

# Trabalho nº 2: Fatorização LU

CPD – Algoritmos Paralelos 2015/2016

Nuno André da Silva Oliveira

A67649

Universidade do Minho

MIEI

a67649@alunos.uminho.pt

Carlos Rafael Cruz Antunes

A67711

Universidade do Minho

MIEI

a67711@alunos.uminho.pt

## I. ALTERAÇÕES AO CÓDIGO

Ao código fornecido (BLAS2LU.m) foram adicionadas as seguintes linhas, que aplicam a pivotação parcial à matriz passada por argumento.

As duas primeiras linhas calculam o pivot e a sua posição na coluna em questão.

```
[M, I] = max( A( i:n, i ) );  
linha = I + i - 1;
```

Seguidamente a linha com o pivot é posicionada no topo das linhas por iterar trocando com a que lá está, caso seja necessário.

```
if(linha > i)  
    A = swapLine( A, linha, i);  
end
```

O ficheiro “BLAS3LU.m” foi alterado de forma a que chamasse uma função auxiliar “BLAS3aux.m”. Esta função trabalha com blocos da matriz, mas quando há trocas de linhas, estas são aplicadas à matriz inteira. O código é apresentado em baixo:

```
function A=BLAS3aux(A,j,o,last)  
m = j-o;  
n = j-last;  
for i=1:min(m-1,n)  
    i2=i+j;  
    [M,I] = max(A(i2:n,i2));  
    linha = I+i2-1;  
    if(linha > i2)  
        A=swapLine(A,linha,i2);  
    end  
    A(i2+1:m,i2)=A(i2+1:m,i2)/A(i2,i2);  
    if i2<n  
        A(i2+1:m,i2+1:n)=A(i2+1:m,i2+1:n)-  
        A(i2+1:m,i2)*A(i2,i2+1:n);  
    end  
end
```

## II. TESTES

Para calcular o erro dos resultados obtidos usamos a função separamos as matrizes L e U da matriz obtida através das funções BLAS2LU, BLAS2LUPP, BLAS3LU e BLAS3LUPP. Desta forma podemos usar a linha de MatLab “norm(P\*A - L\*U)” para obter o erro dos cálculos, e compara-lo entre as versões sem e com pivotação parcial.

## III. RESULTADOS OBTIDOS

Na Tabela 1 podemos ver o erro associado aos diferentes algoritmos. É importante mencionar que não obtivemos resultados para a função BLAS3LUPP, visto que por uma razão desconhecida não eram gerados valores para tamanhos de matrizes superiores a 30.

Em geral podemos verificar que os algoritmos sem pivotação mantêm um erro superior aos algoritmos com pivotação. Por outro lado, o tempo de execução é muito semelhante para o mesmo tamanho da matriz.

Na Tabela 2 temos também uma comparação entre o tamanho escolhido para o bloco e o tempo de execução do algoritmo e o seu erro.

Observou-se que o aumento do tamanho do bloco causa diminuição do tempo de execução até um certo ponto, e que este tempo volta a aumentar passando esse limite. Verificou-se também que à medida que se aumenta o tamanho do bloco o erro aumenta progressivamente.

**Tabela 1. Tempo de execução e erro associado aos diferentes algoritmos.**

<b>BLAS2LU</b>		
Tamanho	Tempo	Erro
1000	2,736772	1,25E-10
2000	22,679022	5,20E-10
3000	78,090144	3,57E-10
4000	180,18586	1,59E-08
<b>BLAS2LUPP</b>		
Tamanho	Tempo	Erro
1000	2,743311	1,62E-13
2000	22,620203	4,07E-13
3000	78,246263	6,92E-13
4000	185,88467	1,11E-12
<b>BLAS3LU (Tamanho do bloco = 100)</b>		
Tamanho	Tempo	Erro
1000	0,147273	2,24E-07
2000	0,67343	2,14E-06
3000	1,998677	1,06E-06
4000	4,308496	1,02E-05

**Tabela 2. Tempo de execução e erro da BLAS3LU com uma matriz de 1000x1000 e um tamanho do bloco variável.**

Tamanho do Bloco	Tempo	Erro
<b>20</b>	0,20372	3,09E-08
<b>40</b>	0,145698	2,74E-08
<b>60</b>	0,128968	1,04E-07
<b>80</b>	0,138056	1,57E-07
<b>100</b>	0,145346	2,24E-07
<b>150</b>	0,17504	4,09E-07
<b>200</b>	0,267975	7,83E-07
<b>250</b>	0,368489	2,44E-06

Por outro lado, a utilização de blocos reduz significativamente o tempo de execução reduzindo o tempo de execução 45 vezes superior na versão se blocos. No entanto observou-se que a utilização desta técnica tem impacto na precisão do resultado aumentando o valor do erro. Esta perda de precisão pode ser controlada com a alteração do tamanho do bloco.

#### IV. CONCLUSÕES

Dados os resultados obtidos nesta pequena experiência em MatLab, concluímos que a pivotação parcial desempenha de facto um papel importante na precisão numérica do algoritmo de factorização LU. Reduzindo o erro quando é implementada esta pivotação, o que leva a um maior nível de confiança nos resultados obtidos não havendo perdas significativas no tempo de execução.