دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

دانشکده ریاضی و علوم کامپیوتر

پروژه کارشناسی درس جبر خطی عدد

عنوان پروژه:

روش های مستقیم محاسبه جواب های دستگاه های ۵-قطری

نوید سیاه کمری- ۴۰۰۱۲۰۲۵

استاد درس: دکتر مهدی دهقان

دی ۱۴۰۳

چکیده

در این پروژه با ما بدنبال حل دستگاه های ۵-قطری بدون استفاده از روش های کلی حل دستگاه ها هستیم. استفاده از روش های کلی برای دستگاه های ۵-قطری هزینه محاسباتی اضافی در بردارد. در این پروژه ۳ الگوریتم معرفی می شوند که هر کدام از آنها دستگاه های مذکور را در زمان خطی حل می کنند. دو الگوریتم اول ماتریس ۵-قطری را با استفاده از یک تبدیل خاص به ماتریس هایی تبدیل می کنند که حل آنها راحت است. این دو الگوریتم در موارد خاص ممکن است دچار شکست شوند؛ پس در ادامه به معرفی نسخه های نمادین آنها می پردازیم که شکست را دور می زنند. الگوریتم سوم ماتریس های ۵-قطری قطر-ثابت را حل می کند. در ادامه به نتایج عددی این الگوریتم ها می پردازیم و آنها را با هم مقایسه می کنیم. همچنین در ضمیمه پیاده سازی این الگوریتم ها در پایتون آورده شده است.

فهرست مطالب

۱.مقدمه ٤

۲. روش PTRANS-I برای حل دستگاه های ۵-قطری ٦

۳. روش PTRANS-II برای حل دستگاه های ۵-قطری ۹

۴. یک روش سریع برای حل دستگاه های ۵-قطری قطر-ثابت ۱۱

۵. نتایج عددی و نتیجه گیری ۱٤

منابع ۱٥

ضمیمه ۱٦

۱.مقدمه

سیستم های ۵-قطری خطی به فرم زیر نوشته می شوند:

$$PX = Y$$

که در آن X ماتریسی ۵-قطری به فرم زیر است:

$$P = \begin{pmatrix} d_1 & a_1 & b_1 & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\ c_2 & d_2 & a_2 & b_2 & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ e_3 & c_3 & d_3 & a_3 & b_3 & 0 & \cdots & \cdots & \cdots & 0 \\ 0 & e_4 & c_4 & d_4 & a_4 & b_4 & 0 & \cdots & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \cdots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & \cdots & 0 & e_{n-2} & c_{n-2} & d_{n-2} & a_{n-2} & b_{n-2} \\ 0 & \cdots & \cdots & \cdots & \cdots & 0 & e_{n-1} & c_{n-1} & d_{n-1} & a_{n-1} \\ 0 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 & e_n & c_n & d_n \end{pmatrix}$$

که $X = (x_1, x_2, \cdots, x_n)^t$ و $Y = (y_1, y_2, \cdots, y_n)^t$ بردار هایی با طول n هستند.

این نوع معادلات در بسیاری از مسائل مهندسی و کاربردی ظاهر می شوند و حل آنها با هزینه محاسباتی پایین از جمله چالش های این زمینه است. استفاده از روش های حذفی گوس و گوس-جردن و تجزیه LU و تجزیه QR برای این نوع دستگاه ها مقرون به صرفه نیست؛ زیرا پیچیدگی محاسباتی این روش ها $O(n^3)$ است. ما بدنبال روش هایی با پیچیدگی محاسباتی $O(n)$ هستیم. در فصل ۲ و ۳ به بررسی این روش ها می پردازیم.

نوع خاصی از ماتریس های ۵-قطری یا به طور کلی ماتریس های قطری، ماتریس های قطر-ثابت هستند که درایه های روی هر قطر آنها ثابت است و به فرم زیر هستند:

$$P = \begin{pmatrix} \alpha & \beta & \gamma & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\ \lambda & \alpha & \beta & \gamma & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\ \sigma & \lambda & \alpha & \beta & \gamma & \cdots & \cdots & \cdots & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \cdots & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \cdots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & \cdots & \cdots & \sigma & \lambda & \alpha & \beta & \gamma \\ 0 & \cdots & \cdots & \cdots & \cdots & \cdots & \sigma & \lambda & \alpha & \beta \\ 0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \sigma & \lambda & \alpha \end{pmatrix}$$

که در فصل ۴ به بررسی روشی سریع تر نسبت به روش های کلی حل دستگاه های ۵-قطری خواهیم پرداخت.

۲. روش PTRANS-I برای حل دستگاه های ۵-قطری

در این روش ماتریس ضرایب را با استفاده از یک تبدیل به ماتریس بالا مثلثی با قطر اصلی ۱ تبدیل می کنیم. پس از تبدیل شدن ماتریس به یک ماتریس بالامثلثی، دستگاه را می توان با استفاده از جایگذاری پسرو به راحتی حل کرد. برای استفاده از این تبدیل نیاز به ۵ بردار جدید داریم که به صورت $\alpha = (\alpha_1, \alpha_2, \cdots, \alpha_{n-1})$ و $\beta = (\beta_1, \beta_2, \cdots, \beta_{n-2})$ و $Z = (z_1, z_2, \cdots, z_n)$ و $\gamma = (\gamma_2, \gamma_3, \cdots, \gamma_n)$ و $\mu = (\mu_1, \mu_2, \cdots, \mu_n)$ تعریف می شوند و به صورت زیر محاسبه می شوند:

$$\alpha_i = \begin{cases} \frac{a_1}{\mu_1} & i = 1 \\ \frac{a_i - \beta_{i-1}\gamma_i}{\mu_i} & i = 2, 3, ..., n-1, \end{cases}$$

$$\beta_i = \frac{b_i}{\mu_i}, \qquad i = 2, 3, ..., n-1,$$

$$z_i = \begin{cases} \frac{y_1}{\mu_1} & i = 1 \\ \frac{y_2 - z_1\gamma_2}{\mu_2} & i = 2 \\ \frac{y_i - z_{i-2}e_i - z_{i-1}\gamma_i}{\mu_i} & i = 3, 4, ..., n, \end{cases}$$

$$\gamma_i = \begin{cases} c_2 & i = 2 \\ c_i - \alpha_{i-2}e_i & i = 3, 4, ..., n, \end{cases}$$

$$\mu_i = \begin{cases} d_1 & i = 1 \\ d_2 - \alpha_1\gamma_2 & i = 2 \\ d_i - \beta_{i-2}e_i - \alpha_{i-1}\gamma_i & i = 3, 4, ..., n. \end{cases}$$

پس بدست آوردن این بردارها می توانیم معادله $PX = Y$ را به صورت زیر بنویسم:

$$\begin{pmatrix} 1 & \alpha_1 & \beta_1 & 0 & \dots & \dots & \dots & \dots & \dots & 0 \\ 0 & 1 & \alpha_2 & \beta_2 & 0 & \dots & \dots & \dots & \dots & 0 \\ 0 & 0 & 1 & \alpha_3 & \beta_3 & 0 & \dots & \dots & \dots & 0 \\ 0 & 0 & 0 & 1 & \alpha_4 & \beta_4 & 0 & \dots & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \dots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & \dots & \dots & 0 & 0 & 0 & 1 & \alpha_{n-2} & \beta_{n-2} \\ 0 & \dots & \dots & \dots & \dots & 0 & 0 & 0 & 1 & \alpha_{n-1} \\ 0 & \dots & \dots & \dots & \dots & \dots & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ \vdots \\ \vdots \\ \vdots \\ x_{n-2} \\ x_{n-1} \\ x_n \end{pmatrix} = \begin{pmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \\ \vdots \\ \vdots \\ \vdots \\ z_{n-2} \\ z_{n-1} \\ z_n \end{pmatrix}$$

دستگاه بالا را می توان با استفاده از جایگذاری پسرو حل کرد. همچنین جواب های دستگاه بالا همان جواب های دستگاه اولیه است زیرا معادله i ام آن با معادله i دستگاه اولیه برابر است. به عنوان مثال معادل بودن معادله اولیه را اثبات می کنیم و معادل بودن معادلات دیگر با همین روش و با استفاده از جایگذاری بدست می آید. باید نشان دهیم که $x_1 + \alpha_1 x_2 + \beta_1 x_3 = z_1$ معادل است با $d_1 x_1 + a_1 x_2 + b_1 x_3 = y_1$. به این منظور کافی است دو طرف معادله دوم را بر μ_1 تقسیم کنیم تا معادله اول حاصل شود.برای اثبات معادل بودن معادله i ام کافی است دو طرف معادله اولیه بر μ_i تقسیم کنیم و سپس کمی عملیات جبری انجام دهیم تا به معادله دوم برسیم. سوال اینجاست که اگر μ_i برابر صفر باشد آیا باز هم می توان این تبدیل را انجام داد؟ جواب خیر است. همانطور هم که در فرمول های تبدیل مشاهده می شود، اگر یک فرمول به صورت کسری باشد، تنها عامل μ_i در مخرج ظاهر شده است. پس اگر یکی از μ_i ها صفر باشد، الگوریتم با شکست مواجه می شود و ماتریس تبدیل یافته حاصل نمی شود. الگوریتمی که توضیح داده شد، الگوریتم PTRANS-I نام دارد.

برای مواجه با مشکل شکست الگوریتم را از حالت عددی به حالت نمادی تبدیل می کنیم. به این منظور کافی است هر جا μ_i برابر صفر شد آن را برابر p قرار داده و الگوریتم را ادامه دهیم. در پایان هر کجا که در بردار های حاصل p شد کافی است به جای آن صفر قرار دهیم. در پایان که جواب ها بر حسب p بدست آمد به جای همه p ها صفر قرار می دهیم. می توان نشان داد که بدون توجه به اینکه p برابر صفر است جواب های معادله بدست می آید. این الگوریتم SPTRANS-I نام دارد.

در تصویر زیر الگوریتم PTRANS-I و مراحل آن توضیح داده شده است.

To find the solution of PLS (1.1) using the transformed system (2.6), we may proceed as follows:

INPUT *order of the matrix* n *and the components* $d_i, a_i, b_i, c_i, e_i, f_i, \quad i = 1, 2, ..., n, (a_n = b_n = b_{n-1} = c_1 = e_1 = e_2 = 0)$.

OUTPUT *The solution vector* $x = (x_1, x_2, ..., x_n)^t$.

Step 1: *Use DETGPENTA algorithm [13] to check the non-singularity of the coefficient matrix of the system (1.3).*

Step 2: *If* $\det(P) = 0$*, then Exit and Print Message ("No solutions") end if.*

Step 3: *Set* $\mu_1 = d_1$, $\alpha_1 = \frac{a_1}{\mu_1}$, $\beta_1 = \frac{b_1}{\mu_1}$, *and* $z_1 = \frac{y_1}{\mu_1}$.

Step 4: *Set* $\gamma_2 = c_2$, $\mu_2 = d_2 - \alpha_1 \gamma_2$, $\alpha_2 = \frac{a_2 - \beta_1 \gamma_2}{\mu_2}$, $\beta_2 = \frac{b_2}{\mu_2}$, *and* $z_2 = \frac{y_2 - z_1 \gamma_2}{\mu_2}$.

Step 5: *For i=3,4,...,n-2 do*

Compute and simplify:

$\gamma_i = c_i - \alpha_{i-2} e_i$,

$\mu_i = d_i - \beta_{i-2} e_i - \alpha_{i-1} \gamma_i$,

$\alpha_i = \frac{a_i - \beta_{i-1} \gamma_i}{\mu_i}$,

$\beta_i = \frac{b_i}{\mu_i}$,

$z_i = \frac{y_i - z_{i-2} e_i - z_{i-1} \gamma_i}{\mu_i}$,

End do.

$\gamma_{n-1} = c_{n-1} - \alpha_{n-3} e_{n-1}$,

$\mu_{n-1} = d_{n-1} - \beta_{n-3} e_{n-1} - \alpha_{n-2} \gamma_{n-1}$,

$\alpha_{n-1} = \frac{a_{n-1} - \beta_{n-2} \gamma_{n-1}}{\mu_{n-1}}$,

$\gamma_n = c_n - \alpha_{n-2} e_n$,

$\mu_n = d_n - \beta_{n-2} e_n - \alpha_{n-1} \gamma_n$,

$z_{n-1} = \frac{y_{n-1} - z_{n-2} e_{n-1} - z_{n-2} \gamma_{n-1}}{\mu_{n-1}}$,

$z_n = \frac{y_n - z_{n-1} e_n - z_{n-1} \gamma_n}{\mu_n}$,

Step 6: *Compute the solution vector* $X = (x_1, x_2, ..., x_n)^t$ *using*

$x_n = z_n$, $x_{n-1} = z_{n-1} - \alpha_{n-1} x_n$.

For i=n-2, n-3, ...,1 do

Compute and simplify:

$x_i = z_i - \alpha_i x_{i+1} - \beta_i x_{i+2}$

End do.

در این الگوریتم با استفاده از الگوریتم DETGPENTA ابتدا دترمینان ماتریس محاسبه شده و اگر دترمینان ماتریس برابر با صفر باشد الگوریتم را متوقف می کند و دستگاه جواب ندارد. در این پروژه الگوریتم DETGPENTA مورد بررسی قرار نمی گیرد. همانطور که در گام ششم و آخر الگوریتم مشاهده می کنید، با استفاده از جایگذاری پسرو، جواب های معادله بدست می آید.

تعداد اعمال حسابی برای انجام این الگوریتم $19n - 29$ است که از مرتبه زمانی $O(n)$ است.

۳. روش PTRANS-II برای حل دستگاه های ۵-قطری

در این روش ماتریس ضرایب را با استفاده از یک تبدیل به ماتریس پایین مثلثی با قطر اصلی ۱ تبدیل می کنیم. پس از تبدیل شدن ماتریس به یک ماتریس پایین مثلثی، دستگاه را می توان با استفاده از جایگذاری پسرو به راحتی حل کرد. برای استفاده از این تبدیل نیاز به ۵ بردار جدید داریم که به صورت $\sigma = (\sigma_2, \sigma_3, \cdots, \sigma_n)$ و $\phi = (\phi_3, \phi_4, \cdots, \phi_n)$ و $\omega = (\omega_1, \omega_2, \cdots, \omega_n)$ و $\rho = (\rho_1, \rho_2, \cdots, \rho_{n-1})$ و $\psi = (\psi_1, \psi_2, \cdots, \psi_n)$ تعریف می شوند و به صورت زیر محاسبه می شوند:

$$\sigma_i = \begin{cases} \frac{c_n}{\psi_n} & i = n \\ \frac{c_i - \phi_{i+1}\rho_i}{\psi_i} & i = n-1, n-2, ..., 2, \end{cases}$$

$$\phi_i = \frac{e_i}{\psi_i}, \qquad i = n, n-1, ..., 3,$$

$$w_i = \begin{cases} \frac{y_n}{\psi_n} & i = n \\ \frac{y_{n-1} - w_n \rho_{n-1}}{\psi_{n-1}} & i = n-1 \\ \frac{y_i - w_{i+2} b_i - w_{i+1}\rho_i}{\psi_i} & i = n-2, n-3, ..., 1, \end{cases}$$

$$\rho_i = \begin{cases} a_{n-1} & i = n-1 \\ a_i - \sigma_{i+2} b_i & i = n-2, n-3, ..., 1, \end{cases}$$

$$\psi_i = \begin{cases} d_n & i = n \\ d_{n-1} - \sigma_n \rho_{n-1} & i = n-1 \\ d_i - \phi_{i+2} b_i - \sigma_{i+1}\rho_i & i = n-2, n-3, ..., 1. \end{cases}$$

پس بدست آوردن این بردارها می توانیم می توانیم معادله $PX = Y$ را به صورت زیر بنویسم:

$$\begin{pmatrix} 1 & 0 & 0 & 0 & \dots & \dots & \dots & \dots & \dots & 0 \\ \sigma_2 & 1 & 0 & 0 & 0 & \dots & \dots & \dots & \dots & 0 \\ \phi_3 & \sigma_3 & 1 & 0 & 0 & 0 & \dots & \dots & \dots & 0 \\ 0 & \phi_4 & \sigma_4 & 1 & 0 & 0 & 0 & \dots & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \dots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & \dots & \dots & 0 & \phi_{n-2} & \sigma_{n-2} & 1 & 0 & 0 \\ 0 & \dots & \dots & \dots & \dots & 0 & \phi_{n-1} & \sigma_{n-1} & 1 & 0 \\ 0 & \dots & \dots & \dots & \dots & \dots & 0 & \phi_n & \sigma_n & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ \vdots \\ \vdots \\ \vdots \\ x_{n-2} \\ x_{n-1} \\ x_n \end{pmatrix} = \begin{pmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \\ \vdots \\ \vdots \\ \vdots \\ w_{n-2} \\ w_{n-1} \\ w_n \end{pmatrix}$$

دستگاه بالا را می توان با استفاده از جایگذاری پسرو حل کرد. این تبدیل بسیار شبیه به تبدیل الگوریتم PTRANS-I است. تنها تفاوت آن در این است که اندیس ها از n شروع می شوند و در فرمول ها جای بردار های a و c و همچنین جای بردار های e و b عوض شده است.نقش ψ_i در این تبدیل همچون نقش

μ_i برای الگوریتم PTRANS-I است.پس موارد شکست و نحوه اثبات معادل بودن دستگاه قبل و بعد از تبدیل همچون PTRANS-I است. همچنین برای مواجه با شکست به جای $p = \mu_i = 0$ از $p = \psi_i = 0$ استفاده می کنیم که این الگوریتم SPTRANS-II نام می گیرد.

در تصویر زیر الگوریتم PTRANS-I و مراحل آن توضیح داده شده است.

INPUT *order of the matrix* n *and the components* $d_i, a_i, b_i, c_i, e_i, f_i,\quad i = 1, 2, ..., n, (a_n = b_n = b_{n-1} = c_1 = e_1 = e_2 = 0)$.

OUTPUT *The solution vector* $x = (x_1, x_2, ..., x_n)^t$.

Step 1: *Use DETGPENTA algorithm [13] to check the non-singularity of the coefficient matrix of the system (1.3).*

Step 2: *If* $\det(P) = 0$, *then Exit and Print Message ("No solutions") end if.*

Step 3: *Set* $\psi_n = d_n$, $\sigma_n = \frac{c_n}{\psi_n}$, $\phi_n = \frac{e_n}{\psi_n}$, *and* $w_n = \frac{y_n}{\psi_n}$.

Step 4: *Set* $\rho_{n-1} = a_{n-1}$, $\psi_{n-1} = d_{n-1} - \sigma_n\rho_{n-1}$, $\sigma_{n-1} = \frac{c_{n-1}-\phi_n\rho_{n-1}}{\psi_{n-1}}$, $\phi_{n-1} = \frac{e_{n-1}}{\psi_{n-1}}$, *and* $w_{n-1} = \frac{y_{n-1}-w_n\rho_{n-1}}{\psi_{n-1}}$.

Step 5: *For i=n-2, n-3, ...,3 do*

Compute and simplify:

$\rho_i = a_i - \sigma_{i+2}b_i,$

$\psi_i = d_i - \phi_{i+2}b_i - \sigma_{i+1}\rho_i,$

$\sigma_i = \frac{c_i-\phi_{i+1}\rho_i}{\psi_i},$

$\phi_i = \frac{e_i}{\psi_i},$

$w_i = \frac{y_i-w_{i+2}b_i-w_{i+1}\rho_i}{\psi_i},$

End do.

$\rho_2 = a_2 - \sigma_4 b_2,$

$\psi_2 = d_2 - \phi_4 b_2 - \sigma_3\rho_2,$

$\sigma_2 = \frac{c_2-\phi_4\rho_2}{\psi_2},$

$\rho_1 = a_1 - \sigma_3 b_1,$

$\psi_1 = d_1 - \phi_3 b_1 - \sigma_2\rho_1,$

$w_2 = \frac{y_2-w_4 b2-w_3\rho_2}{\psi_2},$

$w_1 = \frac{y_1-w_3 b_1-w_2\rho_1}{\psi_1},$

Step 6: *Compute the solution vector* $X = (x_1, x_2, ..., x_n)^t$ *using*

$x_{=}w_1$, $x_2 = w_2 - \sigma_2 x_1$.

For i=3, 4, ...,n do

Compute and simplify:

$x_i = w_i - \sigma_i x_{i-1} - \phi_i x_{i-2}$

End do.

همانطور که در گام ششم و آخر الگوریتم مشاهده می کنید، با استفاده از جایگذاری پسرو، جواب های معادله بدست می آید. همانند الگوریتم PTRANS-I مرتبه زمانی این الگوریتم نیز $O(n)$ است.

۴. یک روش سریع برای حل دستگاه های ۵-قطری قطر-ثابت

دستگاه های که ماتریس ضرایب آنها به صورت زیر است، دستگاه های ۵-قطری قطر ثابت نام دارند.

$$A = \begin{bmatrix} \alpha & \beta & \gamma & & & & & & \\ \lambda & \alpha & \beta & \gamma & & & & & \\ \sigma & \lambda & \alpha & \beta & \gamma & & & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & & & \\ & & \ddots & \ddots & \ddots & \ddots & \ddots & & \\ & & & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & & & \sigma & \lambda & \alpha & \beta & \gamma \\ & & & & & \sigma & \lambda & \alpha & \beta \\ & & & & & & \sigma & \lambda & \alpha \end{bmatrix}$$

هدف این است که به جای حل این دستگاه، با شیفت دادن و سپس بلوک بندی کردن دستگاهی بالا یا پایین مثلثی راحل کنیم و سپس جواب های دستگاه را بدست آوریم. ماتریس زیر را در نظر بگیرد. این ماتریس از شیت دو ستون از ماتریس همانی به سمت راست و دو سطر به سمت بالا حاصل شده است. اگر این ماتریس را در هر ماتریسی ضرب کنیم، ماتریس حاصل ضرب شیف داده شده ماتریس اولیه است.

$$J = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & \cdots & 1 \\ 1 & 0 & 0 & 0 & \cdots & \cdots & 0 \\ 0 & 1 & 0 & 0 & \cdots & \cdots & 0 \end{bmatrix}$$

با ضرب این ماتریس در ماتریس ۵-قطری قطر-ثابت ماتریس حاصل به صورت زیر است:

$$\hat{A} = \left[\begin{array}{ccccccc|c|c} \sigma & \lambda & \alpha & \beta & \gamma & & & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & & & \\ & & \ddots & \ddots & \ddots & \ddots & \ddots & & \\ & & & \ddots & \ddots & \ddots & \ddots & \gamma & \\ & & & & \sigma & \lambda & \alpha & \beta & \gamma \\ & & & & & \sigma & \lambda & \alpha & \beta \\ & & & & & & \sigma & \lambda & \alpha \\ \hline \alpha & \beta & \gamma & & & & & 0 & 0 \\ \hline \lambda & \alpha & \beta & \gamma & & & & 0 & 0 \end{array}\right] \equiv \left[\begin{array}{c|c|c} A_{11} & p & r \\ \hline w^T & 0 & 0 \\ \hline s^T & 0 & 0 \end{array}\right]$$

تجزیه دولیتل ماتریس حاصل به صورت زیر است:

$$\hat{A} = \begin{bmatrix} I & 0 & 0 \\ w^T A_{11}^{-1} & 1 & 0 \\ s^T A_{11}^{-1} & \frac{s^T A_{11}^{-1} p}{w^T A_{11}^{-1}} & 1 \end{bmatrix} \begin{bmatrix} A_{11} & p & r \\ 0 & -w^T A_{11}^{-1} p & -w^T A_{11}^{-1} r \\ 0 & 0 & -s^T A_{11}^{-1} r - \frac{s^T A_{11}^{-1} p}{w^T A_{11}^{-1} p} w^T A_{11}^{-1} r \end{bmatrix}$$

در این ماتریس از آنجا که ضرایب دو واحد به سمت بالا و راست شیفت داده شده است پس بردار های x و b به صورت زیر تعریف شوند تا جواب دستگاه تغییر نکند.

$$x = \left[\begin{array}{c} x_1 \\ \hline x_{n-1} \\ \hline x_n \end{array}\right] \quad , \quad \hat{b} = \left[\begin{array}{c} b_3 \\ \hline b_1 \\ \hline b_2 \end{array}\right]$$

که در آن x_1 و b_3 بردار هایی به طول $(n-2)$ هستند. با ضرب $\hat{A}$ در $\hat{x}$ یک دستگاه معادلات به صورت زیر حاصل می شود:

$$\begin{cases} A_{11} x_1 + x_{n-1} p + x_x r & = b_3 \\ w^T A_{11}^{-1} b_3 - w^T A_{11}^{-1} p x_{n-1} - w^T A_{11}^{-1} r x_n & = b_1 \\ s^T A_{11}^{-1} b_3 - s^T A_{11}^{-1} p x_{n-1} - s^T A_{11}^{-1} x_n & = b_2 \end{cases}$$

برای حل این دستگاه از آنجا که $A_{11}^{-1} b_3$ و $A_{11}^{-1} p$ و $A_{11}^{-1} r$ ظاهر شده است، پس برای حذف این بردار ها باید دستگاه های زیر را حل کنیم:

$$\begin{cases} A_{11} u & = b_3 \\ A_{11} v & = p \\ A_{11} z & = r \end{cases}$$

حل این دستگاه ها با استفاده از الگوریتم جایگذاری پسرو به راحتی انجام می شود. پس از حل این دستگاه ها و جایگذاری در دستگاه اول دستگاه حاصل به صورت زیر است:

$$\begin{cases} x_1 + x_{n-1} v + x_n r & = u \\ w^T u - w^T v x_{n-1} - w^T z x_n & = b_1 \\ s^T u - s^T v x_{n-1} - s^T z x_n & = b_2 \end{cases}$$

که در آن معادله اول یک دستگاه اما اما معادله دوم و سوم هر کدام دستگاه با یک معادله هستند. معادله اول دوم را به صورت زیر می نویسیم:

$$\begin{cases} w^T v x_{n-1} + w^T z x_n & = w^T u - b_1 \\ s^T v x_{n-1} + s^T z x_n & = s^T u - b_2 \end{cases}$$

$$\begin{bmatrix} w^T u & w^T z \\ s^T u & s^T z \end{bmatrix} \begin{bmatrix} x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} w^T u - b_1 \\ s^T u - b_2 \end{bmatrix}$$

ضرایب این معادله اسکالر است پس به راحتی جواب های آن را بدست می آوریم. پس از بدست آوردن جواب ها می توان x_1 را از معادله اول و به این صورت بدست آورد:

$$x_1 \qquad = u - x_{n-1} v - x_n r$$

در نتیجه دستگاه ۵-قطری قطر-ثابت در زمان خطی حل شد.

۵. نتایج عددی و نتیجه گیری

سه الگوریتم بیان شده در فصل های قبلی با پایتون پیاده سازی شده اند، که کد آنها در در لینک گیت هاب ضمیمه شده در پایان گزارش آورده شده است.

n	$\|\|\mathbf{x}-\bar{x}\|\|_\infty$ and CPU time(S)							
	PTRANS-I		**PTRANS-II**		**Algorithm 3**[8]		A\b(MATLAB)	
500	1.5856×10^{-7}	0.0069	0	0.0086	6.8579×10^{-8}	0.0048	9.98×10^{-8}	0.0023
5000	8.3674×10^{-4}	0.0062	0	0.0391	3.0253×10^{-4}	0.0057	2.50×10^{-4}	0.7548
10000	0.0058	0.0114	0	0.0511	0.0052	0.0101	0.0106	4.5464
50000	2.1415	0.0308	0	0.1687	7.9056	0.0119	0.0159	655.51

همانطور که در جدول بالا مشاهده می شود هر دو الگوریتم PTRANS-II و PTRANS-I از توابع متلب که در حالت کلی دستگاه را حل می کنند، بسیار سریع ترند. البته الگوریتم PTRANS-I کمی سریع تر است اما در ابعاد بالا خطا بیشتری نسبت به متلب و PTRANS-II تولید می کند. همانطور که مشاهده می کنید خطای روش PTRANS-II صفر است. پس بهترین روش برای حل دستگاه های ۵-قطری همان PTRANS-II است.

در مورد الگوریتم قطر-ثابت نتایج امیدوار کننده نیست. هر چند ممکن است این الگوریتم نسبت به توابع کلی متلب و پایتون عملکرد بهتری داشته باشد، اما در مقایسه با PTRANS-I و PTRANS-II عملکرد ضعیفی دارد. نتیجه زیرا برای یک دستگاه ۵-قطری قطر ثابت با بعد ۱۰۰۰ ثبت شده است.

```
time of PTRANS-I : 0.010743856430053711
time of PTRANS-II : 0.011032581329345703
time of toeplitz : 0.07966232299804688
```

همچنین نتیجه زیر برای یک دستگاه با مقادیر قطر و سمت راست متفاوت با بعد ۵۰۰ ثبت شده است.

```
time of PTRANS-I : 0.005063772201538086
time of PTRANS-II : 0.005146741867065430
time of toeplitz : 0.030022859573364258
```

در نتیجه با اینکه الگوریتم قطر-ثابت با اینکه روشی سریع تر به نظر می رسید اما در عمل ضعیف تر از الگوریتم های دیگر است که مورد بررسی قرار گرفت.

منابع

[1] جزوه درس کارشناسی جبرخطی عددی ,د. م. دهقان.

[2] S. Hasanbeigi, "A FAST SOLVER FOR PENTADIAGONAL TOEPLITZ SYSTEMS," Tehran, 2024.

[3] A. A. KARAWIA, "On Solving Pentadiagonal Linear Systems via Transformations," 2015.

ضمیمه

آدرس گیت هاب: https://github.com/NavidSiahkamarii/Linear_algebra_final_project