

Reševanje sistemov nelinearnih enačb

Emil Žagar

3. april 2006

Pogosto namesto ene nelinearne enačbe z eno neznanko, rešujemo sistem več nelinearnih enačb z več neznankami. Števili neznank in enačb sta lahko različni, vendar se bomo tukaj omejili le na sisteme, pri katerih je toliko neznank, kot enačb.

Rešujemo torej sistem

$$\begin{aligned}f_1(x_1, x_2, \dots, x_n) &= 0, \\f_2(x_1, x_2, \dots, x_n) &= 0, \\&\vdots \\f_n(x_1, x_2, \dots, x_n) &= 0.\end{aligned}$$

Zaradi krajše pisave uvedemo oznaki

$$\mathbf{F} = (f_1, f_2, \dots, f_n)^T \quad \text{in} \quad \mathbf{x} = (x_1, x_2, \dots, x_n)^T$$

in zgornji sistem zapišemo na kratko

$$\mathbf{F}(\mathbf{x}) = \mathbf{0}. \tag{1}$$

Primer 1 *Oglejmo si primer preprostega nelinearnega sistema z dvema neznankama (navada je, da v primeru, ko je neznank malo, namesto oznak x_i uporabljamo črke x, y, z, \dots),*

$$\begin{aligned}x^3 - 3xy^2 + 1 &= 0, \\3x^2y - y^3 &= 0.\end{aligned}$$

Da se pokazati, da ima sistem tri realne rešitve, namreč

$$(-1, 0), \quad \left(\frac{1}{2}, \frac{\sqrt{3}}{2}\right), \quad \left(\frac{1}{2}, -\frac{\sqrt{3}}{2}\right).$$

Rešitve prejšnjega sistema nelinearnih enačb bi z malo truda lahko poiskali analitično, v splošnem pa je to nemogoče. Zato smo prisiljeni uporabiti numerične postopke. Med množico metod za reševanje nelinearne enačbe z eno neznanko je le malo takih, ki se jih da preprosto posplošiti na reševanje sistemov. Tukaj si bomo ogledali, kako to naredimo za priljubljeno Newtonovo metodo.

Spomnimo se, da je Newtonova iteracija za reševanje enačbe $f(x) = 0$

$$x_{n+1} = x_n - \frac{f(x)}{f'(x_n)}. \quad (2)$$

Izraz (2) lahko preoblikujemo v

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

Izkaže se, da se ta oblika posploši tudi na reševanje sistemov (1). Newtonova iteracija za sistem enačb (ali, kot ji pogosto rečemo, Newtonova iteracija v več dimenzijah) se namreč zapiše v obliki

$$\mathbf{x}_{n+1} = \mathbf{x}_n - J(\mathbf{x}_n)^{-1} \mathbf{F}(\mathbf{x}_n). \quad (3)$$

Pri tem smo z J označili *Jacobijevo matriko* ali *Jacobijan* sistema. Izračunamo jo po predpisu

$$J(\mathbf{x}) := \begin{pmatrix} \frac{\partial f_1}{\partial x_1}(\mathbf{x}) & \frac{\partial f_1}{\partial x_2}(\mathbf{x}) & \dots & \frac{\partial f_1}{\partial x_n}(\mathbf{x}) \\ \frac{\partial f_2}{\partial x_1}(\mathbf{x}) & \frac{\partial f_2}{\partial x_2}(\mathbf{x}) & \dots & \frac{\partial f_2}{\partial x_n}(\mathbf{x}) \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial f_n}{\partial x_1}(\mathbf{x}) & \frac{\partial f_n}{\partial x_2}(\mathbf{x}) & \dots & \frac{\partial f_n}{\partial x_n}(\mathbf{x}) \end{pmatrix}.$$

Začetni približek za rešitev \mathbf{x}_0 moramo seveda spet izbrati sami. Podobno kot pri navadni Newtonovi metodi, nimamo vedno zagotovljene konvergence.

Z numeričnega vidika je pomembno, kako računamo nove približke po formuli (3). Nedopustno je računati inverz matrike J , saj omenjeno formulo lahko prepisemo v ekvivalentno obliko

$$J\Delta\mathbf{x}_n = -\mathbf{F}(\mathbf{x}_n), \quad (4)$$

pri čemer je $\Delta\mathbf{x}_n := \mathbf{x}_{n+1} - \mathbf{x}_n$. Novi približek se potem izračuna kot $\mathbf{x}_{n+1} = \Delta\mathbf{x}_n + \mathbf{x}_n$. V (4) ne računamo inverza matrike J , ampak rešujemo sistem, kar je (predvsem pri večjih n) mnogo hitreje in bolj natančno.

Oglejmo si še primer uporabe Newtonove iteracije v dveh dimenzijah.

Primer 2 Rešujemo enačbo

$$z^3 + 1 = 0, \quad z \in \mathbb{C}. \quad (5)$$

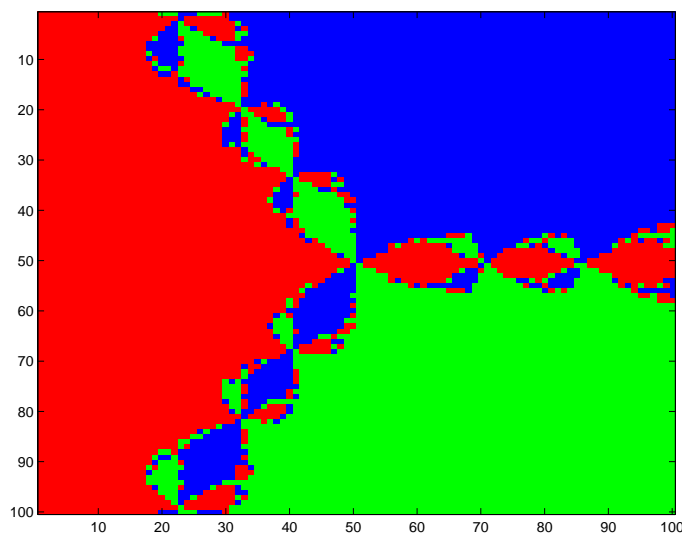
Mnogi programski jeziki ne omogočajo neposrednega računanja s kompleksnimi števili, zato moramo enačbo (5) prevesti v ekvivalentno obliko. To naredimo z uvedbo novih (realnih) spremenljivk x in y ter pišemo $z = x + iy$. Sedaj se (5) prepíše v

$$(x + iy)^3 + 1 = x^3 + 3ix^2y - 3xy^2 - iy^3 + 1 = 0.$$

Na levi morata biti torej realni in imaginarni del enaka nič, kar (ne pozabimo, da sta x in y realna) privede do sistema enačb iz primera 1. Funkcija \mathbf{F} in Jacobijan J sta torej $\mathbf{F}(x, y) = (x^3 - 3xy^2 + 1, 3x^2y - y^3)^T$ in

$$J(x, y) = \begin{pmatrix} 3(x^2 - y^2) & -6xy \\ 6xy & 3(x^2 - y^2) \end{pmatrix},$$

kar uporabimo za reševanje sistema z Newtonovo iteracijo v dveh dimenzijah. Sedaj lahko "narišemo sliko konvergence". Izberemo si območje v ravnini (npr. okrog izhodišča). Vsako od treh rešitev obarvamo z eno od barv. Točko v izbranem območju obarvamo s tisto barvo, kot je obarvana rešitev, h kateri je konvergirala Newtonova metoda z izbrano točko kot začetnim približkom. Tiste točke, ki niso privedle do konvergence, obarvamo črno.



Slika 1: Slika, ki jo dobimo s `slika([-2,2;-2,2],50,1e-5,100,100,1e-4)`.