Εργασία στα πλαίσια του μαθήματος "Εργαλεία Προγραμματισμού" - MATLAB

Δημητρίου Ελευθέριος

Επεξεργαστής: Intel(R) Core(TM) i5-5200U CPU @ 2.20GHz 2.20GHz

1 Άσκηση - Random Variabes

Στην άσκηση αυτή μελετάμε την κατανομή των ψευδοτυχαίων αριθμών που προέρχονται από κανονική κατανομή με μέση τιμή μηδέν (μ=0) και τυπική απόκλιση μονάδα (σ=1).

Αρχικά, δημιουργούμε m = 1000 αριθμούς από κανονική κατανομή, δηλαδή $N(\mu = 0, \sigma^2 = 1)$. Βρίσκουμε την μέση τιμή των μετρήσεων με δύο τρόπους, με το κλασικό άθροισμα διαιρώντας με το πλήθος και στην συνέχεια, με την έτοιμη εντολή του MATLAB "mean(X)" , όπου X = το διάνυσμα που περιλαμβάνει τις μετρήσεις. Παρατηρούμε, όπως είναι λογικό, πως και οι δύο τρόποι δίνουν την ίδια μέση τιμή. Με όμοιο τρόπο υπολογίζουμε και την διασπορά των μετρήσεων, όπου εδώ η έτοιμη εντολή του MATLAB είναι η "var(X)". Ακολουθούν τα αποτελέσματα:

Έστω ότι η διαδικασία μας, μας έδωσε $m = 1000$ τυχαίους αριθμούς τους οποίους συμβολίζουμε με την τυχαία μεταβλητή $X_i = \{x_1, x_2, ..., x_m\}$. Τότε η μέση τους τιμής θα είναι

$$< x >= \frac{x_1 + ... + x_m}{m} = -0.032632$$

ενώ μέσω της έτοιμης εντολής του MATLAB

$$mean(X) = -0.032632$$

Η τυπική απόκλιση με την αναλυτική σχέση αλλά και με την έτοιμη εντολή του MATLAB θα είναι

$$s = \sqrt{\frac{\sum_{i=1}^{m}(x_i - < x >)^2}{m-1}} = 0.998965$$

$$sqrt(var(X)) = 0.998965$$

Στην συνέχεια δημιουργούμε $km = 1000$ πειράματα από $m = 1000$ τυχαίους αριθμούς, κατασκευάζοντας την στοχαστική τυχαία μεταβλητή $X = \frac{\sum_{i=1}^{m} x_i}{m}$. Έστω ότι το σύνολο αυτών των τυχαίων στοχαστικών μεταβλητών το εισάγουμε στο διάνυσμα $\hat{X}$. Τότε, για την μέση τιμή της μεταβλητής θα ισχύουν τα εξής:

$$< \hat{X} >= 0.001038$$

ενώ η τυπική απόκλιση υπολογισμένη θεωρητικά, αλλά και με την έτοιμη εντολή του MATLAB θα είναι

$$s_{\hat{X}} = \frac{s}{\sqrt{(m)}} = 0.031590$$

$$sqrt(var(\hat{X})) = 0.032048$$

Παρατηρούμε πως όσο αυξάνεται το πλήθος των τιμών αλλά και αυτό της εκτέλεσης τους πειράματος, τόσο καλύτερη είναι η σύγκλιση προς τις αναμενόμενες τιμές. Στην συγκεκριμένη περίπτωση, η αναμενόμενη μέση τιμή είναι το μηδέν.

Στην συνέχεια, παρουσιάζονται τα διαγράμματα της άσκησης.

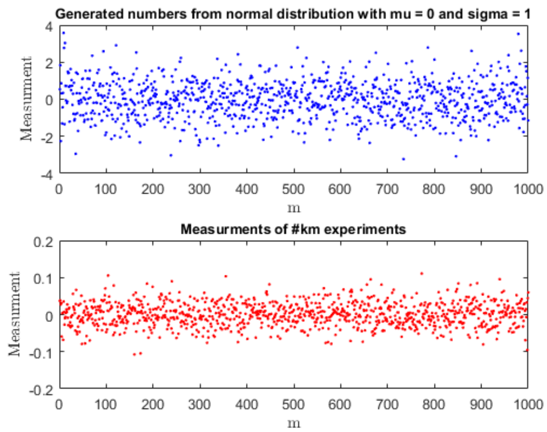

Figure 1: α) Γράφημα της κατανομής των ψευδοτυχαίων τιμών που προέρχονται από κανονική κατανομή με μέση τιμή μηδέν και τυπική απόκλιση μοναδα (km = 1 πείραμα). β) Γράφημα της κατανομής των μέσων τιμών οι οποίες προέρχονται από μετρήσεις από κανονική κατανομή με μέση τιμή μηδέν και τυπική απόκληση μονάδα (km = 1000 πειράματα).

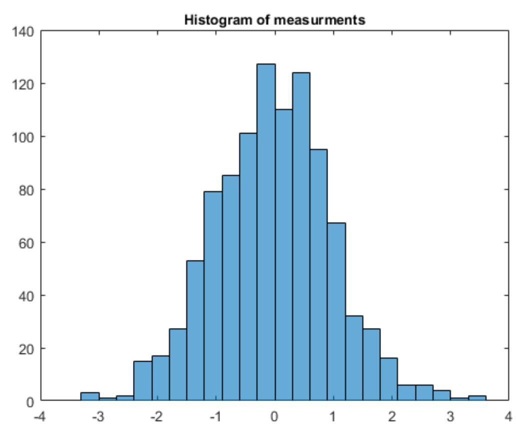

Figure 2: Ιστόγραμμα των μετρήσεων που προέρχονται από κανονική κατανομή με μέση τιμή μηδέν και τυπική απόκλιση μονάδα (km = 1 πείραμα)

2 Άσκηση - Monte Carlo

Σκοπός της άσκησης είναι ο υπολογισμός του π με την καλύτερη δυνατή ακρίβεια. Δημιουργούμε ανύσματα στον τριδιάστατο χώρο έτσι ώστε $x \in [-1,1]$, $y \in [-1,1]$, $z \in [-1,1]$. Οι συντεταγμένες των ανυσμάτων είναι τυχαίοι αριθμοί που κατανέμονται ομοιόμορφα στο διάστημα [-1,1].

Το πλήθος των τυχαίων αριθμών που χρησιμοποιήσαμε για κάθε συντεταγμένη είναι $n = 10^4, 10^5, 10^6, 10^7, 10^8$. Αφού κατασκευάσαμε τις συντεταγμένες στον χώρο, υπολογίσαμε την ακτίνα της σφαίρας κατά τα γνωστά και ελένξαμε πότε αυτό το μέτρο είναι μικρότερο ή ίσο της μονάδας. Για κάθε μήκος εντός της σφαίρας, αύξανε κατά ένα ο μετρητής μας. Στο τέλος, διαιρούσαμε το άθροισμα αυτό με το συνολικό πλήθος (n) ώστε να πάρουμε το ποσοστό των μηκών που βρίσκονται μέσα στην σφαίρα. Ο λόγος των όγκων της σφαίρας και του κύβου ισούται με $\frac{\pi}{6}$, επομένως η προσεγγιστική τιμή του π είναι 6*ποσοστό.

Έστω $\pi_{\pi\epsilon\iota\rho}$ = η τιμή του π από το πείραμά μας και s = το σφάλμα της μέτρησής του, για την αναμενόμενη τιμή θα ισχύει $\pi \in [\pi_{\pi\epsilon\iota\rho} - s, \pi_{\pi\epsilon\iota\rho} + s]$.

Το πρώτο πρόγραμμα πραγματοποιεί το πείραμα με χρήση της for-loop για την κατασκευή των τριών συνιστωσών. Επίσης $[t] = sec$.

Για $n = 10^4$

$$\pi_{\pi\epsilon\iota\rho} = 3.14040 \pm 0.03071$$

$$t = 1.130258$$

Για $n = 10^5$

$$\pi_{\pi\epsilon\iota\rho} = 3.14340 \pm 0.00978$$

$$t = 2.224319$$

Για $n = 10^6$

$$\pi_{\pi\epsilon\iota\rho} = 3.1420140 \pm 0.0030954$$

$$t = 12.079543$$

Για $n = 10^7$

$$\pi_{\pi\epsilon\iota\rho} = 3.14117280 \pm 0.00097961$$

$$t = 108.846326$$

Για $n = 10^8$
$$\pi_{\pi\varepsilon\iota\rho} = 3.141795420 \pm 0.0003098407$$
$$t = 664.122353$$

Για $n = 10^9$: out of memory

Ως δεύτερο τρόπο, πήραμε μόνο μία for-loop η οποία θα ελέγχει μόνο το αν το μήκος είναι μικρότερο ή ίσο της μονάδας, ενώ οι τρεις συνιστώσες δημιουργούνται αυτόματα, χωρίς την for-loop.

Για $n = 10^4$
$$\pi_{\pi\varepsilon\iota\rho} = 3.16260 \pm 0.03126$$
$$t = 0.119232$$

Για $n = 10^5$
$$\pi_{\pi\varepsilon\iota\rho} = 3.14220 \pm 0.00975$$
$$t = 0.195480$$

Για $n = 10^6$
$$\pi_{\pi\varepsilon\iota\rho} = 3.1416420 \pm 0.0031017$$
$$t = 0.402542$$

Για $n = 10^7$
$$\pi_{\pi\varepsilon\iota\rho} = 3.1413630 \pm 0.0009810$$
$$t = 2.999020$$

Για $n = 10^8$
$$\pi_{\pi\varepsilon\iota\rho} = 3.141617880 \pm 0.000309867$$
$$t = 14.241280$$

Για $n = 10^9$: out of memory

Παρατηρούμε στον δεύτερο τρόπο την τεράστια διαφορά στην ταχύτητα των υπολογισμών, αφού χρησιμοποιήσαμε την for-loop στο ελάχιστο. Παρατηρούμε επίσης και στους δύο τρόπους την ακρίβεια στο π να φτάνει μέχρι το τρίτο δεκαδικό.