

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ, ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ ΚΑΙ ΔΙΚΤΥΩΝ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
ΗΥ200: ΕΠΙΣΤΗΜΟΝΙΚΟΣ ΥΠΟΛΟΓΙΣΜΟΣ
ΣΤΕΡΓΙΟΣ ΠΟΥΛΑΡΑΚΗΣ ΑΕΜ:287

ΕΡΓΑΣΙΑ 1: Αριθμητική Γραμμική Άλγεβρα.

(Ημερομηνία Παράδοσης: Κυριακή 10 Απριλίου 2005, (Ώρα: 23:55))

(Η παράδοση της εργασίας θα γίνει μέσα από την σελίδα του μαθήματος στο eClass.)

ΠΙΝΑΚΑΣ ΑΣΚΗΣΕΩΝ 1,2

n_x	n_y	c_1	c_2	c	χρόνος	σφάλμα
3	10	0	0	0	0.015	$8.0005e - 015$
10	3	0	0	0	0.027	$1.6852e - 014$
5	5	0	0	0	0	$3.7321e - 015$
5	5	100	0	0	0.030	$2.6899e - 015$
5	5	0	100	0	0.067	$3.2501e - 015$
5	5	0	0	100	0	$6.3273e - 015$
10	10	0	0	0	0.021	$2.6232e - 014$
20	20	0	0	0	0.30	$2.599e - 013$
30	30	0	0	0	0.8124	$1.0038e - 012$
40	40	0	0	0	3.704	$2.6837e - 012$
50	50	0	0	0	15.102	$7.0978e - 012$
10	10	0	0	100	0.003	$4.9728e - 014$
20	20	0	0	100	0.213	$2.5738e - 013$
30	30	0	0	100	0.721	$1.0141e - 012$
40	40	0	0	100	3.497	$2.8267e - 012$
50	50	0	0	100	12.68	$7.4974e - 012$

Πίνακας 1. Αποτελέσματα με χρήση της lu.

ΠΑΡΑΤΗΡΗΣΕΙΣ - ΣΧΟΛΙΑ:

Άσκηση 1

Το σφάλμα είναι αρκετά μικρό καθώς είναι της τάξης του $e-015$, και επομένως η λύση είναι αρκετά ακριβής. Αυτό οφείλεται στο γεγονός πως η lu είναι άμεση μέθοδος και όχι επαναληπτική, πράγμα που σημαίνει πως όποιο σφάλμα προκύπτει οφείλεται μόνο στην πεπερασμένη αριθμητική του υπολογιστικού συστήματος και έτσι είναι μικρότερο συγκριτικά με το σφάλμα των επαναληπτικών μεθόδων.

Άσκηση 2

Παρατηρούμε ότι καθώς αυξάνονται τα n_x και n_y , αυξάνεται και ο χρόνος εκτέλεσης, επειδή η `lu-run` καλεί την `elliptic`. Η `elliptic` περιέχει βρόγχους που εξαρτώνται από το n_x και το n_y , οπότε όσο αυξάνονται οι τιμές τους, τόσο αυξάνεται και το πλήθος των επαναλήψεων που εκτελούνται για κάθε βρόγχο, άρα και ο χρόνος εκτέλεσης των βρόγχων. Βλέπουμε, ότι όσο αυξάνονται τα n_x και n_y το σφάλμα αυξάνεται, άρα μειώνεται η ακρίβεια της λύσης. Αυτό ισχύει και για $c=0$ και για $c=100$, επειδή η παράμετρος c επηρεάζει μόνο την γραφική παράσταση.

Άσκηση 3

Η `luinc run` καλεί την συνάρτηση `sparse` η οποία μας δίνει τα στοιχεία του αραιού πίνακα a και την θέση τους στον αραιό πίνακα. Έτσι, η `lu run` χρησιμοποιεί τον αραιό πίνακα και εξετάζει όλα τα στοιχεία του ενώ η `luinc run` εξετάζει μόνο τα μη μηδενικά στοιχεία του αραιού πίνακα. Επομένως ο χρόνος εκτέλεσης της `luinc run` θα είναι μικρότερος από τον χρόνο εκτέλεσης της `lu run`.

Άσκηση 4

Παρατηρούμε πως όταν $c=0$ γίνονται όλες οι επαναλήψεις αφού το σφάλμα δεν γίνεται ποτέ μικρότερο από το $\text{eps}=0.5e-5$. Αντίθετα, όταν $c=100$ το σφάλμα είναι μικρότερο του eps και επομένως δεν γίνονται όλες οι επαναλήψεις. Αυτό σημαίνει πως ο βρόγχος `while(iter<nx*ny error>eps)` διακόπτεται σε κάποιο σημείο επειδή $\text{error}<\text{eps}$. Επομένως για $c=0$ το σφάλμα είναι μεγαλύτερο απ' ό,τι για $c=100$ και άρα η ακρίβεια της λύσης είναι καλύτερη για $c=100$. Ακόμη από τον πίνακα φαίνεται ότι για $c=100$ ο χρόνος εκτέλεσης είναι μικρότερος σε σχέση με τον χρόνο εκτέλεσης για $c=0$.

Άσκηση 5

Παρατηρούμε ότι και για $c=0$ και για $c=100$ δεν γίνονται όλες οι επαναλήψεις επειδή το σφάλμα γίνεται μικρότερο του eps . Μόνο στην περίπτωση που έχουμε $n_x=n_y=10$ και $c=0$ γίνονται όλες οι επαναλήψεις. Ακόμη παρατηρούμε ότι είναι πιο ακριβής από την `Jacobi` αφού το αποτέλεσμα κάθε επανάληψης `xnew` εξαρτάται από όλα τα προηγούμενα `xold` ενώ το αποτέλεσμα `xnew` της `Jacobi` εξαρτάται μόνο από το προηγούμενο `xold`. Επομένως ο χρόνος εκτέλεσης της είναι μεγαλύτερος από τον χρόνο εκτέλεσης της `Jacobi`.

Άσκηση 6

Παρατηρούμε ότι η `CG` κάνει λιγότερες επαναλήψεις από τις δύο προηγούμενες μεθόδους και επομένως το σφάλμα της είναι μικρότερο από το σφάλμα των άλλων δύο. Επομένως, η λύση της είναι πιο ακριβής. Επίσης, ο χρόνος εκτέλεσης της είναι πολύ πιο μικρός από το χρόνο εκτέλεσης των προηγούμενων δύο μεθόδων.

ΠΙΝΑΚΑΣ ΑΣΚΗΣΕΩΝ 3,4,5

	$n_x = n_y$	c	χρόνος	επαναλήψεις	σφάλμα	c	χρόνος	επαναλήψεις	σφάλμα
<i>luinc</i>	10	0	0.189		$2.6472e - 014$	100	0		$4.6573e - 014$
	20	0	0		$2.8339e - 013$	100	0.021		$2.7947e - 013$
	30	0	0.27		$1.0538e - 012$	100	0.042		$1.1591e - 012$
	40	0	0.19		$2.851e - 012$	100	0.083		$3.0609e - 012$
	50	0	0.41		$7.4437e - 012$	100	0.59		$7.6968e - 012$
<i>Jacobi</i>	10	0	0.016	100	$2.4734e - 004$	100	0.019	43	$4.5931e - 006$
	20	0	0.301	400	$8.8342e - 005$	100	0.084	139	$4.7231e - 006$
	30	0	3.72	900	$5.1935e - 005$	100	2.7	285	$4.9409e - 006$
	40	0	17.82	1600	$3.6468e - 005$	100	7.4	480	$4.9761e - 006$
	50	0	73.94	2500	$2.8015e - 005$	100	21.07	722	$4.9764e - 006$
<i>G - S</i>	10	0	0.61	100	$8.4316e - 006$	100	0.52	25	$4.3281e - 006$
	20	0	8.375	360	$4.9394e - 006$	100	6.79	77	$4.3936e - 006$
	30	0	45.91	746	$4.9818e - 006$	100	28.1	156	$4.7805e - 006$
	40	0	209.9	1257	$4.9941e - 006$	100	89.6	262	$4.8521e - 006$
	50	0	499.01	1888	$4.9845e - 006$	100	231	393	$4.9679e - 006$
<i>CG</i>	10	0	0	14	$2.544e - 007$	100	0	12	$9.5757e - 007$
	20	0	0.074	29	$2.8347e - 006$	100	0.048	24	$2.7685e - 007$
	30	0	0.60	44	$3.685e - 006$	100	0.53	36	$4.4713e - 007$
	40	0	1.83	59	$3.9598e - 006$	100	1.39	49	$3.7638e - 006$
	50	0	4.52	75	$4.2605e - 006$	100	3.74	62	$3.7857e - 006$

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