

**UNIVERSIDADE FEDERAL DO PARANÁ**  
**GRADUAÇÃO EM BACHARELADO CIÊNCIA DA COMPUTAÇÃO**

**ALEJANDRO GEMIN ROSALES**

**MELHORA DE DESEMPENHO DE SISTEMAS LINEARES POR FATORAÇÃO LU  
APLICADA À INTERPOLAÇÃO POLINOMIAL E AJUSTE DE CURVAS**

CURITIBA  
2021

## **RESUMO**

No presente relatório apresentaremos o efeito das técnicas de otimização nos algoritmos de resolução de sistemas de equações lineares por fatoração LU, aplicado a um problema de interpolação e ajuste de curvas.

**Palavras-chave:**

interpolação; equações; lineares; fatoração; LU; otimização;

## **SUMÁRIO**

<b>RESUMO</b>	<b>2</b>
<b>SUMÁRIO</b>	<b>3</b>
<b>INTRODUÇÃO</b>	<b>4</b>
<b>PROCESSADOR E SUA ARQUITETURA</b>	<b>4</b>
CPU	4
CACHE	4
<b>ALGORITMOS ANALISADOS</b>	<b>5</b>
FUNÇÃO LUDECOMPOSE (sem pivoteamento)	
FUNÇÃO LUSOLVE (com pivoteamento)	
<b>EXPERIMENTOS REALIZADOS</b>	<b>5</b>
<b>OTIMIZAÇÕES</b>	<b>5</b>
<b>GRÁFICOS</b>	<b>6</b>

## 1. INTRODUÇÃO

Neste relatório serão apresentadas as técnicas de otimização que foram aplicadas aos algoritmos que calculam matrizes inversas de matrizes quadradas utilizando o método de fatoração LU. Essas técnicas serão acompanhadas dos testes utilizados para testá-las, além da especificação da máquina para qual foram implementadas.

## 2. PROCESSADOR E SUA ARQUITETURA

Para a realização e avaliação das otimizações é imprescindível o conhecimento da arquitetura do processador. Todos os testes apresentados neste relatório foram realizados na máquina descrita abaixo.

### 2.1. CPU

Nome CPU: Intel(R) Core(TM) i5-7500 CPU @ 3.40GHz

CPU type: Intel Coffelake processor

### 2.2. CACHE

Level:	1
Size:	32 kB
Type:	Data cache
Associativity:	8
Number of sets:	64
Cache line size:	64
Cache type:	Non Inclusive

---

Level:	2
Size:	256 kB
Type:	Unified cache
Associativity:	4
Number of sets:	1024
Cache line size:	64
Cache type:	Non Inclusive

---

Level:	3
Size:	6 MB
Type:	Unified cache
Associativity:	12
Number of sets:	8192
Cache line size:	64
Cache type:	Inclusive

### 3. ALGORITMOS ANALISADOS

Na sessão corrente, apresentaremos breves descrições dos algoritmos que foram submetidos às técnicas de otimização. Versão original e otimizada.

#### 3.1 FUNÇÃO LUDECOMPOSE

A função LUDecompose tem como objetivo a decomposição LU sem pivoteamento. A função de decomposição LU, é determinada por um ponteiro de função.

#### 3.2 FUNÇÃO LUSOLVE

A função LUPdecompose tem como objetivo primordial resolver a inversão da matriz A pelo método de fatoração LU.

### 4. EXPERIMENTOS REALIZADOS

Nesta seção, iremos descrever como foram realizados os testes finais. Para que possamos ter uma garantia de eficácia dos testes, é indispensável que os mesmos sejam realizados em um ambiente controlado. É necessário fixar a frequência do processador. É então executado o comando abaixo:

```
$echo "performance" > sys/devices/system/cpu/cpufreq/policy3/scaling_governor
```

É então nos garantido que a frequência do processador irá se manter a mesma em todos os testes. Foram feitos testes com entradas geradas pelo código gera\_entrada em python. Foram testadas também pelo arquivo sample.

Os códigos são testados com a biblioteca LIKWID para que possamos monitorar o desempenho. Os testes aplicados são:

1. Teste de tempo médio.
2. Banda de memória.
3. Cache L2 miss rate.
4. FLOP\_DP.
5. FLOP\_AVX.

Os testes realizados foram com os seguintes valores de entrada:

10, 32, 50, 64, 100, 128, 200, 256, 300, 400, 512, 1000

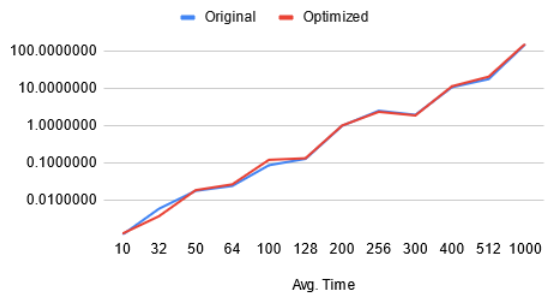
### 5. OTIMIZAÇÕES

Nesta seção iremos falar das otimizações que foram aplicadas aos programas e seus impactos no desempenho dos mesmos.

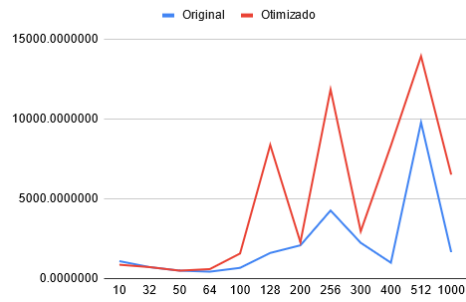
Foi mexido nas duas funções citadas na seção 3. As funções LUdecompose e LUsolve. Nas duas funções tivemos como objetivo de abrir os loops para fazer os cálculos, em buffers que estarão na memória. Também utilizei a forma RowMatriz para quando as matrizes são tratadas com vetores, por exemplo: `mat[i*n+j]` ao invés de `mat[i][j]`.

## 6. GRÁFICOS

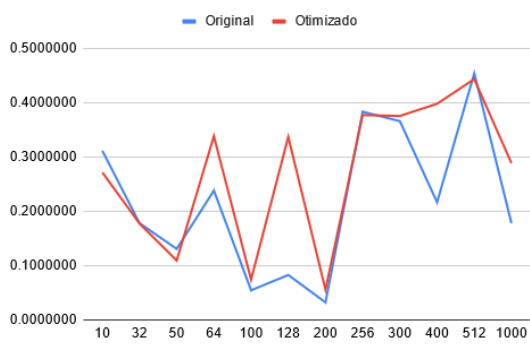
Tempo - Gráfico 1



L3 Bandwidth - Gráfico 3



L2 Miss Rate - Gráfico 2



FLOPS\_DP | FLOPS\_AVX - Gráfico 4

