فاطمه صفری

ش.د: ۹۷۲۸۰۶۳

دکتر قریشی

توضیح پروژه امتحان میان ترم

حل دستگاه معادلات خطی با روش تکرار ژاکوبی

• روش تکرار ژاکوبی

طبق فرمول روش ژاکوبی حل معادلات خطی است که به صورت ترتیبی عمل می کند،

با بررسی تک تک معادلات در سیستم معادلات خطی $Ax = B$ جوابی تقریبی به دست می آوریم.

در معادله i ام، $\sum_{j=1}^{n} a_{ij} = b_i$ را برای i امین مجهول حل میکنیم

معادله $x_i^{(k)} = \left[-\sum_{j=1}^{n} \frac{a_{ij}}{a_{ii}} x_j^{(k-1)} + \frac{b_j}{a_{ii}} \right]$ را برای $1 \le i \le n$ بدست می آوریم.

همینطور فرض می کنیم که تمام عناصر قطری غیر صفر هستند. اگر اینطور نبود، معادلات را به گونه ای مرتب کنید که اینگونه باشد.

• چک کردن غالب قطری

با قطعه کد زیر ابتدا چک میکنیم ماتریس غالب قطری باشد، در غیر این صورت خطا چاپ خواهد شد:

```
For[i = 1, i ≤ Length[A0], i++,
  If[∑_{j=1}^{n} Abs[A[[i,j]]] > 2 Abs[A[[i,i]]], bool = 0;];];
If[bool == 1,
  Print["Diagonally dominant. \n Jacobi iteration is... "];,
  Print["Jacobi iteration Failed"];
  Return[Null];
  Break;];
```

خروجی کد: ”Jacobi iteration Failed“

• تابع و متغیرها

در این قسمت تابع را تعریف می کنیم

- ➔ -*epsilon*: برای ترک کردن حلقه و چاپ جواب آخر باید تفاضل ازین مقدار کوچکتر باشد
- ➔ -*iter:* برای نشان دادن حداکثر جایی که حلقه پیش خواهد رفت
- ➔ -*root:* با توجه به تعداد سطر های ماتریس وارد شده، یک بردار برای مجهول میسازد

```
Jacobiiteration[A0_, B0_, epsilon_, iter_] :=
 Module[{bool = 1, , n = Length[A0], A = N[A0], B = N[B0], eps,
   i, j, k = 0, P = Table[0, {i, 0, Length[A] - 1}], len = Length[A0]},

  root = Table["x"_i, {i, 1, n}];
  Print["Sove with Jacobi, AX = B"];
  Print[MatrixForm[A0], MatrixForm[root], " = ", MatrixForm[B0]];
```

خروجی کد:

```
Solve with Jacobi, AX = B
```

$$\begin{pmatrix} 4 & 3 & -1 \\ 1 & -4 & 1 \\ 1 & 1 & -5 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 4 \\ 1 \\ 1 \end{pmatrix}$$

• تکرار ژاکوبی

```
Print["Initial value is", root, " = ", P];
  X = P;
  eps = 1;
  While[And[eps > epsilon, iter > k],
    X = P;
    For[i = 1, i ≤ len, i++,
     P[[i]] = 1/A[[i,i]] (B[[i]] + A[[i,i]] X[[i]] - Sum[A[[i,j]] X[[j]], {j, 1, len}])];
    eps = Sqrt[(P - X).(P - X)];
    Print["P"_k+1 " = ", P];
    X = P;
    k += 1];
  Print["Result: \n For ", k, " iterations, "];
  Print["A X =", MatrixForm[A0], " ", MatrixForm[X], " = ", MatrixForm[A.X], " ≈ ", MatrixForm[B], " = B"];
  Return[
    P];];
```

- **جواب آخر**

با توجه به شرایط مسئله قرار میدهیم:

```
A = {{4, 3, -1}, {1, -4, 1}, {1, 1, -5}};
B = {4, 1, 1};
Jacobiiteration[A, B, 0.001, 50];
Print["\n\n"]
```

خروجی کد و جواب آخر:

Solve with Jacobi, AX = B

$$\begin{pmatrix} 4 & 3 & -1 \\ 1 & -4 & 1 \\ 1 & 1 & -5 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 4 \\ 1 \\ 1 \end{pmatrix}$$

Diagonally dominant.
Jacobi iteration is...

Initial value is $\{x_1, x_2, x_3\} = \{0, 0, 0\}$

$= P_1\{1., -0.25, -0.2\}$

$= P_2\{1.1375, -0.05, -0.05\}$

$= P_3\{1.025, 0.021875, 0.0175\}$

$= P_4\{0.987969, 0.010625, 0.009375\}$

$= P_5\{0.994375, -0.000664062, -0.00028125\}$

$= P_6\{1.00043, -0.00147656, -0.00125781\}$

$= P_7\{1.00079, -0.00020752, -0.000209766\}$

$= P_8\{1.0001, 0.000145801, 0.00011709\}$

Result:
For 8 iterations,

$$A\ X = \begin{pmatrix} 4 & 3 & -1 \\ 1 & -4 & 1 \\ 1 & 1 & -5 \end{pmatrix} \begin{pmatrix} 1.0001 \\ 0.000145801 \\ 0.00011709 \end{pmatrix} = \begin{pmatrix} 4.00073 \\ 0.999637 \\ 0.999664 \end{pmatrix} \approx \begin{pmatrix} 4. \\ 1. \\ 1. \end{pmatrix} = B$$