

# 1 Sistem nelinearnih enačb

Namesto skalarne funkcije  $f(x) = 0$  imamo zdaj vektorsko funkcijo  $F(x) = 0$ . Spet lahko poiščemo funkcijo  $X_{r+1} = G(X_r)$ , da bo v ničli veljalo  $G(\alpha) = \alpha$ . Takšna iteracija nam bo dala konvergentno zaporedje, če je spektralni radij (največji element matrike po absolutni vrednosti) manjši od 1.

Problem: funkcijo  $G(X)$  je izredno težko najti. Pri Newtonovi metodi smo si pomagali s formulo

$$x_{r+1} = x_r - \frac{f(x_r)}{f'(x_r)},$$

Vendar bi v našem primeru to zahtevalo deljenje z matriko, kar matematično nima smisla.

Na srečo je deljenje ekvivalentno množenju z inverzom, inverz matrike pa znamo izračunati. S tem dobimo Newtonovo metodo v več dimenzijah.

$$X^{(r+1)} = X^{(r)} - J_F^{-1}(X_r)F(X^{(r)})$$

Preoblikujemo, da se izognemo računanju inverza.

$$X^{(r+1)} - X^{(r)} = -J_F^{-1}(X^{(r)})F(X^{(r)})$$

$$J_F(X^{(r)})\Delta X^{(r)} = -F(X^{(r)})$$

Dobili smo sistem linearnih enačb, ki nam da rešitev  $\Delta X_r$ . Iz nje lahko izrazimo  $X^{(r+1)} = X^{(r)} + \Delta X^{(r)}$ .

Zdaj imamo še en problem, ki se mu želimo izogniti, in sicer računanje Jacobijeve matrike. Računsko odvajanje je zamudno.

Rešitev: Kvazi-Newtonova metoda. Namesto, da bi Jacobijevo matriko izračunali na vsakem koraku, jo izračunamo le npr. na vsakih 5 ali 10. Metoda še vedno konvergira, le nekoliko počasneje od Newtonove. Kaj pa, če se želimo popolnoma izogniti odvodom? Obstaja alternativna metoda, imenovana Broydenova metoda, ki pa je precej zahtevna in počasna.

**Linearni problem najmanjših kvadratov.** Začnimo s primerom: imamo hitrosti, ki je linearno odvisna od časa:  $v(t) = \alpha t + \beta$ . Hitrost izmerimo ob več časih  $t_j$ ,  $j = 1, 2, 3, \dots, N$ . S tem dobimo približke  $v_j = v(t_j)$ . Kako na podlagi teh meritev najbolje določiti koeficienta  $\alpha$  in  $\beta$ ? Kaj je najboljša izbira koeficientov je stvar definicije, običajno pa uporabimo metodo najmanjših kvadratov: želimo minimizirati vrednost vsote

$$\sum_{i=1}^N (\alpha t_j + \beta - v_j)^2$$

V splošnem gre za to, da rešujemo sistem enačb

$$A \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_N \end{bmatrix}$$

Sistem nima rešitve, lahko pa poskušamo dobiti minimalno vrednost

$$\|A\mathbf{x} - \mathbf{b}\|$$

Za kvadratno normo, ki se izkaže za najprimernejšo, dobimo zgoraj opisano metodo najmanjših kvadratov. To lahko storimo s poljubno funkcijo.

**Trditev.** Če je  $\text{rang}(A) < n$  in je  $x^* \in \mathbb{R}^n$  rešitev po metodi najmanjših kvadratov, je tudi  $x^* + z$ ,  $z \in \ker(A)$  rešitev. Od zdaj naprej bomo privzemali, da je  $\text{rang}(A) = n$ .

**Normalni sistem enačb.** Izkaže se, da je rešitev po metodi najmanjših kvadratov ravno rešitev sistema

$$A^T A \mathbf{x} = A^T \mathbf{b}$$

To pomeni, da lahko uporabimo LU razcep, razcep Choleskega ali kaj podobnega. Razcep Choleskega gotovo lahko uporabimo, saj je  $A^T A$  očitno simetrična, pozitivno definitna pa zaradi lastnosti skalarnega produkta:

$$\text{Zahtevamo: } x^T(A^T A)x > 0$$

$$x^T(A^T A)x = (x^T A^T)(Ax) = (Ax)^T(Ax) = \|Ax\|_2^2$$

Ta norma je po definiciji večja ali enaka 0, zaradi polnosti ranga matrike  $A$  pa je celo večja od njič, saj velja  $\|Ax\|_2^2 = 0 \Rightarrow Ax = 0 \Rightarrow x \in \ker(A)$ . Tak sistem ima točno eno rešitev, vendar lahko pride do velikih računskih napak, zato raje uporabljamo boljše metode.

**Izpeljava.** Če si  $Ax$  zamislimo kot pravokotno projekcijo vektorja  $b$  na sliko  $A$ , velja  $b - Ax \perp \text{Im}(A)$ . To pomeni, da je vektor  $b - Ax$  pravokoten na vse stolpce matrike  $A$ . Sledi

$$A^T(b - Ax) = 0$$

$$A^T Ax = A^T b$$

**QR razcep.** Gre za razcep  $A = QR$ , kjer je  $Q \in \mathbb{R}^{m \times n}$  matrika z ortogonalnimi stolpci in  $R \in \mathbb{R}^{n \times n}$  zgornje trikotna (ali pa kvazi zgornje trikotna - ne kvadratna matrika, ki ima neničelne vrednosti le nad prvo diagonalo. Temu pravimo razširjeni QR razcep). Če poznamo osnovni QR razcep, lahko naredimo sledeče:

$$A^T Ax = A^T b$$

$$(QR)^T(QR)x = (QR)^T b$$

$$R^T(Q^T Q)Rx = (QR)^T b$$

Vemo, da je  $Q^{-1} = Q^T$ , saj je  $Q$  ortogonalna. Nazadnje na obeh straneh z leve množimo z matriko  $(R^T)^{-1}$ :

$$(R^T)^{-1}R^T Rx = (R^T)^{-1}R^T Q^T b$$

$$Rx = Q^T b$$

Naprej lahko rešujemo z obratnim vstavljanjem. Seveda pa moramo konstruirati še QR razcep.

**Gram-Schmidtova ortogonalizacija.** Matriko  $A$  zapišemo po stolpcih:

$$A = [a_1 \quad a_2 \quad \dots \quad a_n] \in \mathbb{R}^{n \times n}$$

Ortogonaliziramo jo po sledečem postopku:

$$a_1 \rightarrow a'_1 = \frac{1}{\|a_1\|_2} a_1$$

$$a_2 \rightarrow a'_2 = \frac{1}{\|a_1 + \alpha a_2\|} (a_1 + \alpha a_2)$$

Kjer  $\alpha$  določimo tako, da je  $a'_1 \perp a'_2$ . Postopek nadaljujemo za  $a'_3, a'_4, \dots, a'_n$ . Dobljeni vektorji so linearne kombinacije vektorjev  $a_1, a_2, \dots, a_n$ . To linearno kombinacijo opišemo z (zgornje trikotno) matriko  $M$ .

$$A = [a_1 \quad a_2 \quad \dots \quad a_n] = [a'_1 \quad a'_2 \quad \dots \quad a'_n] M$$

S tem smo dobili iskani razcep.

Obstaja tudi modificiran Gram-Schmidtov postopek, ki je nekoliko stabilnejši.

**Givensove rotacije.** Alternativen način QR razcepa je Givensova rotacija. Rotacija vektorja  $\mathbf{x} \in \mathbb{R}^n$  za kot  $\varphi$  je dana z matriko

$$R = \begin{bmatrix} \cos \varphi & \sin \varphi \\ -\sin \varphi & \cos \varphi \end{bmatrix}$$

Matriko  $A$  z leve množimo z  $n$ -dimenzionalnimi rotacijskimi matrikami tako, da pod diagonalo dobimo ničle. točen postopek opisujejo formule

$$r = \sqrt{x_i^2 + x_k^2}$$

$$\cos \varphi = x_i / r$$

$$\sin \varphi = x_k / r$$

Takšne rotacije izvajamo, dokler ne dobimo zgornje-trikotne matrike. Kompozicija teh rotacij pa je gotovo ortogonalna matrika. Na primer:

$$R_{12}^T A = \begin{bmatrix} * & * & \dots & * \\ 0 & * & \dots & * \\ * & * & \dots & * \\ \vdots & \vdots & & \vdots \\ * & * & \dots & * \end{bmatrix}$$

Rotacija spreminja kvečjemu vrstice nad izbranim elementom in kvečjemu stolpce desno izbranega elementa. Sledi, da moramo matriko  $A$  pomnožiti s kompozicijo

$$R_{1,m}^T \dots R_{14}^T R_{13}^T R_{12}^T$$

da vse vrednosti v prvem stolpcu pod diagonalo nastavimo na 0.

$$R_{1,m}^T \dots R_{14}^T R_{13}^T R_{12}^T A = \begin{bmatrix} * & * & \dots & * \\ 0 & * & \dots & * \\ 0 & * & \dots & * \\ \vdots & \vdots & & \vdots \\ 0 & * & \dots & * \end{bmatrix}$$

Nato postopek ponovimo, dokler matriki ne zmanjka stolpcev. Vemo, da bo število stolpcev manjše od števila vrstic, saj imamo predoločen sistem. Na koncu imamo ortogonalno matriko  $Q^T$ :

$$Q^T = R_{n,m}^T \dots R_{24}^T R_{23}^T R_{1,m}^T \dots R_{14}^T R_{13}^T R_{12}^T$$

Vemo že, da je kompozicija ortogonalnih matrik ortogonalna.