دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

دانشکده ریاضی و علوم کامپیوتر
استاد درس: دکتر مهدی دهقان
پاییز ۱۴۰۱

مقدمه ای بر جبرخطی فازی

مهرداد اکثری مهابادی
۹۹۱۲۰۱۰
جبر خطی عددی

فهرست مطالب

۱ چکیده

از گذشته تا کنون دستگاه های معادلات نقش بسیار مهمی در ریاضیات،فیزیک،آمار و دگر علوم مهندسی داشته اند. از آنجایی که در بسیاری از مسائل پارامتر های مسئله به صورت فازی [۱] و نه قطعی [۲] می باشند، توسعه روش های عددی برای حل دستگاه های فازی از اهمیت فراوانی برخوردار است.
در این مقاله ابتدا به بررسی مفاهیم اولیه دستگاه های فازی می پردازیم. در بخش بعد روشی مستقیم برای حل دستگاه های فازی $n \times n$ را بررسی می کنیم. در قسمت آخر پس از معرفی تعمیمی از روش های گاوس_سایدل و ژاکوبی در حل مسائل فازی با ارائه چند مثال عددی، بحث را به پایان می رسانیم.

[1] fuzzy
[2] crisp

۲ مقدمه

مفهوم اعداد فازی[3] برای اولین بار توسط زاده [۲، ۱]مطرح شد. یکی از کاربرد های اصلی اعداد فازی، سیستم های خطی[4] ست که در آن تعدادی یا همه پارامتر ها فازی می باشند.

۱.۲ اعداد فازی

هر دوتایی ترتیبی [5] به صورت $(\underline{u}(r), \overline{u}(r))$، $0 \leq r \leq 1$ که دارای سه شرط زیر باشد را یک عدد فازی تعریف می کنیم.

۱. $\overline{u}(r)$ یک تابع کراندار، چپ پیوسته، نزولی بر روی بازه $[0,1]$ است.

۲. $\underline{u}(r)$ یک تابع کراندار، راست پیوسته، صعودی بر روی بازه $[0,1]$ است.

۳. $\underline{u}(r) \leq \overline{u}(r)$ برای هر $r \in [0,1]$.

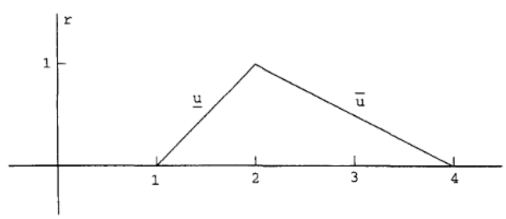
برای هر دو عدد فازی $x = (\underline{x}(r), \overline{x}(r))$ و $y = (\underline{y}(r), \overline{y}(r))$ داریم

۱. $x = y$ اگر و تنها اگر $\underline{x}(r) = \underline{y}(r)$ و $\overline{x}(r) = \overline{y}(r)$

۲. $x + y = (\underline{x}(r) + \underline{y}(r), \overline{x}(r) + \overline{y}(r))$

۳.

$$kx = \begin{cases} (k\underline{x}(r), k\overline{x}(r)) & k \geq 0 \\ (k\overline{x}(r), k\underline{x}(r)) & k < 0 \end{cases}$$



شکل ۱: نمایش عدد فازی $(r-1, 2-\frac{1}{2}r)$

[3]fuzzy numbers
[4]linear systems
[5]ordered pair

۲.۲ اعداد قطعی

اگر داشته باشیم $\underline{u}\left(r\right)=\overline{u}\left(r\right)=\alpha, 0\leq r\leq 1$ به α عددی قطعی [۶] گفته می شود.

۳.۲ دستگاه فازی

دستگاه خطی $n\times n$ زیر

$$\begin{aligned} a_{11}x_1+a_{12}x_2+\ldots+a_{1n}x_n&=y_1\\ a_{21}x_1+a_{22}x_2+\ldots+a_{2n}x_n&=y_2\\ &\vdots\\ a_{n1}x_1+a_{n2}x_2+\ldots+a_{nn}x_n&=y_n \end{aligned} \tag{۱}$$

که ماتریس ضرایب آن $A=\left(a_{ij}\right), 1\leq i,j\leq n$ یک ماتریس با درایه های قطعی و y برداری با اعضای فازی است را یک دستگاه خطی فازی [۷] گفته می شود.
به بردار فازی[۸] $\left(x_1,x_2,\ldots,x_n\right)^T$ که $x_i=\left(\underline{x}_i\left(r\right),\overline{x}_i\left(r\right)\right), 1\leq i\leq n, 0\leq r\leq 1$
جواب دستگاه فازی گفته می شود اگر در شروط زیر صدق کند

$$\begin{aligned} \overline{\sum_{i=0}^{n}a_{ij}x_j}&=\sum_{i=0}^{n}a_{ij}\overline{x_j}=\overline{y_i}\\ \underline{\sum_{i=0}^{n}a_{ij}x_j}&=\sum_{i=0}^{n}a_{ij}\underline{x_j}=\underline{y_i} \end{aligned} \tag{۲}$$

۴.۲ جواب دستگاه فازی

اگر $X=\left\{\left(\underline{x}_i,-\overline{x}_i\right), 1\leq i\leq n\right\}$ جواب یکتا دستگاه ۱۰ باشد، بردار فازی $u=\left\{\left(\underline{u}_i,\overline{u}_i\right), 1\leq i\leq n\right\}$ که به صورت

$$\underline{u}_i(r)=\min\{\underline{x}_i(r),\overline{x}_i(r),\underline{x}_i(1)\} \tag{۳}$$
$$\overline{u}_i(r)=\max\{\underline{x}_i(r),\overline{x}_i(r),\underline{x}_i(1)\} \tag{۴}$$

تعریف می شود، یک جواب فازی[۹] دستگاه $SX=Y$ می نامیم. با توجه به تعریف بالا اگر $\left(\underline{x}_i,\overline{x}_i\right)$ اعداد فازی باشند، به راحتی می توان دید که $\underline{u}_i=\underline{x}_i$ و $\overline{u}_i=\overline{x}_i$. در این حالت به U جواب فازی قوی [۱۰] و در غیر اینضورت جوا ب فازی ضعیف [۱۱] گفته می شود.

[6] crisp
[7] fuzzy linear system(FLS)
[8] fuzzy number vector
[9] fuzzy solution
[10] strong fuzzy solution
[11] weak fuzzy solution

۳ حل دستگاه های فازی

در این قسمت ابتدا طبق رویکرد [۳] روشی عمومی برای حل مستقیم دستگاه ارائه می کنیم. در بخش ۲ به بررسی روش های تکراری گاوس_سایدل و ژاکوبی می پردازیم. در بخش آخر به ارائه مثال های عددی می پردازیم.

۱.۳ روش مستقیم

در اینجا با تبدیل دستگاه $n \times n$ به دستگاه قطعی $2n \times 2n$ روشی برای حل دستگاه فازی ارائه می کنیم. در واقع ما قصد داریم دستگاه را به صورتی بازنویسی کنیم که دیگر عنصر فازی نداشته باشیم. به عبارتی عناصر دستگاه جدید دیگر به فرم زوج مرتبی نخواهند بود و به راحتی می توانیم دستگاه جدید را حل کنیم. بردار $y = (\underline{y}_1, \underline{y}_2, \ldots, \underline{y}_n, \overline{y}_1, \ldots, \overline{y}_n)^T$ را بردار سمت راست و $\underline{x}_i, (-\overline{x}_i)$ متغیر های دستگاه قرار می دهیم. دستگاه معادلات جدید را به صورت زیر فرمول بندی می کنیم.

$$
\begin{aligned}
\underline{y}_1 &= s_{11}\underline{x}_1 + \ldots + s_{1n}\underline{x}_n + s_{1,n+1}\left(\overline{x}_1\right) + \ldots + s_{1,2n}\left(\overline{x}_n\right) \\
\underline{y}_2 &= s_{21}\underline{x}_1 + \ldots + s_{2n}\underline{x}_n + s_{2,n+1}\left(\overline{x}_1\right) + \ldots + s_{2,2n}\left(\overline{x}_n\right) \\
&\vdots \\
\underline{y}_n &= s_{n1}\underline{x}_1 + \ldots + s_{nn}\underline{x}_n + s_{n,n+1}\left(\overline{x}_1\right) + \ldots + s_{n,2n}\left(\overline{x}_n\right) \\
-\overline{y}_1 &= s_{n+1,1}\underline{x}_1 + \ldots + s_{n+1,n}\underline{x}_n + s_{n+1,n+1}\left(\overline{x}_1\right) + \ldots + s_{n+1,2n}\left(\overline{x}_n\right) \\
-\overline{y}_2 &= s_{n+2,1}\underline{x}_1 + \ldots + s_{n+2,n}\underline{x}_n s_{n+2,n+1}\left(\overline{x}_1\right) + \ldots + s_{n+2,2n}\left(\overline{x}_n\right) \\
&\vdots \\
-\overline{y}_n &= s_{2n,1}\underline{x}_1 + \ldots + s_{2n,n}\underline{x}_n + s_{2n,n+1}\left(\overline{x}_1\right) + \ldots + s_{2n,2n}\left(\overline{x}_n\right)
\end{aligned}
\tag{۵}
$$

که S به صورت زیر به دست می آیند

$$
\begin{cases}
s_{ij} = a_{ij}, \quad s_{i+n,j+n} = a_{ij} & a_{ij} \geq 0 \\
s_{i,j+n} = -a_{i,j}, \quad s_{i+n,j} = -a_{ij} & a_{ij} < 0
\end{cases}
\tag{۶}
$$

توجه کنید که مقدار s_{ij} برای مقادیری که توسط رابطه بالا مشخص نمی شوند، صفر است.
با توجه به روابط بالا دستگاه معادلات جدید به صورت $\mathbf{SX} = \mathbf{Y}$ است که در آن $\mathbf{S} = (s_{ij})$ و $1 \leq i, j \leq 2n$ است و

$$
X = \begin{pmatrix} \underline{x}_1 \\ \underline{x}_2 \\ \vdots \\ \underline{x}_n \\ -\overline{x}_1 \\ \vdots \\ -\overline{x}_n \end{pmatrix}, Y = \begin{pmatrix} \underline{y}_1 \\ \underline{y}_2 \\ \vdots \\ \underline{y}_n \\ -\overline{y}_1 \\ \vdots \\ -\overline{y}_n \end{pmatrix}
$$

مثال ۱.۳. دستگاه 2×2 زیر را در نظر بگیرید.

$$x_1 - x_2 = y_1$$
$$x_1 + 2x_2 = y_2$$

با توجه به روابط ۶ می توان ماتریس S را به صورت زیر ساخت.

$$\begin{pmatrix} 1 & 0 & 0 & 1 \\ 1 & 2 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 2 \end{pmatrix}$$

به راحتی می توان دید که فرم ماتریس S به صورت زیر است.

$$S = \begin{pmatrix} B & C \\ C & B \end{pmatrix}$$

که در آن B شامل درایه های مثبت ماتریس ضرایب A و C از قدرمطلق درایه های منفی تشکیل شده. پس می توان گفت $A = B - C$.
دستگاه جدیدی که به دست آوردیم تماما از درایه های قطعی تشکیل شده. تنها در صورتی می توان برای این دستگاه جواب یکتا به دست آورد که ماتریس S نامنفرد باشد. در اینجا با دو سوال رو به رو می شویم.

۱. آیا ماتریس S نامنفرد است؟

۲. آیا جواب به دست آمده می تواند تشکیل برداری فازی برای دستگاه اولیه دهد؟

اگر که پاسخ سوال ۱ مثبت باشد، پاسخ سوال دوم در صورتی مثبت است که برای هر i، $(\underline{x}i\,(r)\,,\overline{x}_i\,(r))$ عددی فازی باشد.
در ادامه ابتدا به سوال اول پاسخ می دهیم. باید بررسی کنیم که تحت چه شرایطی ماتریس S نامنفرد است. از آنجایی که S را با استفاده از ماتریس ضرایب A ساختیم، حدس می زنیم که روابطی بین معکوس پذیری A و S وجود داشته باشد. مثال بعد نشان می دهد که نامنفرد بودن A شرط کافی برای معکوس پذیری S نیست.

مثال ۲.۳. دستگاه 2×2 زیر را در نظر بگیرید.

$$x_1 - x_2 = y_1$$
$$x_1 + x_2 = y_2$$

مشابه مقال قبل ماتریس S را به دست می آوریم.

$$S = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix}$$

می توان دید که S وارون پذیر نیست. در قضیه بعد شرط لازم و کافی برای معکوس پذیری S را بیان میکنیم.

قضیه ۱.۳. ماتریس S نامنفرد است اگر و تنها اگر ماتریس $A = B - C$ و $B + C$ نامنفرد باشند.
اثبات: با توجه به اینکه با جمع ترکیب خطی از سطر ها یا ستون های یک ماتریس تاثیری در دترمینان آن ندارد، می توان به راحتی روابط زیر را به دست آورد.

$$S = \begin{pmatrix} B & C \\ C & B \end{pmatrix} \to S_1 = \begin{pmatrix} B+C & B+C \\ C & B \end{pmatrix} \to S_2 = \begin{pmatrix} B+C & 0 \\ C & B-C \end{pmatrix}$$

$$det(S) = det(S_1) = det(S_2) = det(B+C).det(B-C) = det(B+C).det(A)$$

$$\to \qquad det(S) \neq 0 \longleftrightarrow det(A) \neq 0 \quad and \quad det(B+C) \neq 0$$

قضیه ۱.۳ از اهمیت فراوانی برخوردار است از آنجایی که شرطی لازم و کافی برای معکوس پذیری S بیان می کند. برای حل دستگاه فازی بعد از اینکه فهمیدیم S معکوس پذیر است باید S^{-1} را به دست بیاوریم. در ادامه به بیان قضایای و روش هایی برای به دست آوردن S^{-1} می پردازیم.

قضیه ۲.۳. وارون ماتریس S در صورت وجود، فرمی مشابه S دارد.

$$S^{-1} = \begin{pmatrix} E & D \\ D & E \end{pmatrix}$$

برای دیدن اثبات قضیه فوق به [۳] مراجعه کنید. با توجه به قضیه ۲.۳ S^{-1} را به صورت زیر به دست می آوریم.

$$SS^{-1} = \begin{pmatrix} B & C \\ C & B \end{pmatrix} \begin{pmatrix} D & E \\ E & D \end{pmatrix}$$

$$BD + CE = I, \quad CD + BE = 0 \tag{۷}$$

با جمع و تفریق دو معادله در رابطه ۷ روابط زیر را به دست می آوریم.

$$D + E = (B+C)^{-1}, \quad D - E = (B-C)^{-1} \tag{۸}$$

و نتیجتا

$$E = \frac{1}{2}[(B+C)^{-1} - (B-C)^{-1}], \quad D = \frac{1}{2}[(B+C)^{-1} + (B-C)^{-1}] \tag{۹}$$

با توجه به فرض نامنفرد بودن S، که معادل با نامنفرد بودن $B + C$ و $B - C$است، داریم

$$X = S^{-1}Y \tag{۱۰}$$

که ممکن است X برداری فازی نباشد. قضیه بعد شرط لازم و کافی برای فازی بودن X را بیان می کند.

قضیه ۳.۳. بردار X ر رابطه ۱۰ فازی است، اگر و تنها اگر S^{-1} مثبت باشد، به عبارتی باید داشته باشیم

$$\left(S^{-1}\right)_{ij} \geq 0, \quad 1 \leq i, j \leq n \tag{۱۱}$$

اثبات: قرار می دهیم $S^{-1} = (t)_{ij}\,, 1 \leq i,j \leq n$. از رابطه ۱۰ داریم

$$\underline{x}_i = \sum_{j=1}^{n} t_{ij}\underline{y}_i - \sum_{j=1}^{n} t_{i,n+j}\overline{y}_i, \quad 1 \leq i \leq n \tag{۱۲}$$

$$-\overline{x}_i = \sum_{j=1}^{n} t_{n+i,j}\underline{y}_i - \sum_{j=1}^{n} t_{n+i,n+j}\overline{y}_i, \quad 1 \leq i \leq n \tag{۱۳}$$

با توجه به قضیه ۲.۳ می توان معادلی برای رابطه ۱۱ به صورت زیر نوشت

$$\overline{x}_i = -\sum_{j=1}^{n} t_{i,n+j}\underline{y}_i + \sum_{j=1}^{n} t_{i,j}\overline{y}_i, \quad 1 \leq i \leq n \tag{۱۴}$$

با کم کردن رابطه۱۰ از ۱۲ رابطه زیر را به دست می آوریم

$$\overline{x}_i - \underline{x}_i = \sum_{j=1}^{n} t_{i,j}\left(\overline{y}_i - \underline{y}_i\right) + \sum_{j=1}^{n} t_{i,n+j}\left(\overline{y}_i - \underline{y}_i\right), \quad 1 \leq i \leq n \tag{۱۵}$$

از آنجایی که y برداری فازی است، برای هر $0 \leq i \leq n$ داریم $\overline{y}_i - \underline{y}_i \geq 0$. در نتیجه با توجه به رابطه ۱۳ می توان گفت $\overline{x}_i - \underline{x}_i \geq 0$ اگر و تنها اگر برای هر $0 \leq j \leq n$ داشته باشیم $t_{ij} \geq 0, 1 \leq j \leq n$. پس ثابت کردیم که شرط سوم فازی بودن برقرار است. بعلاوه از روابط ۱۰ و ۱۲ و با توجه به اینکه $\overline{y}_i$ نزولی یکنوا و $\underline{y}_i$ صعودی یکنواست می توان گفت شرط ۱ و ۲ هم برای $\underline{x}_i$ و $\overline{x}_i$ برقرار است. دقت کنید که از چپ و راست پیوسته و کراندار بودن $\underline{x}_i$ و $\overline{x}_i$ به وضوح برقرار است چرا که، با توجه به روابط ۱۰ و ۱۲، ترکیبی خطی از $\underline{y}_i$ و $\overline{y}_i$ ها هستند.

متاسفانه احتمال اینکه S^{-1} شروط قضیه قبل را دارا باشد در عمل کم است.

۲.۳ یک روش جایگزین

از آنجایی که تشکیل یک ماتریس $2n \times 2n$ ممکن از لحاظ محاسباتی هزینه بر باشد، عزتی [۴] با اقتباس از این روش، روشی جدید برای حل دستگاه های فازی ارائه داد که بجای حل یک سیستم $2n \times 2n$ با حل دو دستگاه $n \times n$ جواب دستگاه معادلات را به دست می آورد.

۳.۳ روش های تکراری

در جبرخطی دیدیم معکوس یک ماتریس معمولا هزینه سنگین محاسباتی دارد و برای حل این مشکل از روش های تکراری برای حل دستگاه معادلات استفاده کردیم. در این بخش قصد داریم به معرفی روش های تکراری برای حل دستگاه های فازی بپردازیم.

۱.۳.۳ همگرایی

قضیه ۴.۳. اگر ماتریس ضرایب A غالب قطری اکید باشد، هر دو روش ژاکوبی و گاوس_سایدل در حل معادله دستگاه ۱ همگرا هستند.[۶]

قضیه ۵.۳. ماتریس A غالب قطری اکید است اگر و تنها اگر ماتریس S غالب قطری اکید باشد.

اثبات: با توجه به مطالب گفته شده در قسمت قبل ماتریس B و C به ترتیب عناصر مثبت و منفی A هستند و اگر درایه ای در C ناصفر باشد درایه متناظر با آن در B حتما صفر است. به طور مشابه اگر درایه ای نامنفی در B باشد درایه متناظر آن در C حتما صفر است. در واقع با قراردادن ماتریس B و C روی یکدیگر قدرمطلق ماتریس A را به دست آورده ایم. به این ترتیب واضح است که اگر A غالب قطری اکید باشد،S هم غالب قطری اکید است و بالعکس.

۲.۳.۳ ژاکوبی

بدون از دست رفتن کلیت، فرض کنید $s_{ii} > 0, 1 \leq i \leq 2n$. قرار می دهیم $S = D+L+U$ که در آن

$$D = \begin{pmatrix} D_B & 0 \\ 0 & D_B \end{pmatrix} \quad L = \begin{pmatrix} L_B & 0 \\ C & L_B \end{pmatrix} \quad U = \begin{pmatrix} U_B & C \\ 0 & U_B \end{pmatrix}$$

دستگاه $SX = Y$ را به فرم بلوکی زیر می نویسیم.

$$\begin{pmatrix} B & C \\ C & B \end{pmatrix} \begin{pmatrix} \underline{X} \\ -\overline{X} \end{pmatrix} = \begin{pmatrix} \underline{Y} \\ -\overline{Y} \end{pmatrix}$$

می توان فرم بالا را به صورت زیر نمایش داد.

$$\begin{pmatrix} B & -C \\ -C & B \end{pmatrix} \begin{pmatrix} \underline{X} \\ \overline{X} \end{pmatrix} = \begin{pmatrix} \underline{Y} \\ \overline{Y} \end{pmatrix}$$

با توجه به فرم ژاکوبی داریم

$$\begin{pmatrix} D_B & 0 \\ 0 & D_B \end{pmatrix} \begin{pmatrix} \underline{X} \\ \overline{X} \end{pmatrix} + \begin{pmatrix} L_B + U_B & -C \\ -C & L_B + U_B \end{pmatrix} \begin{pmatrix} \underline{X} \\ \overline{X} \end{pmatrix} = \begin{pmatrix} \underline{Y} \\ \overline{Y} \end{pmatrix}$$

با ضرب در D_B^{-1} روابط تکراری زیر را به دست می آوریم

$$\begin{aligned} \underline{X}^{k+1} &= D_B^{-1}\underline{Y} - D_B^{-1}(L_B + U_B)\underline{X}^k + D_B^{-1}C\overline{X}^k \\ \overline{X}^{k+1} &= D_B^{-1}\overline{Y} - D_B^{-1}(L_B + U_B)\overline{X}^k + D_B^{-1}C\underline{X}^k \end{aligned}$$

می توان روابط بالا را به فرم فشرده ماتریسی هم نوشت. بدین صورت که $X^{k+1} = M_J X^k + C_J$ که M_J و C_J به فرم زیر هستند

$$M_J = \begin{pmatrix} -D_B^{-1}(L_B + U_B) & D_B^{-1}C \\ D_B^{-1}C & -D_B^{-1}(L_B + U_B) \end{pmatrix} \quad C_J = \begin{pmatrix} D_B^{-1}\underline{Y} \\ D_B^{-1}\overline{Y} \end{pmatrix} \quad X = \begin{pmatrix} \underline{X} \\ \overline{X} \end{pmatrix}$$

۳.۳.۳ گاوس_سایدل

با رویکردی مشابه روش ژاکوبی روابط تکراری را برای روش گاوس_سایدل به دست می آوریم. ابتدا مشابه $SX = Y$ را برای فرم گاوس سایدل بازنویسی می کنیم.

$$\begin{pmatrix} L_B + D_B & 0 \\ -C & L_B + D_B \end{pmatrix} \begin{pmatrix} \underline{X} \\ \overline{X} \end{pmatrix} + \begin{pmatrix} U_B & -C \\ 0 & U_B \end{pmatrix} \begin{pmatrix} \underline{X} \\ \overline{X} \end{pmatrix} = \begin{pmatrix} \underline{Y} \\ \overline{Y} \end{pmatrix}$$

کافیست معادلات بالا را به صورت تکراری بازنویسی کنیم

$$\begin{aligned} \underline{X}^{k+1} &= (L_B + D_B)^{-1}\,\underline{Y} - (L_B + D_B)^{-1}\,U_B\underline{X}^k + (L_B + D_B)^{-1}\,C\overline{X}^k \\ \overline{X}^{k+1} &= (L_B + D_B)^{-1}\,\overline{Y} - (L_B + D_B)^{-1}\,U_B\overline{X}^k + (L_B + D_B)^{-1}\,C\underline{X}^k \end{aligned}$$

مشابه قسمت قبل می توان روابط بالا را به فرم ماتریسی هم بازنویسی کرد. به طوری که $X^{k+1} = M_{GS}X^k + C_{GS}$ که در آن M_{GS} و C_{GS} به صورت زیر تعریف می شوند

$$M_{GS} = \begin{pmatrix} -(L_B + D_B)^{-1}\,U_B & (L_B + D_B)^{-1}\,C \\ (L_B + D_B)^{-1}\,C & -(L_B + D_B)^{-1}\,U_B \end{pmatrix} \quad C_{GS} = \begin{pmatrix} (L_B + D_B)^{-1}\,\underline{Y} \\ (L_B + D_B)^{-1}\,\overline{Y} \end{pmatrix}$$

اگر شرایط قضیه ۴.۳ برقرار باشد، هر دو روش برای هر X^0 دلخواهی همگرا هستند.

۴.۳.۳ معیار توقف

در سیستم های فازی با توجه به اینکه اعداد فازی به صورت زوج های مرتب هستند، طبیعی است که معیار توقف به صورت دو نامساوی باشد و زمانی توقف کنیم که شرایط هر دو نامساوی برقرار شده باشد.

$$\frac{||\overline{X}^{k+1} - \overline{X}^k||}{||\overline{X}^k||} < \epsilon \quad \frac{||\underline{X}^{k+1} - \underline{X}^k||}{||\underline{X}^k||} < \epsilon \tag{۱۶}$$

۴.۳ نتایج عددی

مثال ۳.۳. دستگاه 2×2 زیر را در نظر بگیرید.

$$
\begin{aligned}
x_1 - x_2 &= (r, 2-r) \\
x_1 + 3x_2 &= (4+r, 7-2r)
\end{aligned}
$$

با توجه به روابط ۶ می توان ماتریس S را به صورت زیر ساخت.

$$
S = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 1 & 3 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 3 \end{pmatrix}
$$

با توجه به رابطه ۱۰ می توان جواب دستگاه را به صورت زیر به دست آورد

$$
X = \begin{pmatrix} \underline{x}_1 \\ \underline{x}_2 \\ \vdots \\ \underline{x}_n \\ -\overline{x}_1 \\ \vdots \\ -\overline{x}_n \end{pmatrix} = S^{-1}y = \begin{pmatrix} 1.125 & -0.125 & 0.375 & -0.375 \\ -0.375 & 0.375 & -0.125 & 0.125 \\ 0.375 & -0.375 & 1.125 & -0.125 \\ -0.125 & 0.125 & -0.375 & 0.375 \end{pmatrix} \begin{pmatrix} r \\ 4+r \\ r-2 \\ 2r-7 \end{pmatrix}
$$

$$
\begin{aligned}
\underline{x}_1 &= 1.375 + 0.625r, \quad \overline{x}_1 = 2.875 - 0.875r \\
\underline{x}_2 &= 0.875 + 0.125r, \quad \overline{x}_2 = 1.375 - 0.375r
\end{aligned}
$$

می توان دید که دستگاه جواب فازی قوی دارد.
از آنجایی که ماتریس S غالب قطری اکید می باشد، طبق قضیه **۴.۳** هر دو روش ژاکوبی و گاوس سایدل همگرا هستند. برای محاسبات عددی نیاز به حاسبات سمبلک [12] داریم. در اینجا ما از sympy [۷] استفاده کرده ایم. برای دیدن جزئیات پیاده سازی به پیوست مراجعه کنید. برای روش ژاکوبی داریم

$$
M_J = \begin{pmatrix} 0 & 0 & 0 & 1 \\ -\frac{1}{3} & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1/3 & 0 \end{pmatrix} \quad C_J = \begin{pmatrix} r \\ \frac{r}{3} + \frac{4}{3} \\ 2-r \\ \frac{7}{3} - \frac{2r}{3} \end{pmatrix}
$$

$$
X^{k+1} = M_J X^k + C_J
$$

$$
X_0 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}
$$

[12] symbolic

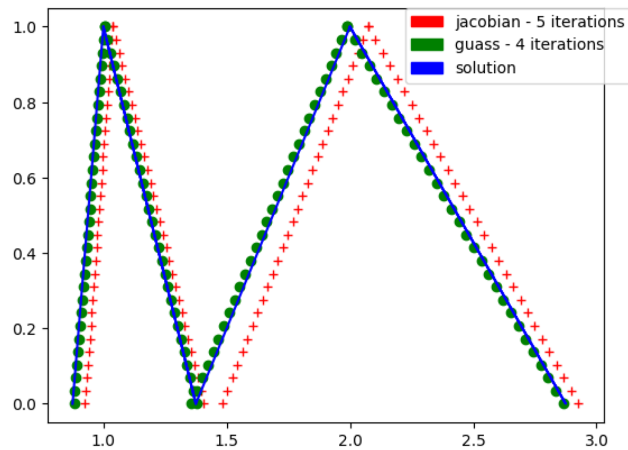
$$X_1 = \begin{pmatrix} r \\ \frac{r}{3} + \frac{4}{3} \\ 2 - r \\ \frac{7}{3} - \frac{2r}{3} \end{pmatrix}$$
$$X_2 = \begin{pmatrix} \frac{r}{3} + \frac{7}{3} \\ \frac{4}{3} \\ \frac{10}{3} - \frac{2r}{3} \\ \frac{5}{3} - \frac{r}{3} \end{pmatrix}$$
$$X_3 = \begin{pmatrix} \frac{2r}{3} + \frac{5}{3} \\ \frac{2r}{9} + \frac{5}{9} \\ \frac{10}{3} - r \\ \frac{11}{9} - \frac{4r}{9} \end{pmatrix}$$
$$X_4 = \begin{pmatrix} \frac{5r}{9} + \frac{11}{9} \\ \frac{r}{9} + \frac{7}{9} \\ \frac{23}{9} - \frac{7r}{9} \\ \frac{11}{9} - \frac{r}{3} \end{pmatrix}$$
$$X_5 = \begin{pmatrix} \frac{2r}{3} + \frac{11}{9} \\ \frac{4r}{27} + \frac{25}{27} \\ \frac{25}{9} - \frac{8r}{9} \\ \frac{40}{27} - \frac{11}{27} \end{pmatrix}$$

می توان دید ژاکوبی با ۵ تکرار به جواب همگرا شده. برای گاوس‌ـ‌سایدل به طور مشابه داریم

$$M_{GJ} = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -\frac{1}{3} \\ 0 & 1 & 0 & 0 \\ 0 & -\frac{1}{3} & 0 & 0 \end{pmatrix} \quad C_{GJ} = \begin{pmatrix} r \\ \frac{4}{3} \\ 2 - r \\ \frac{5}{3} - \frac{r}{3} \end{pmatrix}$$
$$X^{k+1} = M_J X^k + C_J$$
$$X_0 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$
$$X_1 = \begin{pmatrix} r \\ \frac{4}{3} \\ 2 - r \\ \frac{5}{3} - \frac{r}{3} \end{pmatrix}$$
$$X_2 = \begin{pmatrix} \frac{2r}{3} + \frac{5}{3} \\ \frac{r}{9} + \frac{7}{9} \\ \frac{10}{3} - r \\ \frac{11}{9} - \frac{r}{3} \end{pmatrix}$$
$$X_3 = \begin{pmatrix} \frac{2r}{3} + \frac{11}{9} \\ \frac{r}{9} + \frac{25}{27} \\ \frac{25}{9} - \frac{8r}{9} \\ \frac{38}{27} - \frac{10}{27} \end{pmatrix}$$

$$X_4 = \begin{pmatrix} \frac{17r}{27} + \frac{38}{27} \\ \frac{10r}{81} + \frac{70}{81} \\ \frac{79}{27} - \frac{8r}{9} \\ \frac{110}{81} - \frac{10r}{27} \end{pmatrix}$$

در شکل زیر جواب فازی دستگاه معادلات به همراه جواب های تقریبی به دست آمده از روش های ژاکوبی و گاوس_سایدل آمده. می توان دید که گاوس_سایدل تقریب بهتری با تعداد تکرار کمتر داشته.



شکل ۲: مقایسه روش های تقریبی، معیار توقف $\epsilon = 10^{-2}$

مثال ۴.۳. دستگاه 3×3 زیر را در نظر بگیرید.

$$\begin{aligned} x_1 + x_2 - x_3 &= (r, 2-r) \\ x_1 - 2x_2 + x_3 &= (2+r, 3) \\ 2x_1 + x_2 + 3x_3 &= (-2, -1-r) \end{aligned}$$

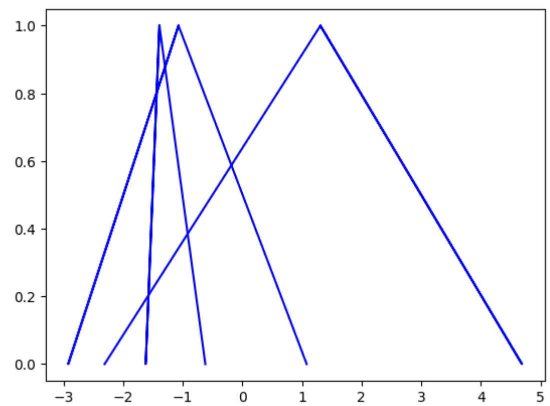
با توجه به روابط ۶ می توان ماتریس S را به صورت زیر ساخت.

$$S = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 2 & 0 \\ 2 & 1 & 3 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 2 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 2 & 1 & 3 \end{pmatrix}$$

مشابه قبل داریم

$$
\begin{aligned}
x_1 &= (-2.31 + 3.62r, 4.69 - 3.38r),\\
x_2 &= (-0.62 - 0.77r, -1.62 + 0.23r),\\
x_3 &= (1.08 - 2.15r, -2.92 + 1.85r).
\end{aligned}
$$

با توجه به شکل ۲ می توان دید که x_1 برخلاف x_2, x_3 یک عدد فازی است



شکل ۳: جواب فازی ضعیف دستگاه مثال ۴.۳.

در نتیجه جواب دستگاه قبل به صورت زیر است.

$$
\begin{aligned}
u_1 &= (-2.31 + 3.62r, 4.69 - 3.38r),\\
u_2 &= (-1.62 + 0.23r - 0.62 - 0.77r),\\
u_3 &= (-2.92 + 1.85r, 1.08 - 2.15r).
\end{aligned}
$$

از آنجایی که ماتریس S غالب قطری اکید نمی باشد، تصمینی برای همگرایی روش های تکرار وجود ندارد. بعلاوه وارون ماتریس $(L_B + D_B)$ وجود ندارد. در نتیجه نمی توانیم برای این مثال از روش های تکراری استفاده کنیم.

مثال ۵.۳. دستگاه 3×3 زیر را در نظر بگیرید.

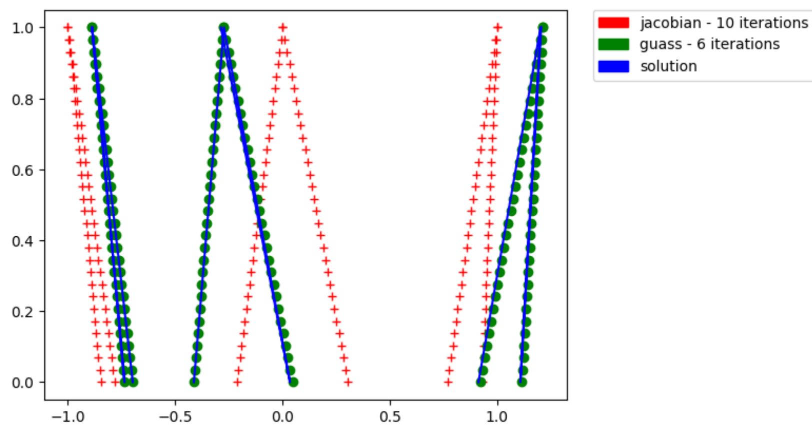
$$
\begin{aligned}
4x_1 + x_2 - x_3 &= (r, 2 - r)\\
-x_1 + 3x_2 + x_3 &= (2 + r, 3)\\
2x_1 + x_2 + 3x_3 &= (-2, -1 - r)
\end{aligned}
$$

$$
S = \begin{pmatrix}
4 & 1 & 0 & 0 & 0 & -1\\
0 & 3 & 1 & -1 & 0 & 0\\
2 & 1 & 3 & 0 & 0 & 0\\
0 & 0 & -1 & 4 & 1 & 0\\
-1 & 0 & 0 & 0 & 3 & 1\\
0 & 0 & 0 & 2 & 1 & 3
\end{pmatrix}
$$

مشابه قبل داریم

$$
\begin{aligned}
x_1 &= (0.1399r - 0.4125, -0.3217r + 0.0351),\\
x_2 &= (0.2894r + 0.9125, 0.0970r + 1.1076),\\
x_3 &= (-0.1897r - 0.6969, -0.1513r - 0.7353).
\end{aligned}
$$

از آنجایی که ماتریس S غالب قطری اکید است و همگرایی تضمین می شود می توان از روش های تکراری بار تقریب پاسخ استفاده کرد. شکل زیر دقت و سرعت همگرایی روش های تقریبی را برای این دستگاه معادلات نشان می دهد. می توانید با مراجعه به پیوست مقدار دقیق X در هر تکرار را تولید کنید.



شکل ۴: مقایسه روش های تقریبی، در اینجا $\epsilon = 10^{-3}$ در نظر گرفته شده. همانطور که مشاهده می شود، گاوس–سایدل با ۴ تکرار توانسته است جواب دقیق دستگاه را به خوبی تقریب بزند.

۴ پیوست

[509]:
```
from sympy import *
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.patches as mpatches
```

[510]:
```
r = symbols('r')
```

[511]:
```
def gauss_seidel(iter_, a, x, y, show_steps=False):

    def init_b(i, j) :
        if a[i, j] >= 0 :
            return a[i, j]
        return 0
    def init_c(i, j) :
        if a[i, j] < 0 :
            return -a[i, j]
        return 0
    b = Matrix(*shape(a), init_b)
    c = Matrix(*shape(a), init_c)

    l_b = b.lower_triangular()
    d_b = l_b.upper_triangular()
    l_b = l_b.lower_triangular(-1)
    u_b = b.upper_triangular(1)

    inv = (l_b + d_b).inv()

    M_gs = Matrix([[- inv * u_b, inv * c],
                    [inv * c, - inv * u_b]])

    C_gs = Matrix([inv * Matrix(y[:len(y) // 2]), inv * Matrix(y[len(y) // 2:
↪])])

    for i in range(iter_) :
        if show_steps :
            display(x)
            print(f'x ({i})')
        x = M_gs * x + C_gs
    return x
```

[515]:
```
# gauss-seidel solving first example
# run this cell with show_steps=True to see each iteration
a = Matrix([[1, -1], [1, 3]])
x = zeros(4, 1)
y = Matrix([r, 4 + r, 2 - r,  7 - 2 * r])
gs_solution = gauss_seidel(5, a, x, y, show_steps=False)
```

[516]:
```
def jacobian(iter_, a, x, y, show_steps=False):

    def init_b(i, j) :
        if a[i, j] >= 0 :
            return a[i, j]
        return 0
    def init_c(i, j) :
        if a[i, j] < 0 :
            return -a[i, j]
        return 0

    b = Matrix(*shape(a), init_b)
    c = Matrix(*shape(a), init_c)

    l_b = b.lower_triangular()
    d_b = l_b.upper_triangular()
    l_b = l_b.lower_triangular(-1)
    u_b = b.upper_triangular(1)

    inv = d_b.inv()

    M_gs = Matrix([[- inv* l_b, inv * c],
                   [inv * c, - inv * l_b]])

    C_gs = Matrix([inv * Matrix(y[:len(y) // 2]), inv * Matrix(y[len(y) // 2:
↪])])
    for i in range(iter_) :
        if show_steps :
            display(x)
            print(f'x ({i})')
        x = M_gs * x + C_gs
    return x
```

[517]:
```
# jacobian solving first example,
# run this cell with show_steps=True to see each iteration
a = Matrix([[1, -1], [1, 3]])
x = zeros(4, 1)
y = Matrix([r, 4 + r, 2 - r,  7 - 2 * r])
j_solution = jacobian(6, a, x, y, show_steps=False)
```

[518]:
```
def plot_fuzzy_number(x1, x2, *args) :
    def line_x_y(u1, u2) :
        x = np.linspace(0, 1, 30)
        return list(u1[0] + u1[1] * x) + list(u2[0] + u2[1] * x), list(x) +␣
↪list(x)
    ax = plt.gca()
    ax.plot(*line_x_y(x1, x2), *args)
```

```python
def plot_fuzzy_solution(x, *args) :
    def get_coef(x) :
        b = x.subs(r, 0)
        a = x.subs(r, 1) - b
        return b, a

    n = len(x) // 2
    for i in range(n) :
        x1 = get_coef(x[i])
        x2 = get_coef(x[n + i])
        plot_fuzzy_number(x1, x2, *args)
```

[ ]:
```python
# comparing results in the first example
real_solution = Matrix([[1.375 + 0.625 * r],
           [0.875 + 0.125 * r],
           [2.875 - 0.875 * r],
           [1.375 - 0.375 * r]])

plot_fuzzy_solution(j_solution, 'r+')
plot_fuzzy_solution(gs_solution, 'go')
plot_fuzzy_solution(real_solution, 'b')
handles = []
handles.append(mpatches.Patch(color='red', label='jacobian - 5 iterations'))
handles.append(mpatches.Patch(color='green', label='guass - 4 iterations'))
handles.append(mpatches.Patch(color='blue', label='solution'))
ax = plt.gca()
ax.legend(handles=[*handles], bbox_to_anchor=(1.05, 1),
          loc='best', borderaxespad=0.)
```

[ ]:
```python
# plotting figure in example 2
real_solution = Matrix([[-2.31 + 3.62 * r],
           [-0.62 - 0.77 * r],
           [1.08 - 2.15 * r],
           [4.69 - 3.38  * r],
           [-1.62 + 0.23 * r],
           [-2.92 + 1.85 * r]])

plot_fuzzy_solution(real_solution, 'b')
```

[ ]:
```python
# comparing iterative methods in the third example
real_solution = Matrix([[0.1399 * r - 0.4125],
            [0.2894 * r + 0.9125],
            [-0.1897 * r - 0.6969],
            [-0.3217 * r + 0.0351],
            [0.0970 * r + 1.1076],
            [- 0.1513 * r - 0.7353]])
```

```python
a = Matrix([[4, 1, -1], [-1, 3, 1], [2, 1, 3]])
x = zeros(6, 1)
y = Matrix([r, 2 + r, -2, 2 - r, 3, - 1 - r])
j_solution = jacobian(10, a, x, y, show_steps=False)
gs_solution = gauss_seidel(4, a, x, y, show_steps=False)
plot_fuzzy_solution(j_solution, 'r+')
plot_fuzzy_solution(gs_solution, 'go')
plot_fuzzy_solution(real_solution, 'b')
handles = []
handles.append(mpatches.Patch(color='red', label='jacobian - 10 iterations'))
handles.append(mpatches.Patch(color='green', label='guass - 4 iterations'))
handles.append(mpatches.Patch(color='blue', label='solution'))
ax = plt.gca()
ax.legend(handles=[*handles], bbox_to_anchor=(1.05, 1),
          loc='best', borderaxespad=0.)
```

مراجع

[1] S.S.L. Chang and L.A. Zadeh, On fuzzy mapping and control, IEEE Trans. Systems Man Cybernet. 2 (1972) 30-34.

[2] L.A. Zadeh, The concept of a linguistic variable and its application to approximate reasoning, Inform. Sei. 8 (1975) 199 -249.

[3] Friedman, M. Ming, M. Kandel, A. Fuzzy linear systems. Fuzzy Sets Syst. (1998), 96, 201–209

[4] Ezzati, R. Solving fuzzy linear systems. Soft Comput 15, 193–197 (2011)

[5] Tofigh Allahviranloo, Numerical methods for fuzzy system of linear equations, Applied Mathematics and Computation, Volume 155, Issue 2, 6 August 2004, Pages 493-502

[6] J.M. Ortega, Numerical Analysis a Second Course, Siam, 1990.

[7] Meurer, Aaron and Smith, Christopher P. and Paprocki, Mateusz and Čertík, Ondřej and Kirpichev, Sergey B. and Rocklin, Matthew and Kumar, AMiT and Ivanov, Sergiu and Moore, Jason K. and Singh, Sartaj and Rathnayake, Thilina and Vig, Sean and Granger, Brian E. and Muller, Richard P. and Bonazzi, Francesco and Gupta, Harsh and Vats, Shivam and Johansson, Fredrik and Pedregosa, Fabian and Curry, Matthew J. and Terrel, Andy R. and Roučka, Štěpán and Saboo, Ashutosh and Fernando, Isuru and Kulal, Sumith and Cimrman, Robert and Scopatz, Anthony, SymPy: symbolic computing in Python, 2017, jan