Rješavanje sustava nelinearnih jednadžbi

1 Newtonova metoda

Promatramo sustava nelinearnih jednadžbi

$$f_i(x_1, x_2, \dots, x_n) = 0, \quad i = 1, \dots, n, \tag{1}$$

odnosno $\mathbf{f}(\mathbf{x}) = \mathbf{0}$, gdje je $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^n$, $\mathbf{f}(\mathbf{x}) = (f_1(x), \dots, f_n(x))^T, \quad \mathbf{x} = (x_1, \dots, x_n)^T$.

Najpoznatija metoda za rješavanje sustava nelinearnih jednadžbi je tzv. Newtonova metoda.[1] Najprije izaberimo početnu aproksimaciju $\mathbf{x}^{(0)} = (x_1^{(0)}, x_2^{(0)}, \dots, x_n^{(0)})^T$ i svaku od funkcija f_i razvijemo u Taylorov red u okolini $x^{(0)}$, te linearnu aproksimaciju označimo s $\tilde{f}_i$:

$$\tilde{f}_i(\mathbf{x}) = f_i(\mathbf{x}^{(0)}) + \sum_{j=1}^{n} \frac{\partial f_i(\mathbf{x}^{(0)})}{\partial x_j}(x_j - x_j^{(0)}), \quad i = 1, \dots, n. \tag{2}$$

Sada umjesto sustava (1) rješavamo sustav

$$\tilde{f}_i(\mathbf{x}) = 0, \quad i = 1, \dots, n,$$

što možemo pisati u matričnom obliku na sljedeći način

$$\mathbf{J}^{(0)}\mathbf{s}^{(0)} = -\mathbf{f}(\mathbf{x}^{(0)}), \tag{3}$$

$$\mathbf{J}^{(0)} = \begin{bmatrix} \frac{\partial f_1(\mathbf{x}^{(0)})}{\partial x_1} & \cdots & \frac{\partial f_1(\mathbf{x}^{(0)})}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n(\mathbf{x}^{(0)})}{\partial x_1} & \cdots & \frac{\partial f_n(\mathbf{x}^{(0)})}{\partial x_n} \end{bmatrix}, \ \mathbf{s}^{(0)} = \begin{bmatrix} x_1 - x_1^{(0)} \\ \vdots \\ x_n - x_n^{(0)} \end{bmatrix}, \ \mathbf{f}(\mathbf{x}^{(0)}) = \begin{bmatrix} f_1(\mathbf{x}^{(0)}) \\ \vdots \\ f_n(\mathbf{x}^{(0)}) \end{bmatrix}.$$

Matricu $\mathbf{J}$ nazivamo Jacobijeva matrica ili Jacobijan sustava. Nova aproksimacija rješenja tada je

$$\mathbf{x}^{(1)} = \mathbf{x}^{(0)} + \mathbf{s}^{(0)}.$$

Općenito, dobivamo iterativni postupak

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \mathbf{s}^{(k)}, \quad k = 0, 1, \dots \tag{4}$$

gdje je vektor smjera kretanja $\mathbf{s}^{(k)}$ rješenje sustava

$$\mathbf{J}^{(k)}\mathbf{s} = -\mathbf{f}(\mathbf{x}^{(k)}). \tag{5}$$

Primijetite da se iterativni postupak (4)–(5) uz pretpostavku regularnosti Jacobijana $\mathbf{J}^{(k)}$ može zapisati u obliku

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \left(\mathbf{J}^{(k)}\right)^{-1} \mathbf{f}(\mathbf{x}^{(k)}), \quad k = 0, 1, \dots, \tag{6}$$

što podsjeća na Newtonovu metodu tangenti za rje˘savanje jednadžbe $f(x) = 0, f : \mathbb{R} \to \mathbb{R}$. Inače, iterativni proces (6) rijetko se koristi jer je postupak (4)–(5) numerički stabilniji.

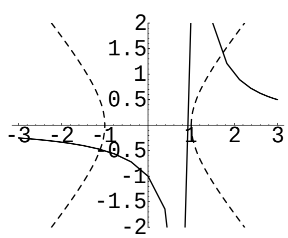
[1] U literaturi ova metoda često se može naći pod nazivom: Newton-Raphson method

Ako početnu aproksimaciju $\mathbf{x}^{(0)}$ izaberemo dovoljno blizu rješenja $\mathbf{x}^{\star}$ i ako je u svakoj iteraciji Jacobijan $\mathbf{J}$ nesingularan, onda Newtonova metoda (4) vrlo brzo (kvadratičnom brzinom) konvergira prema rješenju sustava jednadžbi (1) (više detalja vidi u [4] ili [8]). To zapravo znači da u praksi nemamo garancije da će Newtonova metoda uvijek konvergirati, ali ako smo u stanju početnu aproksimaciju željenog rješenja dobro procijenit, ova metoda će nas vrlo brzo dovesti do pravog rješenja.

Primjer 1 *Treba riješiti sustav jednadžbi:*

$$\begin{aligned} x_2(x_1 - 1) - 1 &= 0 \\ x_1^2 - x_2^2 - 1 &= 0 \end{aligned}$$

Geometrijski, to znači da treba pronači sjecišta odgovarajućih krivulja čiji su grafovi prikazani na *Slici 1*.



Slika 1. $x_2(x_1 - 1) - 1 = 0, \quad x_1^2 - x_2^2 - 1 = 0$

Ovu sliku lako se može nacrtati u *Mathematici*. Najprije treba pozvati *Mathematic-package*:

```
In[1]:= << Graphics‘ImplicitPlot‘
```

a zatim sljedećom naredbom dobivamo *Sliku 1*:

```
In[2]:=ImplicitPlot[{x2(x1 - 1) - 1 == 0, x1^2 - x2^2 - 1 == 0}, {x1, -3, 3},
                    PlotStyle -> {GrayLevel[0], Dashing[{.02}]},
                    PlotRange -> {-2, 2}, AspectRatio -> .7]
```

U ovom slučaju imamo:

$$\mathbf{f}(x_1, x_2) = \begin{bmatrix} x_2(x_1 - 1) - 1 \\ x_1^2 - x_2^2 - 1 \end{bmatrix}, \quad \mathbf{J}(x_1, x_2) = \begin{bmatrix} x_2 & x_1 - 1 \\ 2x_1 & -2x_2 \end{bmatrix}.$$

Funkciju $\mathbf{f}$ i njen Jacobijan $\mathbf{J}$ definirat ćemo na sljedeći način:

```
In[3]:= Clear[f, J, x1, x2]; n = 2;
        f[x_] := {x[2](x[1] - 1) - 1, x[1]^2 - x[2]^2 - 1};
        f[x] /. {x[1] -> x1, x[2] -> x2}
        J[x_] := Table[D[f[x][[i]], x[j]], {i, n}, {j, n}]
        J[x] /. {x[1] -> x1, x[2] -> x2}
```

Počevši primjerice s početnom aproksimacijom $x^{(0)} = (2, 1)^T$ korištenjem jednostavnog programa

```
In[9]:= w = Table[x[i] -> x0[[i]], {i, n}];
        a = H[x] /. w ;
        b = -f[x] /. w;
        Print["x0=", N[x0], "  f0= ", -N[b]]
        s0 = LinearSolve[a, b];
        x1 = x0 + s0; x0 = x1;
```

dobivamo $f^{(0)} = (0, 2)^T$. Ponavljajući ovaj kratnki program dobivamo vrijednosti prikazane u *Tablici 1.*

k	$x_1^{(k)}$	$x_2^{(k)}$	$\mathbf{f}(x_1^{(k)}, x_2^{(k)})^T$	$x_1^{(k)}$	$x_2^{(k)}$	$\mathbf{f}(x_1^{(k)}, x_2^{(k)})^T$
0	2	1	$(0, 2)$	-1	-.5	$(0, -.25)$
1	1.66667	1.33333	(-0.111111, 0.)	-1.11111	-0.472222	(0.003, -0.012)
2	1.71795	1.39744	(-0.003, 0.001)	-1.10692	-0.474621	(-0.00001, 0.00001)
3	1.71667	1.39534	$(2.7^{-6}, -2.7^{-6})$			

Tablica 1

Također za $x^{(0)} = (-1, -.5)^T$ dobivamo $f^{(0)} = (0, -.25)^T$, te u u samo dva koraka rješenje prikazano u *Tablici 1.* U ovom slučaju zadovoljit ćemo se kriterijem da iterativni proces zaustavimo kad vrijenost svih funkcija f_i postanu manje od nekog unaprijed zadanog broja $\varepsilon > 0$, tj. kada za neki k bude ispunjeno

$$\|\mathbf{f}(\mathbf{x}^{(k)})\|_\infty = \max_i |f_i(\mathbf{x}^{(k)})| < \varepsilon.$$

Sada ćemo sagraditi *Mathematica*-modul SNJ koji će rješavati sustav od n nelinearnih jednadžbi s n nepoznanica:

```
In[14]:= SNJ[n_, x0_, eps_, it_, f_] := Module[{a, b, s0, k = 0},
             w = Table[x[i] -> x0[[i]], {i, n}];
             J[x_] := Table[D[f[x][[i]], x[j]], {i, n}, {j, n}];
             a = J[x] /. w ;
             b = -f[x] /. w;
             xs = x0;
             Print["x0=", N[x0], "   f0=", -N[b]];
             While[k = k + 1; Max[Abs[b]] > eps && k < it,
                   s0 = LinearSolve[a, b];
                   xn = xs + s0;
                   xs = xn;
                   w = Table[x[i] -> xs[[i]], {i, n}];
                   a = J[x] /. w ;
                   b = -f[x] /. w;
                   Print["x", k, "=", N[xn], "   f=", -N[b]];
                    ]
               ]
```

Modulu SNJ predajemo sljedeće podatke:

n — broj jedndžbi (odnosno nepoznanica)
$x0$ — vektor početne aproksimacije
eps — zahtijevana točnost
it — maksimalno dozvoljen broj iteracija
f — funkcija $f : \mathbb{R}^n \to \mathbb{R}^n$

Iterativni proces teče i printaju se međurezultati izračunavanja tako dugo dok je $\|\mathbf{f}(\mathbf{x}^{(k)})\|_\infty > eps$ i maksimalno dozvoljen broj iteracija $< it$.

Zadatak 1 *Naći rješenja sustava nelinearnih jednadžbi*

$$\begin{aligned} 3x_1^2 + \tfrac{3}{2}x_2^2 + x_3^2 - 5 &= 0 \\ 6x_1x_2x_3 - x_1 + 5x_2 + 3x_3 &= 0 \\ 5x_1x_3 - x_2x_3 - 1 &= 0 \end{aligned}$$

Rješenje: $x_1 = 1.2844570503, \quad x_2 = 0.1297565120, \quad x_3 = 0.1589186226.$

Zadatak 2 *Naći rješenja sustava nelinearnih jednadžbi*

$$\begin{aligned} 2x_1^2 + x_2^2 - 1 &= 0 \\ x_1^3 + 6x_1^2 x_2 - 1 &= 0 \end{aligned}$$

Rješenje: $x_1 = 0.441615, \quad x_2 = 0.780995$

Zadatak 3 *Naći rješenja sustava nelinearnih jednadžbi*

$$\begin{aligned} 2x_1 - x_2 - 6 \ln x - 3 &= 0 \\ 15x_1 - 10x_2 - 60 \ln x_2 - 6 &= 0 \end{aligned}$$

Rješenje: $x_1 = 0.646061, \quad x_2 = 0.913292.$

Zadatak 4 *Naći rješenja sustava nelinearnih jednadžbi*

$$\begin{aligned} x_1 + 3\log_{10} x_1 - x_2^2 &= 0 \\ 2x_1^2 - x_1 x_2 - 5x_1 + 1 &= 0 \end{aligned}$$

Rješenje: $x_1 = 3.86863, \quad x_2 = 2.99575.$

Zadatak 5 *Za različite vrijednosti od n naći rješenja sustava nelinearnih jednadžbi (vidi* **??***)*

$$\begin{aligned} f_i \equiv x_i - \tfrac{x_{i+1}^2}{10} &= 0, \quad i = 1, \ldots, n-1 \\ f_n \equiv x_n - \tfrac{x_1^2}{10} &= 0 \end{aligned}$$

Rješenje: $x^1 = (0, \cdots, 0)^T, \quad x^2 = (10, \cdots, 10)^T.$

Zadatak 6 *Sljedeći sustav jednadžbi javlja se kod problema modeliranja tranzistora (vidi također [9]).*

$$\begin{aligned} (1 - x_1x_2)x_3\left(\exp(x_5(g_{1k} - g_{3k}x_7^{-3} - g_{5k}x_8^{-3})) - 1\right) - g_{5k} + g_{4k}x_2 &= 0, \quad k = 1,2,3,4 \\ (1 - x_1x_2)x_4\left(\exp(x_6(g_{1k} - g_{2k} - g_{3k}x_7^{-3})) - 1\right) - g_{5k}x_1 + g_{4k} &= 0, \quad k = 1,2,3,4 \\ x_1x_3 - x_2x_4 &= 0, \end{aligned}$$

gdje su $g_{i,j}$ elementi matrice

$$\mathbf{G} = \begin{bmatrix} .485 & .752 & .869 & .982 \\ .369 & 1.254 & .703 & 1.455 \\ 5.2095 & 10.0677 & 22.9274 & 20.2153 \\ 23.3037 & 101.779 & 111.461 & 191.267 \\ 28.5132 & 111.8467 & 134.3834 & 211.4823 \end{bmatrix}.$$

Rješenje:

$$\begin{aligned} x^1 &= (.9, \quad .449987, \quad 1.00001, \quad 2.00007, \quad 7.99997, \quad 7.99969, \quad 5.00003, \quad .999988, \quad 2.00005)^T \\ x^2 &= (.898510, \quad .973968, \quad 11.6486, \quad 10.7462, \quad 3.25083, \quad 6.71055, \quad -8.764, \quad 1.25133, \quad -.525131)^T. \end{aligned}$$

Zadatak 7 *U [14] naveden je sljedeći primjer sustava nelinearnih jednadžbi*

$$\begin{aligned} f_1 &= \arctan(x_1) + 2x_1 - x_2 = 0 \\ f_2 &= -2x_1 + x_2 = 0, \end{aligned}$$

koji ima jedinstveno rješenje $(0,0)^T$. Pokažite da Newtonova metoda za proizvoljni $x_2^{(0)}$ konvergira prema rješenju ako je $|x_1^{(0)}| < 1.391$ i divergira ako je $|x_1^{(0)}| > 1.392$.

Uputa: Razmotrite omjer $\frac{x_1^{(n+1)}}{x_1^{(n)}}$. Poslužite se grafičkim prikazom.

Zadatak 8 *U [5] navedeni su sljedeći test problemi s početnom aproksimacijom $x^{(0)}$ i rješenjem $x^\star$*

(a) K.M. Brown (1969) $n = 5$, $x^{(0)} = (.5, \dots, .5)^T$, $x^\star = (1, \dots, 1)^T$.

$$\begin{aligned} f_i &= -(n+1) + 2x_i + \sum_{j\neq i}^{n} x_j, \quad 1 \le i \le n-1 \\ f_n &= -1 + \prod_{j=1}^{n} x_j = 0 \end{aligned}$$

(b) K.M. Brown (1969) $n = 2$, $x^{(0)} = (.1, 2)^T$, $x^\star = (1.06735, .139228)^T$.

$$\begin{aligned} f_1 &= x_1^2 - x_2 - 1 = 0 \\ f_2 &= (x_1 - 2)^2 + (x_2 - .5)^2 - 1 = 0 \end{aligned}$$

(c) R. Fletcher (1965) $n = 2 \dots, 9$, $x_j^{(0)} = j/(n+1)$, $1 \le j \le n$;
$x_j^\star$ *su permutacije apscisa Čebiševljeve kvadraturne formule reda* n.

$$f_i = \int_0^1 T_i(\zeta)d\zeta - \frac{1}{n}\sum_{j=1}^{n} T_i(x_j),$$

gdje je T_i *i-ti Čebiševljev polinom transformiran na interval* $[0, 1]$, *tj.*

$$T_0(\zeta) \equiv 1, \quad T_1(\zeta) = 2\zeta - 1, \quad T_{i+1}(\zeta) = 2(2\zeta - 1)T_i(\zeta) - T_{i-1}(\zeta), \text{ za } i \ge 1.$$

Primijetite da vrijedi:

$$\int_0^1 T_i(\zeta)d\zeta = \begin{cases} 0, & i \text{ neparan} \\ \frac{-1}{i^2-1}, & i \text{ paran} \end{cases}$$

(d) K. M. Brown, S. D. Conte (1967) $n = 2$, $x^{(0)} = (.6, 3)^T$, $x^\star = (.5, \pi)^T$.

$$\begin{aligned} f_1 &= \tfrac{1}{2}\sin(x_1 x_2) - \tfrac{x_2}{4\pi} - \tfrac{x_1}{2} = 0 \\ f_2 &= \left(1 - \tfrac{1}{4\pi}\right)\left(e^{2x_1} - e\right) + \tfrac{e x_2}{\pi} - 2e x_1 = 0 \end{aligned}$$

(e) K. M. Brown, W. B. Gearhart (1971) $n = 3$, $x^{(0)} = (1, .7, 5)^T$, $x^\star = (0, \sqrt{2}, 6)^T$.

$$\begin{aligned} f_1 &= x_1^2 + 2x_2^2 - 4 = 0 \\ f_2 &= x_1^2 + x_2^2 + x_3 - 8 = 0 \\ f_3 &= (x_1 - 1)^2 + (2x_2 - \sqrt{2})^2 + (x_3 - 5)^2 - 4 = 0 \end{aligned}$$

(f) C. G. Broyden (1965) $n = 5, 10$, $x^{(0)} = (-1, \dots, -1)^T$.

$$\begin{aligned} f_1 &= (.5x_1 - 3)x_1 + 2x_2 - 1 \\ f_i &= x_{i-1} + (.5x_i - 3)x_i + 2x_{i+1} - 1 \quad 2 \le i \le n-1 \\ f_n &= x_{n-1} + (.5x_n - 3)x_n - 1 \end{aligned}$$

Za $n = 5$, $x^\star = (-.968, -1.18696, -1.14848, -.958989, -.594159)^T$

Za $n = 10$, $x^\star = (-1.03011, -1.310444, -1.37992, -1.39071, -1.37963, -1.34993, -1.29066, -1.17748, -.967501, -.596526)^T$

2 Quasi–Newtonova metoda

Vrlo važno mjesto među metodama za rješavanje sustava nelinearnih jednadžbi zauzimaju tzv. Quasi-Newtonove metode, koje je uveo Broyden (1965), a koje su nastale kao generalizacija metode sekanti.

Kao što je pokazano u *t. 1* aproksimaciju funkcije f u okolini točke x^0 možemo zapisati kao

$$f(x) \cong f(x^0) + J(x^0)(x - x^0).$$

Ako s $\bar{B}$ označimo aproksimaciju Jacobijana J u točki x^0, onda ima smisla postaviti zahtjev da $\bar{B}$ zadovoljava jednadžbu

$$f(x) = f(x^0) + \bar{B}(x - x^0),$$

što uz oznake: $s := x - x^0$, $y := f(x) - f(x^0)$ možemo pisati

$$\bar{B}s = y \tag{7}$$

Ako je B poznata aproksimacija od J, Broydenova pretpostavka je da se $\bar{B}$ razlikuje od B samo u smjeru s, tj. da se $\bar{B}$ ne razlikuje od B na ortogonalnom komplementu od s, tj.

$$\bar{B}z = Bz, \text{ ako je } (z, s) = 0. \tag{8}$$

Lako se može pokazati da $\bar{B}$ definiran s

$$\bar{B} = B + \frac{(y - Bs)s^T}{(s, s)} \tag{9}$$

zadovoljava (7) i (8), te da se $\bar{B}$ dobiva od B dodavanjem matrice (korekcije) ranga 1.

Na taj način dobivamo direktnu Broydenovu metodu

$$x_{k+1} = x_k - B_k^{-1} f(x_k), \quad k = 0, 1, \dots , \tag{10}$$

gdje se matrice $B_k \in \mathbb{R}^{n \times n}$ generiraju rekurzivnom formulom

$$B_{k+1} = B_k + \frac{(y_k - B_k s_k)s_k^T}{(s_k, s_k)}, \quad k = 0, 1, \dots , \tag{11}$$

gdje je

$$y_k = f(x_{k+1} - f(x_k), \quad s_k = x_{k+1} - x_k. \tag{12}$$

Izradit ćemo *Mathematica*–program za direktnu Broydenovu metodu i testirati ga na *Primjeru 1*, str. 2. Najprije definirajmo funkciju f, broj jednadžbi n, početnu aproksimaciju x_0, točnost eps, maksimalno dozvoljen broj iteracija it i te jediničnu matricu B_0 reda n kao početnui aproksimaciju jacobijana, a brojač iteracija k postsvimo na 0.

```
In[1]:= f[x_]:= {x[2](x[1] - 1) - 1, x[1]^2 - x[2]^2 - 1};
        n = 2; k = 0;  x0 = {2., 2.}; eps = .00005; it = 100;
        B0 = IdentityMatrix[n];
        Print["x0=", x0, "   f0=", f[x]/.Table[x[i]->x0[[i]], {i,n}]]
```

Nakon toga koristeći algoritam opisan s (10)–(12) napišimo program za direktnu Broydenovu metodu

```
In[2]:= While[k = k + 1;
            w0 = Table[x[i] -> x0[[i]], {i, n}];
            s0 = -Inverse[B0].f[x] /. w0;
            x1 = x0 + s0;
            w1 = Table[x[i] -> x1[[i]], {i, n}];
            f1 = f[x] /. w1; f0 = f[x] /. w0; y = f1 - f0;
            Max[Abs[f1]] > eps && k < it,
            Print["x", k, "=", x1, "     f", k, "=", f1]
            If[Apply[Plus, s0 Inverse[B0].y] == 0, B1 = B0,
            v = s0/Apply[Plus, s0  s0]];
            vv = y - B0.s0; B1 = B0 + Table[vv[[i]] v[[j]], {i, n}, {j, n}];
            x0 = x1; B0 = B1
            ]
```

Računanje inverzne matrice B_k^{-1} može se izbjeći tako da umjesto (10) rješavamo sustav linearnih jednadžbi

$$B_k s_k = -f(x_k).$$

Drugi način da bi izbjegli računanje inverzne matrice B_k^{-1} je primjena *Sherman–Morrisonove leme*[2] Korištenjem Sherman–Morrisonove leme iz (11) izračunat ćemo B_{k+1}^{-1}. Uz oznake

$$u = \frac{y_k - B_k s_k}{(s_k, s_k)}, \quad v = s_k,$$

dobivamo

$$\sigma = 1 + \left(s_k, B_k^{-1} \frac{y_k - B_k s_k}{(s_k, s_k)} \right) = \frac{(s_k, B_k^{-1} y_k)}{(s_k, s_k)},$$

$$B_{k+1}^{-1} = B_k^{-1} - \frac{(s_k, s_k)}{(s_k, B_k^{-1} y_k)} B_k^{-1} \frac{y_k - B_k s_k}{(s_k, s_k)} s_k^T B_k^{-1} = B_k^{-1} + \frac{(s_k - B_k^{-1} y_k) s_k^T B_k^{-1}}{(s_k, B_k^{-1} y_k)}$$

Prema tome ako označimo: $H_k := B_k^{-1}$, onda je $H_{k+1} := B_{k+1}^{-1}$ definiran s

$$H_{k+1} = H_k + \frac{(s_k - H_k y_k) s_k^T H_k}{(s_k, H_k y_k)}, \quad \text{uz uvjet } (s_k, H_k y_k) \neq 0, \tag{13}$$

čime je definiran inverzni Broydenova metoda

$$x_{k+1} = x_k - H_k f(x_k), \quad k = 0, 1, \dots, \tag{14}$$

Ako najprije definiramo potrebne parametre slično kao i kod direktne Broydenove metode

```
In[1]:= f[x_]:= {x[2](x[1] - 1) - 1, x[1]^2 - x[2]^2 - 1};
        n = 2; k = 0;  x0 = {2., 2.}; eps = .00005; it = 100;
        H0 = IdentityMatrix[n];
        Print["x0=", x0, "   f0=", f[x]/.Table[x[i]->x0[[i]], {i,n}]]
```

onda koristeći algoritam opisan s (13)–(14) možemo napisati program za invrznu Broydenovu metodu

[2] Lema (Sherman-Morrison (1949)). Neka su $u, v \in \mathbb{R}^n$ i neka je $A \in \mathbb{R}^{n \times n}$ regularna matrica. Tada je matrica $A + uv^T$ regularna onda i samo onda ako je $\sigma = 1 + (v, A^{-1}u) \neq 0$. Ako je $\sigma \neq 0$, onda vrijedi

$$(A + uv^T)^{-1} = A^{-1} - (1/\sigma) A^{-1} u v^T A^{-1}.$$

```
In[2]:= While[k = k + 1;
            w0 = Table[x[i] -> x0[[i]], {i, n}];
            s0 = -H0.f[x] /. w0;
            x1 = x0 + s0;
            w1 = Table[x[i] -> x1[[i]], {i, n}];
            f1 = f[x] /. w1; f0 = f[x] /. w0; y = f1 - f0;
            Max[Abs[f1]] > eps && k < it,
            Print["x", k, "=", x1, "     f", k, "=", f1]
            If[y == Table[0, {i, n}], H1 = H0,
            v = (Transpose[H0].s0)/Apply[Plus, (H0.y)  s0]];
            vv = s0 - H0.y; H1 = H0 + Table[vv[[i]] v[[j]], {i,n}, {j,n}];
            x0 = x1; H0 = H1
            ]
```

Naravno, rezultati izvođenja Broydenove direktne i inverzne metode moraju biti isti. U *Tablici 2* prikazano je prvih 8 iteracija na *Primjer 1.*.

k	x_1^k	x_2^k	$\|f^k\|_\infty$
0	2	2	1
1	1	3	9
2	0.666667	0	1
3	1.41165	0.847744	.65
4	4.18449	5.07559	15.16
5	1.537	1.00531	.46
6	1.60838	1.13469	.31
7	1.75064	1.45862	.095
8	1.71616	1.38812	.018

Tablica 2 Broydenova metoda

Broydenova metoda je sporija od Newtonove, ali su njene prednosti u tome što nije potrebno izračunavati Jacobijan funkcije f.

Zadatak 9 *Dokažite Sherman–Morrisonovu lemu.*

Zadatak 10 *Usporedite Newtonovu metodu s Broydenovom direktnom i inverznom metodom tako da pratite potrebno vrijeme računanja za ranije navedene primjere. Za $n = 2$ nacrtajte grafove funkcija f_1, f_2, te ucrtajte točke koje pretstavljaju pojedine iteracije.*

Navest ćemo još dvije najpoznatije quasi-Newtonove metode ranga 2 (vid primjerice [3], [4], [6]):

(a) Davidon-Fletcher-Powell (DFP) metoda

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \mathbf{H}_k \mathbf{f}(\mathbf{x}_k), \quad k = 0, 1, \dots ,$$

$$\mathbf{H}_{k+1} = \mathbf{H}_k + \frac{\mathbf{s}_k \mathbf{s}_k^T}{(\mathbf{y}_k, \mathbf{s}_k)} - \frac{\mathbf{H}_k \mathbf{y}_k \mathbf{y}_k^T \mathbf{H}_k}{(\mathbf{y}_k, \mathbf{H}_k \mathbf{y}_k)}, \quad k = 0, 1, \dots ,$$

(b) Broyden-Fletcher-Goldfarb-Schano (BFGS) metoda

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \mathbf{H}_k \mathbf{f}(\mathbf{x}_k), \quad k = 0, 1, \dots ,$$

$$\mathbf{H}_{k+1} = \left(\mathbf{I} - \frac{\mathbf{s}_k \mathbf{y}_k^T}{(\mathbf{y}_k, \mathbf{s}_k)}\right) \mathbf{H}_k \left(\mathbf{I} - \frac{\mathbf{y}_k \mathbf{s}_k^T}{(\mathbf{y}_k, \mathbf{s}_k)}\right) + \frac{\mathbf{s}_k \mathbf{s}_k^T}{(\mathbf{y}_k, \mathbf{s}_k)}, \quad k = 0, 1, \dots ,$$

Više tetalja o ovim metodama, o uvjetima i brzini konvergencije, te njihovim svojstvima može se vidjeti u niže navedenoj literaturi, a posebno u [3], [6]

Zadatak 11 *Izradite programe za DFP i BFGS metode. Testirajte ove metode s drugim na ranije danim primjerima.*

Literatura

[1] G. Dahlquist, A. Björck *Numerische Methoden*, R. Oldenbourg Verlag, München, Wien, 1972. (postoji i engleski prijevod)

[2] E. W. Cheney, *Introduction to Approximation Theory*, AMS Chelsea Publishing, 1998.

[3] J. E. Dennis Jr., J. J. Moré *Quasi–Newton methods, motivation and theory*, SIAM Review **19**(1977), 46–89.

[4] J. E. Dennis, R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, SIAM, Philadelphia, 1996.

[5] D. M. Gay, R. B. Schnabel, *Solving system of nonlinear equations by Broyden's method with projected updates*, in: Nonlinear Programming 3, edited by O. L. Mangasarian, R. R. Meyer, S. M. Robinson, Academic Press, New York, 1978, 245–281.

[6] C. T. Kelley, *Iterative Methods for Optimization*, SIAM, Philadelphia, 1999.

[7] D. Kincaid,W. Cheney, *Numerical Analysis*, Brooks/Cole Publishing Company, New York, 1996.

[8] J. M. Ortega,W. C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, New York, 1970. (postoji i ruski prijevod)

[9] W. L. Price, *A weighted simplex procedure for the solution of simultaneous non-linear equations*, J. Inst. Maths. Applics. **24**(1979), 1–8.

[10] W. H. Press, B. P. Flannery, S. A. Teukolsky, W. T. Vetterling, *Numerical Recipes*, Cambridge University Press, Cambridge, 1989.

[11] H. R. Schwarz, *Numerische Mathematik*, Teubner, Stuttgart, 1986.

[12] R. Scitovski, *Numerička matematika*, Odjel za matematiku, Sveučilište u Osijeku, Osijek, 2000

[13] H. Späth, *Numerik*, Vieweg, 1994.

[14] L. Wegge, *On a discrete version of the Newton–Raphson method*, SIAM J. Numer. Anal. **3**(1966), 134–142.