

3. Võrrandisüsteemide lahendamine

3.1. Vektorfunktsiooni mõiste

Olgu antud m reaalarvulisest komponendist koosnev vektor $x = (x_1, x_2, \dots, x_m)$. Kujutis, mis seab vektorile x vastavusse reaalarvu, on m muutuja funktsioon. Olgu vaatluse all m reaalarvulist m muutuja funktsiooni

$$\begin{aligned}f_1(x) &= f_1(x_1, \dots, x_m), \\f_2(x) &= f_2(x_1, \dots, x_m), \\&\dots \\f_m(x) &= f_m(x_1, \dots, x_m).\end{aligned}$$

Igaüks neist seab vektorile x vastavusse teatud reaalarvu. Kokkuvõttes saame reaalarvudest $f_1(x), \dots, f_m(x)$ moodustada järgmise vektori:

$$F(x) = (f_1(x), \dots, f_m(x)).$$

Seega võib öelda, et F on teatav kujutis, mis seab m komponendist koosnevale vektorile x vastavusse teise m komponendist koosneva vektori. Tegemist on *vektorfunktsiooniga*.

Näiteks olgu $f_1(x) = x_1 - x_2$ ja $f_2(x) = x_1^2 + x_2^2$. Neist kahest funktsioonist moodustub kahekomponendiline vektorfunktsioon $F(x) = (f_1(x), f_2(x))$. Arvutame selle väärtuse $x = (1, 3)$ korral. Kuna $f_1(1, 3) = 1 - 3 = -2$ ja $f_2(1, 3) = 1^2 + 3^2 = 10$, siis saame $F(1, 3) = (-2, 10)$.

3.2. Mittelineaarsed süsteemid ja iteratsioonimeetodid

Mittelineaarne võrrandisüsteem kõige üldisemal kujul on järgmine:

$$\begin{cases} f_1(x_1, x_2, \dots, x_m) = 0 \\ f_2(x_1, x_2, \dots, x_m) = 0 \\ \dots \\ f_m(x_1, x_2, \dots, x_m) = 0, \end{cases} \quad (3.1)$$

kus f_1, \dots, f_m on mingid m muutuja funktsioonid. Kasutades eelmises alampeatükis sisse toodud vektorfunktsiooni mõistet, on võimalik süsteem (3.1) kirjutada ka lühemal kujul järgmiselt:

$$F(x) = 0 \quad (3.2)$$

kus paremal poolel seisab nullvektor, st $0 = (0, \dots, 0)$.

Näiteks süsteemi

$$\begin{cases} \cos(x_1)x_2 + \sqrt{x_1} = x_2 \\ \ln(x_1) + \tan(x_2) = 4 \end{cases}$$

saab esitada kujul

$$\begin{cases} f_1(x_1, x_2) = 0 \\ f_2(x_1, x_2) = 0, \end{cases}$$

kui defineerida järgmised kahe muutuva funktsioonid: $f_1(x_1, x_2) = \cos(x_1)x_2 + \sqrt{x_1} - x_2$ ja $f_2(x_1, x_2) = \ln(x_1) + \tan(x_2) - 4$.

Enamus praktikas kasutatavaid meetodeid süsteemi (3.1) lahendamiseks on *iteratsioonimeetodid*. Iteratsioonimeetodi käivitamiseks valitakse alglähend x^0 . Alglähendi kaudu arvutatakse esimene lähend x^1 , viimase kaudu teine lähend x^2 jne. Iteratsioonimeetodi algoritm on defineeritud, kui on antud valemid, mille põhjal saab x^{n-1} kaudu arvutada x^n .

Erinevalt eelmises peatükis käsitletud võrranditest, on süsteemide lahendamisel tekkivad lähendid vektorid, st

$$\begin{aligned} x^0 &= (x_1^0, x_2^0, \dots, x_m^0), \\ x^1 &= (x_1^1, x_2^1, \dots, x_m^1), \\ x^2 &= (x_1^2, x_2^2, \dots, x_m^2), \\ &\dots \\ x^{n-1} &= (x_1^{n-1}, x_2^{n-1}, \dots, x_m^{n-1}), \\ x^n &= (x_1^n, x_2^n, \dots, x_m^n), \\ &\dots \end{aligned}$$

Seega toimub süsteemi lahendamisel liikumine ühelt vektorilt teisele. Piisava arvu sammude järel jõutakse otsitavale lahendile piisavalt lähedale. Siis võib iteratsiooniprotsessi katkestada.

Kahest võrrandist koosneva süsteemi korral saab alglähendi valimiseks kasutada graafilist meetodit.

3.3. Harilik ja Seideli iteratsioonimeetod

Hariliku ja Seideli iteratsioonimeetodi rakendamiseks peab süsteem (3.1) olema viidud järgmisele kujule:

$$\begin{cases} x_1 = g_1(x_1, x_2, \dots, x_m) \\ x_2 = g_2(x_1, x_2, \dots, x_m) \\ \dots \\ x_m = g_m(x_1, x_2, \dots, x_m), \end{cases} \quad (3.3)$$

kus g_1, \dots, g_m on mingid m muutuva funktsioonid. Võimalusi süsteemi (3.1) teisendamiseks kujule (3.3) on palju. Ühte üldist võimalust vaatleme käesoleva alampeatüki lõpus.

Näiteks üks võimalus eespooltoodud süsteemi

$$\begin{cases} \cos(x_1)x_2 + \sqrt{x_1} = x_2 \\ \ln(x_1) + \tan(x_2) = 4 \end{cases}$$

esitamiseks hariliku ja Seideli iteratsioonimeetodi rakendamiseks sobival kujul on järgmine:

$$\begin{cases} x_1 = e^{4-\tan(x_2)} \\ x_2 = \cos(x_1)x_2 + \sqrt{x_1}. \end{cases} \quad (3.4)$$

Hariliku iteratsioonimeetodi korral toimub x^n arvutamine järgmiste valemitega:

$$\begin{aligned} x_1^n &= g_1(x_1^{n-1}, x_2^{n-1}, \dots, x_m^{n-1}) \\ x_2^n &= g_2(x_1^{n-1}, x_2^{n-1}, \dots, x_m^{n-1}) \\ &\dots \\ x_m^n &= g_m(x_1^{n-1}, x_2^{n-1}, \dots, x_m^{n-1}). \end{aligned} \quad (3.5)$$

Valemities (3.5) leitakse kõigepealt kõigepealt esimene komponent x_1^n . Teise komponendi x_2^n arvutamisel kasutatakse muuhulgas suurust x_1^{n-1} . Tekib loomulik küsimus: miks ei võiks x_2^n arvutamisel kasutada x_1^{n-1} asemel hoopis x_1^n , kuna viimane on juba teada ja x_1^n peaks ju otsitava lähendi esimest komponenti lähendama täpsemini kui x_1^{n-1} ? Kui me kasutame x_2^n arvutamisel x_1^{n-1} asemel suurust x_1^n , siis on vastav valem järgmine:

$$x_2^n = g_2(x_1^n, x_2^{n-1}, \dots, x_m^{n-1}).$$

Analoogiliselt võime x_3^n arvutamisel kasutada suuruste x_1^{n-1} ja x_2^{n-1} asemel suurusi x_1^n ja x_2^n , kuna viimased on kolmanda võrrandi juurde jõudes juba välja arvutatud jne. Kokkuvõttes on sellise meetodi arvutuseeskirjad järgmised:

$$\begin{aligned} x_1^n &= g_1(x_1^{n-1}, x_2^{n-1}, x_3^{n-1}, \dots, x_{m-1}^{n-1}, x_m^{n-1}) \\ x_2^n &= g_2(x_1^n, x_2^{n-1}, x_3^{n-1}, \dots, x_{m-1}^{n-1}, x_m^{n-1}) \\ x_3^n &= g_3(x_1^n, x_2^n, x_3^{n-1}, \dots, x_{m-1}^{n-1}, x_m^{n-1}) \\ &\dots \\ x_m^n &= g_m(x_1^n, x_2^n, \dots, x_{m-1}^n, x_m^{n-1}). \end{aligned} \quad (3.6)$$

See on *Seideli iteratsioonimeetod* (kirjanduses võib kohata ka nimetust *Gauss-Seideli iteratsioonimeetod*).

Näiteks süsteemi (3.4) korral on hariliku ja Seideli iteratsioonimeetodi algoritmid vastavalt järgmised:

$$\begin{aligned} x_1^n &= e^{4-\tan(x_2^{n-1})} \\ x_2^n &= \cos(x_1^{n-1})x_2^{n-1} + \sqrt{x_1^{n-1}} \end{aligned}$$

ja

$$\begin{aligned}x_1^n &= e^{4-\tan(x_2^{n-1})} \\x_2^n &= \cos(x_1^n)x_2^{n-1} + \sqrt{x_1^n}.\end{aligned}$$

Vaatleme üht üldist võimalust süsteemi (3.1) teisendamiseks kujule (3.3). Valime mingid nullist erinevad konstandid C_1, \dots, C_m ja korrutame nendega süsteemi (3.1) võrrandeid:

$$\begin{aligned}C_1 f_1(x_1, x_2, \dots, x_m) &= 0 \\C_2 f_2(x_1, x_2, \dots, x_m) &= 0 \\&\dots \\C_m f_m(x_1, x_2, \dots, x_m) &= 0.\end{aligned}$$

Seejärel liidame i -nda võrrandi vasakule ja paremale poolele muutuja x_i :

$$\begin{aligned}x_1 &= x_1 + C_1 f_1(x_1, x_2, \dots, x_m) \\x_2 &= x_2 + C_2 f_2(x_1, x_2, \dots, x_m) \\&\dots \\x_m &= x_m + C_m f_m(x_1, x_2, \dots, x_m).\end{aligned}$$

Saadud süsteem ongi kujul (3.3), kusjuures

$$g_i(x_1, x_2, \dots, x_m) = x_i + C_i f_i(x_1, x_2, \dots, x_m).$$

3.4. Vektorfunktsiooni Jacobi maatriks

Jacobi maatriks on ühemuutuja funktsiooni tuletise üldistus vektorfunktsioonile. Vaatleme vektorfunktsiooni $F(x) = (f_1(x), f_2(x), \dots, f_m(x))$, kus $x = (x_1, \dots, x_m)$. Selle funktsiooni komponendist $f_i(x)$ saab võtta m osatuletist:

$$\frac{\partial f_i}{\partial x_1}(x), \frac{\partial f_i}{\partial x_2}(x), \dots, \frac{\partial f_i}{\partial x_m}(x).$$

Kuna vektorfunktsioonil $F(x)$ on m komponenti, siis on tema komponentidega seotud osatuletiste koguarv $m \times m$. Moodustame neist osatuletistest $m \times m$ maatriksi:

$$F'(x) = \begin{pmatrix} \frac{\partial f_1}{\partial x_1}(x) & \frac{\partial f_1}{\partial x_2}(x) \dots & \frac{\partial f_1}{\partial x_m}(x) \\ \frac{\partial f_2}{\partial x_1}(x) & \frac{\partial f_2}{\partial x_2}(x) \dots & \frac{\partial f_2}{\partial x_m}(x) \\ \dots & \dots & \dots \\ \frac{\partial f_m}{\partial x_1}(x) & \frac{\partial f_m}{\partial x_2}(x) \dots & \frac{\partial f_m}{\partial x_m}(x) \end{pmatrix}.$$

See on vektorfunktsiooni $F(x)$ nn *Jacobi maatriks*.

Näiteks eelmistes alampeatükkides esinenud kahekomponendilise vektorfunktsiooni $F(x) = (f_1(x), f_2(x))$, kus $f_1(x) = \cos(x_1)x_2 + \sqrt{x_1} - x_2$ ja $f_2(x) = \ln(x_1) + \tan(x_2) - 4$, Jacobi maatriks on järgmine:

$$F'(x) = \begin{pmatrix} \frac{\partial f_1}{\partial x_1}(x) & \frac{\partial f_1}{\partial x_2}(x) \\ \frac{\partial f_2}{\partial x_1}(x) & \frac{\partial f_2}{\partial x_2}(x) \end{pmatrix} = \begin{pmatrix} -\sin(x_1)x_2 + \frac{1}{2\sqrt{x_1}} & \cos(x_1) - 1 \\ \frac{1}{x_1} & \frac{1}{\cos^2(x_2)} \end{pmatrix}.$$

3.5. Newtoni meetod ja selle modifikatsioon

Eelmises peatükis vaatlesime Newtoni meetodit võrrandi $f(x) = 0$ lahendamisel. Tuletame meelde, et selle meetodi algoritm on järgmine:

$$x_n = x_{n-1} - \frac{f(x_{n-1})}{f'(x_{n-1})}$$

ehk

$$x_n = x_{n-1} - [f'(x_{n-1})]^{-1} f(x_{n-1}). \quad (3.7)$$

Newtoni meetodi saab otseselt üldistada mittelineaarsetele süsteemidele (3.2). Selleks asendame algoritmis (3.7) tavalise funktsiooni f vektorfunktsiooniga F ja skalaarsed lähendid vektor-lähenditega. *Newtoni meetodi* algoritm süsteemide jaoks on järgmine:

$$x^n = x^{n-1} - [F'(x^{n-1})]^{-1} F(x^{n-1}), \quad (3.8)$$

kus $[F'(x)]^{-1}$ on vektorfunktsiooni $F(x)$ Jacobi maatriksi pöördmaatriks.

Näiteks süsteemi

$$\begin{cases} \cos(x_1)x_2 + \sqrt{x_1} = x_2 \\ \ln(x_1) + \tan(x_2) = 4 \end{cases}$$

lahendamisel on Newtoni meetodi algoritm järgmine:

$$\begin{pmatrix} x_1^n \\ x_2^n \end{pmatrix} = \begin{pmatrix} -\sin(x_1^{n-1})x_2^{n-1} + \frac{1}{2\sqrt{x_1^{n-1}}} & \cos(x_1^{n-1}) - 1 \\ \frac{1}{x_1^{n-1}} & \frac{1}{\cos^2(x_2^{n-1})} \end{pmatrix}^{-1} \begin{pmatrix} x_1^{n-1} \\ x_2^{n-1} \end{pmatrix}.$$

Newtoni meetodi algoritm nõuab igal sammul Jacobi maatriksi pöördmaatriksi $[F'(x^{n-1})]^{-1}$ arvutamist. Suure maatriksi korral on see üsna töömahukas. *Modifitseeritud Newtoni meetodis* arvutatakse Jacobi maatriksi pöördmaatriksi ainult esimesel sammul. Selle meetodi algoritm on järgmine:

$$x^n = x^{n-1} - [F'(x^0)]^{-1} F(x^{n-1}).$$

Modifitseeritud Newtoni meetodi algoritmisamm on kiirem, kuid ta koondub aeglasemalt kui Newtoni meetod, mistõttu piisavalt täpse lahendi saamiseks tuleb sooritada rohkem algoritmisamme.