



معکوس ماتریس‌های نواری

مریم شمس سولاری^{۱*} و مهران رسولی^۲

دانشکده علوم، گروه ریاضی، دانشگاه پیام‌نور، تهران، ایران

چکیده

در این مقاله سعی داریم الگوریتمی برای یافتن معکوس ماتریس‌های نواری ارائه دهیم. بنابراین ابتدا با تجزیه کردن ماتریس با روش دولیتل LU روابط و فرمول‌هایی را برای محاسبه معکوس ماتریس بدست آورده و در آخر الگوریتم را با استفاده از این روابط پیاده‌سازی می‌کنیم. در پایان، هزینه محاسبات و مثال عددی برای اثبات کارایی الگوریتم در مقایسه با معکوس معمولی در نرم افزار متلب بیان می‌کنیم.

واژه‌های کلیدی: ماتریس معکوس، الگوریتم، تجزیه LU ، ماتریس نواری

کد موضوع‌بندی ریاضی [۲۰۱۰]: 15B36، 15A23، 15A03

۱ مقدمه

شکل کلی یک ماتریس r -قطری $n \times n$ که در آن $(r = 2m + 1 : m = 1, 2, \dots)$ و m تعداد قطرهای ناصفر بالا یا پایین اصلی را نشان می‌دهد، بصورت زیر است:

$$G_{m,n} = \begin{bmatrix} d_1 & a_{1,1} & a_{2,1} & \cdots & a_{m,1} & 0 & \cdots & 0 \\ b_{1,1} & d_2 & a_{1,2} & a_{2,2} & \cdots & a_{m,2} & \ddots & \vdots \\ b_{2,1} & b_{1,2} & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & b_{2,2} & \ddots & \ddots & \ddots & \ddots & \ddots & a_{m,n-m} \\ b_{m,1} & \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & b_{m,2} & \ddots & \ddots & \ddots & \ddots & \ddots & a_{2,n-2} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & a_{1,n-1} \\ 0 & \cdots & 0 & b_{m,n-m} & \cdots & b_{2,n-2} & b_{1,n-1} & d_n \end{bmatrix}. \quad (1)$$

*سخنران. آدرس ایمیل: shamssolary@email.com

شکل کلی یک ماتریس (r, k) -قطری $n \times n$ که در آن $(r = 2m + 1 : m = 1, 2, \dots)$ که m تعداد قطره‌های ناصفر بالا یا پایین قطر اصلی و k فاصله بین این قطره‌های ناصفر را نشان می‌دهد، بصورت زیر است:

$$G_{m,n}^{(k)} = \begin{cases} G_{i,i} = d_i & i = 1, 2, \dots, n, \\ G_{j,k+j} = a_{1,j}, \quad G_{k+j,j} = b_{1,j} & j = 1, \dots, n-k, \\ G_{j,2k+j} = a_{2,j}, \quad G_{2k+j,j} = b_{2,j} & j = 1, \dots, n-2k, \\ \vdots & \vdots \\ G_{j,mk+j} = a_{m,j}, \quad G_{mk+j,j} = b_{m,j} & j = 1, \dots, n-mk, \end{cases} \quad (2)$$

همچنین $a_{j,i}$ و $b_{j,i}$ دنباله‌های متناهی از اعداد غیرصفر به‌ازای $(j = 1, 2, \dots, m)$ و $(i = 1, 2, \dots, n)$ هستند.

۲ نتایج اصلی

در این بخش الگوریتم اصلی را با استفاده از لم زیر و مراجع [۵، ۳، ۱] برای تعیین معکوس هر ماتریس (r, k) -قطری پیاده‌سازی می‌کنیم.

لم ۱.۲. ([۴]) فرض کنید D_k ، برای $k = 1, 2, \dots, n$ مرتبه کهادهای اصلی پیشرو ماتریس $G_{m,n}^{(k)}$ باشند، اگر D_k -ها به‌ازای $k = 1, 2, \dots, n$ مخالف صفر باشند، آنگاه ماتریس $G_{m,n}^{(k)}$ تنها یک تجزیه دولیتل LU بصورت،

$$G_{m,n}^{(k)} = L_{m,n}^{(k)} U_{m,n}^{(k)} \quad (3)$$

دارد که در آن ماتریس‌های L و U به شکل،

$$L_{m,n}^{(k)} = \begin{cases} l_{i,i} = 1 & i = 1, 2, \dots, n, \\ l_{k+j,j} = l_{1,j} & j = 1, \dots, n-k, \\ l_{2k+j,j} = l_{2,j} & j = 1, \dots, n-2k, \\ \vdots & \vdots \\ l_{mk+j,j} = l_{m,j} & j = 1, \dots, n-mk, \end{cases} \quad (4)$$

$$U_{m,n}^{(k)} = \begin{cases} u_{i,i} = t_i & i = 1, 2, \dots, n, \\ u_{j,k+j} = u_{1,j} & j = 1, \dots, n-k, \\ u_{j,2k+j} = u_{2,j} & j = 1, \dots, n-2k, \\ \vdots & \vdots \\ u_{j,mk+j} = u_{m,j} & j = 1, \dots, n-mk, \end{cases} \quad (5)$$

هستند و $\det(G_{m,n}^{(k)}) = \prod_{i=1}^n t_i$.

حال با جایگذاری (۲)، (۴) و (۵) در (۳) روابط زیر را بدست می‌آوریم،

$$t_i = \begin{cases} d_i & i = 1, 2, \dots, k, \\ d_i - \sum_{q=p}^1 l_{q,i-qk} u_{q,i-qk} & p = 1, \dots, m-1, i = pk+1, \dots, (p+1)k \\ d_i - \sum_{q=m}^1 l_{q,i-qk} u_{q,i-qk} & i = mk+1, \dots, n. \end{cases} \quad (6)$$

برای $j = 1, \dots, m-1$ داریم،

$$u_{j,i} = \begin{cases} a_{j,i} & i = 1, \dots, k, \\ a_{j,i} - \sum_{q=p}^1 l_{q,i-qk} u_{j+q,i-qk} & p = 1, \dots, m-j-1, i = pk+1, \dots, (p+1)k, \\ a_{j,i} - \sum_{q=p}^1 l_{q,i-qk} u_{j+q,i-qk} & p = m-j, i = (m-j)k+1, \dots, n-jk, \end{cases} \quad (7)$$

$$u_{m,i} = a_{m,i} \quad i = 1, \dots, n-mk.$$

$$l_{j,i} = \begin{cases} \frac{b_{j,i}}{t_i} & i = 1, \dots, k, \\ \frac{b_{j,i} - \sum_{q=p}^1 l_{j+q,i-qk} u_{q,i-qk}}{t_i} & p = 1, \dots, m-j-1, i = pk+1, \dots, (p+1)k, \\ \frac{b_{j,i} - \sum_{q=p}^1 l_{j+q,i-qk} u_{q,i-qk}}{t_i} & p = m-j, i = (m-j)k+1, \dots, n-jk, \end{cases} \quad (8)$$

$$l_{m,i} = \frac{b_{m,i}}{t_i} \quad i = 1, \dots, n-mk.$$

با استفاده از روابط (۶)، (۷) و (۸) می‌توانیم روابط مربوط به تجزیه دولیتل LU ماتریس $G_{m,n}^{(k)}$ را به‌ازای هر m دلخواه تعیین کنیم.

حال اگر ماتریس (۲) ناکمین باشد آنگاه $(G_{m,n}^{(k)})^{-1} = S = (S_1, S_2, \dots, S_n)^t$ که S_j ، j -امین سطر ماتریس $(G_{m,n}^{(k)})^{-1}$ به‌ازای $j = 1, 2, \dots, n$ است. با استفاده از $G_{m,n}^{(k)}(G_{m,n}^{(k)})^{-1} = E$ (یک ماتریس همانی $n \times n$ است) داریم، $L_{m,n}^{(k)}(U_{m,n}^{(k)}S) = E$ قرار دهید،

$$L_{m,n}^{(k)}X = E \quad (9)$$

آنگاه

$$U_{m,n}^{(k)}S = X \quad (10)$$

که $S = (S_1, \dots, S_n)^t$ و $E = (E_1, \dots, E_n)^t$ ، $X = (X_1, \dots, X_n)^t$ به ترتیب با استفاده از (۹) و (۱۰) داریم،

$$X_i = \begin{cases} E_i & i = 1, 2, \dots, k \\ E_i - \sum_{q=1}^p l_{q,i-qk} X_{i-qk} & p = 1, \dots, m-1, i = pk+1, \dots, (p+1)k, \\ E_i - \sum_{q=1}^m l_{q,i-qk} X_{i-qk} & i = mk+1, \dots, n. \end{cases} \quad (11)$$

$$S_i = \begin{cases} \frac{X_i}{t_i} & i = n, n-1, \dots, n-k+1, \\ \frac{X_i - \sum_{q=1}^p u_{q,i} S_{i+qk}}{t_i} & p = 1, \dots, m-1, i = n-pk, \dots, n-(p+1)k+1, \\ \frac{X_i - \sum_{q=1}^m u_{q,i} S_{i+qk}}{t_i} & i = n-mk, \dots, 1. \end{cases} \quad (12)$$

اکنون با استفاده از (۶)، (۷)، (۸)، (۱۱) و (۱۲) به ساخت الگوریتم می‌پردازیم.

۱.۲ الگوریتمی برای تعیین معکوس هر ماتریس (r, k) -قطری:

الگوریتم ۲.۲. ورودی: مقدار m ، مقدار k و ماتریس $G_{m,n}^{(k)}$.

خروجی: ماتریس معکوس، W .

گام اول:

```

For  $p = 0, 1, \dots, m$  do
  For  $i = pk + 1, \dots, (p + 1)k$  do
     $t_i = G_{i,i} - \sum_{q=p}^1 l_{q,i-qk} u_{q,i-qk}$ 
    If  $t_i = 0$  then  $t_i = z$  end if.
    For  $j = 1, 2, \dots, m - p$  do
       $u_{j,i} = G_{i,jk+i} - \sum_{q=p}^1 l_{q,i-qk} u_{j+q,i-qk}$ 
       $l_{j,i} = \frac{G_{jk+i,i} - \sum_{q=p}^1 l_{j+q,i-qk} u_{q,i-qk}}{t_i}$ 
    end do
    if  $p \geq 1$  then
      For  $j = m, m - 1, \dots, m - p + 1$  do
         $u_{j,i} = G_{i,jk+i} - \sum_{q=m-j}^1 l_{q,i-qk} u_{j+q,i-qk}$ 
         $l_{j,i} = \frac{G_{jk+i,i} - \sum_{q=m-j}^1 l_{j+q,i-qk} u_{q,i-qk}}{t_i}$ 
      end do
    end if
  end do
end do.

```

گام دوم:

$\det(G_{m,n}^{(k)}) = \prod_{i=1}^n t_i$ را محاسبه کن، سپس $z = 0$ را جایگذاری کن اگر $\det(G_{m,n}^{(k)}) = 0$ ماتریس $G_{m,n}^{(k)}$ تکین است. در اینصورت عبارت (ماتریس تکین است) را در خروجی چاپ کن و توقف الگوریتم.

گام سوم:

```

For  $p = 0, 1, \dots, m$  do
  For  $i = pk + 1, pk + 2, \dots, (p + 1)k$  do
     $X_i = E_i - \sum_{q=1}^p l_{q,i-qk} X_{i-qk}$ 
  end do
end do.

```

گام چهارم:

```

For  $p = 0, 1, \dots, m$  do
  For  $i = n - pk, n - pk - 1, \dots, n - (p + 1)k + 1$  do
     $S_i = \frac{X_i - \sum_{q=1}^p u_{q,i} S_{i+qk}}{t_i}$ 
  end do
end do.

```

گام پنجم:

ماتریس معکوس، $W|_{z=0}$ است.

۳ مثال عددی

در این بخش با مثال عددی میزان خطای نسبی الگوریتم را نشان می‌دهیم:

مثال ۱۰۳. معکوس ماتریس یازده‌قطری 11×11 زیر را بیابید.

$$G_{\delta,11}^{(2)} = \begin{bmatrix} 2 & 0 & 1 & 0 & -2 & 0 & -1 & 0 & 3 & 0 & 1 \\ 0 & -1 & 0 & -1 & 0 & 3 & 0 & -1 & 0 & -1 & 0 \\ -1 & 0 & -1 & 0 & -1 & 0 & 2 & 0 & 2 & 0 & 2 \\ 0 & 2 & 0 & 3 & 0 & 2 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 3 & 0 & 2 & 0 & 1 & 0 & -1 & 0 & -1 \\ 0 & 1 & 0 & 4 & 0 & -2 & 0 & 2 & 0 & 1 & 0 \\ 3 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 2 \\ 0 & 3 & 0 & 1 & 0 & -1 & 0 & 3 & 0 & 1 & 0 \\ 4 & 0 & 2 & 0 & -1 & 0 & 3 & 0 & -1 & 0 & -2 \\ 0 & 1 & 0 & 2 & 0 & 1 & 0 & 2 & 0 & 3 & 0 \\ -1 & 0 & 1 & 0 & 2 & 0 & 1 & 0 & 2 & 0 & 3 \end{bmatrix}$$

حل: با استفاده از الگوریتم ۲۰۲ داریم:

$$\bullet \det(G_{\delta,11}^{(2)}) = \prod_{i=1}^{11} t_i = 5250,$$

$$W = \begin{bmatrix} -\frac{2}{35} & 0 & -\frac{72}{35} & 0 & -\frac{11}{5} & 0 & -\frac{32}{35} & 0 & \frac{8}{5} & 0 & \frac{82}{35} \\ 0 & -\frac{11}{25} & 0 & \frac{11}{25} & 0 & -\frac{9}{25} & 0 & \frac{2}{25} & \frac{8}{5} & -\frac{1}{5} & 0 \\ \frac{2}{7} & 0 & \frac{44}{7} & 0 & 7 & 0 & \frac{25}{7} & 0 & -5 & 0 & -\frac{54}{7} \\ 0 & \frac{1}{50} & 0 & \frac{11}{75} & 0 & \frac{16}{75} & 0 & -\frac{7}{50} & 0 & -\frac{1}{15} & 0 \\ -\frac{16}{35} & 0 & -\frac{279}{35} & 0 & -\frac{42}{5} & 0 & -\frac{159}{35} & 0 & \frac{31}{5} & 0 & \frac{344}{35} \\ 0 & \frac{7}{25} & 0 & \frac{4}{75} & 0 & -\frac{1}{75} & 0 & \frac{1}{25} & 0 & \frac{1}{15} & 0 \\ -\frac{6}{35} & 0 & -\frac{29}{35} & 0 & -\frac{7}{5} & 0 & -\frac{29}{35} & 0 & \frac{6}{5} & 0 & \frac{59}{35} \\ 0 & \frac{33}{50} & 0 & -\frac{37}{75} & 0 & \frac{28}{75} & 0 & \frac{19}{50} & 0 & \frac{2}{15} & 0 \\ -\frac{8}{35} & 0 & -\frac{332}{35} & 0 & -\frac{51}{5} & 0 & -\frac{202}{35} & 0 & \frac{28}{5} & 0 & \frac{417}{35} \\ 0 & -\frac{2}{5} & 0 & \frac{1}{15} & 0 & -\frac{4}{15} & 0 & -\frac{1}{5} & 0 & \frac{1}{3} & 0 \\ \frac{12}{35} & 0 & \frac{323}{35} & 0 & \frac{49}{5} & 0 & \frac{198}{35} & 0 & -\frac{37}{5} & 0 & -\frac{398}{35} \end{bmatrix}.$$

میزان خطای نسبی برابر است با،

$$\frac{\|GW - I\|_F}{\|I\|_F} = 29246 \times 10^{-15}.$$

۴ نتایج عددی

الگوریتم ۲۰۲ را برای m, n و k ‌های متفاوت اجرا و نتایج زیر را بدست آوردیم. در جدول زیر زمان برحسب ثانیه و الگوریتم با نرم‌افزار $Matlab(2019)$ انجام شده است و نتایج عددی بدست آمده با نتایج عددی دستور inv در نرم‌افزار $Matlab$ از لحاظ زمان اجرا مقایسه شده‌اند. همانطور که نتایج این بخش نشان می‌دهند الگوریتم ارائه شده در این مقاله برای ماتریس‌هایی با بعد بالا خصوصاً زمانی که $m \ll n$ باشد، نتایج قابل قبولی را نشان می‌دهد.

جدول ۱: مقایسه زمان الگوریتم ۲.۲ با دستور *inv* در *Matlab* و محاسبه خطا

$\frac{\ GW-I\ _F}{\ I\ _F}$	$time(s)_{inv(Matlab)}$	$time(s)_{newAlgorithm}$	k	m	n
$3/3683 \times 10^{-12}$	۰/۷۸۴۲	۰/۷۰۳۷	۶	۹	۳۰۰۰
$5/6838 \times 10^{-11}$	۱/۷۸۲۱	۱/۱۶۹۴	۷	۱۰	۴۰۰۰
$3/9056 \times 10^{-11}$	۳/۳۷۴۶	۳/۱۰۱۸	۱۰	۲۰	۵۰۰۰
$3/۱۳۹۶ \times 10^{-11}$	۵/۶۶۰۹	۴/۲۰۵۶	۸	۲۰	۶۰۰۰
$2/۷۳۱۳ \times 10^{-11}$	۳۱/۶۴۴۳	۱۷/۰۹۶۷	۱۵	۳۰	۱۰۰۰۰
$۱/۱۹۹۱ \times 10^{-10}$	۵۹/۴۰۵۴	۳۴/۸۳۴۳	۲۰	۵۰	۱۲۰۰۰

مراجع

- [۱] م. شمس سولاری، م. رسولی، الگوریتمی برای محاسبه معکوس هر ماتریس r -قطری، موجک ها و جبرخطی ۳(۷) ۷۴-۷۹، ۱۴۰۰.
- [2] J. Jia, S. Li, "Symbolic algorithms for the inverses of general k-tridiagonal matrices," *Computers and Mathematics with Applications*, 70, 3032–3042, 2015.
- [3] M. El-Mikkawy, "A Generalized Symbolic Thomas Algorithm," *Applied Mathematics*, 3, 342-345, 2012.
- [4] M. El-Mikkawy, F. Atlan, "A novel algorithm for inverting a general k-tridiagonal matrix," *Applied Mathematics Letters*, 32, 41-47, 2014.
- [5] F. Diele, L. Lopez, "The use of the factorization of five-diagonal matrices by tridiagonal Toeplitz matrices," *Applied Mathematics Letters*, 11, 61-69, 1998.
- [6] Y. Lin, X. Lin, "A novel algorithm for inverting a k-pentadiagonal matrix," *Journal of Shaanxi University of Science and Technology*, 578-582, 2016.