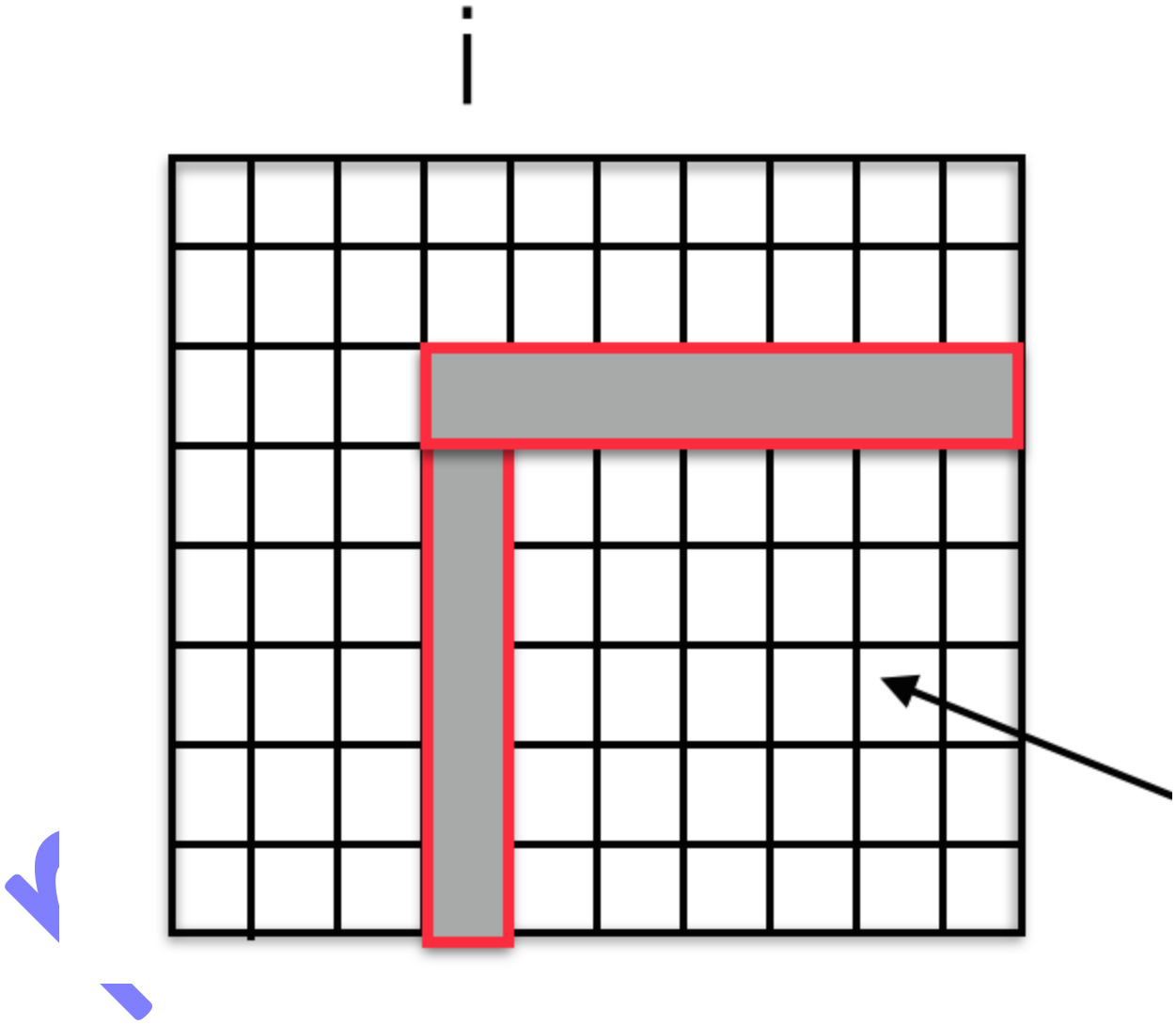
گزارش کامل فارسی متد تجزیه LU (روش دولیتل) به همراه کد متلب و مثال های کاربردی



معرفی

میتوان نشان داد که هر ماتریس مربعی A میتواند به صورت ضرب یک ماتریس پایین مثلثی L و یک ماتریس بالا مثلثی U بیان شود، به عبارتی:

$$A = LU \quad (1)$$

فرآیند محاسبه L و U برای یک ماتریس معلوم A به عنوان تجزیه LU یا فاکتورگیری LU شناخته میشود. تجزیه LU یک عملیات با مقدار مشخص نیست، به عبارت دیگر بدون اعماد قیود خاص بر روی L و U، بی نهایت حالت برای ماتریس های بالا مثلثی و پایین مثلثی ایجاد شده وجود دارد. این قیود اعمالی یک نوع تجزیه را از نوع دیگر متمایز میکند. سه تا از عمومی ترین تجزیه های مورد استفاده در جدول زیر لیست شده اند.

جدول ۱: متدهای تجزیه LU

Name	Constraints
Doolittle's decomposition	$L_{ii} = 1, \quad i = 1, 2, \ldots, n$
Crout's decomposition	$U_{ii} = 1, \quad i = 1, 2, \ldots, n$
Choleski's decomposition	$\mathbf{L} = \mathbf{U}^T$

بعد از تجزیه A ، حل کردن معادله Ax=b ساده است. در ابتدا ما معادله را به صورت Lux = b بازنویسی میکنیم. با استفاده کردن از تبدیل Ux = y، معادله به صورت زیر میشود:

$$Ly = b$$

که به وسیله جایگزینی رو به جلو برای y حل خواهد شد. سپس :

$$Ux = y$$

که با فرآیند جایگزینی از انتها، مقدار x به دست خواهد آمد.

مزیت های روش تجزیه LU نسبت به روش حذفی گوس این است که وقتی یکبار ماتریس A تجزیه شد، ما میتوانیم Ax=b را برای تعداد زیادی از بردارهای ثابت b حل کنیم. هزینه هر حل اضافی نسبتا پایین است، چون که عملیات جایگزینی از ابتدا و انتها نسبت به فرآیند تجزیه خیلی زمان کمتری را مصرف میکند.

فاز حذف

روش تجزیه دولیتل بسیار نزدیک به روش حذفی گوس است. بخ منظور شرح دادن این رابطه ، ماتریس ۳ × ۳ به نام A را در نظر بگیرید و فرض کنید که ماتریس های مثلثی زیر وجود دارند به گونه ای که A=LU.

$$\mathbf{L} = \begin{bmatrix} 1 & 0 & 0 \\ L_{21} & 1 & 0 \\ L_{31} & L_{32} & 1 \end{bmatrix} \qquad \mathbf{U} = \begin{bmatrix} U_{11} & U_{12} & U_{13} \\ 0 & U_{22} & U_{23} \\ 0 & 0 & U_{33} \end{bmatrix}$$

پس از کامل کردن ضرب در سمت راست، ما داریم:

$$\mathbf{A} = \begin{bmatrix} U_{11} & U_{12} & U_{13} \\ U_{11}L_{21} & U_{12}L_{21} + U_{22} & U_{13}L_{21} + U_{23} \\ U_{11}L_{31} & U_{12}L_{31} + U_{22}L_{32} & U_{13}L_{31} + U_{23}L_{32} + U_{33} \end{bmatrix} \qquad (۱)$$

بگذارید اکنون روش حذفی گوس را بر روی معادله ۱ اعمال کنیم. پاس اول روش حذفی شامل انتخاب ردیف اول به عنوان ردیف محوری و اعمال عملیات اولیه است:

$$\text{row } 2 \leftarrow \text{row } 2 - L_{21} \times \text{row } 1 \text{ (eliminates } A_{21})$$

$$\text{row } 3 \leftarrow \text{row } 3 - L_{31} \times \text{row } 1 \text{ (eliminates } A_{31})$$

نتیجه به صورت زیر است:

$$\mathbf{A}' = \begin{bmatrix} U_{11} & U_{12} & U_{13} \\ 0 & U_{22} & U_{23} \\ 0 & U_{22}L_{32} & U_{23}L_{32} + U_{33} \end{bmatrix}$$

در پاس بعدی ما ردیف دوم را به عنوان ردیف محوری در نظر میگیریم و معادلات را به کار میگیریم:

$$\text{row } 3 \leftarrow \text{row } 3 - L_{32} \times \text{row } 2 \text{ (eliminates } A_{32})$$

در پایان:

$$\mathbf{A}'' = \mathbf{U} = \begin{bmatrix} U_{11} & U_{12} & U_{13} \\ 0 & U_{22} & U_{23} \\ 0 & 0 & U_{33} \end{bmatrix}$$

توضیحات زیر دو ویژگی مهم روش تجزیه دولیتل را نشان میدهد:

- ماتریس U با ماتریس بالا مثلثی که از روش حذفی گوس به دست می آید، یکسان است.
- درایه های زیر قطر اصلی ماتریس L ضرایب معادلات محوری هستند که در طول روش حذفی گوس مورد استفاده قرار میگیرند. که L_{ij} ضریبی است که A_{ij} را حذف میکند.

این عمل یک کار مرسوم است که ضرایب را در بخش پایین مثلثی ماتریس ضرایب به وسیله جایگزین کردن آنها با ضرایبی که حذف شده اند، نگه داریم (L_{ij} به جای A_{ij}). المان های قطری ماتریس L نباید نگه داشته شوند، چون همانطور که میدانید همه آنها برابر با ۱ هستند. فر منهایی ماتریس ضرایب به صورت ترکیبی از L و U و به صورت زیر خواهد بود:

(۲)

$$[\mathbf{L}\setminus\mathbf{U}] = \begin{bmatrix} U_{11} & U_{12} & U_{13} \\ L_{21} & U_{22} & U_{23} \\ L_{31} & L_{32} & U_{33} \end{bmatrix}$$

این الگوریتم برای تجزیه دولیتل با روش حذفی گوس تقریبا یکسان است با این تفاوت که هر ضریب λ در قسمت پایین مثلثی ماتریس A ذخیره میشود.

کد فاز تجزیه در روش تجزیه LU دولیتل

در این نسخه از روش تجزیه LU ماتریس اصلی A در ابتدا نابود میشود و سپس با فرم تجزیه شده خود جایگزین میگردد ([L/U]).

```
function A = LUdec(A)
% LU decomposition of matrix A; returns A = [L\U].
% USAGE: A = LUdec(A)
n = size(A,1);
for k = 1:n-1
  for i = k+1:n
    if A(i,k) ~= 0.0
    lambda = A(i,k)/A(k,k);
    A(i,k+1:n) = A(i,k+1:n) - lambda*A(k,k+1:n);
    A(i,k) = lambda;
    end
  end
end
```