

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ, ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ ΚΑΙ ΔΙΚΤΥΩΝ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
ΗΥ200: ΕΠΙΣΤΗΜΟΝΙΚΟΣ ΥΠΟΛΟΓΙΣΜΟΣ

ΕΡΓΑΣΙΑ 1: Αριθμητική Γραμμική Άλγεβρα.

ΑΣΚΗΣΗ 1. α) Καλέστε την `lu_run` για να κάνετε παραγοντοποίηση με την `lu`, και την `\` για να λύσετε τα τριγωνικά συστήματα που προκύπτουν, αν το αρχικό προέρχεται από $n_x = 3, n_y = 10, c_1 = 0, c_2 = 0, c = 0$.

β) Όμοια για $n_x = 10, n_y = 3, c_1 = 0, c_2 = 0, c = 0$.

γ) Όμοια για $n_x = 5, n_y = 5, c_1 = 0, c_2 = 0, c = 0$.

δ) Όμοια για $n_x = 5, n_y = 5, c_1 = 100, c_2 = 0, c = 0$.

ε) Όμοια για $n_x = 5, n_y = 5, c_1 = 0, c_2 = 100, c = 0$.

στ) Όμοια για $n_x = 5, n_y = 5, c_1 = 0, c_2 = 0, c = 100$.

Τι παρατηρείτε ως προς την ακρίβεια της λύσης σας;

ΑΣΚΗΣΗ 2. α) Όμοια για $n_x = 10, n_y = 10, c_1 = 0, c_2 = 0$ και $c = 0$.

β) Όμοια για $n_x = 20, n_y = 20, c_1 = 0, c_2 = 0, c = 0$ και $c = 100$.

γ) Όμοια για $n_x = 30, n_y = 30, c_1 = 0, c_2 = 0, c = 0$ και $c = 100$.

δ) Όμοια για $n_x = 40, n_y = 40, c_1 = 0, c_2 = 0, c = 0$ και $c = 100$.

ε) Όμοια για $n_x = 50, n_y = 50, c_1 = 0, c_2 = 0, c = 0$ και $c = 100$.

Τι παρατηρείτε ως προς την ακρίβεια της λύσης σας και ως προς το χρόνο εκτέλεσης;

ΑΣΚΗΣΗ 3. Δοκιμάστε να λύσετε τα τελευταία 5 συστήματα, της άσκησης 2, χρησιμοποιώντας την βασική ιδιότητα του A , ότι είναι αραιός. Μελετήστε την συνάρτηση `sparse` του Matlab και χρησιμοποιείστε την για μετατρέψετε και να αποθηκεύσετε τον A σε αραιή μορφή. Επαναλάβετε τις εντολές για τη λύση του συστήματος, μόνο που τώρα θα χρησιμοποιήσετε την `luinc` αντί της `lu` μέσα από την `luinc_run`. Τι παρατηρείτε ως προς το χρόνο εκτέλεσης;

Λαμβάνοντας υπόψη σας την διάσπαση του πίνακα A σε $A = D - L - U$ και τις επαναληπτικές μεθόδους Jacobi και Gauss-Seidel για την επίλυση γραμμικών συστημάτων προχωρείστε στις δύο τελευταίες ασκήσεις, έχοντας σαν αρχικό διάνυσμα το $x_0 = 0$, $\epsilon = .5e - 5$ και $maxiter = n_x * n_y$. Οι αλγόριθμοι περιγράφονται στον παρακάτω πίνακα.

Σε μορφή εξισώσεων	Σε μορφή πινάκων
Επαναληπτική μέθοδο Jacobi	
<pre> for k = 1, ..., maxiter $x_i^{(k+1)} = \frac{b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k)} - \sum_{j=i+1}^n a_{ij} x_j^{(k)}}{a_{ii}}, \quad i = 1, \dots, n$ if ($\ x^{(k+1)} - x^{(k)}\ _2 < \epsilon$) stop end </pre>	<pre> for k = 1, ..., maxiter $Dx^{(k+1)} = (L + U)x^{(k)} + b$ if ($\ x^{(k+1)} - x^{(k)}\ _2 < \epsilon$) stop end </pre>
Επαναληπτική μέθοδο Gauss-Seidel	
<pre> for k = 1, ..., maxiter $x_i^{(k+1)} = \frac{b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^n a_{ij} x_j^{(k)}}{a_{ii}}, \quad i = 1, \dots, n$ if ($\ x^{(k+1)} - x^{(k)}\ _2 < \epsilon$) stop end </pre>	<pre> for k = 1, ..., maxiter $(D - L)x^{(k+1)} = Ux^{(k)} + b$ if ($\ x^{(k+1)} - x^{(k)}\ _2 < \epsilon$) stop end </pre>

ΑΣΚΗΣΗ 4. Χρησιμοποιώντας πάλι την ιδιότητα του A , ότι είναι αραιός, καλέστε τη `jacobi`, η οποία θα υλοποιεί την επαναληπτική μέθοδο Jacobi, μέσα από την `jacobi-run`.

Δοκιμάστε να λύσετε τα τελευταία 5 συστήματα από την άσκηση 2. Τι παρατηρείτε ως προς την ακρίβεια της λύσης σας και ως προς το χρόνο εκτέλεσης και πλήθος επαναλήψεων;

ΑΣΚΗΣΗ 5. Τροποποιήστε την `jacobi.m` ώστε να υλοποιήσετε την επαναληπτική μέθοδο Gauss-Seidel μέσα στην `gs.m` την οποία θα καλέσετε κλασικά μέσα από την `gs-run`.

Δοκιμάστε να λύσετε τα ίδια τελευταία 5 συστήματα. Τι παρατηρείτε ως προς την ακρίβεια της λύσης σας και ως προς το χρόνο εκτέλεσης και πλήθος επαναλήψεων;

ΑΣΚΗΣΗ 6. Καλέστε την `cg.m` μέσα από την `cg-run` για να χρησιμοποιήσετε την μέθοδο Συζηγών Κλίσεων(Conjugate Gradients) στην επίλυση των γραμμικών συστημάτων.

Δοκιμάστε να λύσετε τα ίδια τελευταία 5 συστήματα. Τι παρατηρείτε ως προς την ακρίβεια της λύσης σας και ως προς το χρόνο εκτέλεσης και πλήθος επαναλήψεων;

Υπενθυμίζουμε ότι η μέθοδος των Συζηγών Κλίσεων(Conjugate Gradients) μπορεί να δοθεί από τον αλγόριθμο:

Αρχικοποίηση: $x_0 = 0 \Rightarrow r_0 = b - Ax_0$ Λύσε: $D\tilde{r}_0 = r_0 \Rightarrow p_0 = \tilde{r}_0$ <i>for</i> $k = 0, \dots$ $\alpha_k = \frac{(\tilde{r}_k, r_k)}{(p_k, Ap_k)}$ $x_{k+1} = x_k + \alpha_k p_k$ $r_{k+1} = r_k - \alpha_k Ap_k$ $D\tilde{r}_{k+1} = r_{k+1}$ $if(r_{k+1}, r_{k+1}) \leq \epsilon$ $return$ $\beta_k = \frac{(\tilde{r}_{k+1}, r_{k+1})}{(\tilde{r}_k, r_k)}$ $p_{k+1} = \tilde{r}_{k+1} + \beta_k p_k$ <i>end</i>	}	Μέθοδος Συζηγών Κλίσεων.
---	---	---------------------------------

Γενική υπόδειξη: Για να μελετήσετε τα αποτελέσματά σας και να γράψετε τις παρατηρήσεις σας, συγκεντρώστε τα στους παρακάτω πίνακες:

n_x	n_y	c_1	c_2	c	χρόνος	σφάλμα
3	10	0	0	0		
10	3	0	0	0		
5	5	0	0	0		
5	5	100	0	0		
5	5	0	100	0		
5	5	0	0	100		
10	10	0	0	0		
20	20	0	0	0		
30	30	0	0	0		
40	40	0	0	0		
50	50	0	0	0		
10	10	0	0	100		
20	20	0	0	100		
30	30	0	0	100		
40	40	0	0	100		
50	50	0	0	100		

Πίνακας 1. Αποτελέσματα με χρήση της `lu`.

	$n_x = n_y$	c	χρόνος	επαναλήψεις	σφάλμα	c	χρόνος	επαναλήψεις	σφάλμα
<i>luinc</i>	10	0				100			
	20	0				100			
	30	0				100			
	40	0				100			
	50	0				100			
<i>Jacobi</i>	10	0				100			
	20	0				100			
	30	0				100			
	40	0				100			
	50	0				100			
<i>G - S</i>	10	0				100			
	20	0				100			
	30	0				100			
	40	0				100			
	50	0				100			
<i>CG</i>	10	0				100			
	20	0				100			
	30	0				100			
	40	0				100			
	50	0				100			

Πίνακας 2. Αποτελέσματα με χρήση λογισμικού για αραιούς πίνακες και επαναληπτικές μεθόδους.

Καλή επιτυχία.