1 Metoda konjugiranih gradientov s predpogojevanjem

Avtor: Luka Bajić

1.1 Opis problema

Metoda konjugiranih gradientov je postopek za reševanje linearnega sistema enačb $Ax = b$, ob predpostavki, da je matrika A pozitivno definitna. Ker imamo opravka z razpršenimi matrikami, je s stališča prostorske zahtevnosti smiselno, da jih hranimo v posebni strukturi, ki hrani samo neničelne elemente in sestoji iz treh vektorjev: vektor vrednosti v matriki, vektor indeksov po vrsticah in vektor indeksov po stolpcih.

1.2 Opis rešitve

Implementacija ponuja tri metode: *nep_chol*, ki izračuna nepopolni razcep Choleskega za podano matriko A, *conj_grad_baseline*, ki izvede metodo konjugiranih gradientov nad podano matriko A in vektorjem desnih strani b in vrne iskan x, in *conj_grad*, ki prav tako vrne x, le da metodo konjugiranih gradientov izvede s predpogojevanjem s podano spodnjetrikotno matriko L.

1.2.1 Nepopolni razcep Choleskega

1.2.2 Metoda konjugiranih gradientov brez predpogojevanja

1.2.3 Metoda konjugiranih gradientov s predpogojevanjem

1.3 Primer uporabe

Spodnji izsek kode demonstrira razliko med metodo konjugiranih gradientov brez predpogojevanja in s predpogojevanjem.

```
using MKG, Plots, LinearAlgebra, SparseArrays

I = [1., 1, 2, 2, 3, 3, 4, 5, 5]
J = [1., 2, 1, 2, 3, 5, 4, 3, 5]
V = [7., 1.1, 1.1, 2, 3, 3, 0.5, 3, 4.2]
A = sparse(I, J, V)
b = [2, 3, -5, 1, 0.2]
L = nep_chol(A)
x1, it1, res1 = conj_grad_baseline(A, b, vrniresid=true)
x2, it2, res2 = conj_grad(A, b, L, vrniresid=true)

println("MKG brez predpogojevanja se zaustavi po $it1 korakih.")
println("MKG s predpogojevanjem se zaustavi po $it2 korakih.")
```

```
MKG brez predpogojevanja se zaustavi po 5 korakih.
MKG s predpogojevanjem se zaustavi po 1 korakih.
```

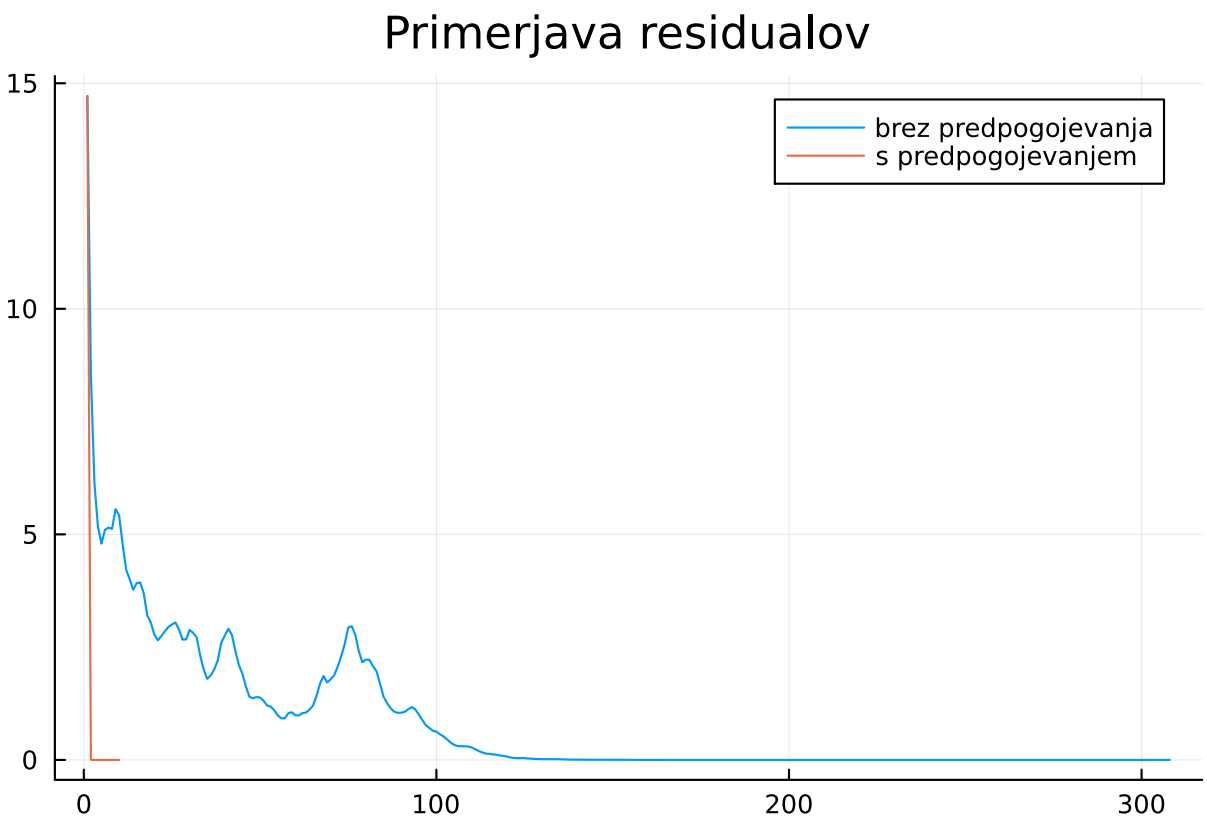
Kot vidimo že na relativno majhnem primeru, predpogojevanje bistveno izboljša konvergenco.

V spodnjem izseku si ogledamo še primer z bistveno večjo matriko, ki jo zgeneriramo psevdonaključno. Z zastavico *vrniresid* nam metodi vrneta tabelo residualov na posameznem koraku iteracije. Residuale izrišemo s paketom Plots in tako dobimo vizualno primerjavo hitrosti konvergence med metodama.

```
using Plots
n=700
I = [1.0]
J = [1.0]
V = [rand()*10.0]
for i=2:n
    push!(V, rand()*80.0)
    push!(I, i)
    push!(J, i)

    if rand() < 0.1
        r = rand()
        push!(V, r)
        push!(V, r)
        r1 = rand(1:n)
        r2 = rand(1:n)
        while r1==i || r2==i
            r1 = rand(1:n)
            r2 = rand(1:n)
        end
        push!(I, r1)
        push!(J, r2)
        push!(I, r2)
        push!(J, r1)
    end
end
A = sparse(I, J, V)
b = rand(n)
x1, it1, res1 = conj_grad_baseline(A, b, vrniresid=true, tol=10e-20)
L = nep_chol(A)
x2, it2, res2 = conj_grad(A, b, L, vrniresid=true, tol=10e-20)

plot(res1, label="brez predpogojevanja", title="Primerjava residualov")
plot!(res2, label="s predpogojevanjem", title="Primerjava residualov")
```


Primerjava residualov
brez predpogojevanja
s predpogojevanjem
15
10
5
0
0
100
200
300