

Επαναληπτικές μέθοδοι για την επίλυση γραμμικών  
συστημάτων

Μιχάλης Δρακόπουλος

Σημειώσεις Αριθμητικής Γραμμικής Άλγεβρας 2012–2013

## Εισαγωγή

Στην αριθμητική επίλυση μαθηματικών εφαρμογών, όπως για παράδειγμα σε μοντέλα διαφορικών εξισώσεων, συχνά καταλήγουμε σε πολύ μεγάλα γραμμικά συστήματα

$$A\mathbf{x} = \mathbf{b}, \quad A \in \mathbb{R}^{n \times n}, \quad \mathbf{x}, \mathbf{b} \in \mathbb{R}^n. \quad (1)$$

Η επίλυση τέτοιων συστημάτων κυριαρχεί στον υπολογιστικό χρόνο που απαιτείται συνολικά. Όπως είναι γνωστό, οι άμεσες μέθοδοι επίλυσης, όπως η απαλοιφή Gauß, κοστίζουν  $O(n^3)$  σε πράξεις και  $O(n^2)$  σε θέσεις μνήμης.

Επιπλέον, τα μαθηματικά μοντέλα συνήθως αντικαθιστούν το συνεχές αρχικό πρόβλημα με μια διακριτή προσέγγισή του πράγμα που σημαίνει ότι κάθε στοιχείο  $x_i$  του  $\mathbf{x}$  που αντιστοιχεί σε μια παράμετρο του προβλήματος, αλληλεπιδρά μόνο με κάποια λίγα “γειτονικά” του  $x_j$ . Αυτό σημαίνει ότι το στοιχείο  $a_{ij}$  του  $A$  θα είναι διάφορο του μηδενός μόνο όταν το  $x_i$  εξαρτάται από το  $x_j$ . Ο  $A$  δηλαδή είναι συνήθως ένας μεγάλος αραιός πίνακας.

Στις περιπτώσεις αυτές, αντί των άμεσων μεθόδων επίλυσης του (1) συχνά προτιμώνται οι επαναληπτικές μέθοδοι. Η βασική ιδέα πίσω από τις επαναληπτικές μεθόδους είναι να ξεκινήσουμε από μια αρχική προσέγγιση  $\mathbf{x}^{(0)}$  (που μπορεί να είναι και αυθαίρετη αν δεν γνωρίζουμε κάτι για τη λύση) και να κατασκευάσουμε διαδοχικές προσεγγίσεις  $x^{(1)}, x^{(2)}, \dots, x^{(k)}$  που να συγκλίνουν στο  $\mathbf{x}$ .

Οι επαναληπτικές μέθοδοι είναι γρηγορότερες και πιο οικονομικές από πλευράς μνήμης από τις αντίστοιχες άμεσες γιατί εκμεταλλεύονται την αραιή δομή του  $A$ . Η επαναληπτική διαδικασία επίσης, δεν χρειάζεται να φτάσει μέχρι την πραγματική λύση, αλλά μπορεί να σταματήσει μόλις επιτευχθεί κάποια επιθυμητή ακρίβεια που σχετίζεται με την ακρίβεια των δεδομένων του προβλήματος.

Στα παρακάτω θα εξετάσουμε δομές αποθήκευσης και αλγόριθμους αραιών πινάκων και μια οικογένεια σχετικά απλών επαναληπτικών μεθόδων.

## Αραιοί πίνακες

Ο Wilkinson ορίζει σαν αραιό ένα πίνακα με αρκετά μηδενικά στοιχεία έτσι ώστε να αξίζει να τα εκμεταλλευτούμε στους υπολογισμούς μας. Αντιμετωπίζοντας ένα πίνακα σαν αραιό μπορούμε να έχουμε:

- ταχύτερους υπολογισμούς αφού δεν εκτελούνται πράξεις με μηδενικά στοιχεία;
- εξοικονόμηση μνήμης αφού τα μηδενικά στοιχεία δεν αποθηκεύονται.

Επειδή όμως η αποθήκευση αραιών πινάκων απαιτεί εξειδικευμένες δομές δεδομένων, θα πρέπει το επιπλέον κόστος διαχείρισης να υπερκαλύπτεται από τα οφέλη σε ταχύτητα λόγω της μη-εκτέλεσης πράξεων με μηδενικά στοιχεία.

Παρακάτω εξετάζουμε δύο από τις πιο γνωστές δομές αποθήκευσης αραιών πινάκων. Έστω  $A$  αραιός πίνακας  $n \times n$  με  $m$  μη-μηδενικά στοιχεία. Για παράδειγμα ο

$$A = \begin{pmatrix} 0 & 3 & 0 & 0 & 1 \\ 4 & 1 & 0 & 0 & 0 \\ 0 & 5 & 9 & 2 & 0 \\ 6 & 0 & 0 & 5 & 3 \\ 0 & 0 & 5 & 8 & 9 \end{pmatrix}, \quad n = 5, \quad m = 13. \quad (2)$$

### Αλγόριθμος 1: $y = Ax$ (σύστημα συντεταγμένων)

```
function y = matvec(val, row, col, x)
m = length(val); n = length(x);
y = zeros(n, 1);
for k = 1:m
    y(row(k)) = y(row(k)) + val(k)*x(col(k));
end
```

### Αποθήκευση με το σύστημα συντεταγμένων

Στο σύστημα αποθήκευσης με *συντεταγμένες*, κάθε μη-μηδενικό στοιχείο του  $a_{ij}$  παριστάνεται σαν  $(i, j, a_{ij})$ , όπου  $i$  η γραμμή,  $j$  η στήλη και  $a_{ij}$  η αριθμητική τιμή του στοιχείου. Οι τριάδες αποθηκεύονται με οποιαδήποτε σειρά.

Είναι ο πιο κατανοητός τρόπος αποθήκευσης, αλλά δεν είναι εύκολο να γίνουν πράξεις ανά γραμμές ή ανά στήλες του πίνακα.

Για την αποθήκευση με το σύστημα συντεταγμένων χρειαζόμαστε 3 διανύσματα μήκους  $m$  (2 ακέραια και 1 πραγματικό). Μια πιθανή αποθήκευση του πίνακα  $A$  στην (2) είναι:

```
val = [3 1 4 1 5 9 2 6 5 3 5 8 9]
row = [1 1 2 2 3 3 3 4 4 4 5 5 5]
col = [2 5 1 2 2 3 4 1 4 5 3 4 5]
```

Ο αλγόριθμος πολλαπλασιασμού αραιού πίνακα (σύστημα συντεταγμένων) με διάνυσμα, σε ψευδοκώδικα MATLAB δίνεται παρακάτω:

### Αποθήκευση συμπιεσμένων γραμμών

Στο σύστημα Compressed Row Storage (CRS) κάθε γραμμή  $i$  ενός αραιού πίνακα  $A$  αποθηκεύεται σαν ένα συμπιεσμένο αραιό διάνυσμα σε ζεύγη  $(j, a_{ij})$  για τα μη-μηδενικά στοιχεία της.

Οι συμπιεσμένες γραμμές αποθηκεύονται συνεχόμενα στο πραγματικό διάνυσμα αριθμητικών τιμών  $val$  και οι αντίστοιχες στήλες τους στο ακέραιο διάνυσμα  $col$ . Τα διανύσματα  $val$  και  $col$  έχουν μήκος  $m$ . Για τον εντοπισμό της αρχικής θέσης κάθε γραμμής  $i$  στα  $val$  και  $col$  χρησιμοποιείται το βοηθητικό ακέραιο διάνυσμα  $start$ . Έτσι για τη γραμμή  $i$ , η διεύθυνση του πρώτου μη-μηδενικού στοιχείου της είναι η  $start(i)$  και η διεύθυνση του τελευταίου η  $start(i+1)-1$  (μια θέση πριν την αρχή της γραμμής  $i+1$ ). Το πλήθος των μη-μηδενικών στοιχείων της γραμμής  $i$  είναι  $start(i+1)-start(i)$ .

Για να επεξεργαστούμε (π.χ. να εμφανίσουμε) τα στοιχεία της γραμμής  $i$  χρησιμοποιούμε ένα βρόχο της μορφής

```
for k = start(i):start(i+1)-1
    disp(val(k));
end
```

Με μια πρώτη ματιά θα περιμέναμε το διάνυσμα  $start$  να έχει  $n$  στοιχεία, όσες δηλαδή και οι γραμμές του  $A$ . Έτσι όμως, ο παραπάνω κώδικας δεν “δουλεύει” για την τελευταία γραμμή (για  $i=n$ ) αφού δεν θα υπάρχει το στοιχείο  $start(n+1)$ . Για να αντιμετωπίσουμε

### Αλγόριθμος 2: $y = Ax$ (σύστημα CRS)

```
function y = matvec(val, col, start, x)
n = length(x); y = zeros(n, 1);
for i = 1:n
    for k = start(i):start(i+1)-1
        y(i) = y(i) + val(k)*x(col(k));
    end
end
```

### Αλγόριθμος 3: Εμπρός αντικατάσταση $Lx = b$ (σύστημα CRS)

```
function x = forsub(val, col, start, b)
n = length(b);
for i = 1:n
    for k = start(i):start(i+1)-1
        b(i) = b(i) - val(k)*x(col(k));
    end
    x(i) = b(i)/val(start(i+1)-1)
end
```

το θέμα αυτό εισάγουμε από σύμβαση ένα επιπλέον στοιχείο στο τέλος του `start` που αντιστοιχεί στη θέση που θα ξεκίναγε η γραμμή  $n+1$  αν υπήρχε. Η τιμή του `start(n+1)` είναι  $m+1$ . Οι απαιτήσεις μνήμης για τη δομή αποθήκευσης CRS είναι επομένως  $m$  πραγματικές και  $m+n+1$  ακέραιες θέσεις.

Η αποθήκευση CRS για τον πίνακα στην (2) είναι:

```
val = [3 1 4 1 5 9 2 6 5 3 5 8 9]
row = [1 1 2 2 3 3 3 4 4 4 5 5 5]
start = [1 3 5 8 11 14]
```

Ακολουθούν αλγόριθμοι σε ψευδοκώδικα MATLAB, για πολλαπλασιασμό πίνακα με διάνυσμα (`matvec`), εμπρός και πίσω αντικατάσταση (`forsub` και `baksub` αντίστοιχα) όταν οι εμπλεκόμενοι πίνακες είναι αποθηκευμένοι σε CRS.

Για την εμπρός και πίσω αντικατάσταση θεωρούμε ότι τα διαγώνια στοιχεία είναι διάφορα του μηδενός. Σε ένα κάτω τριγωνικό πίνακα αποθηκευμένο σε CRS το διαγώνιο στοιχείο της γραμμής  $i$  βρίσκεται στη θέση `start(i+1)-1` (τελευταίο στοιχείο στη γραμμή). Αντίστοιχα για έναν άνω τριγωνικό πίνακα σε CRS το διαγώνιο στοιχείο της γραμμής  $i$  βρίσκεται στη θέση `start(i)` (πρώτο στοιχείο στη γραμμή).

## Επαναληπτικές μέθοδοι σταθερού σημείου

Αν  $x$  είναι η πραγματική λύση του γραμμικού συστήματος (1) και  $\tilde{x}$  μια προσέγγιση της, τότε το σφάλμα  $e = x - \tilde{x}$  και το υπόλοιπο  $r = b - A\tilde{x}$  συνδέονται με τη σχέση

$$r = b - A\tilde{x} = Ax - A\tilde{x} = A(x - \tilde{x}) = Ae.$$

Αλγόριθμος 4: Πίσω αντικατάσταση  $U\mathbf{x} = \mathbf{b}$  (σύστημα CRS)

---

```

function x = baksb(val, col, start, b)
n = length(b);
for i = n:-1:1
    for k = start(i)+1:start(i+1)-1
        b(i) = b(i) - val(k)*x(col(k));
    end
    x(i) = b(i)/val(start(i))
end

```

---

Συνεπώς, αν μπορούσαμε να υπολογίσουμε το σφάλμα  $\mathbf{e}$  που σχετίζεται με την προσέγγιση  $\tilde{\mathbf{x}}$  τότε θα παίρναμε άμεσα και την πραγματική λύση σαν

$$\mathbf{x} = \tilde{\mathbf{x}} + \mathbf{e}. \quad (3)$$

Το σφάλμα επομένως μπορεί να θεωρηθεί σαν μια διόρθωση της προσέγγισης. Ο υπολογισμός του σφάλματος όμως, προκύπτει σαν λύση ως προς  $\mathbf{e}$  του γραμμικού συστήματος  $A\mathbf{e} = \mathbf{r}$ , που είναι ουσιαστικά το αρχικό πρόβλημα (1) που θέλουμε να επιλύσουμε.

Μια επαναληπτική διαδικασία επίλυσης μπορεί να προκύψει με διαδοχικές διορθώσεις τύπου (10), αν αντί του  $\mathbf{e}$  χρησιμοποιήσουμε μια καλή και εύκολα υπολογίσιμη προσέγγιση  $\tilde{\mathbf{e}}$  σύμφωνα με τον ακόλουθο αλγόριθμο

(α) Υπολογισμός του υπολοίπου  $\mathbf{r} = \mathbf{b} - A\mathbf{x}^{\text{old}}$ .

(β) Επίλυση του συστήματος  $M\tilde{\mathbf{e}} = \mathbf{r}$  ως προς  $\tilde{\mathbf{e}}$ .

(γ) Διόρθωση  $\mathbf{x}^{\text{new}} = \mathbf{x}^{\text{old}} + \tilde{\mathbf{e}}$ .

Ο πίνακας  $M$  είναι κάποια προσέγγιση στον  $A$  τέτοια ώστε γραμμικά συστήματα με συντελεστή τον  $M$  να είναι ευκολότερα επιλύσιμα απ' ό τι με τον  $A$ , και τα διανύσματα  $\mathbf{x}^{\text{old}}$  και  $\mathbf{x}^{\text{new}}$  είναι αντίστοιχα η τρέχουσα και η επόμενη προσέγγιση της λύσης  $\mathbf{x}$ .

Τα παραπάνω τρία βήματα του αλγορίθμου μπορούν να συμπυκνωθούν σε ένα. Πολλαπλασιάζοντας τη σχέση  $\mathbf{x}^{\text{new}} = \mathbf{x}^{\text{old}} + \tilde{\mathbf{e}}$  στο βήμα (γ) επί  $M$  έχουμε

$$M\mathbf{x}^{\text{new}} = M\mathbf{x}^{\text{old}} + M\tilde{\mathbf{e}}.$$

Και μιας και  $M\tilde{\mathbf{e}} = \mathbf{r} = \mathbf{b} - A\mathbf{x}^{\text{old}}$  παίρνουμε

$$M\mathbf{x}^{\text{new}} = M\mathbf{x}^{\text{old}} + \mathbf{b} - A\mathbf{x}^{\text{old}} \Leftrightarrow M\mathbf{x}^{\text{new}} = (M - A)\mathbf{x}^{\text{old}} + \mathbf{b}.$$

Θέτοντας  $N = M - A$ , η επανάληψη γίνεται

$$M\mathbf{x}^{\text{new}} = N\mathbf{x}^{\text{old}} + \mathbf{b}. \quad (4)$$

Η λύση  $\mathbf{x}^{\text{new}}$  της εξίσωσης (4) είναι η ίδια που προκύπτει στο βήμα (γ) του παραπάνω αλγορίθμου. Μιας και ο αρχικός πίνακας  $A$  ισούται με τη διαφορά  $M - N$ , οι πίνακες  $M$ ,  $N$  ονομάζονται πίνακες διαχωρισμού. Παρατηρήστε ακόμα ότι αν η επανάληψη συγκλίνει στο όριο  $\mathbf{x}$ , τότε  $\mathbf{x}^{\text{new}} = \mathbf{x}^{\text{old}} = \mathbf{x}$  και από την (4)

$$M\mathbf{x} = N\mathbf{x} + \mathbf{b} \Leftrightarrow (M - N)\mathbf{x} = \mathbf{b} \Leftrightarrow A\mathbf{x} = \mathbf{b}$$

---

### Αλγόριθμος 5: μέθοδος Jacobi

---

```
function [x, iter] = jacobi(A, b, x, epsilon, maxiter)
% A      - pinakas susthmatos
% b      - dexi melos
% x      - sthn eisodo h arqikh ektimhsh, sthn exodo h lush
% epsilon - zhtoumenh akribeia
% maxiter - megistos arijmos epanalhyewn
% iter   - epanalhyeis pou eginan
n = length(b); xnew = zeros(n,1);
for iter = 1:maxiter
    for i = 1:n
        sum = b(i) - A(i,[1:i-1,i+1:n])*x([1:i-1,i+1:n]);
        xnew(i) = sum/A(i,i);
    end
    relerr = norm(xnew-x)/norm(xnew);
    x = xnew;
    if relerr<epsilon, break; end
end
```

---

δηλαδή το σταθερό σημείο της εξίσωσης (4) είναι η λύση του γραμμικού συστήματος (1).

Ανάλογα με την επιλογή του πίνακα  $M$  προκύπτουν διάφορες επαναληπτικές μέθοδοι επίλυσης γραμμικών συστημάτων αυτής της κατηγορίας. Όταν ο είναι αραιός περιμένουμε και οι  $M, N$  των μεθόδων αυτών να είναι επίσης αραιοί πίνακες.

### Η μέθοδος Jacobi

Η μέθοδος προκύπτει εύκολα αν εξετάσουμε καθεμία από τις  $n$  εξισώσεις του γραμμικού συστήματος  $\mathbf{Ax} = \mathbf{b}$ , ανεξάρτητα από τις υπόλοιπες. Σκοπός είναι, για κάθε εξίσωση  $i$ , να μηδενίσουμε το  $r_i = (\mathbf{b} - \mathbf{Ax}^{\text{old}})_i$  “πειράζοντας” μόνο την τιμή  $x_i$  και θεωρώντας τα υπόλοιπα στοιχεία του  $\mathbf{x}$  σταθερά:

$$x_i^{\text{new}} = \left( b_i - \sum_{j \neq i} a_{ij} x_j^{\text{old}} \right) / a_{ii}, \quad i = 1, 2, \dots, n. \quad (5)$$

Παρατηρήστε η μέθοδος Jacobi μεταχειρίζεται τις εξισώσεις ανεξάρτητα τη μια από την άλλη και επομένως η σειρά με την οποία τις εξετάζουμε δεν παίζει ρόλο. Αυτό ισοδυναμεί με το να πάρουμε σαν  $M$  στη σχέση (4) το διαγώνιο πίνακα του  $A$ .

Ένας απλοποιημένος κώδικας MATLAB για την μέθοδο Jacobi είναι ο εξής Παρατηρήστε ότι χρησιμοποιούμε το διάνυσμα  $\mathbf{x}$  για το  $\mathbf{x}^{\text{old}}$  και την αρχική εκτίμηση  $\mathbf{x}^{(0)}$  και το διάνυσμα  $\mathbf{xnew}$  για το  $\mathbf{x}^{\text{new}}$ . Ο αλγόριθμος δεν μπορεί να υλοποιηθεί με ένα μόνο διάνυσμα  $\mathbf{x}$  γιατί σε κάθε επανάληψη χρειαζόμαστε όλες τις τιμές του  $\mathbf{x}^{\text{old}}$  για να μπορέσουμε να υπολογίσουμε το  $\mathbf{x}^{\text{new}}$ .

## Η μέθοδος Gauß-Seidel

Στη μέθοδο Gauß-Seidel προχωράμε όπως και στη Jacobi, με τη διαφορά ότι οι εξισώσεις δεν είναι ανεξάρτητες αλλά εξετάζονται διαδοχικά και χρησιμοποιούμε άμεσα όσες από τις τιμές του  $\mathbf{x}^{\text{new}}$  έχουμε ήδη υπολογίσει. Ξαναγράφοντας την (5) σαν

$$x_i^{\text{new}} = \left( b_i - \sum_{j < i} a_{ij} x_j^{\text{old}} - \sum_{j > i} a_{ij} x_j^{\text{old}} \right) / a_{ii}, \quad i = 1, 2, \dots, n.$$

παρατηρούμε ότι αν εφαρμόζουμε την παραπάνω σχέση παίρνοντας με τη σειρά  $i = 1, 2, \dots, n$  τις εξισώσεις, τότε κατά την επεξεργασία της εξίσωσης  $i$  έχουμε ήδη υπολογίσει τις νέες τιμές  $x_k^{\text{new}}$ ,  $k = 1, \dots, i - 1$  και μπορούμε να τις χρησιμοποιήσουμε άμεσα. Η μέθοδος Gauß-Seidel επομένως υπολογίζει

$$x_i^{\text{new}} = \left( b_i - \sum_{j < i} a_{ij} x_j^{\text{new}} - \sum_{j > i} a_{ij} x_j^{\text{old}} \right) / a_{ii}, \quad i = 1, 2, \dots, n. \quad (6)$$

Αν γράψουμε τον πίνακα του συστήματος σαν

$$A = D - L - U,$$

όπου  $D$  ο διαγώνιος πίνακας,  $-L$  ο αυστηρά κάτω τριγωνικός και  $-U$  άνω τριγωνικός, τότε στην (4)  $M = D - L$ ,  $N = U$  και η μέθοδος Gauß-Seidel με μορφή πινάκων γράφεται

$$(D - L)\mathbf{x}^{\text{new}} = U\mathbf{x}^{\text{old}} + b.$$

Για τον υπολογισμό του  $\mathbf{x}^{\text{new}}$  σε κάθε επανάληψη χρειάζεται μια εμπρός-αντικατάσταση, αφού ο πίνακας  $D - L$  είναι άνω τριγωνικός.

Μια άλλη μορφή της Gauß-Seidel προκύπτει αν εξετάσουμε τις εξισώσεις με αντίστροφη σειρά ξεκινώντας από την  $n$ -οστή και πηγαίνοντας προς την πρώτη. Τότε η επανάληψη σε μορφή πινάκων γράφεται

$$(D - U)\mathbf{x}^{\text{new}} = L\mathbf{x}^{\text{old}} + b$$

και υλοποιείται με μια πίσω-αντικατάσταση σε κάθε επανάληψη.

Ένας απλοποιημένος κώδικας MATLAB για την μέθοδο Gauß-Seidel είναι ο εξής. Ο αλγόριθμος της μεθόδου Gauß-Seidel μπορεί να γραφεί σε πιο συμπαγή μορφή χρησιμοποιώντας μόνο το διάνυσμα  $\mathbf{x}$  για να αποθηκεύει και το  $\mathbf{x}^{\text{old}}$  και το  $\mathbf{x}^{\text{new}}$ . Στον παραπάνω αλγόριθμο το `xold` χρησιμοποιείται μόνο για τον υπολογισμό της `norm(xold-x)` και αυτό μπορεί να αποφευχθεί (βλ. Άσκηση 3).

## Οι μέθοδοι SOR και SSOR

Η επανάληψη Gauß-Seidel (6) μπορεί να γραφτεί σαν

$$x_i^{\text{new}} = x_i^{\text{old}} + \left( b_i - \sum_{j < i} a_{ij} x_j^{\text{new}} - \sum_{j \geq i} a_{ij} x_j^{\text{old}} \right) / a_{ii}, \quad i = 1, 2, \dots, n.$$

Αλγόριθμος 6: μέθοδος Gauß-Seidel

---

```

function [x, iter] = gs(A, b, x, epsilon, maxiter)
% A      - pinakas susthmatos
% b      - dexi melos
% x      - sthn eisodo h arqikh ektimhsh, sthn exodo h lush
% epsilon - zhtoumenh akribeia
% maxiter - megistos arijmos epanalhyewn
% iter   - epanalhyeis pou eginan
n = length(b); xold = x;
for iter = 1:maxiter
    for i = 1:n
        sum = b(i) - A(i, [1:i-1, i+1:n])*x([1:i-1, i+1:n]);
        x(i) = sum/A(i, i);
    end
    relerr = norm(xold-x)/norm(x);
    xold = x;
    if relerr < epsilon, break; end
end

```

---

Η μέθοδος της *SOR* (*Successive Over-Relaxation*) προκύπτει από τη Gauß-Seidel εισάγοντας μια παράμετρο χαλάρωσης  $\omega \geq 1$  με σκοπό την επιτάχυνση της σύγκλισης.

$$x_i^{\text{new}} = x_i^{\text{old}} + \omega \left( b_i - \sum_{j < i} a_{ij} x_j^{\text{new}} - \sum_{j \geq i} a_{ij} x_j^{\text{old}} \right) / a_{ii}, \quad i = 1, 2, \dots, n.$$

Αυτό ισοδυναμεί με την επίλυση του ισοδύναμου γραμμικού συστήματος

$$\omega A \mathbf{x} = \omega \mathbf{b}$$

με πίνακες διαχωρισμού τους

$$M = D - \omega L, \quad N = \omega U + (1 - \omega)D. \quad (7)$$

Η νέα επανάληψη επομένως σχετίζεται με την Gauß-Seidel ( $\mathbf{x}^{\text{GS}}$ ) και την τρέχουσα προσέγγιση  $\mathbf{x}^{\text{old}}$  ως εξής

$$\mathbf{x}^{\text{new}} = \omega \mathbf{x}^{\text{GS}} + (1 - \omega) \mathbf{x}^{\text{old}}.$$

Για  $\omega = 1$  παίρνουμε τη μέθοδο Gauß-Seidel.

Η *SOR*, όπως και η Gauß-Seidel, μπορεί να εφαρμοστεί ξεκινώντας από την τελευταία εξίσωση και πηγαίνοντας προς την πρώτη. Στην περίπτωση αυτή οι αντίστοιχοι πίνακες διαχωρισμού είναι οι

$$M = D - \omega U, \quad N = \omega L + (1 - \omega)D, \quad (8)$$

και αφού ο  $M$  είναι τώρα άνω-τριγωνικός, χρειάζεται να γίνει μια πίσω-αντικατάσταση σε κάθε επανάληψη.

Ο πίνακας επανάληψης

$$G = M^{-1}N$$



είναι συμμετρικός μόνο για τη μέθοδο Jacobi. Σε κάποιες περιπτώσεις είναι επιθυμητό να έχουμε συμμετρικούς πίνακες επανάληψης. Η συμμετρική μορφή της SOR, είναι γνωστή σαν SSOR (Symmetric Successive Over-Relaxation) και προκύπτει αν συνδυάσουμε τις δύο μορφές της SOR που περιγράφηκαν προηγουμένως. Δηλαδή, σε κάθε επανάληψη εκτελούμε ένα βήμα της SOR με τους πίνακες (7) και καταλήγουμε σε μια ενδιάμεση εκτίμηση  $\mathbf{x}^{\text{mid}}$

$$(D - \omega L)\mathbf{x}^{\text{mid}} = (\omega U + (1 - \omega)D)\mathbf{x}^{\text{old}} + \omega \mathbf{b}$$

και ένα δεύτερο βήμα με τους πίνακες (8) για να καταλήξουμε στο  $\mathbf{x}^{\text{new}}$

$$(D - \omega U)\mathbf{x}^{\text{new}} = (\omega L + (1 - \omega)D)\mathbf{x}^{\text{mid}} + \omega \mathbf{b}.$$

## Σύγκλιση

Η γενική μορφή επανάληψης των παραπάνω μεθόδων συνδέει την “παλιά” τιμή  $\mathbf{x}^{(k)}$  στην επανάληψη  $k$  με την “νέα” τιμή  $\mathbf{x}^{(k+1)}$  στην επανάληψη  $k + 1$

$$\mathbf{x}^{(k+1)} = G\mathbf{x}^{(k)} + \mathbf{c}, \quad (9)$$

όπου  $G = M^{-1}N$  και  $\mathbf{c} = M^{-1}\mathbf{b}$ . Ο  $G$  είναι γνωστός σαν πίνακας επανάληψης.

Ας εξετάσουμε τι συμβαίνει στην περίπτωση μιας εξίσωσης με ένα άγνωστο, που τα  $G$  και  $\mathbf{c}$  είναι βαθμωτά μεγέθη. Η λύση της εξίσωσης είναι η τομή των ευθειών  $y = x$  και  $y = Gx + \mathbf{c}$ . Όπως φαίνεται και από το Σχήμα 1, η επαναληπτική μέθοδος συγκλίνει μόνο στην περίπτωση που η απόλυτη τιμή της κλίσης  $G$  της ευθείας  $y = Gx + \mathbf{c}$  είναι μικρότερη από τη μονάδα. Η μέθοδος αποκλίνει για  $|G| > 1$ , ενώ κάνει κύκλους όταν  $|G| = 1$ .

Για τη σύγκλιση στη γενικότερη περίπτωση με περισσότερους από ένα αγνώστους παρατηρούμε κατ’ αρχήν ότι για το σφάλμα σε διαδοχικές επαναλήψεις ισχύει

$$\mathbf{e}^{(k+1)} = G\mathbf{e}^{(k)}$$

και κατ’ επέκταση το σφάλμα στην επανάληψη  $k$  συνδέεται με το αρχικό σφάλμα της αρχικής εκτίμησης  $\mathbf{x}^{(0)}$  με τη σχέση

$$\mathbf{e}^{(k)} = G^k \mathbf{e}^{(0)}. \quad (10)$$

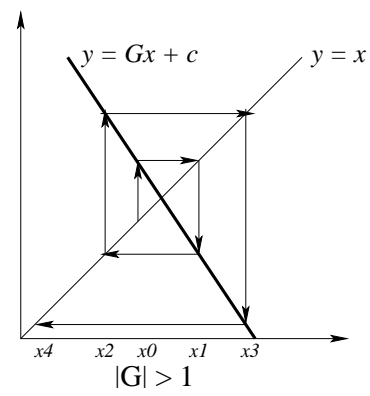
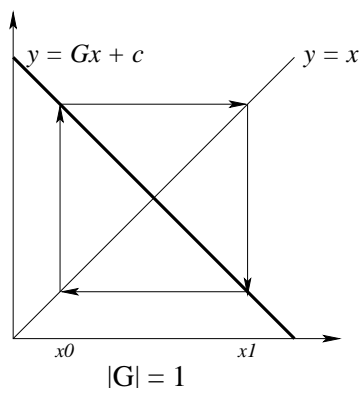
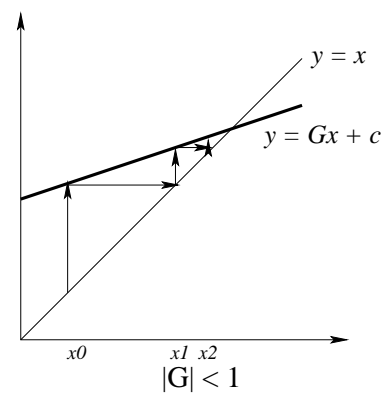
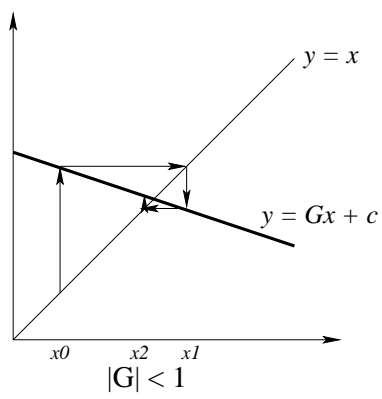
Χρησιμοποιώντας γνωστές ιδιότητες φράσσουμε το  $\|\mathbf{e}^{(k)}\|$

$$\|\mathbf{e}^{(k)}\| = \|G^k \mathbf{e}^{(0)}\| \leq \|G^k\| \|\mathbf{e}^{(0)}\| \leq \|G\|^k \|\mathbf{e}^{(0)}\|$$

και παρατηρούμε ότι για να συγκλίνει η επαναληπτική μέθοδος θα πρέπει  $\|G\| < 1$ , γιατί τότε  $\|G\| \rightarrow 0$ . Η συμπεριφορά αυτή είναι ανάλογη με αυτήν που παρατηρήσαμε στην περίπτωση μιας εξίσωσης με έναν άγνωστο, καθώς η φυσική επέκταση της απόλυτης τιμής σε περισσότερες διαστάσεις είναι η νόρμα.

Ένα πιο αυστηρό φράγμα για το σφάλμα βασίζεται στην φασματική ακτίνα. Χρησιμοποιώντας την (10), θα δείξουμε τώρα ότι αν  $G = V\Lambda V^{-1}$  όπου  $V$  τα ιδιοδιανύσματα και  $\Lambda$  ο διαγώνιος πίνακας ιδιοτιμών του  $G$  τότε

$$\|\mathbf{e}^{(k)}\| \leq \kappa(V)[\rho(G)]^k \|\mathbf{e}^{(0)}\|,$$



Σχήμα 1: Σύγκλιση σε μια διάσταση

όπου  $\kappa(V) = \|V\|\|V^{-1}\|$  ο δείκτης κατάστασης του  $V$  και  $\rho(G)$  η φασματική ακτίνα (η μέγιστη ιδιοτιμή) του  $G$ . Παρατηρούμε ότι

$$G^2 = GG = (V\Lambda V^{-1})(V\Lambda V^{-1}) = V\Lambda^2 V^{-1}$$

και γενικά

$$G^k = V\Lambda^k V^{-1}.$$

Παίρνοντας νόρμες έχουμε

$$\|G^k\| = \|V\Lambda^k V^{-1}\| \leq \|V\|\|V^{-1}\|\|\Lambda\|^k = \kappa(V)\|\Lambda\|^k.$$

Αλλά η νόρμα ενός διαγώνιου πίνακα, όπως ο  $\Lambda$  είναι το μέγιστο διαγώνιο στοιχείο του, άρα η μέγιστη ιδιοτιμή  $\rho(\Lambda)$ .

Με τις παραπάνω παρατηρήσεις μπορούμε να αναλύσουμε τη σύγκλιση των επαναληπτικών μεθόδων αυτής της κατηγορίας για διάφορες κλάσεις πινάκων.

1. Ένας πίνακας  $A$  είναι αυστηρά διαγώνια κυρίαρχος όταν

$$|a_{ii}| > \sum_{j \neq i} |a_{ij}|, \quad \forall i.$$

και τότε η Jacobi και η Gauß-Seidel συγκλίνουν ( $\|G\| < 1$  για κάποια νόρμα).

Έστω

$$r = \max_i \left\{ \sum_{j \neq i} \frac{|a_{ij}|}{|a_{ii}|} \right\} < 1$$

Για τη μέθοδο Jacobi

$$\|G\|_\infty = \max_i \left\{ |g_{ii}| + \sum_{j \neq i} |g_{ij}| \right\} = \max_i \left\{ 0 + \sum_{j \neq i} \frac{|a_{ij}|}{|a_{ii}|} \right\} = r < 1.$$

Για τη Gauß-Seidel, έστω  $\mathbf{x}$  ένα οποιοδήποτε μη-μηδενικό διάνυσμα,  $\mathbf{y}$  ένα διάνυσμα που ικανοποιεί τη σχέση  $M\mathbf{y} = N\mathbf{x}$  για τα  $M, N$  της μεθόδου, δηλαδή  $\mathbf{y} = G\mathbf{x}$ . Τότε θα δείξουμε ότι

$$|y_i| \leq r\|\mathbf{x}\|_\infty, \quad \text{για } i = 1, 2, \dots, n. \quad (11)$$

Για  $i = 1$ ,

$$a_{11}y_1 = -(a_{12}x_2 + a_{13}x_3 + \dots + a_{1n}x_n),$$

οπότε

$$\begin{aligned} |a_{11}| |y_1| &\leq |a_{12}| |x_2| + |a_{13}| |x_3| + \dots + |a_{1n}| |x_n| \leq |a_{12}| \|\mathbf{x}\|_\infty + |a_{13}| \|\mathbf{x}\|_\infty + \dots + |a_{1n}| \|\mathbf{x}\|_\infty \\ &\Rightarrow |y_1| \leq \|\mathbf{x}\|_\infty \sum_{j \neq 1} \frac{|a_{1j}|}{|a_{11}|} \Rightarrow |y_1| \leq r\|\mathbf{x}\|_\infty, \end{aligned}$$

γιατί εξ' ορισμού  $\|\mathbf{x}\|_\infty = \max_i |x_i|$ .

Για  $i = 2$ ,

$$a_{21}y_1 + a_{22}y_2 = -(a_{23}x_3 + \dots + a_{2n}x_n) \Rightarrow a_{22}y_2 = -(a_{21}y_1 + a_{23}x_3 + \dots + a_{2n}x_n),$$

επομένως

$$|a_{22}| |y_2| \leq |a_{21}| |y_1| + |a_{23}| |x_3| + \dots + |a_{2n}| |x_n| \leq |a_{21}| r \|\mathbf{x}\|_\infty + |a_{23}| \|\mathbf{x}\|_\infty + \dots + |a_{2n}| \|\mathbf{x}\|_\infty$$

$$\stackrel{r \leq 1}{\implies} |y_2| \leq \|\mathbf{x}\|_\infty \sum_{j \neq 2} \frac{|a_{2j}|}{|a_{22}|} \implies |y_2| \leq r \|\mathbf{x}\|_\infty,$$

και επαγωγικά καταλήγουμε στην (11). Επομένως

$$\max_i y_i = \|\mathbf{y}\|_\infty \leq r \|\mathbf{x}\|_\infty \implies \|G\mathbf{x}\|_\infty \leq r \|\mathbf{x}\|_\infty \implies \frac{\|G\mathbf{x}\|_\infty}{\|\mathbf{x}\|_\infty} \leq r \implies \|G\|_\infty < 1.$$

2. Αν  $\omega \notin [0, 2]$ , η SOR αποκλίνει.
3. Αν ο  $A$  είναι συμμετρικός και θετικά ορισμένος και  $\omega \in [0, 2]$ , η SOR συγκλίνει πάντα.

## Ασκήσεις

1. Να γράψετε αλγόριθμους πολλαπλασιασμού συμμετρικού αραιού πίνακα με διάλυση, όταν είναι αποθηκευμένο μόνο το άνω τριγωνικό μέρος του (i) στο σύστημα συντεταγμένων και (ii) σε CRS.
2. Το σύστημα αποθήκευσης Compressed Column Storage (CCS) είναι αντίστοιχο του CRS με τη διαφορά ότι τα μη-μηδενικά στοιχεία αποθηκεύονται ανά στήλες. Στην περίπτωση αυτή αντί του col υπάρχει το διάνυσμα row, και το διάνυσμα start καθορίζει τη θέση έναρξης κάθε στήλης. Να γράψετε τους αλγόριθμους matvec, forsub και baksub όταν οι πίνακες τους είναι αποθηκευμένοι σε CCS.
3. Υλοποιήστε τον αλγόριθμο Gauß-Seidel έτσι ώστε να χρησιμοποιεί ένα μόνο διάνυσμα  $\mathbf{x}$  για να αποθηκεύει και το  $\mathbf{x}^{\text{old}}$  και το  $\mathbf{x}^{\text{new}}$ . Υπόδειξη: βρείτε ένα τρόπο να υπολογίζετε τη `norm(xold-x)` ενόσω ενημερώνεται το  $\mathbf{x}$ .
4. Δείξτε ότι ο πίνακας επανάληψης της SSOR είναι συμμετρικός.
5. Έστω  $T$  ένας άνω τριγωνικός πίνακας με μηδενική διαγώνιο. Δείξτε ότι η επαναληπτική μέθοδος  $\mathbf{x}^{\text{new}} = T\mathbf{x}^{\text{old}} + \mathbf{c}$  συγκλίνει.
6. Έστω  $T$  ένας άνω τριγωνικός πίνακας με  $t_{ii} = i/(i+1)$ . Να εξετάσετε αν συγκλίνει η επαναληπτική μέθοδος  $\mathbf{x}^{\text{new}} = T\mathbf{x}^{\text{old}} + \mathbf{c}$ .
7. Δείξτε ότι η επαναληπτική μέθοδος με πίνακα επανάληψης  $G = [0.9990.998; 00]$  συγκλίνει πάντα.
8. Για ποιες τιμές του  $\tau$  συγκλίνει η επαναληπτική μέθοδος για το γραμμικό σύστημα

$$0.9x_1 + x_2 = 1$$

$$1.5x_1 + \tau x_2 = 4.$$

9. Έστω το γραμμικό σύστημα

$$\begin{aligned}x_1 + x_2 &= 1 \\x_1 + 6x_2 + 3x_3 &= 3 \\x_2 + 3x_3 &= 5.\end{aligned}$$

Εξετάστε αν συγκλίνουν οι μέθοδοι Jacobi και Gauß-Seidel υπολογίζοντας τις αντίστοιχες φασματικές ακτίνες. Τι μπορείτε να πείτε για τις ταχύτητες σύγκλισης?

10. Έστω το γραμμικό σύστημα

$$\begin{aligned}5x_1 - x_2 + x_3 &= 10 \\2x_1 + 8x_2 - x_3 &= 11 \\-x_1 + x_2 + 4x_3 &= 3.\end{aligned}$$

(i) Να εκτελέσετε 3 επαναλήψεις της μεθόδου Jacobi, με αρχική εκτίμηση  $\mathbf{x}^{(0)} = \mathbf{0}$  και να υπολογίσετε τις αντίστοιχες προσεγγίσεις στο  $\mathbf{x}$ . Συγκλίνει η επανάληψη στη λύση? (ii) Να εκτελέσετε 3 επαναλήψεις της μεθόδου Gauß-Seidel, με αρχική εκτίμηση  $\mathbf{x}^{(0)} = \mathbf{0}$  και να υπολογίσετε τις αντίστοιχες προσεγγίσεις στο  $\mathbf{x}$ . Συγκλίνει η επανάληψη στη λύση? (iii) Βρείτε εκτιμήσεις σφαλμάτων για τα  $\mathbf{x}^{(3)}$  και των δύο μεθόδων χρησιμοποιώντας τη νόρμα απείρου.