yra sukurtos dviem atvejams: viengubo ir dvigubo tikslumo skaičiavimams. IMSL MATH/LIBRARY procedūros daugiausia skirtos taikomųjų tyrimų skaičiavimams atlikti bei matematinei analizei. Kiekviena procedūra yra smulkiai aprašyta, todėl jų panaudojimas neturėtų sudaryti didelių sunkumų.

Norint pasinaudoti sukurtais skaičiavimo schemomis, pirmiausia reikia parašyti programą FORTAN kalba ir šioje programoje iškviesti MATH/LIBRARY procedūrą. Praktiškai visos procedūros yra paprogramių forma, todėl jos iškviečiamos su CALL. Šios bibliotekos nėra standartinės, todėl jos neįjungiamos į sąrašą, kuriame kompiatorių ieško reikalingų programos modulių. Norint įtraukti

biblioteką į kompiatoriaus naudojamų elementų sąrašą, reikia naudoti modulių prijungimo komandą USE, pavyzdžiui,

```
use msimsl
```

Su MSIMSL paprogramių biblioteka galima susipažinti Fortan Power Station kairiajame lange suradus nuorodą IMSL Libraries Reference. MSIMSL biblioteką sudaro trys dalys: a) matematinės paprogramės, b) statistinės paprogramės ir c) specialiosios funkcijos.

Norint pasinaudoti šiomis funkcijomis reikia:

a) žinoti sprendžiamo uždavinio problemą;

b) iš bibliotekos surasti duotą uždavinį sprendžiamą paprogramę ir susipažinti su jos aprašymu: kokius duomenis ir kokiame formate juos pateikti bei kokia forma paprogramė pateikia atsakymą;

c) pasinaudoti paprograme. Aiškintis, kaip veikai konkreti paprogramė vartotojui nebūtina, svarbu suprasti, kokių pradinių duomenų reikalauja naudojama paprogramė ir kokie kintamieji išsaugo skaičiavimų rezultatus.

Išnagrinėkime du paprogramių pavyzdžius.

Matricų išvedimas su WRRRN

Kad būtų lengviau atspausdinti skaičių matricas, naudojama paprogramė WRRRN. Jos trumpas aprašymas pateiktas žemiau, o kitame pavyzdyje parodyta, kaip naudojant šią paprogramę, atspausdinama matrica.

```
! CALL WRRRN (TITLE, NRA, NCA, A, LDA, ITRING)
! Argumentai
! TITLE — simbolių seka – tiesiog pavadinimas (Input)
! NRA — matricos A eilučių skaičius. (Input)
! NCA — matricos A stulpelių skaičius. (Input)
! A — NRA × NCA matrica. (Input)
! LDA — A matricos dimensija (Input)
! ITRING — trikampio pasirinkimas.. (Input)
```

1 pav. MSIMSL paprogramės aprašymas.

Norėdami į ekraną išvesti matricos A turinį, sukuriame programėlę kaip parodyta 2 pavyzdyje. Atkreipkite dėmesį, kad matricos A nariai sukonstruojami dvigubu DO ciklu taip, kad gauto realaus skaičiaus pirmas skaitmuo rodytų eilutės numerį, o antrasis skaitmuo – stulpelio, kuriame yra šis skaičius, numerį.

```

! ***** wrrrn_pvz.f90
implicit none
character (8) title
integer NRA, NCA, LDA
integer :: itring = 0 ! 0, 1, 2, -1, -2
real, allocatable, dimension(:,:) :: A
integer i,j

title = 'matrica'
NRA = 5
NCA = 3
LDA = 5
allocate(A(NRA,NCA))

do i = 1, NRA
  do j = 1, NCA
    A(i,j) = i*10 + j
  enddo
enddo
CALL WRRRN (title, NRA, NCA, A, LDA, itring)
END

```

2 pav. MSIMSL paprogramės WRRRN naudojimas.

Kokia prasmė kintamojo itring sužinokite patys keisdami šio kintamojo vertes kaip nurodyta pavyzdyje.

Tiesinių lygčių sistemos sprendimas su LSLRG

Žemiau pateiktame pavyzdyje išnagrinėtas elektrinės grandinės sprendimo pavyzdys. Norint sužinoti elektros srovės stiprius atskirose grandinės atšakose, reikia išspręsti tiesinę lygčių sistemą. Sprendimui gali būti panaudota MSIMSL bibliotekos paprogramė LSLRG. Jos trumpas aprašymas pateiktas pavyzdyje.

```

! CALL LSLRG (N, A, LDA, B, IPATH, X)
! N — lygčių skaičius
! A — dydžio NxN matrica, kurioje sudėti tiesinių lygčių
koeficientai.
! LDA — matricos A dimensija
! B — yra N ilgio vektorius, kuriame surašyti lygčių sistemos po
lygybes esantys skaičiai
! PATH — sprendimo būdas: IPATH = 1 sprendžiama AX = B,
! IPATH = 2 sprendžiama ATX = B.
! X — yra N ilgio vektorius, kuriame išsaugomas atsakymas

```

Error! No text of specified style in document..**3 pav. LSLRG paprogramės aprašymas.**

Tarkime, spręsimė trijų dimensijų uždavinį. Visi koeficientai turi būti surašyti tokia forma:

!		A × X =	B
!	a11 a12 a13	I1	b1
!	a21 a22 a23	I2	b2
!	a31 a32 a33	I3	b3

A yra dvimatė matrica, I1, I2, I3 yra nežinomieji – matrica X. Tiesinių lygčių sistemos koeficientai – vienmatė matrica B. Pažiūrėkime, kaip dauginant šias matricas gaunama lygtis. Dauginam pirmą eilutę su stulpeliu X tokiu būdu:

$$A_{11} \times I_1 + A_{12} \times I_2 + A_{13} \times I_3 = B_1$$

Panašiai gaunamos ir kitos dvi lygtys. Tarkime, kad pradiniai duomenys yra tokie:

!		A		X =	B
	-1	-1	-1	I1	0
!	1300	-500	0	I2	10
!	0	500	-5000	I3	-4

Tuomet išspręsti pateiktą tiesinių lygčių sistemą galima taip:

```
CALL LSLRG (3, A, 3, B, 1, X)
CALL WRRRN ('sroves', 3, 1, X, 1, 0)
```

Antroji eilutė atspausdins matricą X, t.y., nežinomus dydžius, I1, I2 ir I3.

4. Tyrimo metodika

Elektros grandinių uždavinys

Elektros srovė vadinamas bet koks tvarkingas elektrinių krūvių judėjimas. Elektros srovė, kuri atsiranda laidžiose aplinkose dėl tvarkingo laisvų elektronų judėjimo veikiant elektriniam laukui vadinama *laidumo srove*. Laidumo srovės pavyzdžiai gali būti srovė metaluose ir puslaidininkiuose, kuriuose juda laisvi elektronai; srovė elektrolituose, kuriuose tvarkingai juda priešingų ženklų jonai.

Srovės stiprių vadinamas fizikinis dydis, lygus per tiriamą paviršių pernešto krūvio dq ir trukmės, per kurią tas krūvis buvo perneštas, santykiui:

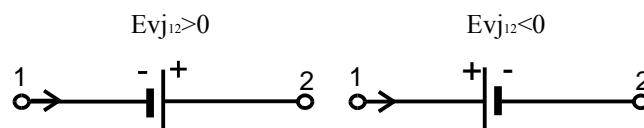
$$I = \frac{dq}{dt} \quad (1)$$

Elektros srovė vadinama *nuolatine*, jei srovės stipris ir jos kryptis nekinta laike. tai yra nuolatinei srovei galioja lygybė $I = q / t$, kur q yra per baigtinį laiko tarpą t perneštas elektros krūvis pro tiriamą paviršių.

Pašalinėmis jėgomis vadinamos neelektrostatinės jėgos, kurių veikiami elektros krūvinikai pradeda tvarkingą judėjimą laidininke; be to, tokios jėgos palaiko nuolatinę srovės tekėjimą grandinėje. Pašalinės jėgos, skirtingai negu kuloninės jėgos, ne sujungia skirtingo ženklo krūvius, o sąlygoja tų krūvių atsiskyrimą ir palaiko potencialų skirtumą laidininko galuose. Stacionarus pašalinių jėgų elektrinis laukas sukuriamas elektrinės energijos šaltiniais: galvaniniais elementais, elektriniais generatoriais.

Pašalinės jėgos apibūdinamos elektrovaros jėga, kuri skaitiškai lygi darbui, kurį atlieka pašalinės jėgos, pernešdamos vienetinį teigiamą krūvį nuo vieno taško į kitą. Darbas atliekamas naudojant šaltinio energiją, todėl E_{vj} vadinamas *elektros energijos šaltinio elektrovaros jėga*.

E_{vj} ženklas nustatomas tokia taisykle: jeigu šaltinio viduje eina nuo katodo į anodą, t.y. pašalinių jėgų stiprumas šaltinyje sutampa su srove toje grandinės dalyje (1 pav.), tai E_{vj} laikoma teigiama. Kai srovė šaltinyje teka nuo anodo į katodą, tai E_{vj} laikoma neigiama.



4 pav. E_{vj} šaltinio ženklo nustatymo taisyklė

Grandinės dalies 12 *įtampa* U_{12} vadinamas fizikinis dydis lygus darbui, kuris atliekamas toje grandinės dalyje pernešant vienetinį teigiamą krūvį. Įtampa grandinės galuose sutampa su potencialų skirtumu tik tuo atveju, jei toje grandinės dalyje nėra E_{vj} šaltinių.

Realiose grandinėse jungiamųjų laidų varža yra maža, todėl laikoma, kad ji lygi 0. Apibendrintas Omo dėsnis grandinės daliai yra toks:

$$I R_{12} = U_{12}. \quad (2)$$

Srovės stiprio sandauga su tos dalies varža yra lygi sumai potencialų skirtumo ir visų elektrinės energijos šaltinių E_{vj} , įjungtų toje grandinės dalyje.

Neišsišakojusioje uždaroje elektrinėje grandinėje srovės stipris visuose skerspjūviuose vienodas. Todėl tokio grandinės galuose $j_1 = j_2$ ir $R_{12} = R$, kur R - bendra visos grandinės varža. Tada (2) virsta Omo dėsnio grandinės daliai:

$$I R = E_{vj}, \quad (3)$$

kur E_{vj} - algebrinė visų grandinės elektrinės energijos šaltinių suma.

Jeigu uždara grandinę sudaro E_{vj} šaltinis, kurio vidinė varža r , o išorinės grandinės varža R , tai Omo dėsnis tokiai grandinei yra toks

$$I (R+r) = E_{vj}. \quad (4)$$

Sudėtingų (išsišakojusių) grandinių skaičiavimo tikslas - surasti srovės skirtingose grandinės atšakose, kai žinomos visos grandinės varžos ir E_{vj} šaltiniai. Išsišakojusios grandinės turi taškus, kurie jungia daugiau kaip du laidininkus; tokie taškai vadinami *mazgais*. *Pirma Kirchhofo taisyklė* yra tokia: algebrinė n srovių, sutekančių į vieną mazgą, suma lygi nuliui:

$$\boxed{}. \quad (5)$$

Čia n - laidininkų skaičius, susikertančių viename mazge, I_i - srovė laidininke. Teigiama laikoma srovė, kuri įteka į mazgą, o neigiama - kuri išteka.

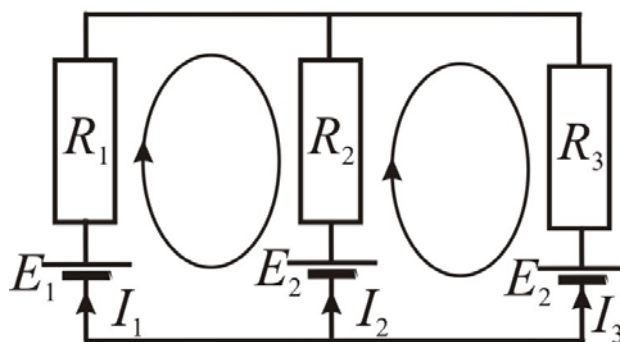
Antroji Kirchhofo taisyklė (kontūrų taisyklė) yra tokia: bet kuriame laisvai pasirinktame išsišakojusios elektrinės grandinės kontūre algebrinė visų srovių I_i sandaugų su atitinkamų grandinės dalių varžomis R suma lygi visų kontūro E_{vj} algebrinei sumai:

$$\sum_{i=1}^{i=n} I_i R_i = \sum_{i=1}^{i=m} E_{vj}. \quad (6)$$

Čia n - kontūro dalių, į kurias kontūrą išdalina mazgai, skaičius; kiekvienoje kontūro dalyje teka skirtingos srovės. Norint taikyti antrą Kirchhofo taisyklę, būtina pasirinkti kontūro apėjimo kryptį. Tada teigiamomis laikomos srovės, kurių kryptis sutampa su kontūro apėjimo kryptimi, ir neigiamomis, kurių kryptis priešinga apėjimo kryptčiai. E_{vj} šaltinių ženklas laikomas teigiamu, jei jie kuria srovės, kurių kryptys sutampa su kontūro apėjimo kryptimi.

Išsišakojusi grandinė skaičiavimas vyksta tokia tvarka:

- 1) laisvai pasirenkamos srovių kryptys visose grandinės atšakose;
- 2) jei grandinė turi m mazgų, tai užrašoma $m - 1$ nepriklausoma pirmos Kirchhofo taisyklės lygtis;
- 3) laisvai išskiriami kontūrai grandinėje, pasirenkama kontūro apėjimo kryptis. Kontūrai sudaromi taip, kad kiekvienas naujas kontūras turėtų bent vieną grandinės dalį, kuris nepateko į prieš tai sudarytus kontūrus;
- 4) užrašomos antros Kirchhofo taisyklės lygtys. Jei grandinėje yra p dalių tarp mazgų ir m mazgų, tai galima užrašyti $(p - m + 1)$ nepriklausomą antros Kirchhofo taisyklės lygtį.



5 pav. Evj šaltinio ženklų nustatymo taisyklė

Pagal I taisyklę - srovėms. Grandinėje du mazgai, vadinasi, galima užrašyti tik vieną nepriklausomą lygtį mazgams. Pagal pasirinktas srovių kryptis gauname:

$$-I_1 - I_2 - I_3 = 0 \quad (7)$$

Pagal II taisyklę:

Kairiajam kontūriui:

$$I_1 R_1 - I_2 R_2 = E_1 - E_2 \quad (8)$$

Dešiniajam kontūriui:

$$I_2 R_2 - I_3 R_3 = E_2 - E_3 \quad (9)$$

Pastaba. Kairiajame kontūre E_1 įjungtas kontūro apėjimo kryptimi, o E_2 – priešinga kryptimi.

Dešiniame kontūre E_2 įjungtas kontūro apėjimo kryptimi, o E_3 – priešingai. (7 – 8) formulės sudaro tiesinių lygčių sistemą, kurią reikia išspręsti: trys lygtys, trys nežinomieji. Tokią lygtį išspręsti galima analitiškai, tačiau sprendžiant sudėtingesnę sistemą reikia ieškoti efektyvesnių sprendimo metodų. Vienas tokių metodų yra panaudoti tiesinių lygčių sprendimui paprogrames.

Elektros grandinių uždavinio sprendimo pavyzdys

Priede Nr. 1 pateiktas trijų kontūrų elektros grandinę aprašančios lygčių sistemos sprendimas, kai pradiniai duomenys yra tokie (pridedamo duomenų failo grand.data turinys):

1300 500 5000
6 -4 0
R1 R2 R3 (omais)
E1 E2 E3 (voltais)

5. Tyrimo eiga

1. Laboratorinis darbas atliekamas kompiuterių klasėje.
2. Išanalizuokite gautą elektros grandinės brėžinį. Nubraižykite srovių kontūrus ir pagal jas užrašykite tiesinių lygčių sistemą.
3. Parašykite programą, su kuria galima būtų apskaičiuoti sroves, tekančias visomis grandinės šakomis. Programa turi pradinius duomenis nuskaityti iš duomenų failo, o rezultatus surašyti į rezultatų failą. Greitai peržiūrai programa rezultatus turėtų parodyti ir ekrane.
4. Programoje turi naudojama bent viena studento sukurta funkcija ir paprogramė.
5. Atliktie reikalingus skaičiavimus ir atliktą darbą pateikti parašytą tekstiniu redaktoriumi. Atsiskaitymo dokumente turėtų būti:

- a) Užduoties sąlyga.
- b) Brėžinys

- c) Fizikinis uždavinio sprendimas – lygčių sistema.
- d) Trumpai aprašytas programos algoritmas – bendrais bruožais išdėstyti rašomos programos struktūrą.
- e) Pateikti veikiančios programos tekstą su komentarais. Programa turi būti parašyta suprantamu ir tvarkingu kodu pagal anksčiau pateiktus nurodymus.

6. Kontroliniai klausimai

1. Omo dėsnis grandinės daliai ir uždarai grandinei
2. Kirchhofo taisyklių taikymas.
3. Fortrano procedūros MSIMSL bibliotekoje ir jų naudojimas.
4. MSIMSL Fortrano procedūros tiesinių lygčių sprendimui.
5. Paprogramės LSLRG naudojimas.

7. Literatūra

1. A. Kanapickas. Paskaitų konspektas. 7 skyrius.
2. Tamašauskas A. , Vosylius J. Fizika, T. 2. - Vilnius: Mokslas, 1989. – 195 p.
3. Fortran for scientists and engineers. Chapter 16.1: introduction to numerical methods – equations.

Priedas Nr. 1. Užduoties sprendimo pavyzdys

```
use msimsl
implicit none
integer, PARAMETER ::
IPATH=1

real, allocatable, dimension(:, :) ::
A
real, allocatable, dimension(:) ::
Voltai, Omai, B, X
integer i, Nlygciu, LDA, N
integer ios          ! galimos I/O
klaidos Nr
integer :: unit = 3    ! ivedimo
irenginys

open(unit, file = 'kirch3.dat', status
= 'old', iostat = ios)
if(ios /= 0) stop 'kirch3.dat
nepavyko atidaryti'
read(unit, *) Nlygciu
!read(unit, *) nexam
print *, 'Grandziu (lygciu)
skaicius: ', Nlygciu

ALLOCATE(X(Nlygciu))
ALLOCATE(B(Nlygciu))
ALLOCATE(Voltai(Nlygciu))
ALLOCATE(Omai(Nlygciu))
ALLOCATE(A(Nlygciu,
Nlygciu))          !
!
do i = 1, Nlygciu          !
PRADEDAM SKAITYTI
TOLIAU ESANCIUS DUOMENIS
read(unit,*, iostat = ios)
Voltai(i), Omai(i)
if(ios /= 0) stop 'IO klaida'
end do
close (unit)

print *, "
print *, 'Saltiniu itamos: '
print *, Voltai
print *, "
print *, 'Varzos: '
print *, Omai

print *, "
```

```
print *, 'Formuojam matrica A... '
A(1,1) = Omai(1)
A(1,2) = Omai(2) * (-1)
A(1,3) = 0
A(2,1) = 0
A(2,2) = Omai(2)
A(2,3) = Omai(3) * (-1)
A(3,1) = 1
A(3,2) = 1
A(3,3) = 1

print *, "
print *, 'Lygciu koeficientai
(matrica A)'
do i = 1, Nlygciu
print *, A(i, 1:Nlygciu)
end do

print *, "
print *, 'Formuojam matrica B... '

B(1) = Voltai(1) - Voltai(2)
B(2) = Voltai(2) - Voltai(3)
B(3) = 0

print *, "
print *, 'Matrica B:'
print *, B(1:Nlygciu)
pause

LDA = Nlygciu
N = Nlygciu
CALL LSLRG (N, A, LDA, B,
IPATH, X)

print *, 'Sroves: matrica X'
CALL WRRRN ('X', 1, N, X, 1,
0)

open(unit, file = 'kirch3.dat',
access = 'append', status = 'old',
iostat = ios)
if(ios /= 0) stop 'kirch3.dat
nepavyko atidaryti'
write (unit,*) 'Sroves: '
write (unit,*) X
END
```