

# Iterirani Ritzov postupak za rješavanje sustava linearnih algebarskih jednadžbi

---

Dvornik, Josip; Lazarević, Damir

Source / Izvornik: **Građevinar**, 2017, 69, 521 - 535

Journal article, Published version

Rad u časopisu, Objavljena verzija rada (izdavačev PDF)

<https://doi.org/10.14256/JCE.2036.2017>

Permanent link / Trajna poveznica: <https://urn.nsk.hr/urn:nbn:hr:237:006494>

Rights / Prava: [In copyright](#) / [Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2025-01-23**

Repository / Repozitorij:

[Repository of the Faculty of Civil Engineering,  
University of Zagreb](#)



Primljen / Received: 10.3.2017.

Ispravljen / Corrected: 25.5.2017.

Prihvaćen / Accepted: 20.6.2017.

Dostupno online / Available online: 10.8.2017.

# Iterirani Ritzov postupak za rješavanje sustava linearnih algebarskih jednačbi

## Autori:



Prof.emer.dr.sc. **Josip Dvornik**, dipl.ing.građ.  
Sveučilište u Zagrebu  
Građevinski fakultet  
Zavod za tehničku mehaniku  
[dvornik@grad.hr](mailto:dvornik@grad.hr)



Prof.dr.sc. **Damir Lazarević**, dipl.ing.građ.  
Sveučilište u Zagrebu  
Građevinski fakultet  
Zavod za tehničku mehaniku  
[damir@grad.hr](mailto:damir@grad.hr)

Izvorni znanstveni rad

**Josip Dvornik, Damir Lazarević**

## Iterirani Ritzov postupak za rješavanje sustava linearnih algebarskih jednačbi

U radu je opisano stanje razvoja nove iteracijske metode za rješavanje sustava linearnih algebarskih jednačbi. Metoda je pogodna za izrazito velike sustave slabo popunjenih matrica. Osim vlastitih obilježja, posjeduje i svojstvo općenitosti, jer su mnogi iteracijski postupci samo poseban slučaj ovoga pristupa. Algoritam je realiziran samostalno, a potom je pridružen programu otvorena koda FEAP. Provedene su i raznolike provjere, posebice na praktičnim modelima. Premda je postupak tek djelomice istražen, već pokazuje dobre rezultate.

### Ključne riječi:

iteracijski postupak, Jacobijeva metoda, Gauß – Seidelova metoda, metoda najstrmijega silaska, metoda konjugiranih gradijenata

Original scientific paper

**Josip Dvornik, Damir Lazarević**

## Iterated Ritz Method for solving systems of linear algebraic equations

The paper describes research state of a new iterative method for solving systems of linear algebraic equations. The method is suitable for extremely large systems with sparse matrices. In addition to its own characteristics, it also has a feature of generality, as many iterative methods are only special cases of this approach. The algorithm was developed independently, and then implemented into the open source code program FEAP. Also, various checks were conducted, especially on practical models. Although the method has been only partially studied, good results have already been obtained.

### Key words:

iterative method, Jacobi method, Gauß – Seidel method, method of steepest descent, conjugate gradient method

Wissenschaftlicher Originalbeitrag

**Josip Dvornik, Damir Lazarević**

## Iteriertes Ritz-Verfahren zur Lösung von linearen Gleichungssystemen

In der Arbeit wird der aktuelle Entwicklungsstand des neuen Iterationsverfahrens für die Lösung von linearen Gleichungssystemen beschrieben. Das Verfahren eignet sich insbesondere für sehr große Systeme mit schwach gefüllten Matrizen. Es besitzt neben eigenen Merkmalen auch das Merkmal der Allgemeinheit, da viele Iterationsverfahren nur ein Spezialfall dieses Ansatzes sind. Der Algorithmus wurde selbständig ermittelt und danach dem FEAP Programm mit einem offenen Programmcode zugeordnet. Es wurden auch zahlreiche Prüfungen vorgenommen, insbesondere an praktischen Modellen. Obwohl das Verfahren erst teilweise untersucht wurde, zeigt es gute Ergebnisse.

### Schlüsselwörter:

Iterationsverfahren, Jacobi-Verfahren, Gauß-Seidel-Verfahren, Methode des steilsten Abstiegs, Methode der konjugierten Gradienten

### 1. Kratki teorijski uvod

Većina numeričkih postupaka proračuna inženjerskih modela završava na (često vrlo velikom) sustavu algebarskih jednadžbi

$$\mathbf{K}\mathbf{u}=\mathbf{f} \tag{1}$$

koji treba učinkovito riješiti. Ako je riječ o metodi pomaka, sustav jednadžbi ima značenje uvjeta ravnoteže. Tada je  $\mathbf{K}$  matrica krutosti,  $\mathbf{u}$  vektor nepoznatih pomaka, a  $\mathbf{f}$  vektor zadanoga opterećenja. U slučaju simetrične i pozitivno definitne matrice sustava rješenje je ekvivalentno minimizaciji kvadratne forme koja predstavlja potencijalnu energiju statičkoga sistema:

$$\Pi(\mathbf{u})=\frac{1}{2}\mathbf{u}^T\mathbf{K}\mathbf{u}-\mathbf{u}^T\mathbf{f} \tag{2}$$

Prvi je član potencijalna energija deformacija, a drugi rad (potencijal) vanjskih sila. Ovaj zapis predstavlja diskretiziranu aproksimaciju Lagrangeova funkcionala energije kontinuiranoga (matematičkoga) modela linearno elastična tijela. Ploha konstantne razine energije  $\Pi(\mathbf{u}) = c$  predstavlja elipsoid (hiperelipsoid) koji možemo prikazati jednadžbom

$$(\mathbf{u}-\mathbf{u}_0)^T\mathbf{K}(\mathbf{u}-\mathbf{u}_0)=1 \tag{3}$$

Radi se o tijelu  $n$  - dimenzijskoga prostora, gdje je  $n$  broj nepoznatih stupnjeva slobode. Geometrijski gledano, komponente vektora  $\mathbf{u}$  su koordinate točaka prostora, a vektora  $\mathbf{u}_0$  središta elipsoida. Matricu  $\mathbf{K}$  (reda  $n$ ) možemo prikazati umnoškom  $\mathbf{Q}^T\mathbf{D}\mathbf{Q}$ . Ortogonalnu matricu  $\mathbf{Q}$  čine stupci koji predstavljaju osi elipsoida, odnosno vlastite vektore, a dijagonalna matrica  $\mathbf{D}$  sadrži vrijednost omjera  $(2c/\lambda_i)^2$ , gdje je  $c$  razina energije, a  $\lambda_i$  vlastite vrijednosti. Duljine poluosi elipsoida iznose  $2c/\lambda_i$ . U njegovu je središtu  $c = 0$ . Na slikama 1.a i 1.b skicirane su inačice u ravnini ( $n = 2$ ) i prostoru ( $n = 3$ ). Prema teoremu o minimumu potencijalne energije tijela u stabilnoj ravnoteži, točka prostora kojoj pripada najmanji iznos energije predstavlja rješenje problema, a leži u središtu svih elipsoida. Možemo ju poistovjetiti s degeneriranim elipsoidom kojemu su duljine svih glavnih osi jednake nuli. U numeričkoj se

realizaciji, ovisno o prihvaćenom kriteriju točnosti, ipak radi o vrlo malom elipsoidu kojemu pripada nešto viša razina energije od minimalne.

### 2. Iteracijska zamisao

Približno rješenje, zadržimo oznaku  $\mathbf{u}$ , možemo pronaći uzastopnom primjenom diskretiziranoga Ritzova postupka. Zamisao temeljimo na izboru linearno nezavisnih koordinatnih vektora  $\phi_i$  u smjeru kojih, uz pripadajuće skalare  $a_i$ , rastavljamo trenutni prirast rješenja [1]:

$$\Delta\mathbf{u} = \sum_{i=1}^m a_i\phi_i \tag{4}$$

U klasičnoj realizaciji postupka nepoznanice  $a_i$  i vektore  $\phi_i$  zovemo Ritzovim koeficijentima i Ritzovim vektorima. Broj vektora mora ispuniti nejednakost  $1 \leq m \leq n$ . Ako definiramo pravokutnu matricu s  $m$  stupaca i  $n$  redaka, tako da stupci sadrže koordinatne vektore, odnosno

$$\Phi = [\phi_1 \phi_2 \dots \phi_m] \tag{5}$$

a skalarne varijable smjestimo u vektor

$$\mathbf{a} = [a_1 a_2 \dots a_m]^T \tag{6}$$

prirast rješenja možemo zapisati u obliku

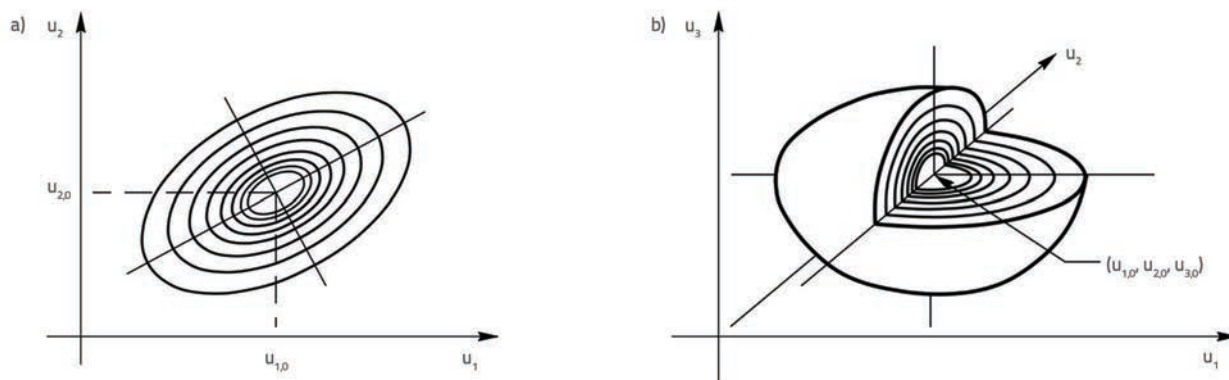
$$\Delta\mathbf{u} = \Phi \mathbf{a} \tag{7}$$

Sada u duhu iteracijskoga postupka vrijedi

$$\mathbf{u}_{i+1} = \mathbf{u}_i + \Delta\mathbf{u}_i \tag{8}$$

gdje indeksi označavaju dva uzastopna koraka iteracije. Ako je energija u  $i$ - tomu koraku

$$\Pi_i = \frac{1}{2} \mathbf{u}_i^T \mathbf{K} \mathbf{u}_i - \mathbf{u}_i^T \mathbf{f} \tag{9}$$



Slika 1. Elipsa i elipsoid energije

tada je zbog (7) i (8) u sljedećemu koraku

$$\Pi_{i+1} = \frac{1}{2} (\mathbf{u}_i^T + \mathbf{a}_i^T \Phi_i^T) \mathbf{K} (\mathbf{u}_i + \Phi_i \mathbf{a}_i) - (\mathbf{u}_i^T + \mathbf{a}_i^T \Phi_i^T) \mathbf{f} \quad (10)$$

Množenjem podizraza u zagradama i sređivanjem nastalih članova dobivamo

$$\Pi_{i+1} = \Pi_i + \frac{1}{2} \mathbf{a}_i^T \Phi_i^T \mathbf{K} \Phi_i \mathbf{a}_i - \mathbf{a}_i^T \Phi_i^T \mathbf{r}_i \quad (11)$$

gdje je neuravnoteženo opterećenje, odnosno rezidual

$$\mathbf{r}_i = \mathbf{f} - \mathbf{K} \mathbf{u}_i \quad (12)$$

Budući da energija  $\Pi_i$  ne ovisi o prirastu pomaka  $\Delta \mathbf{u}_i$ , a time niti o varijablama  $\mathbf{a}_i$ , možemo ju izostaviti iz postupka minimizacije, odnosno dovoljno je derivirati prirast energije

$$\Delta \Pi_i = \frac{1}{2} \mathbf{a}_i^T \Phi_i^T \mathbf{K} \Phi_i \mathbf{a}_i - \mathbf{a}_i^T \Phi_i^T \mathbf{r}_i \quad (13)$$

Ako uvedemo poopćenu Ritzovu matricu krutosti

$$\mathbf{K}_i = \Phi_i^T \mathbf{K} \Phi_i \quad (14)$$

koja je također simetrična, a u slučaju linearno nezavisnih koordinatnih vektora i pozitivno definitna, te poopćeni Ritzov vektor opterećenja

$$\bar{\mathbf{r}}_i = \Phi_i^T \mathbf{r}_i \quad (15)$$

prirast energije možemo zapisati u skraćenu obliku

$$\Delta \Pi_i = \frac{1}{2} \mathbf{a}_i^T \mathbf{K}_i \mathbf{a}_i - \mathbf{a}_i^T \bar{\mathbf{r}}_i \quad (16)$$

kojemu nakon minimizacije (derivacije) po  $\mathbf{a}_i$  očito pripada sustav jednažbi

$$\mathbf{K}_i \mathbf{a}_i = \bar{\mathbf{r}}_i \quad (17)$$

Njime, u smislu Ritzova postupka, aproksimiramo polazni sustav (1). Aproksimacija najčešće nije zadovoljavajuća, zbog čega i radimo iteracijsko poboljšanje. Rješenjem sustava (17), simboličkoga zapisa

$$\mathbf{a}_i = \mathbf{K}_i^{-1} \bar{\mathbf{r}}_i \quad (18)$$

dobivamo koeficijente  $\mathbf{a}_i$ , a zatim prema (7) približni prirast pomaka  $\Delta \mathbf{u}_i$  i u skladu s (8) novi pomak  $\mathbf{u}_{i+1}$ . Postupak prekidamo nakon ispunjenja kriterija konvergencije. Uobičajeno je upotrijebiti euklidsku normu reziduala, odnosno iteracijski tijek završiti ako je

$$\|\mathbf{r}_i\|_2 \leq \varepsilon \|\mathbf{r}_0\|_2 \quad (19)$$

gdje je  $\mathbf{r}_0$  početni rezidual ( $\mathbf{r}_0 = \mathbf{f}$  ako je  $\mathbf{u}_0 = \mathbf{0}$ ), a  $\varepsilon$  vrlo mali

pozitivan broj. Istaknimo, trenutačni rezidual možemo odrediti i iz rekurzivne formule

$$\mathbf{r}_i = \mathbf{r}_{i-1} - \mathbf{K} \Delta \mathbf{u}_{i-1} \quad (20)$$

dobivene primjenom izraza (12) u koji prema (8) treba uvrstiti  $\mathbf{u}_i = \mathbf{u}_{i-1} + \Delta \mathbf{u}_{i-1}$  i prepoznati da je  $\mathbf{r}_{i-1} = \mathbf{f} - \mathbf{K} \mathbf{u}_{i-1}$ . Izraz (20) ubrzava postupak, jer nasuprot izrazu (12), nije potrebno u svakom koraku računati ukupni pomak. Odredimo ga tek na kraju, nakon ispunjenja kriterija konvergencije. Međutim, ako primjenjujemo (20), zbog gomilanja pogrešaka zaokruživanja rezidual treba povremeno obnoviti iz uvjeta ravnoteže (12). Usput, taj uvjet trebamo i na samomu početku proračuna (za  $i=0$ ), radi proračuna početnoga reziduala ako  $\mathbf{u}_0$  nije nul-vektor. Istaknimo još da prilikom realizacije postupka treba ograničiti i maksimalan broj koraka čime, u slučaju problema s konvergencijom, sprječavamo dugotrajan proračun.

### 3. Prekoračenje (podkoračenje) pomaka

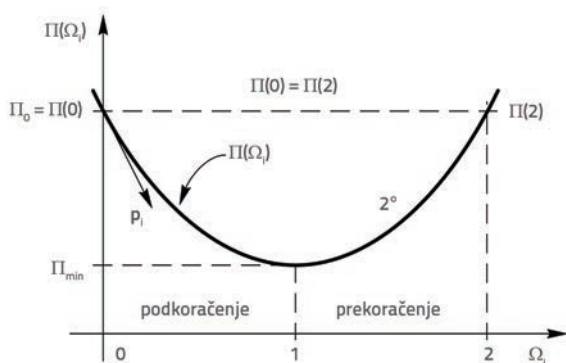
Motivirani poznatim postupkom uzastopnoga (stalnog) prekoračenja, nazvanim i postupkom uzastopne prekomjerne relaksacije (engl. *successive overrelaxation*), proračun možemo ubrzati primjenom faktora relaksacije  $\Omega_i$  (engl. *relaxation factor*), kojim množimo prirast rješenja. Tada umjesto primjenom (8), novi pomak tražimo kao

$$\mathbf{u}_{i+1} = \mathbf{u}_i + \Omega_i \Delta \mathbf{u}_i \quad (21)$$

Naime, lokalni minimum energije (unutar podprostora), najčešće nije globalno optimalan. Konvergenciju često možemo poboljšati prekoračenjem pomaka, odnosno izborom  $\Omega_i > 1$  (engl. *overrelaxation*) ili podkoračenjem pomaka, za  $\Omega_i < 1$  (engl. *underrelaxation*). U oba slučaja sustav (17) neće biti zadovoljen, pa energija neće postići minimum unutar podprostora. Međutim, premda takav korak nije lokalno optimalan, može ubrzati globalnu konvergenciju postupka. Na žalost, nije jednostavno od koraka do koraka odrediti dobru vrijednost faktora relaksacije. Samo znamo da vrijedi  $0 < \Omega_i < 2$ . Premda postoji matematički dokaz ovih granica [2], možemo mu dati jasnu interpretaciju. Budući da je  $\Pi$  kvadratna funkcija, potragu za minimumom duž nekoga smjera traženja  $\mathbf{p}_i$  (slika 2.), koji se najčešće podudara sa  $\Delta \mathbf{u}_i$ , možemo opisati funkcijom

$$\Pi(\Omega_i) = \Pi_{\min} + (\Pi_0 - \Pi_{\min})(\Omega_i - 1)^2$$

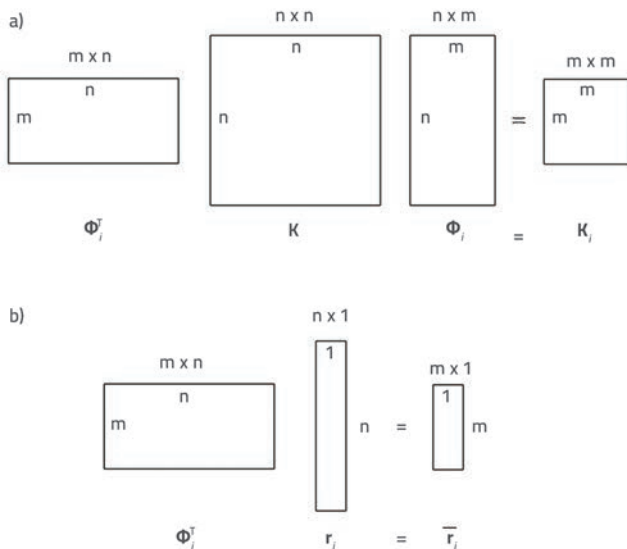
gdje je  $\Pi_0$  iznos energije u hvatištu smjera traženja (na početku koraka),  $\Pi_{\min}$  lokalni minimum duž toga smjera, a  $\Pi(\Omega_i)$  iznos u nekoj točki kvadratne funkcije. Iz uvjeta monotone konvergencije postupka  $\Pi(\Omega_i) < \Pi_0$  možemo pisati  $(\Omega_i - 1)^2 < 1$  ili  $0 < \Omega_i < 2$ . Na granicama intervala energija ostaje jednaka početnoj, odnosno  $\Pi(0) = \Pi(2)$ , pa postupak ne konvergira. Za vrijednosti  $\Omega_i$  izvan tih granica energija raste i postupak divergira.



Slika 2. Ovisnost energije o faktoru relaksacije

### 4. Iteriranje primjenom maloga sustava

Postupak počinjemo izborom početne aproksimacije  $u_0$ . Zatim u svakom koraku definiramo niz koordinatnih vektora matrice  $\Phi_i$ , a potom određujemo vektor  $\Delta u_i$ . Taj prirast (kojemu u podprostoru koji razapinju koordinatni vektori pripada vektor  $a_i$ ), daje najveće smanjenje energije  $\Delta \Pi_i$  unutar podprostora. Zbog toga treba riješiti sustav linearnih algebarskih jednadžbi (17), ali s malim brojem nepoznanica, jednakim broju koordinatnih vektora. Na taj način rješavanje izvornoga sustava koji sadrži  $n$  jednadžbi svodimo na višestruko rješavanje sustava od  $m$  jednadžbi. U nekim našim primjerima  $n$  je iznosio oko  $10^7$ , a  $m$  najviše deset. Poznate sheme matičnih transformacija koje uzrokuju nastanak maloga sustava skicirane su na slici 3.



Slika 3. Tvorba maloga sustava: a) matrice  $K_i$ , b) vektora  $\bar{r}_i$

Rješenje takvoga sustava (matrica  $K_i$  je najčešće puna), možemo potražiti bilo kojim izravnim postupkom. Mi smo primijenili rastav Choleskoga. Ako je postupak proračuna konvergentan, suma rješenja malih sustava teži prema rješenju velikoga sustava, a suma prirasta energije malih sustava monotono pada i teži prema minimumu energije velikoga sustava. Konvergencija je

ugrožena u slučaju pojave linearno zavisnih koordinatnih vektora (kada podprostor degenerira), zatim reziduala ortogonalnoga na trenutni podprostor i učinka pogreške zaokruživanja pri kraju iteracijskoga postupka (ako je kriterij prekida proračuna prestrog). Ponešto o ovim problemima u poglavlju 7.

### 5. Osnovni pseudokod

Lako je uočiti vrlo prirodnu i ne odveć tešku zamisao ovoga algoritma. Unatoč tomu, ovaj pristup i interpretacija nisu naišli na širu primjenu među istraživačima koji se bave učinkovitim iteracijskim postupcima rješavanja velikih sustava [3]. Teorijski gledano i njega, poput mnogih drugih iteracijskih algoritama, možemo smatrati Krilovljevim postupkom [4, 5], a neke sličnosti sa zapažanjima u ovom radu postoje u [6]. Zanimljivo je, iteracija po podprostorima (engl. *subspace iteration*), što je u biti i ovaj postupak, ima dosta široku primjenu kod rješavanja problema vlastitih vrijednosti. Glavne sastavnice opisanoga postupka sažeto su prikazane jednostavnim pseudokodom, tablica 1.

Tablica 1. Iterirani Ritzov postupak

<b>Potrebno:</b> $K, f, \varepsilon$ matrica krutosti, vektor opterećenja, kriterij prekida
<b>1. Rezultat:</b> $u$ vektor pomaka
2. $i \leftarrow 0$ brojač koraka
3. $u_i \leftarrow 0$ početno rješenje nul - vektor
4. $r_i \leftarrow f$ rezidual jednak opterećenju
<b>5. repeat</b>
6. $\Phi_i \leftarrow [\phi_{i,1} \phi_{i,2} \dots \phi_{i,m}]$ definiranje koordinatnih vektora
7. $K_i \leftarrow \Phi_i^T K \Phi_i$ tvorba "male" matrice sustava
8. $\bar{r}_i \leftarrow \Phi_i^T r_i$ tvorba "maloga" vektora desne strane
9. $a_i \leftarrow K_i^{-1} \bar{r}_i$ rješavanje "maloga" sustava
10. $\Delta u_i \leftarrow \Phi_i a_i$ određivanje prirasta rješenja
11. $u_{i+1} \leftarrow u_i + \Delta u_i$ proračun novoga pomaka
12. $r_{i+1} \leftarrow f - K u_{i+1}$ novi rezidual
13. $i \leftarrow i + 1$ povećanje brojača koraka
14. <b>until</b> $\ r_i\ _2 \leq \varepsilon \ r_0\ _2$

Učinkovitost postupka stvar je određenoga kompromisa. Upotrijebimo li veći broj dobro odabranih koordinatnih vektora (koji razapinju podprostor u kojemu leži kvalitetna aproksimacija prirasta rješenja), smanjenje energije po koraku bit će veće, trebat ćemo manje koraka za dobivanje rješenja, ali će trajanje koraka biti dulje. Manji broj vektora znači veću brzinu koraka, ali manje učinkoviti podprostor i više koraka za postizanje rješenja. Optimalno bi bilo pronaći kvalitetan podprostor male dimenzije, tako da i stroži kriterij konvergencije dostignemo nakon maloga broja brzih koraka.

## 6. Posebni slučajevi postupka

Pravokutnu matricu  $\Phi_i$  zovemo matricom podprostora. Istaknuli smo, njezini stupci su koordinatni vektori  $\phi_{1,i}$  do  $\phi_{m,i}$  koji ga razapinju. Ovisno o njihovu izboru razlikujemo brojne iteracijske postupke. Nećemo ih posebno objašnjavati [7-11], nego ćemo poznatije prikazati kao posebne slučajeve ovoga pristupa. Treba samo upotrijebiti prikladne koordinatne vektore. Kratki opisi koji slijede ne služe bržoj realizaciji tih postupaka u odnosu na uvriježene načine [12], nego samo doprinose pojašnjenju i ističu općenitost ovoga iteracijskog algoritma. Spomenimo još, danas popularno jednostruko ili višestruko preduvjetovanje (engl. *preconditioning*) sustava [13, 14] možemo interpretirati primjenom (jednoga ili više) koordinatnih vektora. Za ovaj postupak preduvjetovanje ne znači nikakvu bitnu promjenu algoritamskoga slijeda (vidjeti podpoglavlje 7.2.3).

### 6.1. Jacobijev postupak

Dobivamo ga ako u svakom koraku definiramo samo jedan koordinatni vektor oblika  $\phi_{m,i} = \mathbf{D}^{-1}\mathbf{r}_i$ , gdje je  $\mathbf{D}$  dijagonalna matrica koja sadrži dijagonalne članove od  $\mathbf{K}$ . Time matrica  $\Phi_i$  postaje matrica stupac, odnosno  $\Phi_i = [\phi_i]$ , a  $\mathbf{K}_i$  i  $\bar{\mathbf{r}}_i$  degeneriraju u skalare. Znači, mali sustav (17) reducira se na jednu jednadžbu kojom u svakom koraku određujemo prirast pomaka.

### 6.2. Gauß - Seidelov postupak i postupak uzastopnog prekoračenja

Postupak možemo opisati nizom ciklusa, odnosno "prolaza" kroz sve stupnjeve slobode. Tada se ciklus sastoji od  $n$  koraka. U svakom koraku (redosljedom pobrojavanja čvorova) rješavamo samo jedan stupanj slobode. Proces često zovemo relaksacijom, a pripada mu matrica  $\Phi_i$  koja u svakom koraku ima jedan koordinatni vektor jednak ortu, odnosno  $\phi_i = \mathbf{e}_i$ . Komponenta koja odgovara stupnju slobode koji trenutačno relaksiramo ima vrijednost jedan, a sve ostale su jednake nuli. Matrica  $\mathbf{K}_i$  i vektor  $\mathbf{r}_i$  opet degeneriraju u skalar, što se ponovo svodi na rješavanje jedne jednadžbe. Time smo zadovoljili ravnotežu samo pripadajućeg stupnja slobode, jer istodobno poremetimo rezidual prethodne jednadžbe. Tijekom konvergentnoga iteracijskog postupka iznos poremećaja postaje sve manji. Ciklus završava kada riješimo svih  $n$  jednadžbi (koraka), a u ravnoteži je samo posljednja. Zatim počinje novi ciklus, s istim koordinatnim vektorima. Kako napreduju koraci, jedinica se opet premiješta od prve prema zadnjoj komponenti vektora.

Ako razmišljamo matricno (po ciklusima koji odgovaraju koracima u drugim postupcima) i upotrijebimo rješavanje redom pobrojavanja čvorova, svakomu ciklusu pripada jedan koordinatni vektor oblika  $\phi_i = \mathbf{L}^{-1}\mathbf{r}_i$ , gdje je  $\mathbf{L}$  donja trokutna matrica (sadrži donji trokutni dio matrice  $\mathbf{K}$  i dijagonalne članove). U ovoj interpretaciji postupak uzastopnog prekoračenja sadrži istu matricu, samo su dijagonalni članovi pomnoženi lokalnim faktorom relaksacije  $\omega$ , koji treba razlikovati od globalnoga  $\Omega$ .

Tada ne vrijedi niti ravnoteža onoga stupnja slobode čiji smo jednadžbu upravo riješili. (Kod Gauß - Seidelova je postupka  $\Omega = \omega = 1$ ). Ako rješavamo suprotno od redosljeda pobrojavanja čvorova, donju trokutnu matricu zamijenimo gornjom, odnosno  $\phi_i = \mathbf{U}^{-1}\mathbf{r}_i$ .

Postoje i drugi načini obilaženja čvorova, primjerice poznata "šahovska shema" na pravokutnoj mreži. U prvoj polovini ciklusa rješavamo "crne", a u drugoj "bijele" čvorove. Obilaziti možemo i pobrojavanjem s desna na lijevo (umjesto klasičnoga s lijeva na desno). Uvedemo li i dijagonalnu povezanost među čvorovima, raste broj mogućih obilaženja. Ako još stupci i redci zamijene mjesta, dobivamo nove načine pristupanja. U mreži koja se ortogonalno širi u svim koordinatnim smjerovima postoji još više mogućnosti, posebice ako sadrži dijagonalne veze u koordinatnim ravninama i duž prostornih dijagonala. Nepravilne mreže iz metode konačnih elemenata još su složenije i imaju golemi broj smislenih veza među čvorovima, a time i putova širenja informacija o opterećenju (rezidualu).

Zanimljiva je zamisao ići od (po apsolutnoj vrijednosti) najvećega prema najmanjemu rezidualu tekućega ciklusa. Međutim, taj redosljed treba korigirati nakon rješenja svake jednadžbe (provedbe jednoga koraka), zbog promjene reziduala u trenutačnu čvoru i njegovim topološkim susjedima. Naravno, stalna korekcija (sortiranje) reziduala zahtijeva dodatni utrošak vremena, što utječe na brzinu proračuna.

U načelu, ovim načinima redefinirani algoritam treba manji broj koraka za dostizanje rješenja. Uostalom, poznati postupci Crossa, te Wernera i Csonke, bolje konvergiraju ako proračun počnemo od čvora u kojemu smo ustanovili najveći rezidual. Tada obnovimo vektor reziduala i opet uravnotežimo čvor s najvećim rezidualom. Postupak rješavanja više ne ovisi o načinu pobrojavanja čvorova, nego o razdiobi reziduala (opterećenja). Zbog toga pojam ciklusa gubi pravo značenje, jer prije nego što neke čvorove uravnotežimo prvi put, druge rješavamo više puta.

### 6.3. Metoda najstrmijega silaska

I ovdje u svakom koraku upotrebljavamo samo jedan koordinatni vektor, jednak vektoru reziduala. Dakle,  $\phi_i = \mathbf{r}_i$ . Ujedno je riječ i o gradijentu funkcionala energije s negativnim predznakom, koji uobičajeno upotrebljavamo prilikom klasične realizacije postupka, odnosno  $\mathbf{r}_i = -\nabla\Pi(\mathbf{u})$ . Ako je matrica  $\mathbf{K}$  simetrična i pozitivno definitna, metoda je konvergentna, ali je konvergencija obično spora, posebice u slučaju loše uvjetovane matrice. Premda ova metoda nije učinkovita, služi kao pedagoški uvod i motivacija za poboljšanja drugih iteracijskih postupaka.

### 6.4. Metoda konjugiranih gradijenata

U prvom koraku imamo samo jedan koordinatni vektor, rezidual, kao u metodi najstrmijega silaska. Nakon toga mu se pridružuje i vektor prethodnoga prirasta rješenja koji pospješuje konvergenciju. Zbog toga je ovaj postupak puno brži od prethodnoga. Možemo pisati  $\phi_{1,i} = \mathbf{r}_i$  i  $\phi_{2,i} = \Delta\mathbf{u}_{i-1}$ . Znači mala

matrica je reda dva, odnosno  $\Phi_i = [\phi_{1,i}, \phi_{2,i}]$ , pa u svakom koraku rješavamo sustav s dvije nepoznanice. Ovaj opis se razlikuje od uobičajenih u kojima se upotrebljava  $\mathbf{K}$  - ortogonalizacija vektora. Međutim, ta ortogonalizacija je istoznačna rješavanju spomenutoga sustava s dvije nepoznanice.

Zbog gomilanja pogrešaka zaokruživanja postupno se gubi svojstvo ortogonalnosti ovih dvaju vektora. Unatoč tomu, i približna ortogonalnost iz bliskih koraka jako ubrzava konvergenciju u odnosu na metodu najstrijemijega silaska. Numerički pokusi na relativno malim sustavima potvrđuju teoriju. Rezidualna norma najkasnije u  $n$  - tomu koraku skoči na nulu (preciznije, numeričku aproksimaciju nule). S druge strane, pokusi na velikim sustavima pokazuju glatko ponašanje u  $n$  - tomu koraku, kada bismo teorijski očekivali spomenuti skok. Zbog toga postupak mora imati kriterij prekida. Odnosno, ponaša se kao uobičajeni iteracijski postupak.

Premda je metoda relativno brza, a u slučaju simetrične i pozitivno definitne matrice sustava ima i praktičnu primjenu, nismo baš uvijek zadovoljni s konvergencijom, posebice ako je matrica loše uvjetovana. Zbog toga se često upotrebljavaju brojni načini preduvjetovanja, pa se govori o preduvjetovanoj metodi konjugiranih gradijenata. Tada postupak može završiti (ispuniti kriterij prekida), nakon puno manjega broja koraka od teorijski potrebnih  $n$ . Zaključimo ovaj odjeljak pomalo optimističnim citatom [4]: "Izbor [iteracijskoga] postupka je osjetljiv problem. Ako je matrica [sustava]  $A$  simetrična i pozitivno definitna, izbor je lagan: konjugirani gradijenti." To bi u skladu s našim postupkom značilo da ne postoji bolji trenutačni podprostor koordinatnih vektora od ravnine? Čini se da bi moralo biti prostora za napredak.

## 7. O izboru učinkovitoga podprostora koordinatnih vektora

Mi bismo htjeli odrediti matricu  $\Phi_i$  tako da postupak mnogo brže konvergira od klasičnih. Premda želimo da broj koordinatnih vektora  $m$  ostane malen, čini nam se da je samo jedan vektor (kao kod Jacobija, Gauß - Seidela, uzastopnog prekoračenja i najstrijemijega silaska), ili dva (kao kod konjugiranih gradijenata), manje od optimuma. Jasno, budući da je broj vektora puno manji od  $n$ , približni prirast  $\Delta u_i$  može u ranoj fazi postupka samo s vrlo malom vjerojatnošću dostići rješenje  $\mathbf{u}$ .

Možemo zamisliti da dvama koordinatnim vektorima metode konjugiranih gradijenata dodamo treći. Time podprostor proširujemo na tri vektora, pa je u usporedbi s podprostorom koji sadrži samo dva vektora razumno očekivati veće smanjenje energije po koraku. U najgoremu slučaju, doprinos dodatnoga vektora može biti jednak nuli. Možemo zaključiti i ovako: minimum funkcije energije u širem podprostoru ne može biti veći (više razine) od minimuma u manjem podprostoru. Razmišljajući na ovaj način možemo dodati četvrti, peti i sljedeće vektore, te očekivati još veće smanjenje energije po koraku. Na taj bi način trebalo statičkomu sustavu u par koraka oduzeti većinu energije i "prigušiti ga" prema najnižoj točki - rješenju.

Očito je zamisao o proširenju podprostora privlačna, ali samo do neke mjere. S jedne strane, generiranje vektora ne smije biti vremenski zahtjevno, a s druge strane, pretjerivanje s njihovim brojem povećava mali sustav koji treba riješiti u svakom koraku. U krajnjemu slučaju, ako bi dimenzija podprostora bila jednaka broju nepoznanica, minimizirali bismo ukupnu energiju i odredili rješenje u prvom koraku (zbog pogrešaka zaokruživanja možda u drugom), ali po "cijeni" jednako ili većoj od one za izravno rješavanje sustava.

### 7.1. Nužni uvjeti izbora

Mala matrica  $\mathbf{K}_i$  može biti singularna (ili skoro singularna - loše uvjetovana), ako su neki koordinatni vektori točno (ili približno) linearno zavisni. Podsjetimo se, vektori  $\phi_i$  su linearno nezavisni ako izraz

$$\sum_{i=1}^m a_i \phi_i = \mathbf{0} \quad (22)$$

vrijedi samo u slučaju svih  $a_i = 0$ . U numeričkoj realizaciji ovaj uvjet treba postrožiti: vektori ne smiju biti ni "skoro" linearno zavisni. Tada euklidska norma linearne kombinacije vektora smije biti manja od maloga pozitivnog broja  $\delta$ , odnosno

$$\left\| \sum_{i=1}^m a_i \phi_i \right\|_2 < \delta \quad (23)$$

samo ako su svi koeficijenti  $|a_i| < \delta$ . Generiranjem koordinatnih vektora različitim postupcima (o čemu više u nastavku), kršenje uvjeta (22), a posebice (23) nije automatski spriječeno. Vektore koji ne zadovoljavaju te uvjete treba odbaciti, čime doduše smanjujemo dimenziju podprostora, ali mala matrica postaje regularna i bolje uvjetovana. Ako su više od dva vektora međusobno linearno zavisni, nije jednoznačno definirano koji treba odbaciti. Tada bismo mogli razmišljati o točnijoj ili približnoj ortogonalizaciji takvih vektora. U svakomu slučaju, situacija u kojoj se vektori generiraju, a potom odbacuju, ortogonaliziraju ili možda zamjenjuju drugima poskupljuje korak, pa treba nastojati da se događa što rjeđe. Treba razmišljati o postupku tvorbe kojim nije (ili je rijetko) moguće generirati (gotovo) zavisne vektore. Radi lakše realizacije ove zamisli i tijekom proračuna možemo mijenjati način generiranja vektora. Ako se zavisnost ipak dogodi, možemo se "spasiti u zadnji čas", jer u postupku dekompozicije neke uporišne komponente (engl. *pivots*) matrice  $\mathbf{K}$ , postaju jednake (bliske) nuli. To možemo prepoznati i upotrijebiti za izbacivanje pripadajućih jednadžbi iz postupka proračuna maloga sustava. Ovo se smanjenje podprostora pokazalo brzim i jednostavnim rješenjem problema linearne zavisnosti.

Međutim, tvorba podprostora je jedno, a njegova kvaliteta je nešto posve drugo. Primjerice, ako su koordinatni vektori okomiti na vektor reziduala, podprostor nije upotrebljiv. Tada je desna strana maloga sustava (Ritzov vektor opterećenja)

$\bar{r}_i = \Phi_i^T r_i = \mathbf{0}$ . Posljedica je  $\mathbf{a}_i = \mathbf{0}$  i  $\Delta \Pi_i = 0$ , pa nema smanjenja energije. Znači, postupak ne konvergira. Ako želimo izbjeći ovaj problem potrebno je da norma  $\|\Phi_i^T r_i\|_2 / \|r_i\|_2$  bude veća od neke male konstante. Za to je dovoljan jedan koordinatni vektor  $\phi_i$  koji nije ortogonalan na rezidual  $r_i$ . Primjerice, možemo ga odrediti množenjem odabrane pozitivno definitne matrice  $\mathbf{P}$  (dimenzije  $n$ ), vektorom reziduala:  $\phi_i = \mathbf{P} r_i$ . Tada je prema definiciji pozitivne definitnosti  $\phi_i^T r_i = r_i^T \mathbf{P} r_i > 0$ , osim ako je  $r_i$  nul - vektor, ali to znači da je postignuto rješenje. Radi pojašnjenja zanimljivo je istaknuti još nešto. Ako bismo prilikom proračuna novoga reziduala  $r_{i+1}$ , zadržali stare koordinatne vektore  $\Phi_i$  (iz prethodnoga koraka), dobili bismo upravo istaknuti nepovoljni slučaj, jer je novi vektor reziduala uvijek ortogonalan na stari podprostor (Tvrdnja ne vrijedi u slučaju prekoračenja ili podkoračenja pomaka, nego samo za  $\Omega = 1$ ), odnosno vrijedi  $\Phi_i^T r_{i+1} = \mathbf{0}$ . Uz oznake i odnose uvedene prilikom opisa postupka, jednostavan dokaz ide redom:

$$\begin{aligned} \Phi_i^T r_{i+1} &= \Phi_i^T (\mathbf{f} - \mathbf{K} u_{i+1}) \\ &= \Phi_i^T \mathbf{f} - \Phi_i^T \mathbf{K} (u_i + \Delta u_i) \\ &= \Phi_i^T \mathbf{f} - \Phi_i^T \mathbf{K} u_i - \Phi_i^T \mathbf{K} \Delta u_i \\ &= \Phi_i^T (\mathbf{f} - \mathbf{K} u_i) - \Phi_i^T \mathbf{K} \Phi_i \mathbf{a}_i \\ &= \Phi_i^T r_i - \mathbf{K}_i \mathbf{a}_i = \Phi_i^T r_i - \bar{r}_i \\ &= \Phi_i^T r_i - \Phi_i^T r_i = \mathbf{0} \end{aligned} \tag{24}$$

U usporedbi s Ritzovim koordinatnim funkcijama na kontinuumu, u ovoj diskretnoj inačici ne tražimo ispunjenje uvjeta neprekinutosti i geometrijskih rubnih uvjeta. Ako poznajemo samo sustav jednadžbi, a ne i statički sistem od kojega je nastao, takva svojstva nisu niti definirana. Jedino znamo da su sadržana u matrici sustava. Ipak, intuitivno možemo smatrati neki koordinatni vektor "glađim" ako sadrži manji relativni udio "visokih tonova" - vlastitih vektora matrice  $\mathbf{K}$  kojima pripadaju velike vlastite vrijednosti. Takav vektor tvori realističnije koeficijente Ritzove matrice, jer nastaje rezidual koji također ima mali udio visokih tonova. Dijagonalni element male matrice koji pripada takvom koordinatnom vektoru manji je, odnosno manja je odgovarajuća "poopćena krutost". Nasuprot njemu, pregubi vektor tvori preveliku krutost Ritzove matrice i uzrokuju zaostajanje konvergencije (engl. *locking*) reziduala. To je lako dokazati ako koordinatni vektor  $\phi_i$  rastavimo u bazi vlastitih vektora matrice:

$$\phi_i = \sum_{j=1}^n h_j \mathbf{v}_j \tag{25}$$

U slučaju normiranih vektora ( $\mathbf{v}_j^T \mathbf{v}_j = 1$ ), pripadajući dijagonalni element male matrice (označimo je nakratko kao i matricu sustava), možemo zapisati kao

$$k_{i,j} = \phi_i^T \mathbf{K} \phi_i = \sum_{j=1}^n h_j^2 \lambda_j \tag{26}$$

Uočavamo, koeficijenti  $h_j^2$  (kojima pripada veći  $\lambda_j$ ), množe se većom vlastitom vrijednošću  $\lambda_j$  i time više doprinose (povećanju) krutosti  $k_{i,j}$ . Usput, pregrubi vektori imaju i veliku kvadratnu normu reziduala  $\|r_i\|_2^2$ . S napredovanjem koraka, glatki koordinatni vektori učinkovito smanjuju doprinos niskih vlastitih vektora. Nakon toga se postupak ponaša kao da utjecaja tih vektora nema. Time broj uvjetovanosti matrice sustava postaje manji, pa raste brzina konvergencije. Ako bismo dodali još sličnih vektora, smanjenje energije bi trebalo biti veće. S teorijskoga gledišta svojstvo "glatkoće" nije nužno, ali smo ga svrstali u ovaj odjeljak, jer je bitno za učinkovitu realizaciju postupka. Bez njega postupak nije konkurentan.

Premda istaknuta ograničenja sužavaju izbor, skup iz kojega možemo odabrati dobre koordinatne vektore i dalje ostaje iznimno velik. Na žalost, nisu nam unaprijed poznati (i koliko smo upućeni nitko ne zna) kriteriji za izbor općeučinkovitih vektora. Niti je dovoljno poznata pozadinska teorija koja bi olakšala biranje. Nevolja je što je mogućnosti mnogo i možemo se samo zadovoljiti provedbom i usporedbom numeričkih pokusa na brojnim primjerima. U pravilu, određeni skup vektora za neke modele radi dobro, a za druge ne. U ovim uvjetima bili bismo zadovoljni izborom dobrih vektora, a najbolji skup (nekoliko brzih i o modelima neovisnih vektora) bio bi ravan velikomu otkriću.

## 7.2. Nekoliko prijedloga tvorbe

Postoje dva temeljna pristupa tvorbi koordinatnih vektora: opći i posebni. U prvomu nisu potrebni nikakvi dodatni podaci o modelu. Matrica sustava i vektor desne strane dovoljni su za generiranje vektora. Ako je izbor vektora dobar u većini smo praktičnih slučajeva zadovoljni s konvergencijom postupka. U drugomu pristupu upotrebljavamo neke posebnosti koja vrijede samo za model koji rješavamo. Tada očekujemo odličnu konvergenciju, ali samo za taj model (i možda za mali skup njemu bliskih modela). Pokažimo u nastavku nekoliko načina tvorbe koordinatnih vektora, najprije prema prvomu, a potom i drugomu pristupu.

### 7.2.1. Izbor konstantnih vektora

Najjednostavnije je unaprijed odabrati skup linearno nezavisnih vektora. Takav slučaj imamo kod Gauß - Seidelova postupka ili postupka uzastopnoga prekoračenja. Broj vektora podudara se s brojem nepoznanica, a njihov redoslijed ciklički ponavljamo do ispunjenja kriterija konvergencije, kao što je navedeno u poglavlju 6.

### 7.2.2. Izbor temeljen na trenutačnu rezidualu

Dobar skup vektora možemo odrediti primjenom trenutačnog reziduala  $r_i$ . Naime, budući da nas "skupi" proračun početne pogreške (posudili smo oznaku za ort)  $\mathbf{e}_0 = \mathbf{K}^{-1} r_0$ , odmah vodi točnomu rješenju, možemo upotrijebiti matricu  $\mathbf{P}$ , koja uz što manje računanja dobro aproksimira  $\mathbf{K}^{-1}$ . Dobar je izbor ako je pozitivno definitna, ali to nije nužno. Smije biti nesimetrična i



loše uvjetovana, pa čak i singularna, jer to ne znači singularitet matrice  $\mathbf{K}$ . Naravno, ne smijemo generirati sve koordinatne vektore matricama istoga singulariteta.

Kako smo već istaknuli, koordinatni vektor generiramo kao  $\mathbf{P} \mathbf{r}_r$ . Pri tomu postoji i dodatna prednost u primjeni vektora trenutačnoga reziduala, jer osigurava neortogonalnost koordinatnog podprostora na taj vektor, što je važno za konvergenciju postupka. Primjerice, ako  $\mathbf{K}$  aproksimiramo jediničnom matricom ( $\mathbf{P} = \mathbf{I}$ ) dobivamo  $\phi_i = \mathbf{r}_r$ , što je zapravo metoda najstrijemijega silaska. Jacobijevu iteraciju temeljimo na malo boljoj aproksimaciji dijagonalnom matricom ( $\mathbf{P} = \mathbf{D}^{-1}$ ). Već smo isticali, tada je  $\phi_i = \mathbf{D}^{-1} \mathbf{r}_r$ . Ovi postupci nisu učinkoviti, jer (doduše "jeftina") zamjenjujuća matrica  $\mathbf{P}$  sadrži premalo podataka o inverznoj  $\mathbf{K}$ . Treba istaknuti: ako upotrebljavamo više koordinatnih vektora, nije nužno da pojedini vektor dobro aproksimira  $\mathbf{K}^{-1} \mathbf{r}_r$ , nego da podprostor koji razapinju svi vektori sadrži što bolju aproksimaciju toga umnoška.

Bolje koordinatne vektore možemo odrediti primjenom jednoga (ili nekoliko) ciklusa Gauß - Seidelova postupka ili postupka uzastopnoga prekoračenja. Pri tomu vrijedi probati različite načine "obilaženja" čvorova. Ali za razliku od klasične realizacije, radi kraćega trajanja trebalo bi raditi nepotpuno: zabraniti povratak na već relaksirane čvorove. Zamisao je dobra, ali poskupljuje ciklus. Možda jedino u prvomu ciklusu sortirati čvorove (prema apsolutnim iznosima komponenata reziduala), a kasnije samo čuvati ili rijetko popravljati poredak. Među ciklusima (možda i koracima) je poželjno uvesti lokalni faktor relaksacije  $\omega$ , ali bi optimalan iznos trebalo istražiti. Označimo s  $\mathbf{L}_\omega$  i  $\mathbf{U}_\omega$  donju i gornju trokutnu matricu od  $\mathbf{K}$ , čije smo dijagonalne elemente pomnožili s  $\omega$ . Ove matrice jednostavno i brzo invertiramo i množimo vektorom. Koordinatni vektor određen jednim ciklusom postupka uzastopnoga prekoračenja, redosljedom pobrojavanja nepoznanica (odozgo), možemo prikazati u obliku  $\phi_i = \mathbf{L}_\omega^{-1} \mathbf{r}_i$ . Primjenom istoga postupka suprotnim redosljedom (odozdo), dobivamo  $\phi_i = \mathbf{U}_\omega^{-1} \mathbf{r}_i$ . U oba slučaja ciklus počinjemo nul-vektorom. Koordinatni vektor možemo generirati višestrukom primjenom zbroja ili umnoška ovih dvaju pristupa. Dakle:

$$\phi_i = (\mathbf{L}_\omega^{-1} + \mathbf{U}_\omega^{-1})(\mathbf{L}_\omega^{-1} + \mathbf{U}_\omega^{-1}) \dots \mathbf{r}_i \quad (27)$$

ili:

$$\phi_i = \mathbf{L}_\omega^{-1} \mathbf{U}_\omega^{-1} \mathbf{L}_\omega^{-1} \mathbf{U}_\omega^{-1} \dots \mathbf{r}_i \quad (28)$$

Prvi način može između zagrada imati i matricu  $\mathbf{K}$ . Lako je u ovim jednadžbama prepoznati  $\mathbf{P}$ . Ako se ovakvi vektori upotrebljavaju samostalno, manji faktor relaksacije (blizak jedinici) čini konvergenciju sporijom, ali jamči brže "izgladivanje". Zbog toga ih je učinkovito upotrijebiti kao nadopunu drugim koordinatnim vektorima. S porastom broja ciklusa (uzastopnih izmjena matrica  $\mathbf{L}_\omega^{-1}$  i  $\mathbf{U}_\omega^{-1}$ ), glatkoća vektora raste (opada mu doprinos visokih vlastitih vektora i raste doprinos konvergenciji u području niskih vlastitih vrijednosti). Ovim je načinima moguće generirati koordinatne vektore i uporabom drugih iteracijskih postupaka.

Čak se može raditi o algoritmima koji nisu ni konvergentni niti numerički stabilni. Zato lokalni faktor relaksacije ne mora ležati unutar teorijski utvrđenih granica globalnoga faktora (koji mora biti između 0 i 2).

Zanimljive koordinatne vektore možemo odrediti postupkom izgladivanja vektora reziduala. Tada govorimo o "filtriranju". Komponenta takvoga vektora jednaka je sumi komponenata vektora reziduala u susjednim čvorovima, pomnoženih težinskim faktorima. Ovaj pristup je bio učinkovit u nekim primjerima s velikim skokovima funkcije reziduala. Oni nastaju zbog loše prognoze pomaka u pojedinim koracima, što se često događa u području ležajeva i slobodnih rubova modela.

Uglavnom, ako rezidual rastavimo na vlastite vektore matrice  $\mathbf{K}$ , trebalo bi formirati dva ili tri koordinatna vektora koji izgladuju donji dio spektra reziduala (niskih vlastitih vrijednosti), a zatim dodati još dva specijalizirana za gornji dio spektra. Njima bi trebalo dodati i prethodni prirast pomaka  $\Delta \mathbf{u}_{i-1}$ , što je bit uspjehnosti metode konjugiranih gradijenata.

U ovomu radu koordinatni su vektori generirani primjenom simetričnoga postupka prekomjerne relaksacije (engl. *Symmetric successive overrelaxation*). Dakle, prvi je vektor određen kao

$$\phi_1 = \mathbf{L}_\omega^{-1} \mathbf{D} \mathbf{U}_\omega^{-1} \mathbf{r}_i \quad (29)$$

a za ostale smo upotrijebili rekurzivnu formulu

$$\phi_j = \mathbf{L}_\omega^{-1} \mathbf{D} \mathbf{U}_\omega^{-1} (\mathbf{K} \phi_{j-1}), j = 2, \dots, m \quad (30)$$

Njima je pridružen i prethodni prirast pomaka. Proveli smo proračune s različitim brojem koordinatnih vektora (odjeljak 9). Odabrali smo nešto veći lokalni koeficijent relaksacije  $\omega = 1,65$ . Globalni je iznosio  $\Omega = 1$ . Pojasnimo malo umnožak u zagradi. Budući da jedan korak u smjeru vektora  $\phi_1$  daje trenutačni prirast pomaka  $\alpha \phi_1$ , gdje je  $\alpha$  neki broj, prema (20) dobivamo rezidual  $\mathbf{r}_i - \alpha \mathbf{K} \phi_1$ . Ako na njega primijenimo simetričnu prekomjernu relaksaciju, zbog (29) dobivamo drugi vektor kao  $\phi_2 = \phi_1 - \alpha \mathbf{L}_\omega^{-1} \mathbf{D} \mathbf{U}_\omega^{-1} (\mathbf{K} \phi_1)$ . Vektor  $\phi_1$  već sudjeluje u tvorbi podprostora, a  $\alpha$  utječe samo na duljinu novoga vektora (ne mijenja podprostor koji taj vektor proširuje), pa  $\phi_1$  i  $\alpha$  možemo izostaviti. Time dobivamo oblik (30). Zanimljiva, brža realizacija ovoga pristupa može biti da umjesto  $\mathbf{K}$  upotrijebimo  $\mathbf{D}$ . I ova razmišljanja vrijede općenito. Odnosno, ovako možemo generirati koordinatne vektore primjenom bilo kojega iteracijskog postupka, ne samo simetrične relaksacije. Čak i za svaki vektor možemo upotrijebiti drugačiji postupak. Primjerice, za tvorbu prvoga vektora upotrijebimo nepotpuni rastav Choleskoga (engl. *incomplete Cholesky factorization*), za drugoga i trećega prekomjernu relaksaciju odozgo i odozdo, za četvrtoga i petoga simetričnu inačicu tih postupaka i tako dalje. Slično opisanom učinku jednoga vektora, ovdje svi vektori dobiveni relaksacijom izgladuju podprostor. S porastom njihova broja (ciklusa kod tvorbe jednoga vektora), glatkoća reziduala raste, što doprinosi bržoj konvergenciji. Sličan učinak ima i vektor Jacobijeva postupka, s komponentama  $r_i/k_{ii}$ .

### 7.2.3. Izbor u skladu s postupcima preduvjetovanja

Brojne zamisli koje se upotrebljavaju za klasično preduvjetovanje sustava jednadžbi mogu se primijeniti za generiranje koordinatnih vektora. Osim matrica koje pripadaju isticanim iteracijskim postupcima, nepotpuni rastav Choleskoga ili matrični polinom o  $\mathbf{K}$ , također služe tvorbi matrice preduvjetovanja, označimo je s  $\mathbf{M}$ . Njome nastojimo smanjiti broj uvjetovanosti matrice sustava. Simbolički, umjesto rješavanja  $\mathbf{K}\mathbf{u} = \mathbf{f}$  posredno rješavamo

$$\mathbf{M}^{-1}\mathbf{K}\mathbf{u} = \mathbf{M}^{-1}\mathbf{f} \quad (31)$$

pri čemu matricu  $\mathbf{M}$  treba moći brzo invertirati. S našega gledišta  $\mathbf{M}^{-1}$  nije ništa drugo nego  $\mathbf{P}$ , a koordinatni vektor je opet  $\mathbf{Pr}$ . Zanimljivo, neki naši primjeri su bolje konvergirali tvorbom koordinatnih vektora pobrojavanjem prema nazad (engl. *backward*). Međutim, u literaturi je uobičajeno postupke preduvjetovanja upotrijebiti samo pobrojavanjem prema naprijed (engl. *forward*). Zbog toga bi vrijedilo pokušati realizaciju i prema nazad. Radi generiranja nekoliko koordinatnih vektora možemo upotrijebiti dva ili više načina preduvjetovanja (matrica  $\mathbf{P}$ ) istodobno. To je istoznačno višestruku preduvjetovanju. Tada učinkovitost koraka može porasti. Prema tomu, matrične transformacije koje služe preduvjetovanju sustava nisu potrebne, odnosno u smislu ovoga postupka preduvjetovanje možemo smatrati načinom tvorbe koordinatnih vektora. Prilikom generiranja i oni mogu ispasti (točno ili približno) linearno zavisni, pa jedan ili više njih treba odbaciti.

### 7.2.4. Izbor temeljen na prethodnomu prirastu pomaka

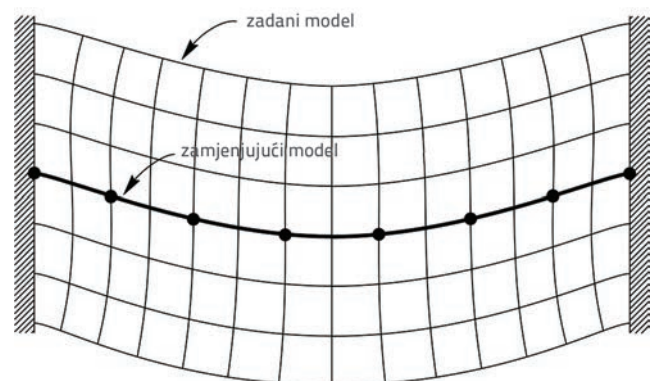
Dobar skup koordinatnih vektora temeljimo na "recikliranju povijesti" (engl. *history recycling*), odnosno uporabi vektora prethodnoga prirasta  $\Delta\mathbf{u}_{r-1}$ . Poznato je, taj vektor izrazito poboljšava konvergenciju konjugiranih gradijenata u odnosu na najstrijmiji silazak, a omogućava i uporabu faktora relaksacije, pa može povećati učinkovitost postupka. U našoj interpretaciji, jedan od vektora podprostora jest  $\phi_i = \Delta\mathbf{u}_{r-1}$  i ima sličan učinak, premda se u odnosu na metodu konjugiranih gradijenata gubi rekurzivna ortogonalnost na prethodne priraste. Kod našega postupka (za razliku od brojnih), postoji samo ortogonalnost među uzastopnim (ne udaljenim) prirastima rješenja, rezidualima ili podprostorima, jer ortogonalizacija "poskupljuje" korak. Ako je  $\Omega \neq 1$ , nema ni uzastopne ortogonalnosti. Zato tijekom proračuna mali sustav može postati slabije uvjetovan, što usporava konvergenciju. Eventualno bismo mogli razmišljati o ortogonalizaciji koordinatnih vektora nakon određenoga broja koraka, čemu nismo skloni. Dodatne vektore možemo generirati množenjem posljednjega vektora prirasta nekom matricom, odnosno  $\phi_i = \mathbf{S}\Delta\mathbf{u}_{r-1}$ . Dodavanje ranijih vektora prirasta  $\Delta\mathbf{u}_{r-2}$ ,  $\Delta\mathbf{u}_{r-3}$  i tako dalje, nije bilo dovoljno djelotvorno. Osim posljednjega vektora prirasta i trenutačni vektor rješenja  $\mathbf{u}_i$  može na sličan način poslužiti tvorbi koordinatnih vektora. "Recikliranje" toga vektora ima smisla upravo zbog gubitka ortogonalnosti, a mogao bi biti učinkovit za rješavanje nelinearnih sustava kod kojih

takvo svojstvo ne postoji. Spomenimo za kraj, ova se skupina vektora ne upotrebljava samostalno.

### 7.2.5. Izbor temeljen na podacima o modelu

Za drugi način generiranja koordinatnih vektora upotrebljavamo neke konkretne podatke o modelu koji rješavamo. Najčešće upotrebljavamo približnu geometriju i pojednostavljena svojstva modela, a često je dovoljan samo po nečemu sličan problem. Primjerice, možemo isključiti manje važne stupnjeve slobode, upotrijebiti istu geometriju i mrežu elemenata, ali jednostavniju razdiobu krutosti, iskoristiti hijerarhijsko ponašanje složenoga modela (kao kod ručnoga proračuna), upotrijebiti grublju mrežu elemenata i slično. Uglavnom, pomaci ovih modela pod rezidualnim opterećenjem mogu poslužiti kao dobri koordinatni vektori. Najčešće se radi o grubljim vektorima, koji najviše vrijede u prvim koracima proračuna. Kasnije treba izglediti rješenje, pa je bolje primijeniti neke od ranije opisanih, preciznijih pristupa. Unatoč tomu, takvi vektori mogu omogućiti iznimno brzo rješavanje mnogih konkretnih problema, ali postupak tvorbe ima dodatnu teškoću: nedostatak općenitosti. Svaki tip jednadžbi zahtijeva poseban pristup.

Primjerice, zamjenjujući model visoke grede može biti klasična, niska greda, a ljske membrana. Ako poznamo rješenje zamjenjujućega modela  $\mathbf{u}_z$  tada je koordinatni vektor  $\phi = \mathbf{N}\mathbf{u}_z$ , gdje je  $\mathbf{N}$  matrica interpolacijskih funkcija koja veže stupnjeve slobode zadanoga i zamjenjujućega modela. U slučaju grede zamjenjujući model je štapni (može biti i zglobno oslonjen), a zadani je definiran mrežom konačnih elemenata u ravnini (slika 4.). Točke obaju modela koje leže na osi vezane su polinomima trećega stupnja, a one zadanoga modela (smještene izvan osi), vežemo za zamjenjujući model hipotezom ravnoga poprečnog presjeka štapa. Polinomima i hipotezom tvorimo stupce matrice  $\mathbf{N}$ . Ovako određena matrica je singularna, jer su pomaci među modelima linearno zavisni, ali koordinatni vektor je korektan i s napretkom postupka vodi prema rješenju niskoga nosača. Naravno, to nije dovoljno dobro rješenje problema. Zbog toga treba uvesti i vektor reziduala (ili neki drugi "popravak"), kao dodatni koordinatni vektor koji će tijekom postupka korigirati pretpostavku ravnih presjeka i dovesti do dobrog rješenja visokoga nosača.



Slika 4. Model obostrano upetoga nosača

Za tvorbu koordinatnoga vektora možemo upotrijebiti i iteracijski postupak proračuna u skladu s hijerarhijom krutosti modela. Koordinatni vektor određujemo kao rješenje nakon samo nekoliko koraka iteracije među dijelovima modela s različitim krutostima. Jasno, to bi za konačan rezultat (koji bismo dobili nakon većega broja koraka), bila vrlo gruba aproksimacija, ali je za brzo dobivanje koordinatnoga vektora sasvim prihvatljiv. Vektore možemo generirati modelima s grubljom mrežom, primjenom par koraka nekoga višemrežnog postupka [15, 16], ali i izravnom uporabom analitički definiranih (kontinuiranih) koordinatnih funkcija nad približnim područjem proračuna, koje zadovoljavaju geometrijske rubne uvjete. Tu svakako spadaju Ritzove i kao zanimljivo proširenje R - funkcije [17], koje možemo množiti polinomima. Čak i slobodnije odabrane glatke funkcije koje ne ispunjavaju prirodne i geometrijske rubne uvjete (opet ističemo polinome), mogu poslužiti za koordinatne vektore. Otvore i slične nepravilnosti nije potrebno precizno uzeti u obzir. U ovim su slučajevima komponente koordinatnih vektora jednake vrijednostima funkcija u čvorovima modela. Jasno, takvi vektori su "grubi", a elementi pripadajuće Ritzove matrice su "prekruti", ali mogu poslužiti u prvim koracima proračuna. Ako ih ipak zadržimo, rješenje će se izgledati kasnijim tijekom postupka, ali će konvergencija postupka biti lošija. Zbog toga bi bilo bolje takve vektore unaprijed donekle izglatiti tako da sa svakim provedemo jedan ciklus prekomjerne relaksacije s malim  $\omega$  po rastućem, a potom i po padajućem redosljedu pobrojavanja nepoznanica.

I primjenom analogija možemo generirati koordinatne vektore. Primjerice, ako rješavamo problem ploče, upotrijebimo rješenje problema magnetskoga ili električnog polja, torzije, membrane, roštilja i slično. Vrijedi i rješenje iste ploče s drugim opterećenjem, ili čak "od oka" skicirano polje pomaka. Tada mjerenjem sa slike odredimo pomake kao komponente vektora. Sva takva rješenja mogu biti stupci matrice  $\phi_i$ .

Koordinatne vektore možemo primijeniti za učinkovito zadavanje kinematičkih ograničenja. Primjerice, spomenutu hipotezu ravnih presjeka štapa možemo smatrati takvim ograničenjem. Zapišemo li ih u obliku  $\mathbf{T}\mathbf{u} = \mathbf{f}$ , gdje je  $\mathbf{T}$  matrica ograničenja, početna aproksimacija rješenja  $\mathbf{u}_0$  mora zadovoljiti nehomogena ograničenja, odnosno  $\mathbf{T}\mathbf{u}_0 = \mathbf{f}$ , a koordinatni vektori samo homogena, dakle  $\mathbf{T}\phi_i = \mathbf{0}$ . Navedimo za kraj i mješoviti pristup tvorbi koordinatnih vektora. Tako prirast pomaka  $\Delta\mathbf{u}_i$  ili trenutačnu aproksimaciju  $\mathbf{u}_i$  pomnožimo nekim funkcijama koordinata. Može i Ritzovim ili R - funkcijama, pa i polinomima koje smo ranije istaknuli.

## 7.2.6. Koordinatni vektori kao ulazni podaci

Konačno, ako se netko sjeti boljšega skupa (s jednim ili više) koordinatnih vektora, bez teškoća može formirati matricu  $\phi_i$  i upotrijebiti ovdje opisani postupak. Na taj način predloženi algoritam možemo smatrati općim pristupom iteracijskomu rješavanju sustava, kojemu kao ulazne podatke (osim nužnih koje trebaju svi iteracijski postupci), zadajemo i koordinatne vektore.

## 8. Ukratko o realizaciji

Priloženi pseudokod realiziran je programskim jezikom gfortran [18]. Upotrijebljene su 64 bitna Ubuntu inačica 5.3.1 i OS X inačica 6.1.0. Nakon provjere programa na malim sustavima jednadžbi, generirali smo matrice krutosti i opterećenja velikoga broja ravninskih i prostornih rešetkastih nosača. Tvorbe modela proveli smo na dva načina. S jedne strane, štapove smo postavljali klasično, tako da rešetke tvore dobar statički sistem. S druge strane, radi svjesnoga pogoršanja uvjetovanosti pripadajućeg sustava, odnosno broja  $cond(\mathbf{K})$ , štapovima velikih razlika u krutostima nepravilno smo spajali udaljene čvorove. Time smo tvorili nelogične rešetke koje ne možemo smatrati konstrukcijama. Na taj smo način program testirali na, s numeričkoga gledišta, dobrim i lošim modelima. Upotrijebili smo zapis matrice u obliku potpunoga knjiženja (engl. *full bookkeeping*) po stupcima, a probali smo i po retcima [19]. Vrlo slična inačica zapisa postoji i u programu otvorena koda za realizaciju metode konačnih elemenata FEAP [20]. Time smo odmah vodili računa o spajanju s tim programskim paketom. Upotrijebili smo inačicu 8.4.1 [21]. Spajanje, programsko prevođenje i povezivanje s FEAP-om također su realizirani jezikom gfortran.

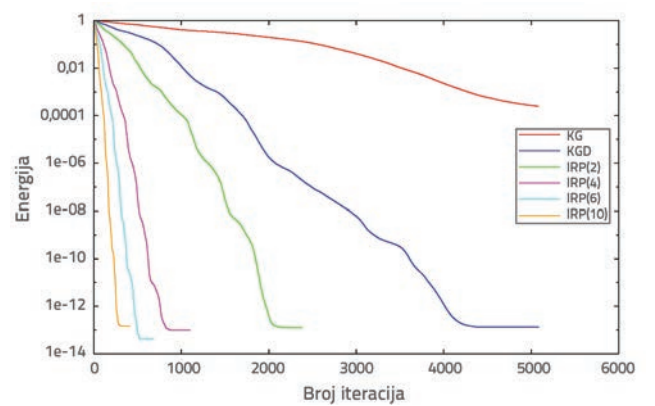
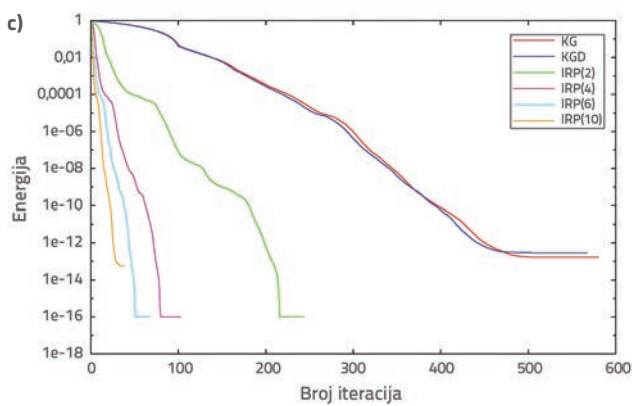
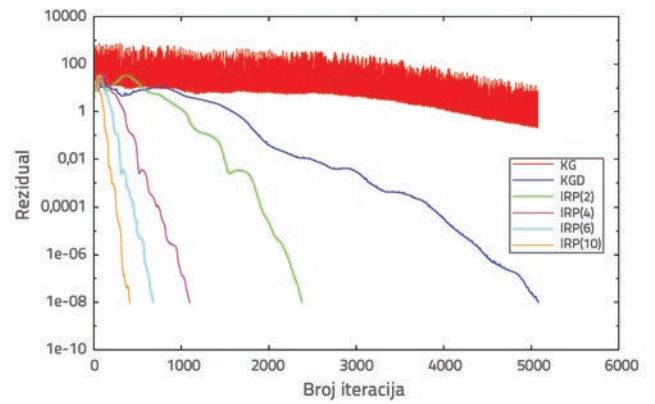
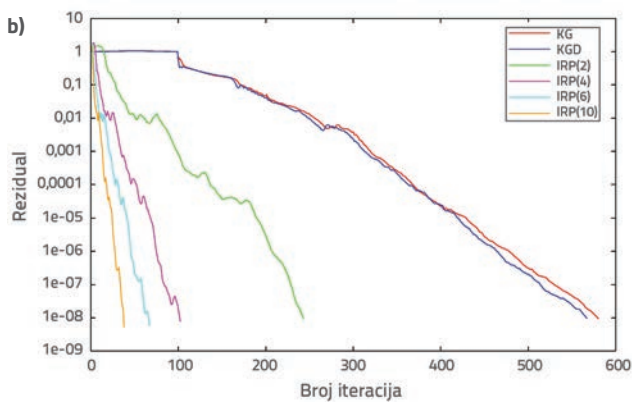
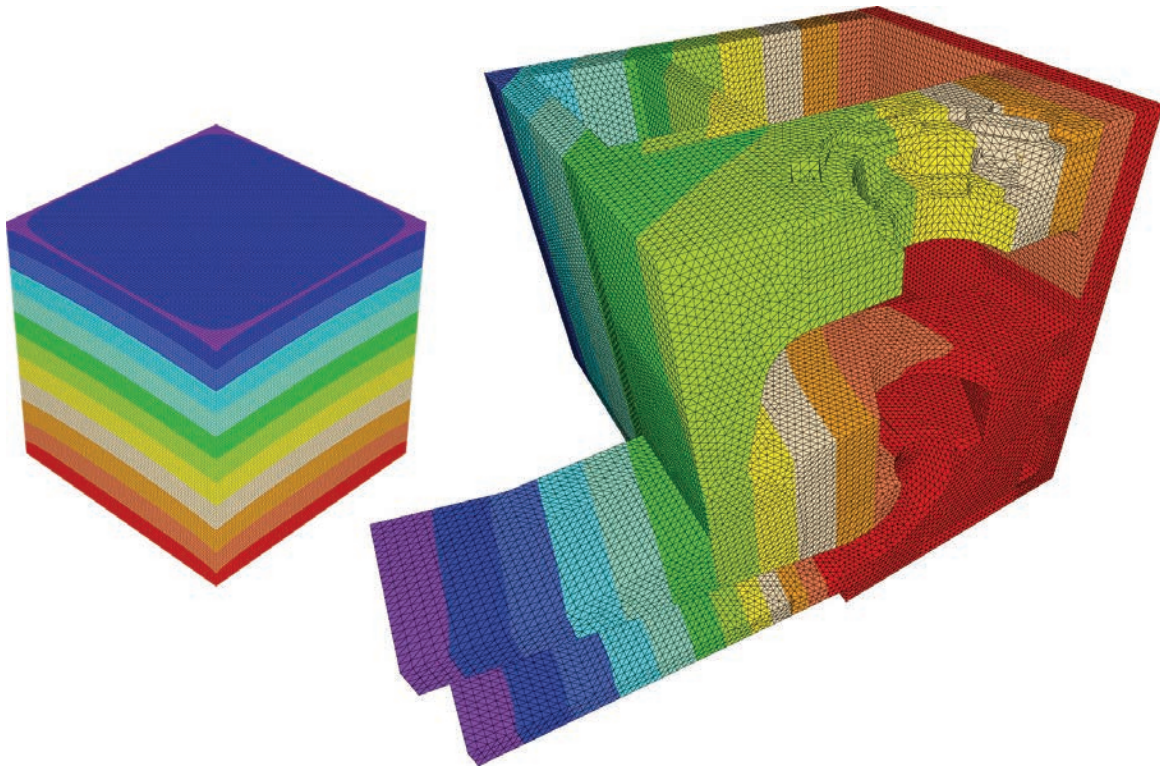
## 9. Rezultati proračuna praktičnih modela

Nakon temeljnih provjera, analizirali smo nekoliko modela iz konstruktorske prakse na kojima smo radili prijašnjih godina (slike 5. do 7.). Radi jasnoće, na slikama su izostavljeni prikazi opterećenja i ležajeva. Modeli sadrže različite vrste i oblike konačnih elemenata. Nepoznanice su pomaci, a ponegdje i kutovi zaokreta. Osnovni podaci o modelima priloženi su u tablici 2. Popunjenost matrice određena je kao omjer broja članova štednoga zapisa i broja svih elemenata matrice.

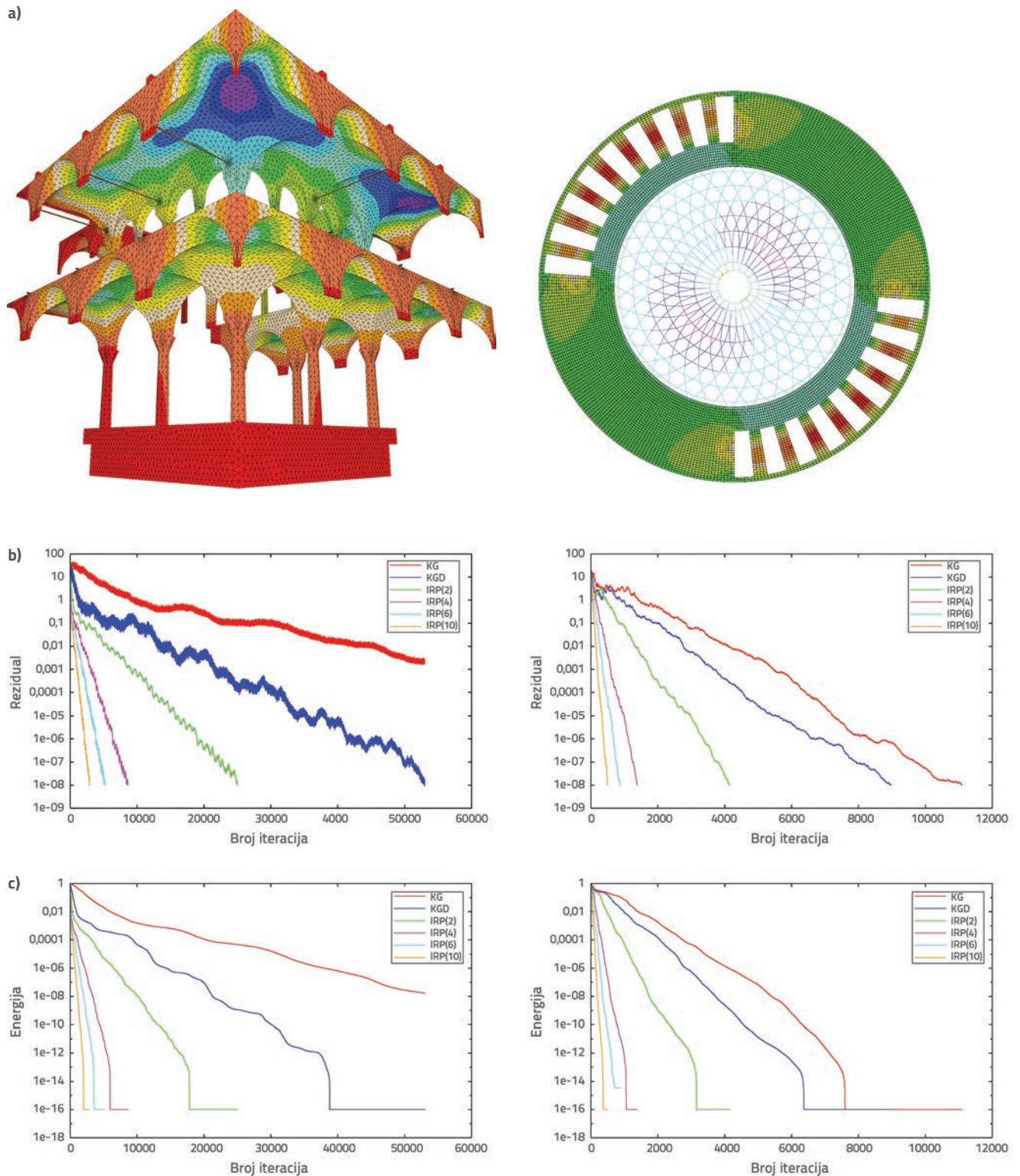
Tablica 2. Osnovni podaci o numeričkim modelima

	Broj slike	Broj čvorova	Broj elemenata	Broj nepoznanica	Broj članova štednoga zapisa	Popunjenost matrice
5.	lijevo	1.030.301	1.000.000	3.060.300	123.026.091	$1,31 \cdot 10^{-5}$
	desno	276.244	1.461.134	820.446	17.723.235	$2,63 \cdot 10^{-5}$
6.	lijevo	71.307	278.499	206.527	3.826.156	$8,97 \cdot 10^{-5}$
	desno	11.844	11.664	69.984	1.635.876	$3,34 \cdot 10^{-4}$
7.	lijevo	3.018.960	2.918.728	8.955.164	358.190.300	$4,47 \cdot 10^{-6}$
	desno	486	1.782	2.754	54.594	$7,20 \cdot 10^{-3}$

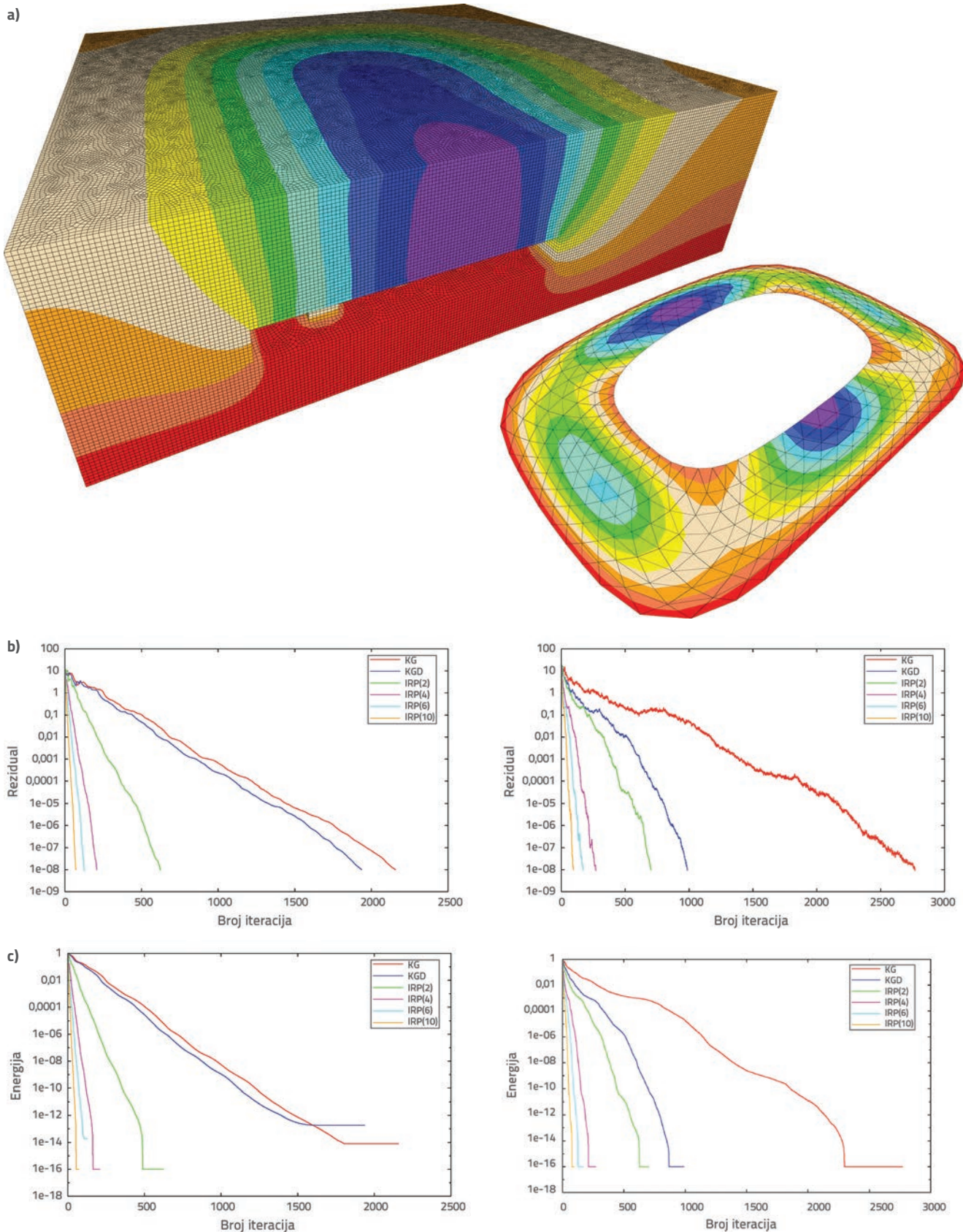
a)



Slika 5. Proračun modela kocke (lijevo) i strojnice HE Rama (desno; prikazana je samo polovina modela) [22]: a) razdioba vertikalnih pomaka; b) opadanje reziduala; c) smanjenje energije



Slika 6. Proračun atrija Kneževa dvora u Dubrovniku (lijevo) [23] i kupole dvorane "Krešimir Ćosić" u Zadru (desno) [24]: a) razdioba vertikalnih pomaka; b) opadanje reziduala; c) smanjenje energije



Slika 7. Proračun podzemnoga kamenoloma Kanfanar s okolnim područjem (lijevo; unutarnje prostorije i stupovi nisu vidljivi) [25] i krovšta budućega stadiona Kantrida na Rijeci (desno) [26]: a) razdioba vertikalnih pomaka; b) opadanje reziduala; c) smanjenje energije

Tablica 3. Usporedba postupaka prema broju koraka

Broj slike		Broj koraka do ispunjenja kriterija					
		KG	KGD	IRP(2)	IRP(4)	IRP(6)	IRP(10)
5.	lijevo	580	567	243	102	67	38
	desno	$>10^4$	5083	2381	1097	682	410
6.	lijevo	$>10^5$	53.002	24.995	8608	5166	2 871
	desno	11 091	8966	4142	1381	888	498
7.	lijevo	2 157	1936	623	208	126	71
	desno	2 769	987	703	269	169	94

Raznovrsne su provjere iteracijskih algoritama nužne, jer je poznata njihova ovisnost o naravi problema. Odnosno, mogu se prilagoditi za izrazito brzo rješavanje tipičnih primjera, s unaprijed poznatim podacima kojima namjestimo ključne parametre postupka. Međutim, čim problem odstupa od očekivanoga, učinkovitost pada. Na slikama su priložene razdiobe vertikalnih pomaka modela [pod a)], te ovisnosti logaritma omjera reziduala  $\|r_i\|_2/\|r_0\|_2$  i energije  $(\Pi_0 - \sum \Delta \Pi_i)/\Pi_0$  o broju koraka [pod b) i c)]. Crvena boja označava područje najmanjih, a ljubičasta najvećih pomaka. Provedeni su proračuni iteriranim Ritzovim postupkom (IRP) sa dva, četiri, šest i deset koordinatnih vektora (argument uz IRP), te metodom konjugiranih gradijenata bez preduvjetovanja (KG) i s dijagonalnim preduvjetovanjem (KGD). Na slikama pod b) i c) možemo uočiti: s povećanjem podprostora opada broj potrebnih koraka, odnosno veće je smanjenje reziduala i energijske norme po koraku. Čak i za dva koordinatna vektora postupak brže konvergira od KG i KGD (koje također možemo interpretirati dvama vektorima). Treba priznati, postoji i bolje preduvjetovanje metode konjugiranih gradijenata, primjerice nepotpunim rastavom Choleskoga, ali iz rezultata je vidljivo da za veći broj koordinatnih vektora imamo pristojnu zalihu (što svakako treba provjeriti). To potvrđuje i tablica 3. koja sadrži broj koraka potreban za smanjenje omjera reziduala na  $10^{-8}$ . Istaknimo još, rezultati svih primjera se podudaraju s rješenjima dobivenima izravnim i iteracijskim postupcima koji standardno postoje unutar FEAP-a.

## 10. Zaključak

U skladu s opisanim svojstvima i iskustvima autora ovog rada, iterirani Ritzov postupak može imati prednosti kod rješavanja velikih linearnih sustava, slabo popunjenih matrica. Neki naši primjeri su sezali do  $10^7$  nepoznanica, popunjenosti oko  $10^{-6}$ . Tumačenje postupka je blisko inženjerskomu načinu razmišljanja, odnosno Ritzovoj energijskoj interpretaciji, za razliku od primjerice metode konjugiranih gradijenata koja se uobičajeno pojašnjava geometrijski, u apstraktnomu  $n$  - dimenzijskom prostoru.

Ovaj postupak ne bi smio biti lošiji od metode konjugiranih gradijenata (s preduvjetovanjem i bez njega). Odnosno, dobro proširenje podprostora ne donosi pogoršanje konvergencije. Pri tomu možemo upotrijebiti više načina generiranja koordinatnih vektora. Kao da radimo s više iteracijskih postupaka istodobno. Dobrim izborom vektora konvergencija je puno brža od konvergencije pojedinačnoga postupka. Preduvjetovanje koje transformira sustav ovdje nije potrebno, ali se algoritmi razvijeni za preduvjetovanje mogu uspješno primijeniti za tvorbu vektora. Ponovno pokretanje (engl. *restart*), poznato iz klasične metode konjugiranih gradijenata, kojim se zbog gubitka ortogonalnosti ponovo započinje postupak, ovdje nije opravdano. Iz prethodnoga koraka nasljeđujemo samo prirast pomaka koji poboljšava rješenje u trenutačnom koraku. Budući da ne zahtijevamo ortogonalnost među iteracijskim veličinama, dodatno ubrzanje treba očekivati u primjeni na nelinearne probleme. Naime, tada nema svojstava ortogonalnosti na kojima ustrajavaju brojni iteracijski algoritmi. Zbog toga se postupak može uspješno primijeniti i u području optimizacije.

Jasno je, izrazito velike sustave nije ekonomično rješavati jednim procesorom. Tada paralelan pristup postaje nužnost. Zanimljivo bi bilo svakomu koordinatnom vektoru pridružiti njegov procesor. Budući da smo, radi brojnih mogućnosti, još uvijek u intenzivnoj istraživačkoj fazi algoritma (već samo izmjene iznosa  $\Omega$  i  $\omega$  izrazito utječu na brzinu konvergencije), nema smisla mjeriti trajanje proračuna i usporedbu utrošenoga vremena u odnosu na druge izravne i iteracijske postupke. To je moguće napraviti tek nakon brižljivoga programiranja, programskoga prevođenja i povezivanja konačne inačice programa, s uključenim optimizacijskim opcijama koje doprinose brzini izvođenja koda. Zbog toga je ovaj rad više metodološke (algoritamske), nego praktične naravi.

## Zahvale

Ovaj je rad financirala Hrvatska zaklada za znanost projektom IP - 2014 - 09 - 2899. Zahvaljujemo doc. dr. sc. Mariu Urošu na pomoći prilikom pripreme numeričkih modela.

## LITERATURA

- [1] Dvornik, J.: Generalization of the CG Method Applied to Linear and Nonlinear Problems, *Computers & Structures*, 10 (1979) 1/2, pp. 217-223.
- [2] Varga, R.S.: *Matrix Iterative Analysis*, Second Edition, Springer Series in Computational Mechanics, Springer - Verlag, Berlin, 2000, <https://doi.org/10.1007/978-3-642-05156-2>
- [3] Saad, Y., van der Vorst, H.: Iterative solution of linear systems in the 20th century, *Journal of Computational and Applied Mathematics*, 123 (2000), pp. 1-33.
- [4] van der Vorst, H.A.: *Iterative Krylov Methods for Large Linear Systems*, Cambridge University Press, Cambridge, 2003, <https://doi.org/10.1017/CBO9780511615115>
- [5] Liesen, J., Strakoš, Z.: *Krylov Subspace Methods. Principles and Analysis*, Oxford University Press, Oxford, 2013.
- [6] Brezinski, C.: Multiparameter descent methods, *Linear Algebra and its Applications*, 296 (1999) 1-3, pp. 113.-141.
- [7] Axelsson, O.: *Iterative Solution Methods*, Cambridge University Press, New York, 1994, <https://doi.org/10.1017/CBO9780511624100>
- [8] Greenbaum, A.: *Iterative Methods for Solving Linear Systems*, Siam, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, 1997, <https://doi.org/10.1137/1.9781611970937>
- [9] Young, D.M.: *Iterative Solution of Large Linear Systems*, Dover Publications, Inc., Mineola, New York, 2003.
- [10] Saad, Y.: *Iterative Methods for Sparse Linear Systems*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, 2003, <https://doi.org/10.1137/1.9780898718003>
- [11] Olshanskii, M.A., Tyrtshnikov, E.E.: *Iterative Methods for Linear Systems. Theory and Application*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, 2014, <https://doi.org/10.1137/1.9781611973464>
- [12] Barrett, R., Berry, M., Chan, T.F., Demmel, J., Donato, J., Dongarra, J., Eijkhout, V., Pozo, R., Romine, C., van der Vorst, H.: *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, 1987.
- [13] Benzi, M.: Preconditioning techniques for large linear systems: A survey, *Journal of Computational Physics*, 182 (2002), pp. 418-477.
- [14] Korneev, V.G., Langer, U.: Domain Decomposition Methods and Preconditioning, u: *Encyclopedia of Computational Mechanics* (Editors: Erwin Stein, René de Borst, Thomas J. R. Hughes), Chapter 22, Volume 1: Fundamentals, John Wiley & Sons, Ltd, Chichester, 2004., pp. 617-649.
- [15] Trottenberg, U., Oosterlee, C. W., Schüller, A.: *Multigrid*, Academic Press, London, 2001.
- [16] Hackbusch, W.: *Multigrid Methods for FEM and BEM Applications*, *Encyclopedia of Computational Mechanics* (Editors: Erwin Stein, René de Borst, Thomas J. R. Hughes), Fundamentals, John Wiley & Sons, Ltd, Chichester, 1 (2004) 20, pp. 577-596.
- [17] Dvornik, J.: Generiranje glatkih funkcija na složenom području pomoću R - funkcija, *KoG*, 1 (1996), pp. 27-30.
- [18] The gfortran team: *Using GNU Fortran, For GCC version 7.0.0 (pre - release)*, (GCC), Free Software Foundation, Boston, 2016.
- [19] Nour - Omid, B., Taylor, R. L.: *An Algorithm for Assembly of Stiffness Matrices into a Compacted Data Structure*, Report No. UCB/SESM - 84/06, Structural Engineering and Structural Mechanics, Department of Civil Engineering, University of California, Berkeley, 1984.
- [20] Taylor, R.L.: *FEAP - A Finite Element Analysis Program, Version 8.4, User Manual*, Department of Civil and Environmental Engineering, University of California at Berkeley, Berkeley, 2013.
- [21] Taylor, R.L.: *FEAP - A Finite Element Analysis Program, Version 8.4, Programmer Manual*, Department of Civil and Environmental Engineering, University of California at Berkeley, Berkeley, 2014.
- [22] Statički i dinamički proračun postojećih temelja agregata na nova opterećenja i djelovanja, knjiga G3, glavni projekt (projektant konstrukcije za ovu projektnu obradu Milan Crnogorac), Javno poduzeće Elektroprivreda hrvatske zajednice Herceg bosne d. d. Mostar, Zagreb, 2012.
- [23] Lazarević, D., Dvornik, J., Fresl, K.: Analiza oštećenja atrija Kneževa dvora u Dubrovniku, *GRAĐEVINAR*, 56 (2004) 10, pp. 601-612.
- [24] Lazarević, D., Anđelić, M., Uroš, M.: Oblikovanje i proračun kupole dvorane "Krešimir Ćosić" u Zadru, *Građevinar*, 62 (2010) 10, pp. 875.-886.
- [25] Hrženjak, P., Petzel, M., Vujec, S.: Kontrolna mjerenja i numeričke analize za dimenzioniranje stupova i komora pri podzemnom otkopavanju arhitektonsko-građevnog kamena, *Znanstvenostručno savjetovanje s međunarodnim sudjelovanjem "Mehanika stijena i tuneli"*, (urednici Jašarević, I., Hudec, M., Vujec, S.), 1 (1999), pp. 127-133.
- [26] Uroš, M., Gidak, P., Lazarević, D.: Optimization of stadium roof structure using force density method, *Proceedings of the third international conference on structures and architecture (ICSA2016) - Structures and Architecture - Beyond their Limits* (editor Paulo J.S. Cruz), CRC Press/Balkema, Taylor & Francis Group, Guimaraes, 2016. pp. 693-700, <https://doi.org/10.1201/b20891-95>