1η Υποχρεωτική Εργασία LaTeX

Ονοματεπώνυμο: Στέφανος Καραμπέρας
ΑΕΜ: 2910

15 Δεκεμβρίου 2019

Άσκηση 1

Γενικές παρατηρήσεις

Ο κώδικας που έχει συνταχθεί για την άσκηση 1 βρίσκεται εντός του φακέλου "Code/task-1" .
Οι ζητούμενοι αλγόριθμοι που εφαρμόζουν τις τεχνικές προσέγγισης ρίζας με τη μέθοδο της διχοτόμησης, τη μέθοδο Newton-Raphson και τη μέθοδο της τέμνουσας, έχουν υλοποιηθεί εντός του αρχείου "root_estimation_algorithms.py" Επιπρόσθετα, εντός του αρχείου f_function.py έχουν δημιουργηθεί οι συναρτήσεις calculate_f(x), calculate_der_1_f(x) και calculate_der_2_f(x). Οι συναρτήσεις αυτές δέχονται ως όρισμα έναν αριθμό ξ και επιστρέφουν την ρίζα της $f(x)$, $f'(x)$ και $f''(x)$ αντίστοιχα, για την δοθείσα συνάρτηση $f(x)$ της εκφώνησης της άσκησης 1. Τέλος, με δεδομένο πως η ζητούμενη ακρίβεια ορίζεται στο 5ο δεκαδικό ψηφίο, η μεταβλητή t_err στο αρχείο "root_estimation_algorithms.py" αρχικοποιείται με την τιμή t_err = 0.000005 (ακρίβεια 5ου δεκαδικού ψηφίου με στρογγυλοποίηση).

Αλγόριθμος για τη μέθοδο Διχοτόμησης

Η συνάρτηση που υλοποιήθηκε για την εκτέλεση του αλγορίθμου που εφαρμόζει τη μέθοδο της διχοτόμησης είναι η "partitioning_root_estimation()".
Η συνάρτηση αρχικά υπολογίζει το πλήθος των απαιτούμενων επαναλήψεων της μεθόδου διχοτόμησης για την επίτευξη της ζητούμενης ακρίβειας. Στη συνέχεια, ξεκινά να εκτελεί το ορισμένο πλήθος επαναλήψεων, σταματώντας σε περίπτωση που βρει ακριβή ρίζα της $f(x)$ και τυπώντας τη ρίζα, ή συνεχίζοντας μέχρι να ολοκληρώσει το ορισμένο πλήθος επαναλήψεων, τυπώνοντας τελικά την προσέγγιση της ρίζας και το πλήθος επαναλήψεων που απαιτήθηκαν για την εκτίμησή της.

Αλγόριθμος για τη μέθοδο Newton-Raphson

Η συνάρτηση που υλοποιήθηκε για την εκτέλεση του αλγορίθμου που εφαρμόζει τη μέθοδο Newton-Raphson είναι η 'newton_raphson_root_estimation()".
Η συνάρτηση αρχικοποιεί το x_{n-1} χρησιμοποιώντας το πρώτο στοιχείο του πεδίου ορισμού [-2,2] της $f(x)$, δηλαδή το -2. Ακολούθως, υπολογίζει διαδοχικά το x_n μέχρις ότου η απόλυτη τιμή της διαφοράς x_{n-1}-x_{n-1} να είναι μικρότερη ή ίση του καθορισμένου αποδεκτού σφάλματος (ακρίβεια), δηλαδή $|x_n$ - $x_{n-1}|$ $\leq$ t_err. Στο τέλος κάθε επανάληψης, η τιμή του q_{n-1} ανανεώνεται με την τιμή του q_n. Η συνάρτηση καταγράφει σε μια μεταβλητή iter_count το πλήθος επαναλήψεων που πραγματοποιεί κατά την εκτέλεσή της. Με τη λήξη των επαναλήψεων λόγω επίτευξης της επιθυμητής ακρίβειας, τυπώνεται το πλήθος επαναλήψεων που εκτελέστηκαν και η προσέγγιση της ρίζας που υπολογίστηκε.

Αλγόριθμος για τη μέθοδο Τέμνουσας

Η συνάρτηση που υλοποιήθηκε για την εκτέλεση του αλγορίθμου που εφαρμόζει τη μέθοδο της τέμνουσας είναι η "secant_root_estimation()".
Η συνάρτηση αρχικοποιεί το x_{n-1} χρησιμοποιώντας το πρώτο στοιχείο του πεδίου ορισμού [-2,2] της $f(x)$, δηλαδή το -2. Επιπρόσθετα, αρχικοποιεί το x_n χρησιμοποιώντας το τελευταίο στοιχείο του πεδίου ορισμού [-2,2] της $f(x)$, δηλαδή το 2.
Ακολούθως, υπολογίζει διαδοχικά το x_{n+1} μέχρις ότου η απόλυτη τιμή της διαφοράς x_{n+1} - x_n να είναι μικρότερη ή ίση του καθορισμένου αποδεκτού σφάλματος (ακρίβεια), δηλαδή $|x_{n+1}$ - $x_n| \leq$ t_err. Στο τέλος κάθε επανάληψης, η τιμή του q_{n-1} ανανεώνεται με την τιμή του q_n, ενώ η τιμή του q_n ανανεώνεται με την τιμή του q_{n+1}. Η συνάρτηση καταγράφει σε μία μεταβλητή iter_count το πλήθος επαναλήψεων που πραγματοποιεί κατά την εκτέλεσή της. Με τη λήξη των επαναλήψεων λόγω επίτευξηςτης επιθυμητής ακρίβειας, τυπώνεται το πλήθος επαναλήψεων που εκτελέστηκαν και η προσέγγιση της ρίζας που υπολογίστηκε.

Άσκηση 2

Γενικές παρατηρήσεις

Ο κώδικας που έχει συνταχθεί για την άσκηση 2 βρίσκεται εντός του φακέλου "Code/task-2" .
Οι ζητούμενοι αλγόριθμοι που εφαρμόζουν τις τεχνικές προσέγγισης ρίζας με την τροποποιημένη μέθοδο Newton-Raphson, την τροποποιημένη μέθοδο διχοτόμησης και την τροποποιημένη μέθοδο της τέμνουσας, έχουν υλοποιηθεί εντός του αρχείου "modified_root_estimation_algorithms.py" Επιπρόσθετα, εντός του αρχείου f_function.py έχουν δημιουργηθεί οι συναρτήσεις calculate_f(x), calculate_der_1_f(x) και calculate_der_2_f(x). Οι συναρτήσεις αυτές δέχονται ως όρισμα έναν αριθμό ξ και επιστρέφουν την ρίζα της $f(x)$, $f'(x)$ και $f''(x)$

αντίστοιχα, για την δοθείσα συνάρτηση $f(x)$ της εκφώνησης της άσκησης 2. Τέλος, με δεδομένο πως η ζητούμενη ακρίβεια ορίζεται στο 5ο δεκαδικό ψηφίο, η μεταβλητή t_err στο αρχείο "root_estimation_algorithms.py" αρχικοποιείται με την τιμή t_err = 0.000005 (ακρίβεια 5ου δεκαδικού ψηφίου με στρογγυλοποίηση).

Αλγόριθμος για την τροποποιημένη μέθοδο Newton-Raphson

Η συνάρτηση που υλοποιήθηκε για την εκτέλεση του αλγορίθμου που εφαρμόζει τη μέθοδο Newton-Raphson είναι η 'modified_newton_raphson_root_estimation()". Η συνάρτηση αρχικοποιεί το x_n χρησιμοποιώντας το πρώτο στοιχείο του πεδίου ορισμού [-2,2] της $f(x)$, δηλαδή το -2. Ακολούθως, υπολογίζει διαδοχικά το x_{n+1} μέχρις ότου η απόλυτη τιμή της διαφοράς x_{n+1}-x_n να είναι μικρότερη ή ίση του καθορισμένου αποδεκτού σφάλματος (ακρίβεια), δηλαδή $|x_{n+1}$ - $x_n|$ $\leq$ t_err. Η συνάρτηση καταγράφει σε μια μεταβλητή iter_count το πλήθος επαναλήψεων που εκτελεί κατά την εκτέλεσή της. Με τη λήξη των επαναλήψεων λόγω επίτευξης της επιθυμητής ακρίβειας, τυπώνεται το πλήθος επαναλήψεων που εκτελέστηκαν και η προσέγγιση της ρίζας που υπολογίστηκε.

Αλγόριθμος για την τροποποιημένη μέθοδο διχοτόμησης

1 Άσκηση 1 (Λύση)

$$\scriptstyle{ABΓ\Delta EZH}\Theta I \kappa\lambda\mu\nu\xi o\pi\rho\varsigma$$

2 Άσκηση 2 (Λύση)

Normal Italics ***Bold***
E*mphasized* *Underlined*

3 Άσκηση 3 (Λύση)

$$a^2 + b^2 = c^2$$

$$e^{i\pi} = -1$$

$$\pi = \frac{c}{d}$$

$$\frac{d}{dx}\int_a^x f(s)ds = f(x)$$

$$f(x) = \sum_{i=0}^{\infty} \frac{f^{(i)}(0)}{i!} x^i$$

$$\mathbf{Ax} = \mathbf{b}$$

$$||x + y|| \leq ||x|| + ||y||$$

$$\mathbf{I} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{1}$$

$$\mathbf{I} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}$$

$$\mathbf{I} = \begin{Bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{Bmatrix}, \quad \mathbf{I} = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}, \quad \mathbf{I} = \begin{Vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{Vmatrix} \tag{3}$$

4 Άσκηση 4 (Λύση)

Τέφας	*2*	*3*
Πήτας	*5*	*6*
Λάσκαρης	*8*	*9*

Κοτρόπουλος	*2*	*3*
Πήτας	*5*	*6*
Νικολαΐδης	*8*	*9*

1	*2*	*3*
4	*5*	*6*
7	*8*	*9*

1	*2*	*3*
4	*5*	*6*
7	*8*	*9*

Μέλη ΔΕΠ Πληροφορικής		
Λέκτορες	*VD*	*Δραζιώτης Κωνσταντίνος*
Επίκουροι	*LN*	*Λάσκαρης Νικόλαος*
	TG	*Τσουμάκας Γρηγόριος*
Αναπληρωτές	*TA*	*Τέφας Αναστάσιος*
	PN	*Πλέρος Νίκος*
	PA	*Παπαδόπουλος Απόστολος*
Καθηγητές	*KC*	*Κοτρόπουλος Κωνσταντίνος*
	PI	*Πήτας Ιωάννης*
	VI	*Βλαχάβας Ιωάννης*

5 Άσκηση 5 (Λύση)

- *Τέφας*
- *Μπουζάς*
- *Μπρούζα*
- *Λάσκαρης*
- *Κοτρόπουλος*
- *Πήτας*
- *Νικολαΐδης*

1. *Τέφας*
2. *Μπουζάς*
3. *Μπρούζα*
4. *Λάσκαρης*
5. *Κοτρόπουλος*
6. *Πήτας*
7. *Νικολαΐδης*

(α) *Τέφας*

(β) *Μπουζάς*

(γ) *Μπρούζα*

(δ) *Λάσκαρης*

(ε) *Κοτρόπουλος*

(ζ) *Πήτας*

(η) *Νικολαΐδης*