

Numerical Computations

Computer Assignment 2

Professor : Jamal Kazazi
Shahaboddin Sheybani
810101454

2023

Q1a)

Main orders :

برای برازش تعدادی داده با یک خط باید دستگاه ماتریسی زیر را حل کنیم

$$\begin{bmatrix} [xx] & [x] \\ [x] & N \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} [xy] \\ [y] \end{bmatrix}$$

عناصر درون ماتریس به معنای جمع هر کدام از فرم داده ها می باشد.
پاسخ این ماتریس که a و b را به ما می دهد خط برازش شده را به فرم زیر
به دست می آورد

$$Y = ax + b$$

خط 1 تا 4: تعریف ماتریس داده های صورت سوال که p مقدار تقاضا و q مقدار قیمت را نشان می دهد

خط 5 : ریختن طول ماتریس داده ها یعنی تعدادشان در n

خط 6 و 7 : تعریف تابع $p(q)$ که برابر خروجی تابع برازش می باشد

خط 8 : تعریف ماتریسی که خطای قیمت را ذخیره کند

خط 9 تا 11 : مقداردهی عناصر مختلف ماتریس خطا با یک حلقه for
به طول n که عناصر برابر با قدر مطلق اختلاف داده های سوال و مقدار
خروجی تابع برازش شده می باشد

خط 12 تا 15 : برای محاسبه مجموع مزدوج خطاها متغیر sum_error
را برابر صفر تعریف می کنیم و با یک حلقه for در مرحله آپدیت می شود
خط 16 تا 19 : رسم دو نمودار درواقع تابع برازش شده و داده های
سوال و رسم آنها در یک کادر

Curve Fitness function :

خط 20 : تعریف تابع برازش کننده که با دریافت دو ماتریس ورودی داده
ها یک تابع خروجی دارد

خط 21 و 22 : ریختن جمع مقادیر x در متغیر sum_x با استفاده از تابع $\text{sum}(x)$ و همینطور برای Y

خط 23 : ریختن طول ماتریس ورودی یعنی تعداد داده ها در N
خط 24 تا 27 : محاسبه مجموع مجذور x ها و ریختن در ماتریس sum_xx

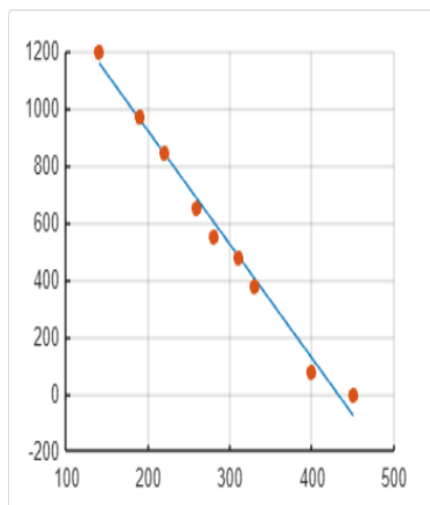
خط 28 تا 31 : محاسبه مجموع ضرب y و x ها و ذخیره در sum_xy
خط 32 : تعریف ماتریس ضرایب A که در ابتدای توضیحات مشخص شد

خط 33 : ذخیره دترمینان ماتریس در d
خط 34 : تعریف ماتریس معلومات که همان B می باشد
خط 35 : تعریف ماتریس وارون یک ماتریس 2×2 که می دانیم چگونه محاسبه می شود
خط 36 : ماتریس coeff که دارای a, b می باشد را برابر ضرب وارون A در B قرار می دهیم

$$AX = B \quad , \quad X = A^{-1}B$$

خط 38 تا 42 : تعریف تابع برازش شده به صورت $a + b \cdot p$ که برابر با خروجی یعنی output قرار داده می شود

Result :



```
q x p x
1x1 symfun

val(q) =

1713.4415954415972009883262217045 - 3.9790598290598290598290598290598*x
```

Q1b)

Main Orders:

برای برازش داده ها با تابعی درجه 2 روش کار به شکل قبل است فقط کمی فرم ماتریس به شکل زیر تغییر می کند

$$\begin{bmatrix} [x^4] & [x^3] & [x^2] \\ [x^3] & [x^2] & [x] \\ [x^2] & [x] & N \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} [x^2 y] \\ [xy] \\ [y] \end{bmatrix}$$

تابع خروجی به فرم $y = ax^2 + bx + c$ می باشد
خط 1 تا 19 : دستورات کاملا مشابه بخش قبلی می باشد

Curve Fitness Function:

خط 20 : تعریف تابع curve_fit که با دریافت دو ماتریس داده خروجی تابع داشته باشد

خط 21 و 22 : برابر قرار دادن جمع عناصر y و x برابر با y_1 و x_1

خط 24 تا 27 : تعریف متغیر x_2 که برابر مجموع توان دو های X ها می باشد و آپدیت با یک حلقه for

خط 28 تا 31 : تعریف متغیر x_3 که برابر مجموع توان سه های X ها می باشد و آپدیت با یک حلقه for

خط 32 تا 35 : تعریف متغیر x_4 که برابر مجموع توان چهار های X ها می باشد و آپدیت با یک حلقه for

خط 36 تا 39 : تعریف متغیر xy که برابر مجموع ضرب های y و x ها می باشد و آپدیت با یک حلقه for

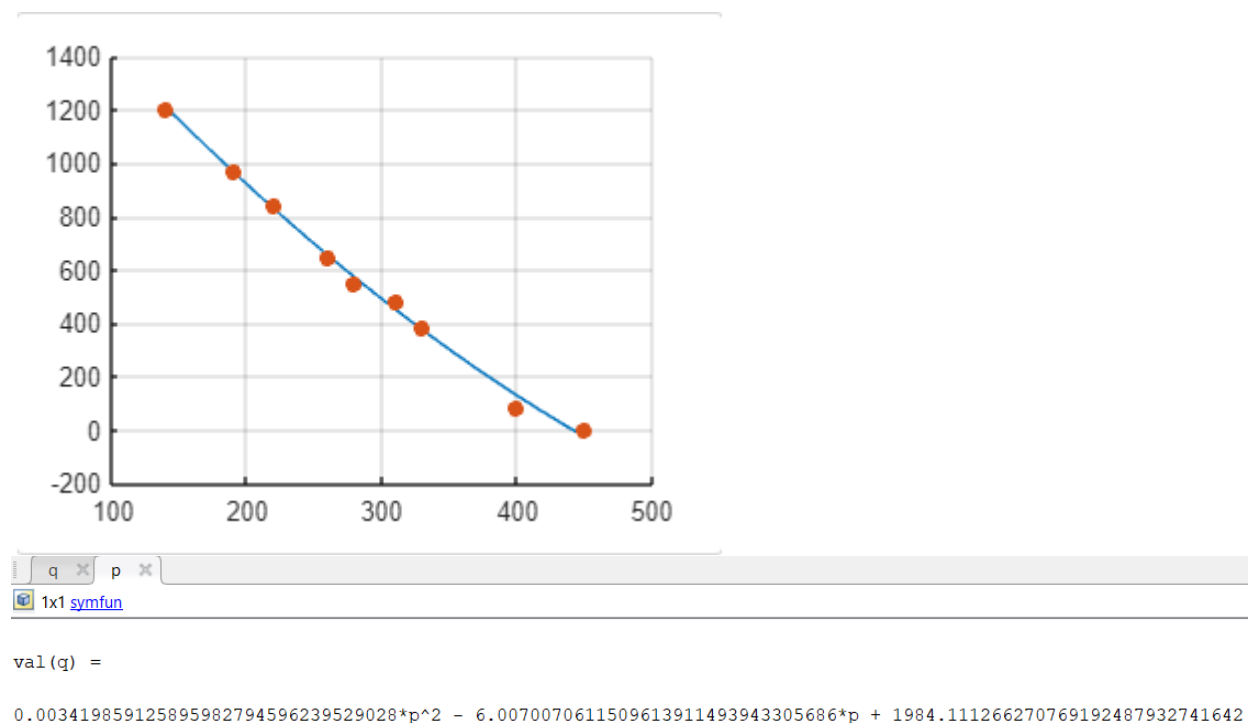
خط 40 تا 43 : تعریف متغیر $x^2 y$ که برابر مجموع ضرب های y و x^2 ها می باشد و آپدیت با یک حلقه for

خط 44 تا 45 : تعریف ماتریس A و B به همان فرمی که در ابتدا ذکر شد

خط 45 : محاسبه دستگاه ماتریس با استفاده از تابع حذفی گاوس که در

پروژه قبلی توضیحات آن داده شد
 خط 48 تا 53 : پیدا کردن ضرایب تابع برازش شده و تعریف تابع $q(p)$ و
 برابر قرار دادن output با $a + b*p + b*p^2$
 خط 55 تا 76 : دستورات تابع حذفی گاوس که توضیحات آن در پروژه
 قبل قرار دارد

Result:



Q1c)

با توجه به مقادیر ماتریس p_error در بخش 1 می فهمیم که فقط به
 ازای 845 و 480 دارای خطای زیر 10 است

	1	2	3	4	5	6	7	8	9
1	43.6268	12.5798	6.9516	28.8860	49.3048	0.0670	20.3519	41.8177	77.1353
2									

با توجه به مقادیر ماتریس p_error در بخش 2 می فهمیم که فقط به
 ازای 970 و 650 و 380 دارای خطای زیر 10 است

	1	2	3	4	5	6	7	8	9	10
1	10.1595	3.7632	16.9091	3.4719	20.2662	29.4125	5.7784	48.4859	26.5204	
2										

Q2)

Main Orders :

طبق فرمول تیلور $o(h^2)$ مشتق هر نقطه از رابطه زیر به دست می آید
$$f'(x_i) = \frac{f(x_{i+1}) - f(x_{i-1}))}{2h}$$
 و برای نقطه آخر از $\frac{f(x_i) - f(x_{i-1}))}{h}$ می رویم
خط 1 تا 4 : تعریف ماتریس های x و y و t برابر با طول n
خط 5 : برابر قرار دادن v_x برابر با خروجی تابع velocity با دریافت x
خط 6 : برابر قرار دادن v_y برابر با خروجی تابع velocity با دریافت y
خط 7 : تعریف ماتریس v_t که سرعت کل می باشد به طول $n-1$
خط 8 تا 10 : مقداردهی عناصر ماتریس v_t برابر جذر مجموع مجذور
های هر عنصر v_x و v_y که مقدار کل سرعت است

Velocity Function :

خط 11 : تعریف تابع velocity با دریافت دو تابع x و t
خط 12 : تعریف h برابر با اختلاف دو تا از t ها
خط 13 : تعریف متغیر n برابر با طول x
خط 14 : تعریف ماتریس v به طول $n-1$
خط 15 تا 17 : از آنجایی که سرعت در نقطه اول را نیاز نداریم از $i=2$
شروع میکنیم تا $n-1$ زیرا نقطه آخر را از فرمول دوم استفاده می کنیم
خط 18 : سرعت نقطه آخر را از فرمول دوم که در بالا گفته شد استفاده
می کنیم
خط 19 : خروجی را برابر ماتریس v قرار می دهیم

Result :

Variables - v_t											
v_t											
1x10 double											
	1	2	3	4	5	6	7	8	9	10	11
1	0.9638	1.2879	1.3277	1.1595	1.0580	1.5760	2.7124	4.2707	6.2475	7.3142	
2											
3											