

Um Algoritmo Sistólico para Decomposição LU

Kunio Okuda*

Universidade de São Paulo

Instituto de Matemática e Estatística

Departamento de Ciência da Computação

18 de setembro de 1992

Sumário

Neste trabalho apresentamos um algoritmo sistólico para decomposição LU de uma matriz densa $n \times n$. O tempo de execução é de $3n$ passos enquanto o algoritmo proposto por Kung e Leiserson é de $4n$. O seu funcionamento é inverso ao sistema sistólico para multiplicação de matrizes do mesmo autor. O sistema proposto usa uma disposição em forma de matriz quadrada de células básicas e é totalmente modular, usando apenas um tipo de célula, ao passo que o de Kung e Leiserson usa dois tipos de célula. Após a descrição detalhada do sistema proposto, apresentamos também uma prova da sua correção.

Abstract

In this work we present a systolic algorithm for LU decomposition of a dense $n \times n$ matrix. The execution time is $3n$ steps while the algorithm proposed by Kung e Leiserson uses $4n$ steps. The way it works is the inverse of the systolic system for matrix multiplication by the same author. The proposed system uses a mesh-connected layout of the basic cells which is totally modular, using only one type of cell, while the algorithm of Kung e Leiserson uses two types of cells. After a detailed description of the proposed system, we also present a proof of its correctness.

*O autor é mestre em matemática aplicada, professor assistente do IME-USP e membro do projeto de pesquisa em Computação Paralela do IME/USP, que conta com apoio do BID/USP-área científica (projeto 30.01), CNPq/PROTEM (Projeto Temático em Ciência da Computação) e da Fapesp-processo 88/3095-9.

1 Decomposição LU

O problema de *decomposição LU* consiste em decompor uma matriz quadrada A em um produto de uma matriz triangular inferior L com uma matriz triangular superior U (ver fig. 1). Seja $Ax = b$ um sistema de equações lineares. Uma vez que A é decomposta em LU , o sistema $Ax = b$ é equivalente a $LUx = b$ que é resolvido de seguinte maneira:

1. Resolve-se o sistema linear triangular $Ly = b$, obtendo-se y
2. Em seguida, resolve-se o sistema linear triangular $Ux = y$, obtendo-se x .

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ l_{21} & 1 & 0 & 0 \\ l_{31} & l_{32} & 1 & 0 \\ l_{41} & l_{42} & l_{43} & 1 \end{pmatrix} \times \begin{pmatrix} u_{11} & u_{12} & u_{13} & u_{14} \\ 0 & u_{22} & u_{23} & u_{24} \\ 0 & 0 & u_{33} & u_{34} \\ 0 & 0 & 0 & u_{44} \end{pmatrix}$$

Figura 1: Decomposição LU para uma matriz 4×4

Temos o seguinte teorema sobre decomposição LU [8].

Teorema 1 *Seja A uma matriz $n \times n$ tal que $\det(A) \neq 0$, então existe uma matriz \tilde{A} obtida por permutação de linhas (pivotamento) de modo que $\tilde{A} = LU$ onde L e U são matrizes triangulares inferior e superior respectivamente.*

Vamos supor que A tem a propriedade de que sua decomposição LU possa ser feita sem pivotamento (por exemplo, A matriz simétrica positiva definida, irredutível ou diagonal dominante).

A decomposição LU de uma matriz A , onde $L = (l_{ij})$ e $U = (u_{ij})$ são matrizes triangulares inferior e superior respectivamente, é obtida com a seguinte recorrência:

$$\begin{aligned} a_{ij}^{(1)} &= a_{ij} \\ a_{ij}^{(k)} &= a_{ij}^{(k-1)} + l_{ik}(-u_{kj}) \\ l_{ik} &= \begin{cases} 0 & \text{se } i < k \\ 1 & \text{se } i = k \\ a_{ik}^{(k)} u_{kk}^{-1} & \text{se } i > k \end{cases} \\ u_{kj} &= \begin{cases} 0 & \text{se } k > j \\ a_{kj}^{(k)} & \text{se } k \leq j \end{cases} \end{aligned}$$

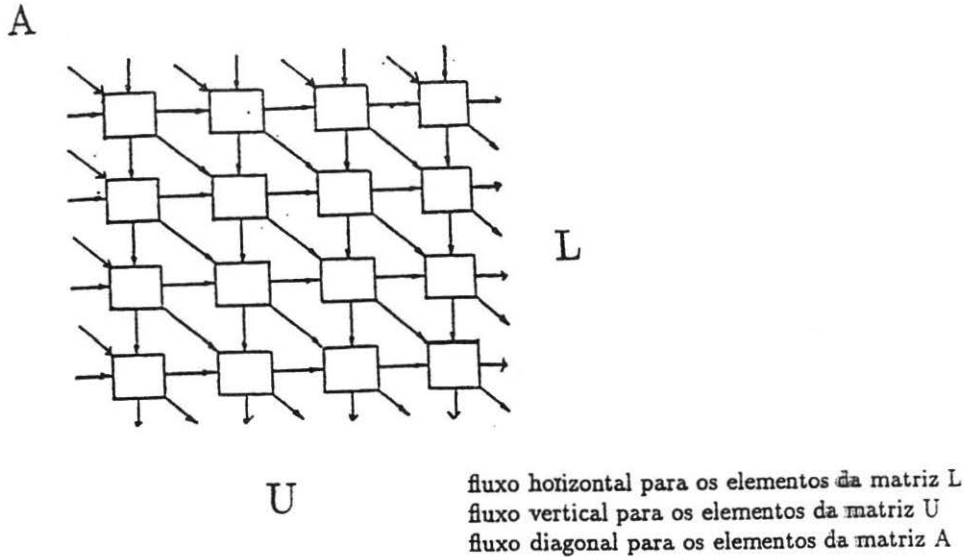


Figura 2: Sistema Sistólico para Decomposição LU

2 Sistema Sistólico para Decomposição LU

Em 1979, junto com seus outros trabalhos pioneiros que introduziram o conceito de *sistema sistólico*, Kung e Leiserson [2] apresentaram uma solução sistólica para a decomposição LU de uma matriz de banda. Naquela solução, em cada direção uma das três células está ativa em cada instante e o tempo requerido é de $4 \times n$ passos para uma matriz densa de $n \times n$.

A seguir mostraremos um sistema sistólico alternativo usando um dispositivo retangular de n^2 células básicas com 3 fluxos de dados conforme a fig. 2.

Mostraremos ainda que este sistema realiza a decomposição LU em $3n$ passos.

O funcionamento básico do sistema é o seguinte:

Pela fig. 2, os coeficientes da matriz A entram no sistema na direção diagonal. Cada célula (ij) , isto é, da linha i e coluna j , começa a atuar após receber a_{ij} e irá gerar u_{ij} ou l_{ij} , dependendo da sua coordenada (i, j) . Após a geração de u_{ij} ou l_{ij} , a célula (i, j) cessará a sua atividade, servindo apenas para passagem de dados.

Este sistema só produz l_{ij} para $i > j$ e u_{ij} para $i \leq j$ pois somente estes elementos nos interessam.

Informalmente, pode-se dizer que à medida que os elementos da matriz A caminham na direção diagonal, cada célula obtém os elementos das matrizes U e L da maneira inversa do produto $L \times U$. Assim, a operação de decomposição LU pode ser vista como a inversa da

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix} = \begin{pmatrix} u_{11} & u_{12} & u_{13} & u_{14} \\ l_{21}u_{11} & l_{21}u_{12} + u_{22} & l_{21}u_{13} + u_{23} & l_{21}u_{14} + u_{24} \\ l_{31}u_{11} & l_{31}u_{12} + l_{32}u_{22} & l_{31}u_{13} + l_{32}u_{23} + u_{33} & l_{31}u_{14} + l_{32}u_{24} + u_{34} \\ l_{41}u_{11} & l_{41}u_{12} + l_{42}u_{22} & l_{41}u_{13} + l_{42}u_{23} + l_{43}u_{33} & l_{41}u_{14} + l_{42}u_{24} + l_{43}u_{34} + u_{44} \end{pmatrix}$$

Figura 3: Matriz A re-escrita

operação de multiplicação de matrizes. Não é de se surpreender, portanto, que o sistema aqui proposto tenha tanta semelhança com o sistema sistólico proposto por Okuda e Song[4] para multiplicação de matrizes.

Por efeito de ilustração usaremos $n = 4$. A matriz A pode ser reescrita como na fig. 3.

Pelas Fig.2 e fig.3 e observando que cada célula (i, j) vai gerar u_{ij} ou $l_{i,j}$, podemos ver facilmente que:

1. Cada célula $(1, j)$, para $1 \leq j \leq n$, da primeira linha simplesmente vai transferir o valor de a_{1j} como u_{1j} . Uma vez gerado, u_{1j} caminha na direção vertical, de cima para baixo.
2. Cada célula $(i, 1)$, para $2 \leq i \leq n$, da primeira coluna (exceto a célula $(1, 1)$) vai gerar $l_{i,1}$ utilizando u_{11} que caminha na direção vertical de cima para baixo, após ser gerado na célula $(1, 1)$. Após a sua geração, $l_{i,1}$ caminha para direita.
3. Quanto às células da sub-matriz (i, j) , para $2 \leq i, j \leq n$, elas vão gerar $u_{i,j}$ ou $l_{i,j}$, dependendo da sua coordenada, utilizando $l_{i,j}$ s geradas nas células que estão à esquerda e $u_{i,j}$ s geradas nas células que estão acima.

Por exemplo, a célula $(2, 2)$ da fig.3 vai gerar u_{22} utilizando u_{12} gerado na célula $(1, 2)$ e l_{21} na célula $(2, 1)$: $u_{22} = a_{22} - l_{21}u_{12}$. A célula $(3, 2)$ vai gerar l_{32} utilizando u_{12} , u_{22} e l_{31} gerados nas células $(1, 2)$, $(2, 2)$ e $(3, 1)$ respectivamente: $l_{32} = (a_{32} - l_{31}u_{12})/u_{22}$.

A fig.4 mostra esquematicamente a observação acima.

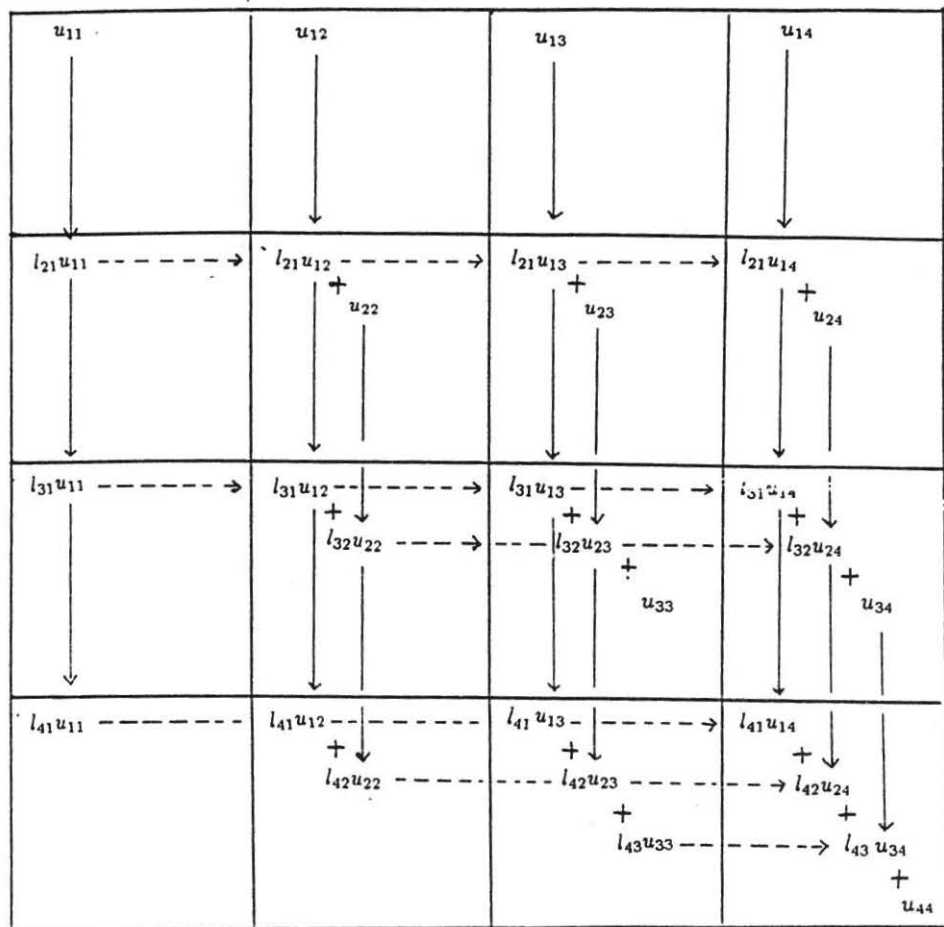
Assim, a computação de u_{ij} ou l_{ij} utiliza a quantidade auxiliar $t = a_{ij} - \sum_{k=1}^h l_{ik}u_{kj}$ ($h = \min(i, j)$), que será calculada na célula (i, j) .

3 Descrição da célula básica

Cada célula (i, j) apresenta registradores denominados a, l, u, t correspondentes aos coeficientes das matrizes A, L, U e a quantidade auxiliar descrita anteriormente.

Fazemos as seguintes observações:

1. Cada a_{ij} participa em cálculo somente na célula (i, j) . Não há nenhuma computação com ele nas demais células, antes e depois, por onde ele passa.

Figura 4: geração e utilização de elementos u e l

2. O tipo de operações que cada célula executa em cada instante depende de presença de determinados dados.
3. Após a geração de u_{ij} ou l_{ij} a célula é desativada e serve apenas para passagem de dados.

Conforme as observações acima, temos 5 registradores binários para controlar o comportamento de cada célula:

V_e e V_x ambos estacionários em cada célula com valores iniciais 1.

V_a, V_u e V_l cujos valores caminham com os fluxos de dados a_{ij}, u_{ij}, l_{ij} respectivamente.

Os seus valores têm o seguinte significado.

$$\begin{aligned}
 V_e &= \begin{cases} 1 & \text{a célula está ativa, i.e. ainda não gerou } u_{ij} \text{ ou } l_{ij} \\ 0 & \text{a célula não está ativa, i.e. já gerou } u_{ij} \text{ ou } l_{ij} \end{cases} \\
 V_x &= \begin{cases} 1 & \text{a célula } (i, j) \text{ não recebeu } a_{ij} \\ 0 & \text{a célula } (i, j) \text{ já recebeu } a_{ij} \end{cases} \\
 V_a &= \begin{cases} 1 & a_{ij} \text{ não foi usado pela célula } (i, j) \\ 0 & a_{ij} \text{ já foi usado pela célula } (i, j) \end{cases} \\
 V_u &= \begin{cases} 1 & \text{quando acompanha } u_{ij} \text{ gerado} \\ 0 & \text{quando acompanha dado inválido (a ser ignorado)} \end{cases} \\
 V_l &= \begin{cases} 1 & \text{quando acompanha } l_{ij} \text{ gerado} \\ 0 & \text{quando acompanha dado inválido (a ser ignorado)} \end{cases}
 \end{aligned}$$

Cada célula realiza o seguinte algoritmo:

```

 $a_{out} \leftarrow a_{in};$ 
if  $V_e V_a V_x$ 
  then begin
     $t \leftarrow a_{in};$ 
     $V_x \leftarrow 0;$ 
     $V_{a_{out}} \leftarrow 0;$ 
  end;

if  $V_e \bar{V}_x \bar{V}_u \bar{V}_l$ 
  then begin
     $u_{out} \leftarrow t;$ 
     $V_{u_{out}} \leftarrow 1;$ 
     $V_e \leftarrow 0;$ 
  end;

if  $V_e \bar{V}_x V_u \bar{V}_l$ 

```

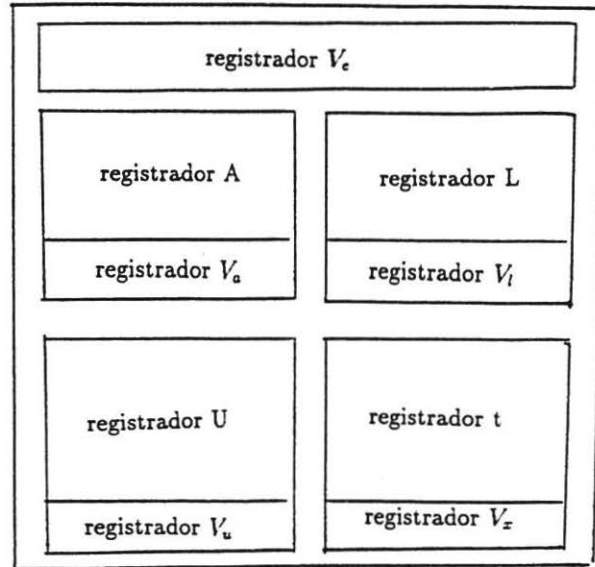


Figura 5: estrutura de célula básica

```

then  $t \leftarrow t - l_{in}u_{in}$ ;

if  $V_e\bar{V}_xV_u\bar{V}_l$ 
then begin
   $l_{out} \leftarrow t/u_{in}$ ;
   $V_{out} \leftarrow 1$ ;
   $V_e \leftarrow 0$ ;
end;
```

Chamamos a atenção ao fato de que o algoritmo acima é apenas para ilustrar o funcionamento do circuito. Por exemplo, as condições do segundo ao quarto if são exclusivas e a seleção de uma das três é feita num só passo no circuito que não é sequencial como indicado pelo algoritmo.

Esquematicamente cada célula tem estrutura como na fig. 5. O circuito completo de cada célula é visto na fig. 6 e nesta figura as posições relativas de registradores são alteradas em relação à fig. 5 de propósito.

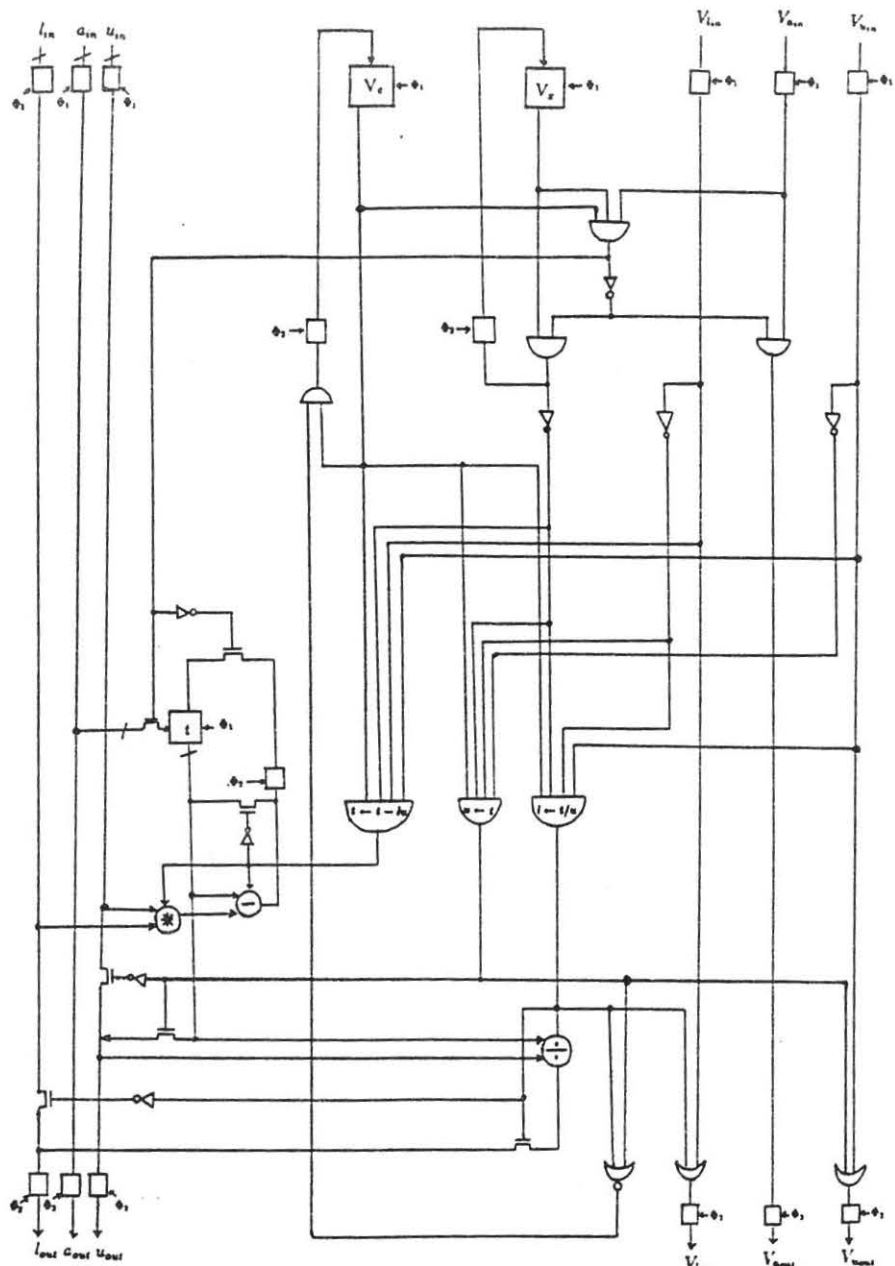


Figura 6: Circuito

4 Correção

Lema 1 Com o algoritmo descrito na seção 2, cada célula gera:

u_{ij} no $[(i-1) \times 2 + j]^{\text{º}}$ passo, se $i \leq j$

ou

l_{ij} no $[i + (j-1) \times 2]^{\text{º}}$ passo, se $i > j$

onde

$$u_{ij} = \begin{cases} a_{ij} & \text{se } i = 1 \\ a_{ij} - \sum_{k=1}^{i-1} l_{ik}u_{kj} & \text{se } i < j \end{cases}$$

e

$$l_{ij} = \begin{cases} a_{ij}/u_{11} & \text{se } j = 1 \\ (a_{ij} - \sum_{k=1}^{i-1} l_{ik}u_{kj})/u_{jj} & \text{se } j > 1 \end{cases}$$

Demonstração

A demonstração do Lema é feita por indução. O passo da indução é feita provando duas proposições que por sua vez também usam indução, i.e., usaremos indução dupla. O sistema sistólico para decomposição LU de uma matriz $A(n \times n)$ é formado por n^2 células em disposição matricial $n \times n$. Definimos região R_k (fig. 7) como segue:

$R_k = (k, 1), (k, 2), \dots, (k, k), (1, k), (2, k), \dots, (k-1, k)$

Então $\bigcup_{k=1}^n R_k$ cobre todas as células do sistema sistólico. (fig.7)

A demonstração do Lema será feita por indução em k .

1. base de indução:
é trivial que célula $(1, 1)$ gera u_{11} no passo 1.
2. Vamos supor que o Lema é valido para qualquer $h, h < k$.
3. Passo da indução:
Consideremos a região R_k e partiremos R_k em duas sub-regiões L_k e C_k onde $L_k = (k, 1), (k, 2), \dots, (k, k-1)$ e $C_k = (1, k), (2, k), \dots, (k, k)$.

Provemos as duas proposições seguintes.

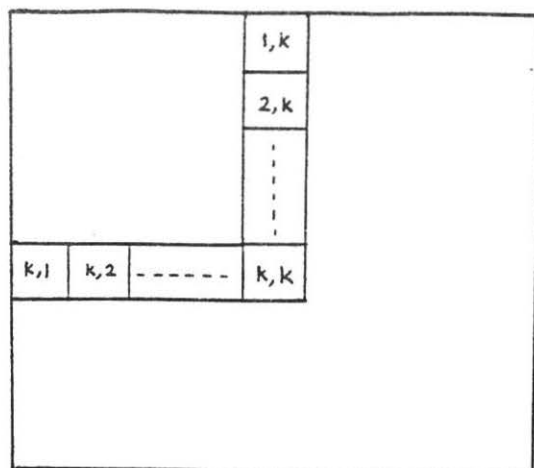
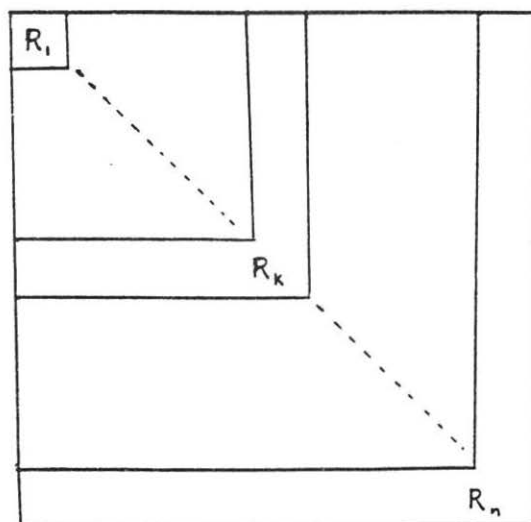
Proposição 1 Em L_k , a célula (k, j) gera l_{kj} no $[k + (j-1) \times 2]^{\text{º}}$ passo.

Proposição 2 Em C_k , a célula (i, k) gera u_{ik} no $[(i-1) \times 2 + k]^{\text{º}}$ passo.

4.1 Demonstração da proposição 1

Indução sobre j .

1. $j = 1$:
é trivial que célula $(k, 1)$ gera l_{k1} no $k^{\text{º}}$ passo.

Figura 7: região R_k Figura 8: União de R_k s

	S	I	II	III	IV
(1, s)	$u_{1,s}$	s	$k-1$	$k+s-1$	
(2, s)	$u_{2,s}$	$s+2$	$k-2$	$k+s$	
⋮	⋮	⋮	⋮	⋮	
(i-1, s)	$u_{s-1,s}$	$3s-4$	$k-s+1$	$k+2s-3$	
(s, s)	$u_{s,s}$	$3s-2$	$k-s$	$k+2s-2$	
	⋮				

* não vão gerar u

K	(k, 1)	(k, 2)	(k, 3)	...	(k, s-1)	(k, s)	(k, s+1)
I	l_{k1}	l_{k2}	l_{k3}	...	$l_{k,s-1}$		
II	k	$k+2$	$k+4$...	$k+2(s-2)$		
III	$s-1$	$s-2$	$s-3$...	1		
IV	$k+s-1$	$k+s$	$k+s+1$...	$k+2s-3$		

* como $i > j$ estas células não geram l_{ij} e não tem influência sobre célula (k,s)

I elemento gerado (por hipótese de indução)

II passo em que o elemento é gerado (por hipótese de indução)

III distância a (k,s)

IV passos em que o elemento chega a (k,s) (II+III)

Figura 9: tabela

2. Vamos supor que vale para qualquer $t, t < s$.

3. passo da indução:

Considere a célula (k, s) . Como foi observado no item 2, a célula (k, s) recebe a_{ks} no $(k + s - 1)^{\text{a}}$ passo e daí começa a sua atividade. Por outro lado, $a_{ks} = \sum_{m=1}^s l_{km} u_{ms}$. Pela hipótese de indução podemos construir a seguinte tabela.(fig.9)

- A célula (k, s) recebe a_{ks} no $(k + s - 1)^{\text{a}}$ passo. Temos $V_e V_a V_x = \text{verdadeiro}$ e logo os comandos: $t \leftarrow a_{in}$; $V_x \leftarrow 0$; e $V_a \leftarrow 0$; produzem $t = a_{ks}$, $V_x = 0$ e $V_a = 0$.
- Pela fig.9, neste momento l_{k1} e u_{1s} também entram nesta célula e portanto $V_e \bar{V}_x V_u V_l = \text{verdadeiro}$. Assim o comando do algoritmo $t \leftarrow t - l_{in} u_{in}$ produz $t = a_{ks} - l_{k1} u_{1s}$
- do $(k + s - 1)^{\text{a}}$ passo até $(k + 2 \times s - 3)^{\text{a}}$ passo, temos sempre $V_e \bar{V}_x V_u V_l = \text{verdadeiro}$ e calculamos $t = a_{ks} - \sum_{m=1}^{s-1} l_{km} u_{ms}$
- No $(k + 2 \times s - 2)^{\text{a}}$ passo, só entra u_{ss} . Assim $V_e \bar{V}_x V_u \bar{V}_l = \text{verdadeiro}$ e calcula-se $b/u_{ss} = l_{ks}$ e a célula (k, s) fica destivada.

Assim demonstramos a proposição 1.

Observamos também que no $(3 \times n - 2)^{\text{a}}$ passo u_{nn} , o último elemento a ser gerado pelo sistema sistólico, será gerado e transferido para fora do sistema sistólico. Neste momento todos os elementos gerados já estão fora e não depositados nas células do sistema como ocorre nos outros exemplos de sistema sistólico (nestes casos geralmente vão levar mais n passos para que todos os elementos gerados saiam do sistema)

4.2 Demonstração da proposição 2

Análoga a demonstração da proposição 1.

Com o Lema acima a corretude do sistema é demonstrada.

5 Conclusão

Destacamos os seguintes pontos.

1. O sistema proposto é totalmente modular, facilmente extensível para n qualquer e é constituído de um só tipo de célula básica.
2. O tempo de processamento é de $3n$ passos não só para geração dos elementos mas também para transferir os resultados para fora do sistema. Existem outros sistemas sistólicos para decomposição LU com tempo de processamento $3n$ mas seus resultados ficam estacionados nas células^[3].

3. Neste trabalho não apenas foi dado o esquema de interconexão entre células e o algoritmo das células básicas mas também detalhamos todo o esquema interno da célula na fig.6.
4. Além disso foi mostrada a correção do sistema sistólico proposto.
5. Como o tempo de processamento é de $3n$, este é um algoritmo sistólico ótimo para decomposição LU ([6]).

Observa-se que o detalhamento da célula básica e a demonstração da correteza não costumam ser tratadas em trabalhos no gênero.

6 Agradecimento

O autor agradece ao Prof. Dr. Siang Wun Song pelas valiosas sugestões e cuidadosas leituras do artigo.

Referências

- [1] Kung,H.T.
Why Systolic Architecture?
Computer 15,1 (Jan,1982)
- [2] Kung,H.T. & Leiserson,C.E.
Systolic Array for VLSI in Duff,I.S.e Stewart,G.W.(ed.),
Sparse Matrix Proceeding 1978,SIAM,1979.
- [3] Moldovan,D.I.
On the Design of Algorithm for VLSI Systolic Array
Proceeding of the IEEE Vol.71 No.1,Jan,1983
- [4] Okuda,K. & Song,S.W.
Um Algoritmo de Multiplicação de Matrizes para Implementação em VLSI,
Anais do I Congresso da Sociedade Brasileira de Microeletrônica,Campinas,
Julho,1986.
- [5] Okuda,K.
Desenvolvimento Formal de Algoritmos Paralelos Sistólicos
dissertação de mestrado,MAC,IME,USP,1989
- [6] Robert,Y.
The Impact of Vector and Parallel Architecture on the Gaussian Elimination Algorithm
Manchester University Press,1990

- [7] Song, S.W.
Algoritmos Paralelos e Arquitetura VLSI
IV Escola de Computação, São Paulo, 1984
- [8] Stern, J.M.
Fatoração L-U e Aplicações
Relatório técnico RT-MAP-8606, MAP-IME-USP, 1986