''

Επιτάχυνση του αλγόριθμου του Lanczos χρησιμοποιώντας μηχανές γραφικών

Παρουσίαση 1

Ιάσονας Παυλίδης
michailpg [at] ece.auth.gr

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης

9/5/2021

Ορισμός Ιδιοτιμών

Ορισμός 1

Έστω $A \in R^{nxn}$ συμμετρικός πίνακας τάξης n. Το διάνυσμα $v \in R^n$ καλείται ιδιοδιάνυσμα του A αν υπάρχει λ τέτοιο ώστε να ισχύει,

$$Av = \lambda v, v \neq 0$$

Το λ καλείται ιδιοτιμή του A. Το σύνολο όλων των ιδιοτιμών ενός πίνακα αποτελεί το φάσμα του.

- Οι ιδιοτιμές των τριγωνικών πινάκων είναι οι τιμές των διαγώνιων στοιχείων τους.
- Ένας πίνακας τάξης n έχει ακριβώς n μη μηδενικές ιδιοτιμές.
- Αν λ_i είναι ιδιοτιμές του A, τότε οι ιδιοτιμές του A^{-1} είναι $\frac{1}{\lambda_i}$

Υποχώροι Krylov

Ορισμός 2

Έστω πίνακας $A \in R^{nxn}$ και διάνυσμα $x \in R^n$. Ο Krylov υποχώρος ορίζεται ως

$$K_j(A, x) = span\{x, Ax, ..., A^{j-1}x\}$$

Αποδεικνύεται πως αν $\{u_1, u_2, ..., u_j\}$ είναι μια ορθοκανονική βάση του $K_j(A, x)$ και $U_k = [u_1, u_2, ..., u_k]$,

$$AU_k = U_k H_k + u_{k+1} h_{k+1} e_k^T,$$

$$e_k^T = [0\ 0\ ...\ 1] \in R^k$$

Ο πίνακας H είναι άνω Hessenberg στη γενική περίπτωση και τριδιαγώνιος όταν ο A είναι συμμετρικός.

Αλγόριθμος του Lanczos

Έστω συμμετρικός πίνακας $A \in R^{nxn}$ και ενα μη μηδενικό διάνυσμα $q \in R^n$. Ο αλγόριθμος του Lanczos παράγει τα ορθοκανονικά διανύσματα $\{q_1, q_2, ..., q_j\}$ για τα οποία ισχύει

$$span\{q_1, q_2, ..., q_j\} = K_j(A, x)$$

Χρησιμοποιώντας τον πίνακα $Q_j = [q_1 \ q_2 \ ... \ q_j]$ προκύπτει

$$AQ_j = Q_j T_j + q_{j+1}\beta_{k+1}e_k^T$$

Μετά απο n βήματα θα έχουμε,

$$AQ_n = Q_n T_n$$

$$A = Q_n T_n Q_n^T$$

Άρα οι ιδιοτιμές του A είναι ίδιες με του T!

Block Lanczos

Στην *block* εκδοχή του αλγόριθμου αντικαθιστούμε το αρχικό διάνυσμα με ένα βλοσκ διανυσμάτων και σε κάθε επανάληψη παράγουμε ένα νέο. Αν το μέγεθος ενός βλοσκ είναι b και $Q_0 = [q_1\ q_2\ ...\ q_b]$, παράγεται μια βάση για το χώρο

$$K_j(A, Q_0) = span\{Q_0, AQ_0, ..., A^{j-1}Q_0\},\ jb \ll n$$

Σε σύγκριση με την απλή μέθοδο, η block εκδοχή:

- Είναι πιο αποδοτική σε άποψη χρόνου και μνήμης cache
- Μπορεί να υπολογίσει ιδιοτιμές με πολλαπλότητα μέχρι το μέγεθος του block
- Έχει πιο γρήγορη σύγκλιση ως προς τον αριθμό των επαναλήψεων

Randomized Block Lanczos

Η randomized block εκδοχή του αλγόριθμου βασίζεται στη γενικότερη φιλοσοφία των randomized αλγορίθμων που αναπτύσσονται πολύ την τελευταία δεκαετία.

Αντί να ξεκινήσουμε με ένα τυχαίο μπλόκ, επιλέγουμε το $Q_0 = A\Omega$ όπου $\Omega \in R^{n \times b}$ ένας πίνακας κανονικής κατανομής. Με αυτόν τον τρόπο το αρχικό μπλόκ περιέχει ένα κομμάτι του χώρου των στηλών του A.

$$K_j(A, Q_0) = span\{Q_0, AQ_0, ..., A^{j-1}Q_0\},\ jb \ll n$$

Η σύγκλιση του RBL είναι "ὑπεργραμμική" (superlinear) όταν το φάσμα του πίνακα A φθίνει σχετικά γρήγορα προς το 0.

Χρησιμότητα του Αλγόριθμου και Ακρίβεια Ιδιοτιμών

Σκοπός μας είναι να υπολογίσουμε κάποιες ακραίες ιδιοτιμές του πίνακα A χρησιμοποιώντας λίγες επαναλήψεις $j \ll n$.

$$A = Q_j T_j Q_j^T + E \approx Q_j T_j Q_j^T$$

Έτσι, υπολογίζουμε τις ιδιοτιμές ενός μεγάλου πίνακα A ($O(n^3)$), λύνοντας το ιδιοπρόβλημα ενός πολύ μικρότερου τριδιαγώνιου T ($O(n^2)$).

Οι ιδιοτιμές του πίνακα T ονομάζονται Ritz τιμές του A γιατί είναι οι Rayleigh-Ritz-Galerkin προσεγγίσεις των πραγματικών ιδιοτιμών του A.

Ακρίβεια Ιδιοτιμών

Έστω μ μια ιδιοτιμή του T και x το ιδιοδιάνυσμα που της αντιστοιχεί. Αν $\upsilon = Qx \in R^n$,

$$\|A\upsilon - \mu\upsilon\|_2 = \|B_{j+1} x_b\|_2$$

όπου x_b είναι τα b τελευταία στοιχεία του διανύσματος x.

Ο Αλγόριθμος

Ο αλγόριθμος αποτελείται ουσιαστικά από 4 βήματα:

1. Επιλογή του αρχικού μπλόκ, $V = A\Omega \in R^{n \times b}$
2. Δημιουργία μιας ορθοκανονικής βάσης για τον $K_j(A, V)$ αποθήκευση στον $Q_j \in R^{n \times jb}$
3. Δημιουργία του μπλόκ τριδιαγώνιου πίνακα $T_j \in R^{jb \times jb}$
4. Υπολογισμούς ιδιοτιμών και ιδιοδιανυσμάτων του T_j

Όλοι οι πίνακες μπορούν να συνδυαστούν στην ακόλουθη αναδρομική σχέση

$$Q_{j+1}B_{j+1} = AQ_j - Q_jA_j - Q_{j-1}B_j^T$$

Όπου

$$\boldsymbol{A_j} = Q_j^T AQ_j$$

$$Q_{j+1}\boldsymbol{B_{j+1}} = qr(AQ_j - Q_jA_j - Q_{j-1}B_j^T)$$

$$T_j = \begin{bmatrix} A_1 & B_2{}^T & 0 & \dots & 0 \\ B_2 & A_2 & B_3{}^T & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & B_{j-1} & A_{j-1} & B_j{}^T \\ 0 & \dots & 0 & B_j & A_j \end{bmatrix}$$

Προβλήματα Αριθμητικής Ευστάθειας

Ο αλγόριθμος του Lanczos έχει προβλήματα ευστάθειας όταν εκτελείται σε περιβάλλον μη ακριβούς αριθμητικής, όπως είναι οι υπολογιστές. Πιο συγκεκριμένα,

- Υπάρχει απώλεια στην καθετότητα των δύο πιο πρόσφατων μπλόκ διανυσμάτων της βάσης, Q_{j+1} και Q_j, λόγω σφαλμάτων στρογγυλοποιήσης. Το αντιμετωπίζουμε με τη χρήση του *local reorthogonalization*.
- Υπάρχει απώλεια στην καθετότητα όλων των διανυσμάτων της βάσης Q, εξαιτίας της σύγκλισης της 'Μεθόδου της Δύναμης', (Power Method). Το αντιμετωπίζουμε με τη χρήση του *partial reorthogonalization*.

Local/Partial Reorthogonalization

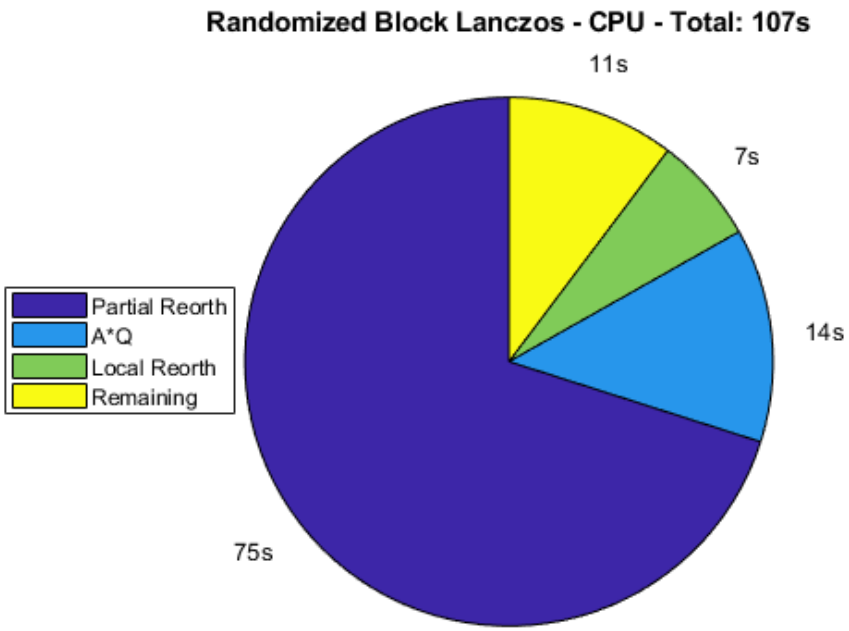
```
function V  = loc_reorth(U1,U2)
    temp = U2.'*U1;
    V = U1 - U2*temp;
    [V,~] = qr(V,0);
end
```

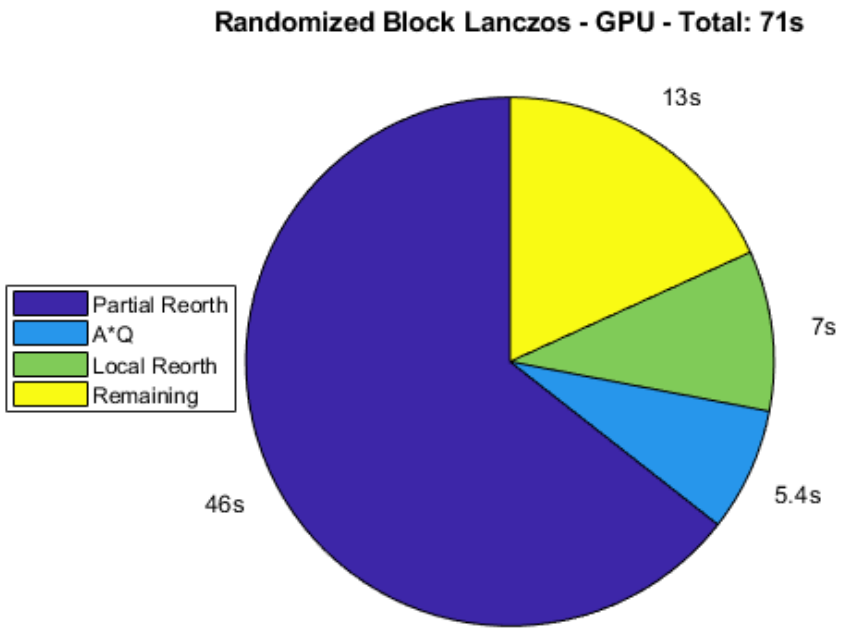
```
function [V1,V2] = part_reorth(U,i,b)
    V1 = U(:,(i-1)*b+1:i*b);
    V2 = U(:,(i-2)*b+1:(i-1)*b);
    for j=1:i-2
        Uj = U(:,(j-1)*b+1:j*b);
        temp = Uj.'*V1;
        V1 = V1 - Uj*temp;
        temp = Uj.'*V2;
        V2 = V2 - Uj*temp;
    end
end
```

Ανάλυση Χρόνου Εκτέλεσης

Για το παράδειγμα μας, χρησιμοποιήθηκε ο συμμετρικός πίνακας,

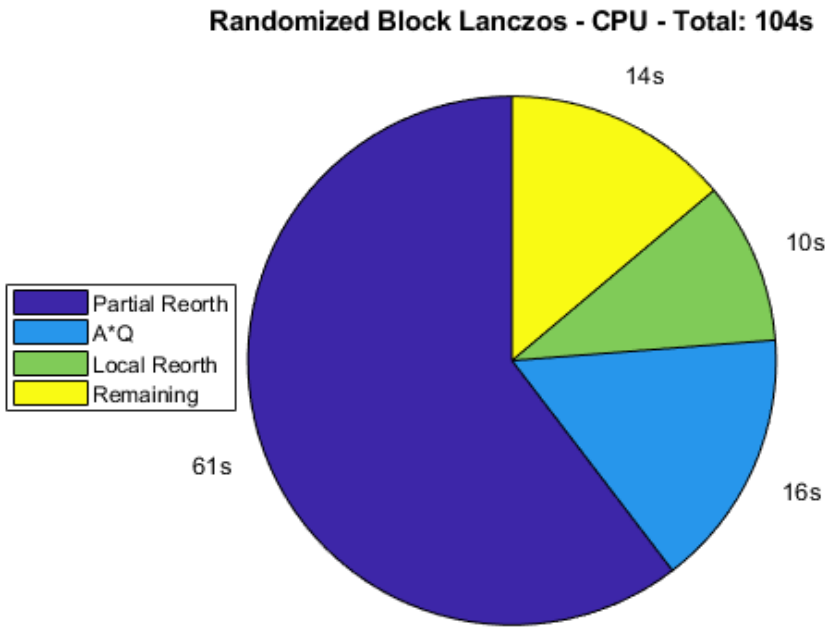
- $1,391,349$ x $1,391,349$
- $64,131,971$ μη μηδενικά στοιχεία
- 50 μεγαλύτερες ιδιοτιμές

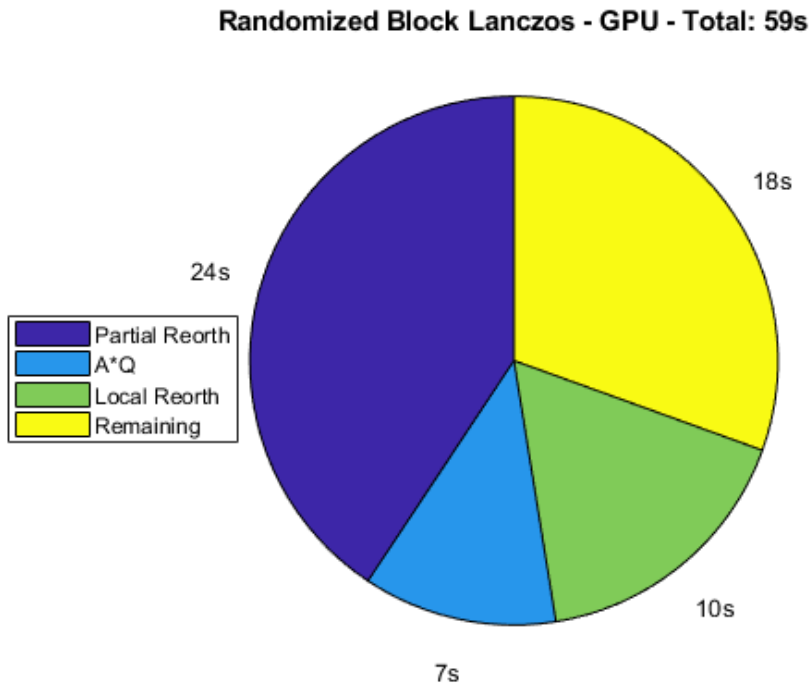




Ανάλυση Χρόνου Εκτέλεσης

Αυξάνοντας το μέγεθος του μπλόκ από 4 σε 7, έχουμε καλύτερη αξιοποίηση της μνήμης cache και καλύτερο λόγο $\frac{computation}{communication}$. Επιλέον, η επιθυμητή σύγκλιση επιτυγχάνεται με λιγότερες επαναλήψεις.





Σημεία Βελτίωσης

- Καλύτερη αξιοποίηση της μνήμης RAM και GPU VRAM
- Μεταφορά λιγότερων αλλά μεγαλύτερων πινάκων από και πρός τη GPU για βελτίωση του λόγου $\frac{computation}{communication}$
- Αξιοποίηση των Tensor Cores για βέλτιστα αποτελέσματα σε γινόμενα *Dense x Dense*