بسمه تعالی

دانشکده مهندسی کامپیوتر، دانشگاه صنعتی شریف

- مشخصات : عطیه جمشیدپور
- شماره دانشجویی:۹۴۱۰۳۸۳۵

تمرین دوم محاسبات عددی

موعد تحویل :۱۲/۲۸

تمرین اول

n	a	b	Xn=(a+b)/2	sign f(a)f(x)	f(Xn)	f(a)
1	0	1	0.5	1	-0.37758	-1
2	0.5	1	0.75	-1	0.01831	-0.37758
3	0.5	0.75	0.625	1	-0.18596	-0.37758
4	0.625	0.75	0.6875	1	-0.08533	-0.18596
5	0.6875	0.75	0.71875	1	-0.03388	-0.08533
6	0.71875	0.75	0.73438	1	-0.00787	-0.03388
7	0.73438	0.75	0.74219	-1	0.0052	-0.00787
8	0.73438	0.74219	0.73829	1	-0.00133	-0.00787
9	0.73829	0.74219	0.74024	-1	0.00193	-0.00133
10	0.73829	0.74024	0.73927	-1	0.00031	-0.00133

Illustration 1: 4 جدول محاسبه تقریب D

$0.00031<0.001\,;\alpha:0.7393$

تمرین دوم

الف)

n	a	b	f(a)	f(b)	x=(a*f(b)-b*f(a))/(f(b)-f(a))	sign of f(x)f(a)	f(x)
1	-1	0	0.36788	-1	-0.73106	-1	-0.05648
2	-1	-0.73106	0.36788	-0.05648	-0.76685	-1	-0.00183

$n=2\,;x=-0.7668$

ب)

n	x	f(x)
1	1	-0.6
2	1.5	1.425
3	1.1481	-0.179897524
4	1.1875	-0.044970703
5	1.2006	0.0021852242
6	1.2	0

$0<0.002\,;x=1.2$

تمرین سوم

الف)

n	x	f(x)	f_prime(x)
1	1	0.37	1.37
2	0.729927007	0	1.37

$x = 0.72992701$

ب)

n	x	g(x)=exp(-x/2)	f(x)
1	0.7	0.704688089718713	-0.013261173339466
2	0.704688089718713	0.703038203683446	0.004699096004082
3	0.703038203683446	0.703618409426139	-0.001648525721527
4	0.703618409426139	0.703414317310518	0.000580374094603
5	0.703414317310518	0.703486101631198	-0.000204071290242
6	0.703486101631198	0.703460852448368	7.17868972359703E-05

$x^2 e^x - 1 = 0 ; x^2 = e^{(-x)} ; g(x) = x = e^{(-x/2)} ; x \in [0,1] ; e^{(-x/2)} \in [0.6066, 1]$

$g'(x) = -0.5 e^{(-x/2)} ; x \in [0, 1] ; g'(x) \in [-0.5, -0.3032] ; |g'(x)| < 1$

$7.17868972359703\text{E-}05 < 1\text{E-}04 ; x = 0.703486101631198$

تمرین چهارم

الف)

$g(x) = g(\alpha) + g'(\alpha)(x-\alpha) + (\frac{1}{2}) g''(c)(x-\alpha)^2 ; with\, c\, between\, x\, end\, \alpha$

$x = x_n ; x_{(n+1)} = g(x_n), g(\alpha) = \alpha ; g'(\alpha) = 0$

$x_{(n+1)} = \alpha + (\frac{1}{2}) g''(c_n)(x_n - \alpha)^2$

$\alpha - x_{(n+1)} = -\frac{1}{2} g''(c_n)(\alpha - x_n)^2 ; with\, c_n\, between\, a\, end\, x_n$

با توجه به روابط فوق میتوان گفت در این صورت روش تکرار ساده همگرا است. در واقع در صورتی که مشتق دوم g در نقطه α صفر نباشد، همگرایی با درجه دوم رخ خوهاد داد[1].

ب)

$$NEWTON\,'\,S\,METHOD : x_{(n+1)} = x_n - \frac{f(x_n)}{f'(x_n)}$$

در صورتیکه مشتق اول در ریشه برابر با صفر باشد، در برخی موارد استفاده از روش نیوتون منجر به همگرایی نخواهد شد.زیرا هنگامی که مشتق نزدیک به صفر باشد، خط مماس تقریباً افقی است و از این رو ممکن است نسبت به ریشه مورد نظر بیشتر جهش کند[3].

در برخی موارد ممکن است این روش همچنان همگرا باشداما درجه همگرایی درجه دوم نیست، حتی اگر تابع در همه جا مشتق پذیر باشد. مثال زیررا در نظر بگیرید[3, 2].

$$f(x) = x^K ; f'(x) = Kx^{(k-1)} ; x - \frac{f(x)}{f'(x)} = x^k/2$$

تمرین پنجم

$$f(x) = 0 ; -\lambda f(x) = 0 ; x = x - \lambda f(x) ; g(x) = x - \lambda f(x) ;$$

$$g'(x) = 1 - \lambda f'(x) ; -1 < g'(x) < 1, |g'(x)| < q < 1$$

همان طور که مشاهده می‌شود شرط اول از تخمین به روش تکرار ساده ارضا شده است. حال میبایست بررسی کنیم. با توجه به اینکه x=0 ریشه f(x) لحاظ شده است بازه (a, b)را به عنوان بازه شامل صفر در نظر می‌گیریم. از طرفی چون مشتق اول تابع همواره مثبت است، نتیجه میگیریم این تابع اکیدا صعودی است. از طرفی چون این تابع در صفر ریشه دارد و a<0 میتوان نتیجه گرفت f(a)<0

$$a < 0 ; g(a) = a - \lambda f(a), checking\ if : g(a) > a ; a - \lambda f(a) > a, as\ we\ know\ \lambda > 0 : f(a) < 0, True$$

با توجه به اکیدا و صعودی بودن تابع و ریشه داشتن آن در صفر و اینکه b>0 میتوان نتیجه گرفت f(b)>0

$$b > 0 ; g(b) = b - \lambda f(b), checking\ if : g(b) < b ; b - \lambda f(b) < b, as\ we\ know\ \lambda > 0 : f(b) > 0, True$$

لذا شرط دوم همگرایی روش تکرار ساده نیز ارضا شده و داریم

$$a < x < b ; a < g(x) < b$$

بنابراین روش تکرار ساده برای یافتن ریشه x=0 برای f(x) =0 همگرا است. لذا میتوان نتیجه گرفت که g(x) به x=0 که ریشه معادله است همگرا است [4] لذا :

$$g(x) \rightarrow x = 0 ; x - \lambda f(x) \rightarrow 0$$

تمرین ششم

الف)

الگوریتم هورنر پیاده‌سازی شده و در ضمیمه این سند در attachment/code/hw2-1.py قرار دارد.

این الگوریتم به ازای مقادیر ورودی داده شده خروجی زیر را دارد :

$P(3) = -9125$

ب)

الگوریتم تخمین ریشه به روش وتری پیاده‌سازی شده و در ضمیمه این سند در attachment/code/hw2-2.py قرار دارد [5].

ورودی این برنامه به صورت یک آرایه در ابتدای برنامه قرار دارد و به صورت زیر است :

```
# inputs = [function: string, a: float, b: float, N: int]
inputs = ['x**3 + 5', -10.5, 11, 10]
```

- المان اول رشته ای است که در واقع تابعی از x است.
- المان دوم ابتدای بازه‌ایست که ریشه در آن قرار دارد.
- المان سوم انتهای بازه‌ایست که ریشه در آن قرار دارد.
- المان چهارم تعداد مراحل است.

منابع

[1] : http://homepage.divms.uiowa.edu/~whan/3800.d/S3-4.pdf, page 18

[2] : https://en.wikipedia.org/wiki/Newton%27s_method

[3] : http://bit.ly/33sOjZ3

[5] : https://www.math.ubc.ca/~pwalls/math-python/roots-optimization/secant/