

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ, ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ ΚΑΙ ΔΙΚΤΥΩΝ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
HY200: ΕΠΙΣΤΗΜΟΝΙΚΟΣ ΥΠΟΛΟΓΙΣΜΟΣ

Ράμμου Μαρία-Αικατερίνη

AEM:278

username:marammou

ΕΡΓΑΣΙΑ 1: Αριθμητική Γραμμική Άλγεβρα.

Γενικά σχόλια:

Η διαφορική εξίσωση

$$-\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) + c_1 \frac{\partial u}{\partial x} + c_2 \frac{\partial u}{\partial y} + cu = f, \text{ όπου } x, y \in (0, 1) \times (0, 1),$$
$$u(x, y) = 0, \text{ για } (x, y) \text{ στο σύνορο.}$$

έχει συντελεστές c_1 , c_2 και c . Οι c_1 και c_2 είναι συντελεστές της πρώτης παραγώγου. Αυτό σημαίνει πως αν έστω ένας από τους δύο ισούται με μηδέν η γραφική παράσταση θα αποσβένει, δεν θα έχει δηλαδή συμμετρική μορφή όπως όταν $c_1 = c_2 \neq 0$ όπου η γραφική παράσταση έχει συμμετρική μορφή. Το c δείχνει την καμπή της γραφικής παράστασης. Όσο αυξάνονται τα n_x και n_y τόσο πιο λεπτομερής είναι η γραφική παράσταση γιατί δίνουμε περισσότερα σημεία.

Άσκηση 1.

Στις 2 πρώτες περιπτώσεις όπου n_x και n_y είναι μικρότερα του 5 δεν θα εμφανιστεί ο πίνακας *nonzero*. Οι χρόνοι είναι μηδενικοί γιατί του δίνουμε μικρές τιμές για τα n_x και n_y άρα δεν καθυστερεί ιδιαίτερα. Η λύση μας είναι αρκετά ακριβής αφού το σφάλμα είναι αρκετά μικρό (της τάξης του $e-15$).

Άσκηση 2.

Όσο αυξάνονται τα n_x και n_y περιμένουμε πως η λύση θα είναι πιο ακριβής. Ωστόσο το σφάλμα αυξάνεται άρα έχουμε το αντίθετο αποτέλεσμα (αυτό ισχύει και για $c = 0$ και για $c = 100$). Η αλλαγή του c από 0 σε 100 δεν επιφέρει κάποια αλλαγή στην ακρίβεια της λύσης. Οι χρόνοι όπως είναι αναμενόμενο αυξάνουν καθώς αυξάνουν τα n_x και n_y .

Άσκηση 3

Ο χρόνος εκτέλεσης μειώνεται αισθητά. Ειδικά για τις περιπτώσεις όπου τα n_x και n_y μεγαλώνουν πολύ (δηλαδή πάνω από 30) οι χρόνοι είναι κατά πολύ μικρότεροι σε σχέση με την προηγούμενη μέθοδο. Η ακρίβεια είναι μικρή όπως και στην προηγούμενη άσκηση αφού το σφάλμα είναι της τάξης του $e-14$ για μικρές τιμές των n_x και n_y και της τάξης του $e-12$ για μεγαλύτερες τιμές τους.

Άσκηση 4

Για $c = 0$ το σφάλμα είναι της τάξης του $e-5$. Αυτό σημαίνει πως έχει μειωθεί η ακρίβεια. Επίσης έχει αυξηθεί ο χρόνος εκτέλεσης και εκτελούνται όλες οι επαναλήψεις (δηλαδή $n_x * n_y$). Όταν το c είναι 100 οι επαναλήψεις είναι λιγότερες. Γι' αυτό τελειώνει σε συντομότερο χρονικό διάστημα. Επίσης το σφάλμα είναι μικρότερο (δηλαδή της τάξης του $e-6$) άρα η ακρίβεια είναι μεγαλύτερη.

Άσκηση 5

Ενώ στην αντίστοιχη περίπτωση η Jacobi εκτελούσε όλες τις δυνατές επαναλήψεις, η Gauss-Seidel εκτελεί λιγότερες. Αυτό γίνεται γιατί το σφάλμα είναι της τάξης του $e-6$ άρα παύει να ισχύει η προϋπόθεση $error > eps$ (αφού $eps = 0.5e-5$) και έτσι βγαίνουμε από το βρόγχο πριν προλάβουν να ολοκληρωθούν όλες οι επαναλήψεις. Η Gauss-Seidel έχει πολύ μεγαλύτερους χρόνους εκτέλεσης (φτάνει μέχρι 558s για $n_x = n_y = 50$). Αυτό συμβαίνει γιατί η Jacobi είναι πλήρως παραλληλίσιμη μέθοδος ενώ η Gauss-Seidel εξαρτάται από όλα τα προηγούμενα στοιχεία που έχει υπολογίσει. Επομένως χρειάζεται περισσότερο χρόνο για να υλοποιηθεί.

Άσκηση 6

Όπως και με τη Gauss-Seidel δεν εκτελούνται όλες οι επαναλήψεις γιατί το σφάλμα είναι της τάξης του $e-7$ και στις τελευταίες περιπτώσεις $e-6$. Ωστόσο με την Conjugate Gradients έχουμε πολύ μικρούς χρόνους (ο μεγαλύτερος είναι 4.234s), και πολύ λιγότερες επαναλήψεις σε σχέση με τις άλλες δύο μεθόδους. Συνεπώς είναι πιο ακριβής και πιο σύντομη σε σχέση με τις Jacobi και Gauss-Seidel.

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	0	$8.0005e - 015$
10	3	0	0	0	0	$1.6852e - 014$
5	5	0	0	0	0	$3.7321e - 015$
5	5	100	0	0	0	$2.6899e - 015$
5	5	0	100	0	0	$3.2501e - 015$
5	5	0	0	100	0	$6.3273e - 015$
10	10	0	0	0	0.02	$2.6232e - 014$
20	20	0	0	0	0.2010	$2.599e - 013$
30	30	0	0	0	0.7310	$1.0038e - 012$
40	40	0	0	0	3.284	$2.6837e - 012$
50	50	0	0	0	14.8020	$7.0978e - 012$
10	10	0	0	100	0	$4.9728e - 014$
20	20	0	0	100	0.15	$2.5738e - 013$
30	30	0	0	100	0.711	$1.0141e - 012$
40	40	0	0	100	3.004	$2.8267e - 012$
50	50	0	0	100	10.295	$7.4974e - 012$

Πίνακας 1. Αποτελέσματα με χρήση της `lu`.

	$n_x = n_y$	c	χρόνος	επαναλήψεις	σφάλμα	c	χρόνος	επαναλήψεις	σφάλμα
<i>luinc</i>	10	0	0.171		$2.6472e - 014$	100	0		$4.6573e - 014$
	20	0	0.01		$2.8339e - 013$	100	0.02		$2.7947e - 013$
	30	0	0.131		$1.0588e - 012$	100	0.04		$1.1591e - 012$
	40	0	0.12		$2.851e - 012$	100	0.11		$3.0609e - 012$
	50	0	0.22		$7.4437e - 012$	100	0.231		$7.6968e - 012$
<i>Jacobi</i>	10	0	0.016	100	$2.4734e - 004$	100	0.015	43	$4.5931e - 006$
	20	0	0.297	400	$8.8342e - 005$	100	0.109	139	$4.7231e - 006$
	30	0	3.313	900	$5.1935e - 005$	100	1.094	285	$4.9409e - 006$
	40	0	18.844	1600	$3.6468e - 005$	100	5.719	480	$4.9761e - 006$
	50	0	73.188	2500	$2.8015e - 005$	100	20.938	722	$4.9764e - 006$
<i>G - S</i>	10	0	0.36	100	$8.4316e - 006$	100	0.3440	25	$4.3281e - 006$
	20	0	7.797	360	$4.9394e - 006$	100	4.5940	77	$4.3936e - 006$
	30	0	45.937	746	$4.9818e - 006$	100	26.157	156	$4.7805e - 006$
	40	0	191	1257	$4.9941e - 006$	100	96.953	262	$4.8521e - 006$
	50	0	558.375	1888	$4.984e - 006$	100	257.063	393	$4.9679e - 006$
<i>CG</i>	10	0	0	14	$2.544e - 007$	100	0	12	$9.5757e - 007$
	20	0	0.047	29	$2.8347e - 006$	100	0.032	24	$2.7658e - 007$
	30	0	0.328	44	$3.685e - 006$	100	0.266	36	$4.4713e - 007$
	40	0	1.359	59	$3.9598e - 006$	100	1.125	49	$3.7638e - 006$
	50	0	4.234	75	$4.2605e - 006$	100	3.5	62	$3.7857e - 006$

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