Advanced Optimization

Homework 2.

—

Ali Izadi

810199102

سوالات پیاده سازی

سوال ۱

الگوریتم گرادیان مزدوج مطابق با الگوریتم زیر پیاده سازی شد:

مقادیر اولیه: $r_0 = Ax_0 - b$، $p_0 = -r_0$ و $k = 0$

تا وقتی که $r_k \neq 0$ انجام بده:

- $\alpha_k = -\frac{r_k^T p_k}{p_k^T A p_k}$
- $x_{k+1} = x_k + \alpha_k p_k$
- $r_{k+1} = A x_{k+1} - b$
- $\beta_{k+1} = \frac{r_{k+1}^T A p_k}{p_k^T A p_k}$
- $p_{k+1} = -r_{k+1} + \beta_{k+1} p_k$
- $k = k + 1$

کد کلاس اصلی گرادیان مزدوج در زیر آورده شده است:

```python
class ConjugateGradient:
    def __init__(self, A, b, x0, max_iter=200000, epsilon=1e-6):
        self.A = A
        self.b = b
        self.x0 = x0
        self.max_iter = max_iter
        self.epsilon = epsilon

    def solve(self):

        rs = []

        A = self.A
        b = self.b

        x = self.x0

        r = A @ x - b
        p = - r

        rs.append(np.linalg.norm(r))

        for i in range(self.max_iter):

            alpha = - (r.T @ p) / (p.T @ (A @ p))
            x = x + alpha * p
            r = A @ x - b
            B = (r.T @ A @ p) / (p.T @ A @ p)
            p = - r + B * p

            rs.append(np.linalg.norm(r))

            if np.linalg.norm(r) < self.epsilon:
                print("Converged at iteration: ", i + 1)
                break

        return x, rs
```

نرخ همگرایی الگوریتم بالا بر اساس condition number ماتریس A طبق رابطه زیر به دست می آید:

$$\|x_k - x^*\|_A \leq 2\left(\frac{\sqrt{\kappa(A)}-1}{\sqrt{\kappa(A)}+1}\right)^k \|x_0 - x^*\|_A$$

بنابراین انتظار داریم هر چه condition number زیاد باشد الگوریتم دیرتر همگرا شود.

برای محاسبه condition number مقادیر ویژه ماتریس A را با استفاده از

np.linalg.eigvals(A)

به دست آورده و نسبت بزرگترین به کوچکترین مقادیر ویژه را به عنوان condition number در نظر گرفتیم.

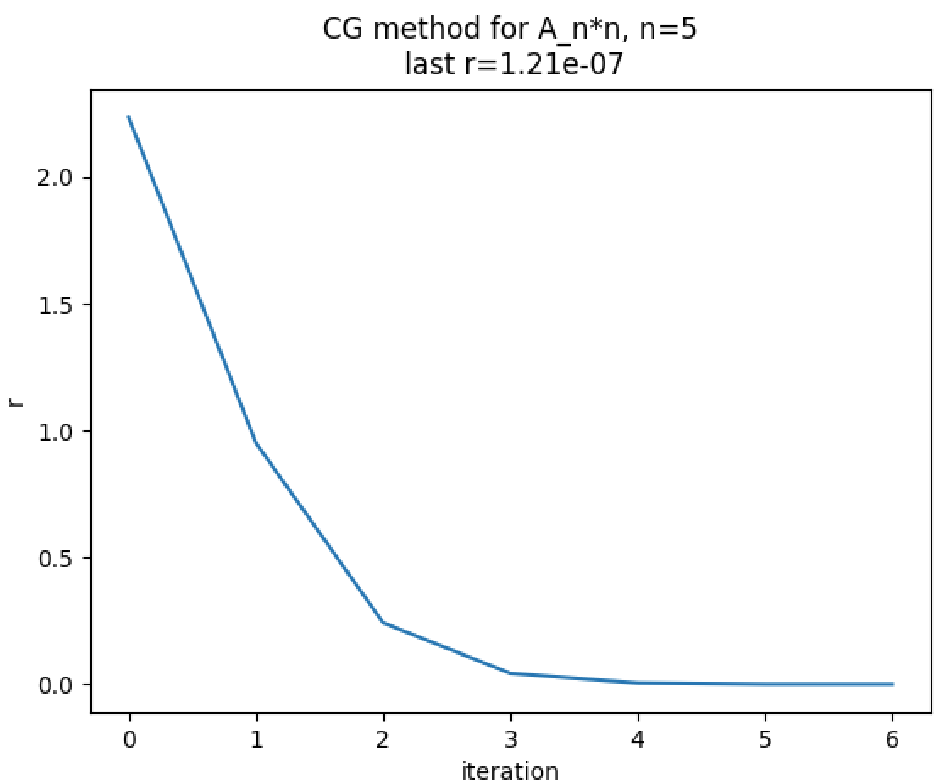
در جدول زیر اطلاعات زیر آورده شده است.

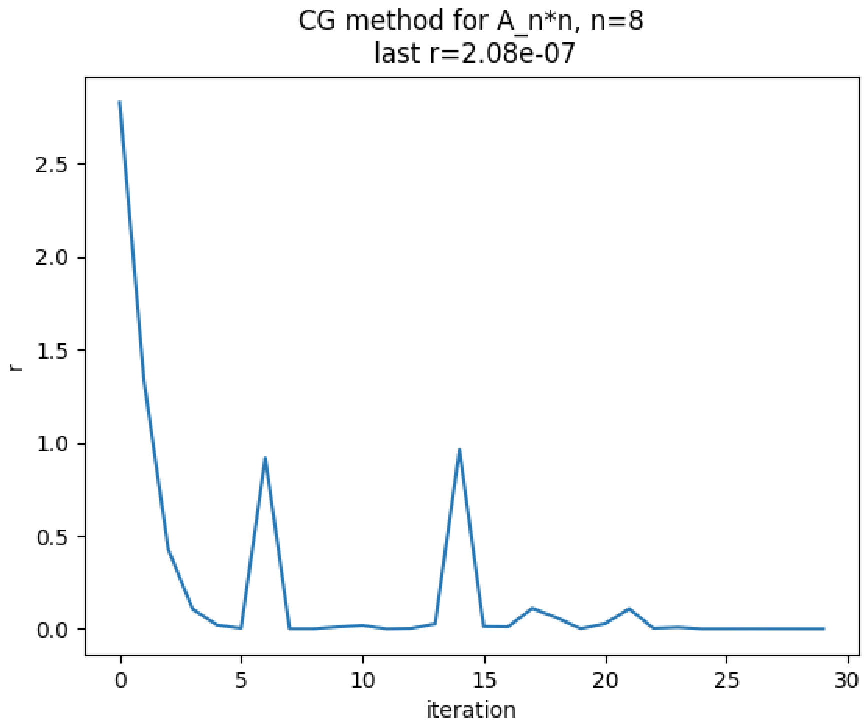
- تعداد iteration مورد نیاز برای همگرایی
- تعداد ابعاد ماتریس A
- عدد حالت
- آخرین r به دست آمده

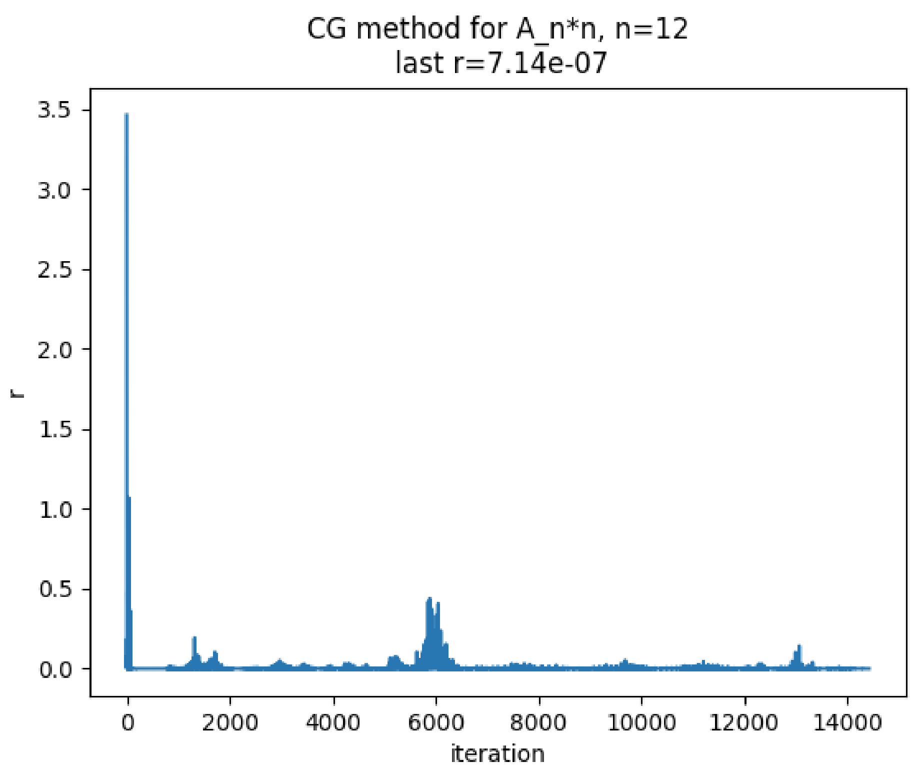
n	Iterations needed	Condition number	Last r
5	6	476607.25	1.21e-07
8	29	15257575797.52	2.08e-07
12	14427	1.6558444762	7.14e-07

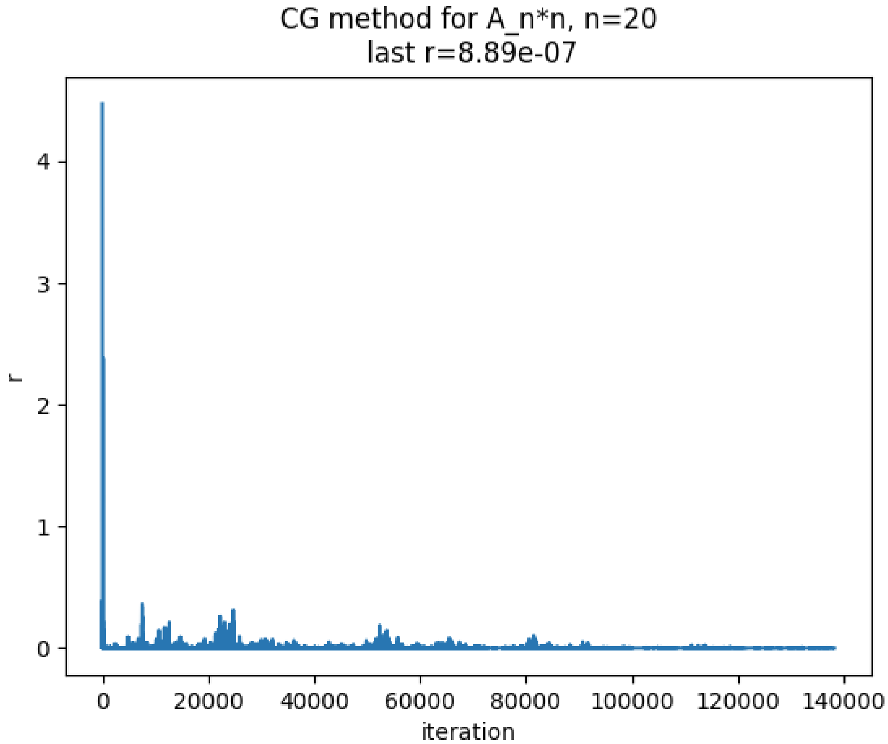
		684286e+16	
20	138333	-2.4593500896 239245e+17	8.89e-07

همان طور که مشاهده میشود با افزایش condition number تعداد iterationهای مورد نیاز برای همگرایی نیز افزایش یافته است. در زیر نمودارهای همگرایی برای مقادیر مختلف n آورده شده است




CG method for A_n*n, n=8
last r=2.08e-07
r
iteration
0.0
0.5
1.0
1.5
2.0
2.5
0
5
10
15
20
25
30


CG method for A_n*n, n=12
last r=7.14e-07
r
iteration
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
0
2000
4000
6000
8000
10000
12000
14000


CG method for A_n*n, n=20
last r=8.89e-07
r
4
3
2
1
0
0
20000
40000
60000
80000
100000
120000
140000
iteration

سوال ۲

از الگوریتم پیاده سازی شده در سوال اول برای این سوال نیز استفاده شد.

چون ماتریس A قطری است پس مقادیر ویژه این ماتریس برابر همان قطر اصلی آن است.

برای تولید ماتریس اول مقادیر قطر اصلی آن را به صورت عدد uniform بین ۱ تا ده نمونه برداری میکنیم.

```
n = 50
A1 = np.diag(np.random.uniform(1, 10, n))
```

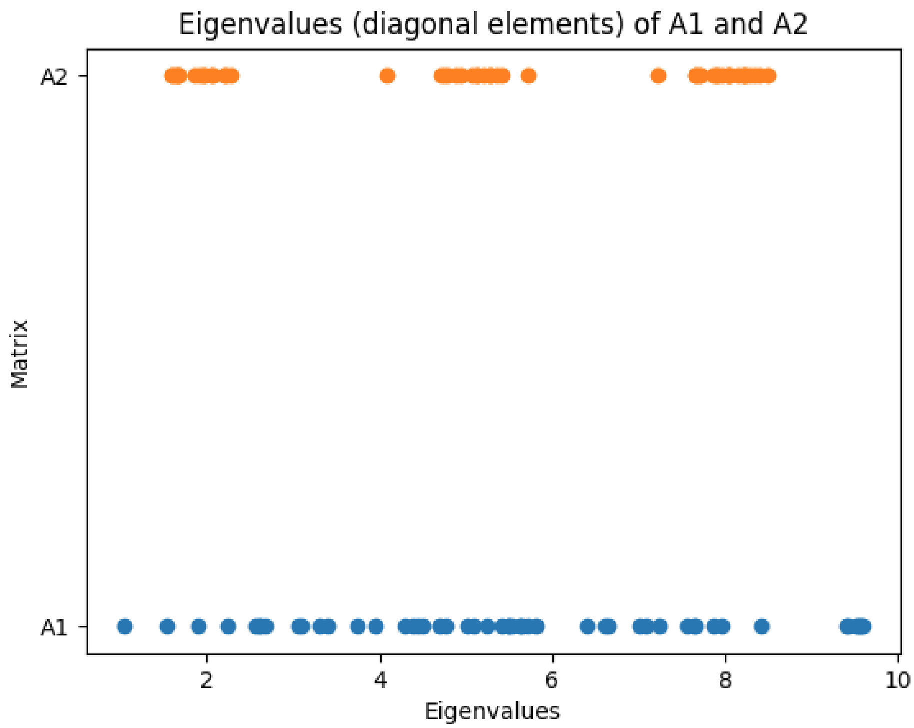
برای تولید ماتریس دوم نیز اطراف سه نقطه ۲و ۵و ۸ به ترتیب ۱۵و ۱۵و ۲۰ از توزیع نرمال با میانگین این نقاط و انحراف معیار ۰.۳ نمونه برداری میکنیم.

```
e1 = np.random.normal(2, 0.3, 15)
e2 = np.random.normal(5, 0.3, 15)
e3 = np.random.normal(8, 0.3, 20)

assert (e1 > 0).all() and (e2 > 0).all() and (e3 > 0).all()

A2 = np.diag(e1.tolist() + e2.tolist() + e3.tolist())
```

شکل زیر مقادیر قطر اصلی دو ماتریس را نشان میدهد.



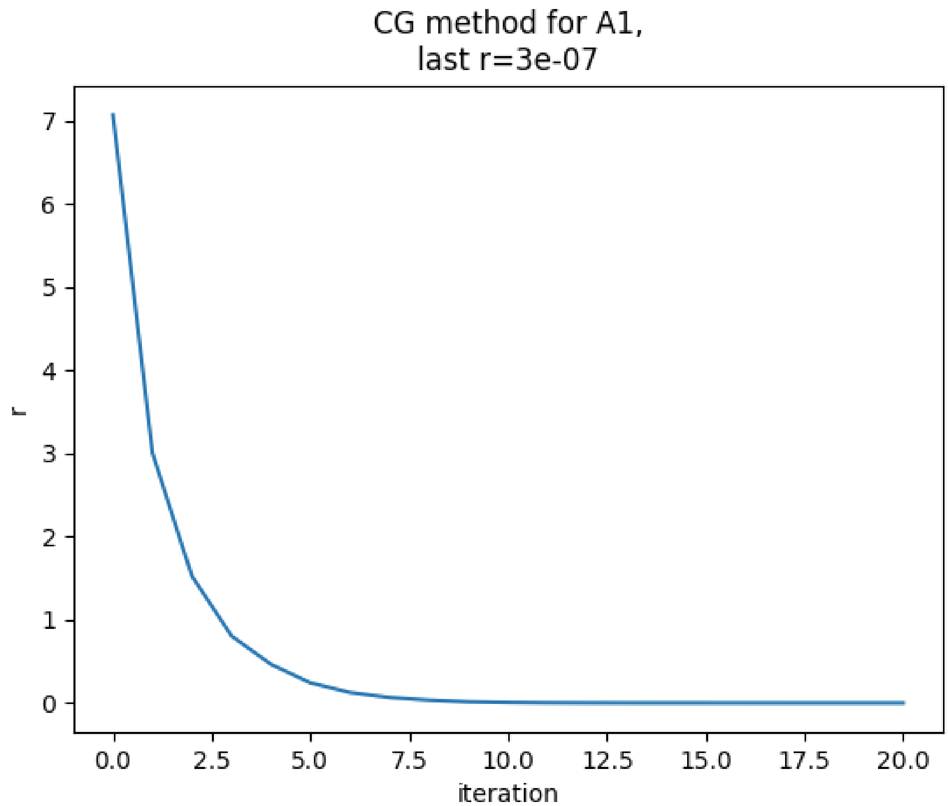
در جدول زیر اطلاعات بهینه سازی دو حالت زیر آورده شده است.

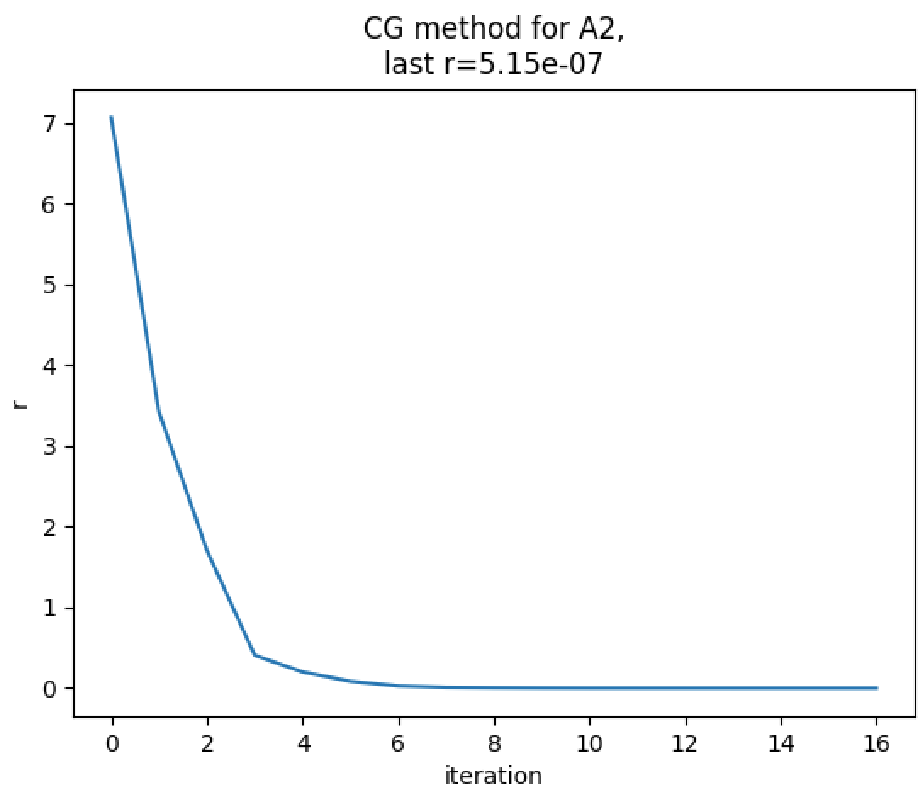
- تعداد iteration مورد نیاز برای همگرایی
- نوع ماتریس
- عدد حالت
- آخرین r به دست آمده

A	Iterations needed	Condition number	Last r
A1	20	9.29	3e-07
A2	16	5.35	5.15e-07

همان طور که مشاهده میشود ماتریس دوم چون تغییرات انحنای کمتری و اطراف نقاط کلاستر شده دارد condition number آن نیز کمتر است و به خاطر وابستگی نرخ همگرایی روش گرادیان مزدوج به condition number در دو ماتریس بالا معادله خطی ماتریس A2 به خاطر condition number کمتر سریع تر همگرا شده است.

در زیر نمودارهای همگرایی برای مقادیر دو نوع ماتریس آورده شده است:


CG method for A1,
last r=3e-07
r
iteration
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
0
1
2
3
4
5
6
7


CG method for A2,
last r=5.15e-07
r
iteration
0
2
4
6
8
10
12
14
16
0
1
2
3
4
5
6
7

سوال ۳

سوالات تحلیلی

سوال ۱

بردارهای $\{p_0, p_1, \ldots, p_l\}$ نسبت به ماتریس معین مثبت A مزدوج هستند اگر:

$$p_i^T A p_j = 0 \quad \forall i \neq j$$

فرض میکنیم بردار های p از مستقل نیستند (وابسته اند).

پس داریم:

به ازای $a_1, a_2, \ldots, a_l$ که همه صفر نباشند

$$\Sigma\, a_i p_i = 0$$

عبارت بالا را ابتدا در ماتریس A ضرب میکنیم. پس خواهیم داشت.

$$\Sigma a_i A p_i = 0$$

۱) سپس ماتریس p_j^T را در عبارت بالا ضرب میکنیم. چون $p_i^T A p_j = 0 \quad \forall i \neq j$ پس همه عبارات صفر خواهند شد به جز jامین عبارت و خواهیم داشت:

$$a_j p_j^T A p_j = 0$$

چون ماتریس A معین مثبت است پس $p_j^T A p_j > 0$ و طبق عبارت بالا خواهیم داشت:

$$a_j = 0$$

برای jهای دیگر نیز فرآیند ۱) را تکرار کنیم نتیجه میگیریم که همه‌ی aها مساوی صفر هستند که با فرض همه صفر نبودن ضریب های a برای وابسته بودن بردار ها در تناقض است.

پس نتیجه میگیریم بردارها از هم مستقل هستند.

سوال ۲

ابتدا اثبات عبارت 5.32 کتاب Nocedal توضیح داده میشود که کمک میکند تا بتوانیم از آن برای پیدا کردن نرخ همگرایی روش conjugate gradient استفاده کنیم.

۱) روش گرادیان مزدوج نقطه k+1ام مطابق با زیر به دست می آید.

$$x_{k+1} = x_0 + \alpha_0 p_0 + \cdots + \alpha_k p_k$$

چون میدانیم که جهت های تولید شده روی فضای زیر هستند:

$$\text{span}\{p_0, p_1, \ldots, p_k\} = \text{span}\{r_0, Ar_0, \ldots, A^k r_0\},$$

پس خواهیم داشت:

$$\begin{aligned} x_{k+1} &= x_0 + \alpha_0 p_0 + \cdots + \alpha_k p_k \\ &= x_0 + \gamma_0 r_0 + \gamma_1 A r_0 + \cdots + \gamma_k A^k r_0, \end{aligned}$$

در عبارت بالا یک چند جمله ای درجه K ماتریس A ظاهر شده است پس داریم:

$$x_{k+1} = x_0 + P_k^*(A) r_0.$$

که P برابر است با:

$$P_k^*(A) = \gamma_0 I + \gamma_1 A + \cdots + \gamma_k A^k,$$

حال r_0 ظاهر شده در عبارت را با استفاده از x^* حذف میکنیم تا عبارات بر حسب x به دست آیند.

داریم:

$$r_0 = Ax_0 - b = Ax_0 - Ax^* = A(x_0 - x^*),$$

ودر نتیجه جایگذاری آن در رابطه x_{K+1} و هم چنین کم کردن x^* از دو طرف تساوی خواهیم داشت.

$$x_{k+1} - x^* = x_0 + P_k^*(A)r_0 - x^* = [I + P_k^*(A)A](x_0 - x^*).$$

چون ماتریس معین مثبت بوده مقادیر ویژه ماتریس A مثبت هستند و داریم.

$$A = \sum_{i=1}^{n} \lambda_i v_i v_i^T.$$

چون بردارهای ویژه تشکیل یک span را میدهند داریم پس میتوان نقاط فضا را بر حسب ترکیب خطی بردارهای ویژه نوشت و داریم:

$$x_0 - x^* = \sum_{i=1}^{n} \xi_i v_i,$$

میتوان نشان داد که بردارهای ویژه ماتریس A بردارهای ویژه چند جمله ای درجه K ماتریس A نیز هستند. پس داریم.

$$P_k(A)v_i = P_k(\lambda_i)v_i,$$

بنابراین طبق این نکته و با جایگذاری

$$x_0 - x^* = \sum_{i=1}^{n} \xi_i v_i ,$$

در :

$$x_{k+1} - x^* = x_0 + P_k^*(A) r_0 - x^* = [I + P_k^*(A)A](x_0 - x^*).$$

خواهیم داشت:

$$x_{k+1} - x^* = \sum_{i=1}^{n} [1 + \lambda_i P_k^*(\lambda_i)] \xi_i v_i .$$

میدانیم که

$$\|z\|_A^2 = z^T A z = \sum_{i=1}^{n} \lambda_i (v_i^T z)^2;$$

و بنابراین با محاسبه نرم A رابطه بالا خواهیم داشت:

$$\|x_{k+1} - x^*\|_A^2 = \sum_{i=1}^{n} \lambda_i [1 + \lambda_i P_k^*(\lambda_i)]^2 \xi_i^2 .$$

روش گرادیان مزدوج نرم بالا را بین چند جمله ای های مختلف minimum میکند بنابراین داریم:

$$\|x_{k+1} - x^*\|_A^2 = \min_{P_k} \sum_{i=1}^{n} \lambda_i [1 + \lambda_i P_k(\lambda_i)]^2 \xi_i^2.$$

هم چنین داریم:

$$x_0 = \sum_{i=1}^{n} \xi_i v_i$$

$$Ax_0 = \sum_{i=1}^{n} \xi_i A v_i = \sum_{i=1}^{n} \xi_i \lambda_i v_i$$

$$f(x_0) = \frac{1}{2} x_0^T A x_0 = \left(\frac{1}{2}\right) \left(\sum_{i=1}^{n} \xi_i v_i\right)^T \sum_{i=1}^{n} \xi_i \lambda_i v_i = \frac{1}{2} \sum_{i=1}^{n} \lambda_i \xi_i^2$$

بنابراین عبارت بالا در عبارت زیر ظاهر شده است.

$$f(x^{k+1}) \leq \tfrac{1}{2} \sum_{i=1}^{n} \left(1 + \lambda_i P^k(\lambda_i)\right)^2 \lambda_i \xi_i^2,$$

بنابراین خواهیم داشت:

$$f(x^{k+1}) \leq \max_i \left(1 + \lambda_i P^k(\lambda_i)\right)^2 f(x^0),$$

حال چون مقادیر ویژه در بازه a, b قرار دارند چند جمله ای درجه k را میتوان به صورت زیر نوشت.

$$1 + \lambda P^k(\lambda) = \frac{2}{(a+b)\lambda_1 \cdots \lambda_k}\left(\frac{a+b}{2} - \lambda\right)(\lambda_1 - \lambda)\cdots(\lambda_k - \lambda).$$

با جایگذاری این چند جمله ای در نامساوی به دست آمده خواهیم داشت:

و چون

$$1 + \lambda_i P^k(\lambda_i) = 0,$$

خواهیم داشت:

$$f(x^{k+1}) \leq \max_{a \leq \lambda \leq b} \frac{\left(\lambda - \frac{1}{2}(a+b)\right)^2}{\left(\frac{1}{2}(a+b)\right)^2} f(x^0)$$

و در نتیجه:

$$f\left(x^{k+1}\right) \leq \left(\frac{b-a}{b+a}\right)^2 f\left(x^0\right)$$

سوال ۳

تابع f برابر است با:

$$f(x,\, y) = (1.5 - x + xy)^2 + \left(2.25 - x + xy^2\right)^2 + \left(2.625 - x + xy^3\right)^2$$

گرادیان آن برابر است با:

$$\nabla f = \begin{bmatrix} 2(1.5 - x + xy)(-1 + y) + 2\left(2.25 - x + xy^2\right)\left(-1 + y^2\right) + 2\left(2.625 - x + xy^3\right)\left(-1 + y^3\right) \\ 2(1.5 - x + xy)x + 2\left(2.25 - x + xy^2\right)x + 2\left(2.625 - x + xy^3\right)x \\ \end{bmatrix}$$

مطابق با الگوریتم زیر سه مرحله BFGS طی شد.

مقدار اولیه H نیز ماتریس identity و alpha نیز 0.1 در نظر گرفته شد.

Given starting point x_0, convergence tolerance $\epsilon > 0$,
    inverse Hessian approximation H_0;
$k \leftarrow 0$;
while $\|\nabla f_k\| > \epsilon$;
    Compute search direction

$$p_k = -H_k \nabla f_k;$$

    Set $x_{k+1} = x_k + \alpha_k p_k$ where α_k is computed from a line search
        procedure to satisfy the Wolfe conditions (3.6);
    Define $s_k = x_{k+1} - x_k$ and $y_k = \nabla f_{k+1} - \nabla f_k$;
    Compute H_{k+1} by means of (6.17);
    $k \leftarrow k + 1$;
end (while)

تابع H نیز بر اساس رابطه زیر به روز رسانی شد:

$$H_{k+1} = (I - \rho_k s_k y_k^T) H_k (I - \rho_k y_k s_k^T) + \rho_k s_k s_k^T$$

که در آن:

$$s_k = x_{k+1} - x_k$$

و

$$y_k = \nabla f_{k+1} - \nabla f_k$$

و

$$\rho_k = \frac{1}{y_k^T s_k}.$$

مقادیر H_{k+1}, y_k, s_k, x_{k+1}, p_k محاسبه شده با چهار رقم اعشار در هر مرحله آورده شده اند:

(1) **p0:** [2.3389 -5.8187] **x1:** [2.33389 -0.18187] **s0:** [[0.23389] [-0.58187]] **y0:** [[4.7688] [-10.4304]] **H1:** [[0.8524 0.3206] [0.3673 0.2024]]	**(2)** **p1:** [-0.5927 0.0409] **x2:** [2.27462 -0.17778] **s1:** [[-0.05927] [0.00409]] **y1:** [[-0.4232] [0.999]] **H2:** [[0.2589 1.9977] [0.0552 0.9202]]
(3) **p2:** [6.6976 3.2136] **x3:** [2.94438 0.14358] **s2:** [[0.66976] [0.32136]] **y2**: [[1.6039] [-7.9644]] **H3:** [[0.8374 -2.5197] [0.2696 -0.9334]]	