Προσεγγιστική λύση Γραμμικών Συστημάτων με την μέθοδο Gauss-Seidel

Δημιουργία κώδικα στο Matlab

Χατζηγεωργίου Αντώνης
Νοέμβριος 2013

Περιεχόμενα

1. Αρχικό πρόβλημα.

Στο παρών έγγραφο, για την ανάπτυξη-παρουσίαση του κώδικα που επιλύει προσεγγιστικά, γραμμικά προβλήματα με την μέθοδο των Gauss-Seidel, θα χρησιμοποιηθεί το παρακάτω σύστημα:

$$2x_1 - x_2 + 0x_3 = 2$$
$$x_1 - 3x_2 + x_3 = -2$$
$$-x_1 + x_2 - 3x_3 = -6$$

Ουσιαστικά αυτό το σύστημα αποτελείτε από τον πίνακα των συντελεστών των αγνώστων(A), τον πίνακα των αγνώστων(x_i), και τον πίνακα των αποτελεσμάτων(B).

A =

2	-1	0
1	-3	1
-1	1	-3

x_i =

x_1

x_2

x_3

B =

2
-2
-6

2. Εφαρμογή της θεωρίας.

Η γενική μορφή ενός γραμμικού συστήματος, με την βοήθεια των πινάκων είναι:

$$\begin{bmatrix} \alpha_{11} & \alpha_{12} & \cdots & \alpha_{1v} \\ \alpha_{21} & \alpha_{22} & \cdots & \alpha_{2v} \\ \vdots & \vdots & & \vdots \\ \alpha_{\mu 1} & \alpha_{\mu 2} & \cdots & \alpha_{\mu v} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_v \end{bmatrix} = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_\mu \end{bmatrix}.$$

Δηλαδή

$$Ax = B$$

Υπάρχουν διάφοροι μέθοδοι επίλυσης αυτών των συστημάτων. Η μέθοδος των Gauss-Seidel που χρησιμοποιώ εγώ, είναι μια επαναληπτική μέθοδος. Απαιτεί μια αρχική τιμή $x^{(0)}$ και στη συνέχεια, τη δημιουργία μιας ακολουθίας διανυσμάτων $x^{(0)}, x^{(1)}, x^{(2)}$,..... ώστε τελικά να γίνει σύγκλιση στο x .

Ο αλγόριθμος που περιγράφει την διαδικασία είναι:

$$x_i = \left(-\sum_{j=1}^{i-1} a_{ij} x_j \; - \sum_{j=1}^{n} a_{ij} x_j^0 + b_i \right) \Big/ a_{ii} \qquad (1)$$

Ο παραπάνω αλγόριθμος, είναι μια εξέλιξη της μεθόδου του Jacobi. Περιγράφει, πως από το πρώτο βήμα της εφαρμογή της μεθόδου, κάθε άγνωστος που υπολογίζεται, μπαίνει στο αμέσως επόμενο βήμα υπολογισμού, του επόμενου αγνώστου.

Εκτός από τον παραπάνω αλγόριθμο που μπορεί να χρησιμοποιηθεί άμεσα, ο πίνακας A, μπορεί να γραφτεί και ως:

$$A = D + L + U$$

Όπου στην περίπτωση του συγκεκριμένου προβλήματος:

```
D =

     2     0     0
     0    -3     0
     0     0    -3
```

```
L =

     0     0     0
     1     0     0
    -1     1     0

U =

     0    -1     0
     0     0     1
     0     0     0
```

Συνεπώς, το $Ax = B$ μπορεί να γραφτεί ως $(D + L + U)x = B$

άρα

$$(D + L)x = -Ux + B \;\; =>$$

$$x = -(D + L)^{-1} * Ux + (D + L)^{-1} * B \qquad (2)$$

Έτσι λοιπόν, έχοντας τον τύπο (2) στο μυαλό μας, προκύπτει η παρακάτω επαναληπτική σχέση:

$$x^{(i+1)} = -(D + L)^{-1} * Ux^{(i)} + (D + L)^{-1} + B \qquad (3)$$

όπου $i = 0\,,1\,,2$ ....

3. Ανάπτυξη κώδικα στο Matlab.

Κατά την δημιουργία του κώδικα, αυτό που έπρεπε να κάνω, είναι να φτιάξω μια Function, όπου εισήγαγα μέσα:

- Το αρχικό σύστημα, με όλους τους πίνακες που αναλύουν τον πίνακα των συντελεστών των αγνώστων.
- Επίσης, ένα πίνακα με τις αρχικές τιμές

$$x0 = [0; 0; 0]$$

- Τον μέγιστο αριθμό των επαναλήψεων που θέλω να κάνει το πρόγραμμα.
- Τον τύπο (3) μέσα σε μία επαναλαμβανόμενη διαδικασία.
- Την επιθυμητή ακρίβεια μέχρι την οποία θέλω να φτάσει το πρόγραμμα.

Έχοντας όλα τα παραπάνω, δημιουργήθηκε ο παρακάτω κώδικας

```
 function Gauss_Seidel_method(A,B,D,L,U)
A=input('Matrix A=');
B=input('Matrix B=');
D=diag(diag(A));
L=tril(A-diag(diag(A)));
U=triu(A-diag(diag(A)));
x=zeros(length(B),1);
N=input('Maximum number of iterations=');
e=input('Accuracy=');
for k = 1:N
    xold = x;
     x=-((D+L)^-1)*U*x+((D+L)^-1)*B;
   if norm( x - xold ) < e
       break;
   end
   format long
end
k
x
```

3.1 Επεξήγηση του κώδικα.

Είναι φανερό, ότι οι 2 πρώτες γραμμές, αναφέρονται στο να εισάγει ο χρήστης τους πίνακες A και B.

Με τις εντολές δεξιά στα D , L , και U είναι η διάσπαση του πίνακα A, όπως προανέφερα παραπάνω. Δηλαδή το D είναι η διαγώνιος, το L είναι τα στοιχεία κάτω από την διαγώνιο(η εντολή `tril` , είναι ουσιαστικά `tri` (lower)) , και το U είναι τα στοιχεία πάνω από την διαγώνιο(κι εδώ αντίστοιχα η εντολή `triu` είναι ουσιαστικά `tri(upper))`.

Η αμέσως επόμενη εντολή είναι η : `x=zeros(length(B),1);`
Αυτό που λέω στο πρόγραμμα, είναι να φτιάξει έναν πίνακα, ίδιου μεγέθους με αυτόν των αποτελεσμάτων (B). Αυτός ο πίνακας θα είναι μηδενικός, και θα εκφράζει τις αρχικές τιμές.

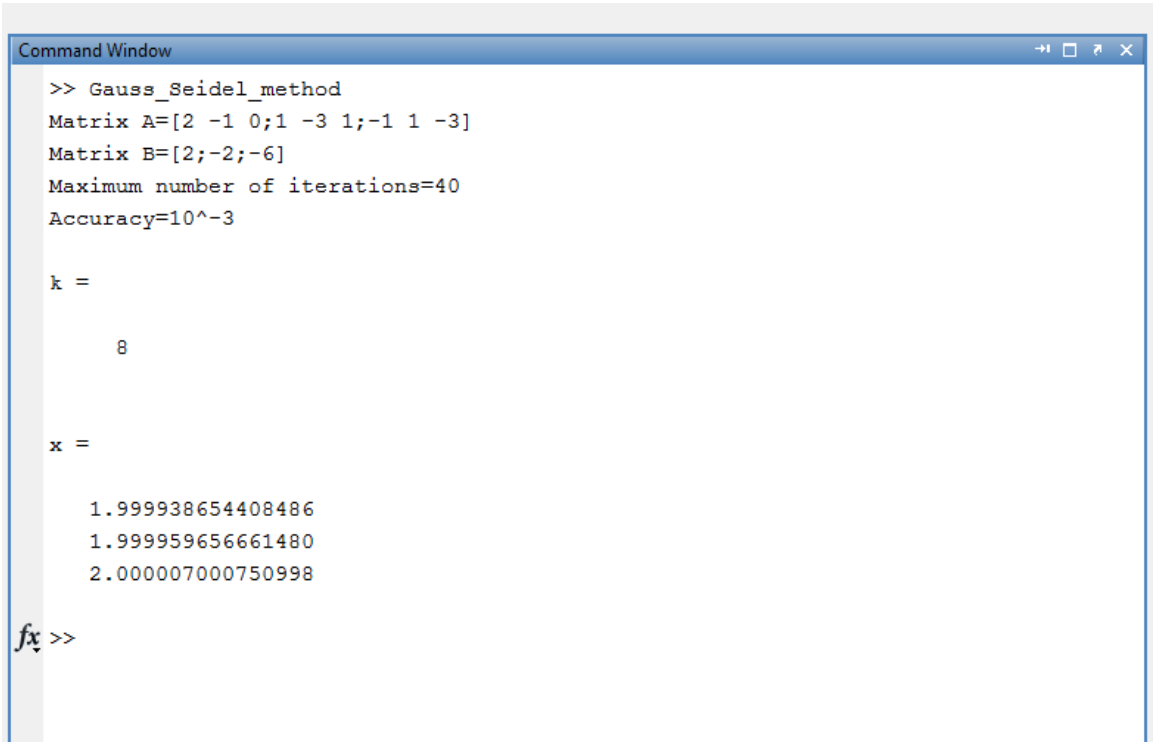
Στην συνέχεια, όρισα στο πρόγραμμα να ρωτάει τον χρήστη τον μέγιστο αριθμό των επαναλήψεων και την μέγιστη ακρίβεια που επιθυμεί.

Το μόνο που χρειάζεται πλέον, είναι να διατυπωθεί ο γενικός τύπος, μέσα σε μία επαναληπτική διαδικασία. Σε μία λούπα. Αυτό γίνεται με ένα For το οποίο θα ξεκινάει από την πρώτη επανάληψη, έως τις μέγιστες επαναλήψεις που του έχουμε ορίσει.

Το πρόγραμμα ολοκληρώνεται, με την έκφραση της ακρίβειας. Δηλαδή, μπορεί εμείς να έχουμε πει μια συγκεκριμένη ακρίβεια, αλλά το πρόγραμμα δεν καταλαβαίνει σε ποια επανάληψη θα φτάσει αυτή την ακρίβεια. Αυτό γίνεται με μια norm μέσα σε ένα IF . Η συνθήκη είναι, πως μόλις η διαφορά, των τελικών τιμών των X από τις αρχικές μηδενικές τιμές `( x - xold )` γίνει μικρότερη από την ακρίβεια που έχουμε ορίσει, τότε το πρόγραμμα να σταματήσει(break), και να βγει από το For. Να σταματήσει δηλαδή τις πράξεις.

Στο τέλος, γράφοντας X και K , χωρίς ερωτηματικό δίπλα, του λέω να μου δώσει τα αποτελέσματα των X, καθώς επίσης και τις επαναλήψεις που χρειάστηκε να κάνει, μέχρι την ακρίβεια που του έχουμε ορίσει.

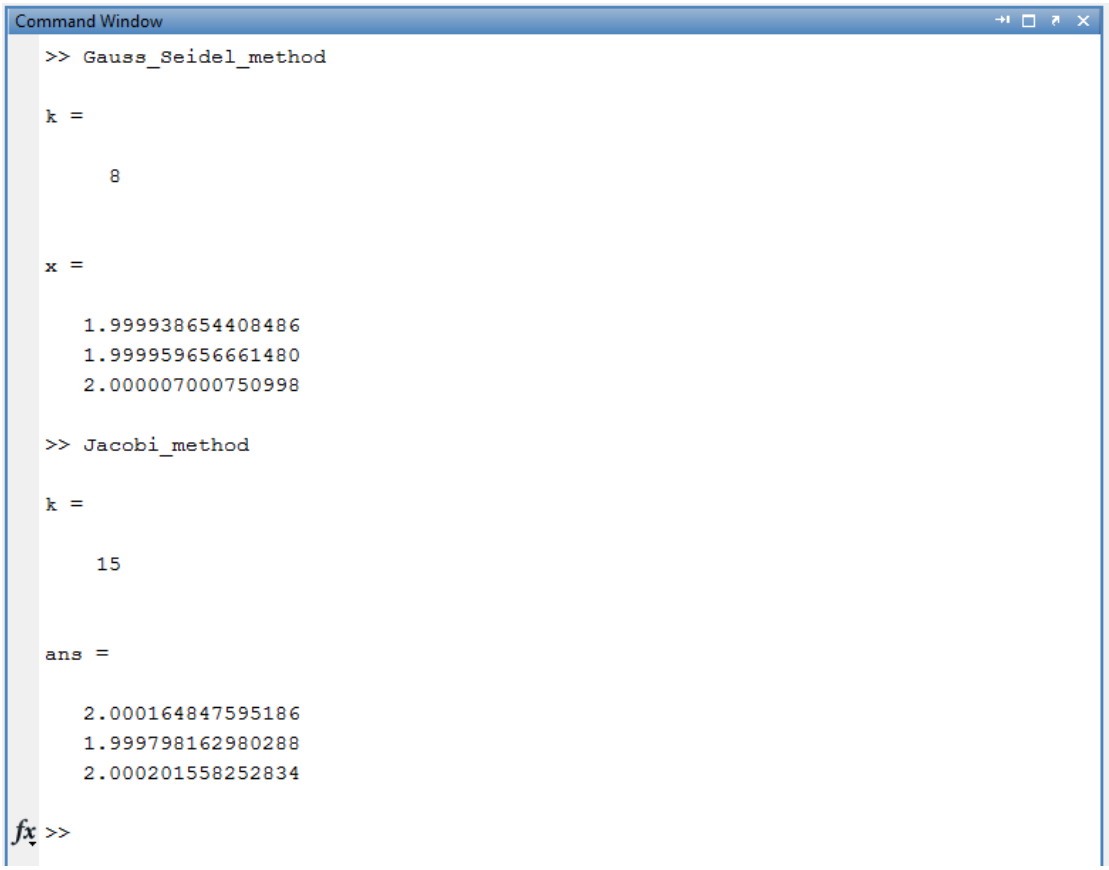
Τρέχοντας την function, εμφανίζονται στο Command Window τα παρακάτω:



Για το συγκεκριμένο παράδειγμα, η μέθοδος αυτή χρειάστηκε 8 επαναλήψεις για να φτάσει στην ακρίβεια του 10^{-3} . Τα αποτελέσματα συγκεντρωτικά φαίνονται και στον παρακάτω πίνακα:

X_1	**1.99993865**
X_2	**1.99995966**
X_3	**2.00000700**
Επαναλήψεις	**8**

Στην παρακάτω εικόνα, εμφανίζονται τα αποτελέσματα των τιμών των Χ, έχοντας χρησιμοποιήσει αντίστοιχη Function με την μέθοδο του Jacobi(το συγκεκριμένο πρόγραμμα δεν θα αναπτυχθεί στα παρών έγγραφο), χάριν της σύγκρισης την ταχύτητας των δυο μεθόδων.



Έτσι λοιπόν, έχοντας τρέξει και τα δυο προγράμματα ταυτόχρονα, παρατηρούμε πως η μέθοδος των Gauss –Seidel είναι κατά πολύ γρηγορότερη από αυτή του Jacobi. Παρακάτω, παρατίθεται συγκεντρωτικός πίνακας.

	Μέθοδος Jacobi	**Μέθοδος Gauss-Seidel**
X_1	**2.00016**	**1.99994**
X_2	**1.99980**	**1.99996**
X_3	**2.00020**	**2.00001**
Επαναλήψεις	**15**	**8**