



UNIVERSIDADE FEDERAL DO PARANÁ
CIÊNCIA DA COMPUTAÇÃO

ANDERSON APARECIDO DO CARMO FRASÃO
ERICK ECKERMANN CARDOSO

ANÁLISE DE DESEMPENHO

CURITIBA
2022

ANDERSON APARECIDO DO CARMO FRASÃO
ERICK ECKERMANN CARDOSO

ANÁLISE DE DESMPENHO

Relatório apresentado a disciplina Introdução à computação Científica, do curso de Ciência da Computação da Universidade Federal do Paraná – UFPR – como requisito parcial para aprovação na disciplina, sob a orientação dos professores Me. Armando Luiz N. Delgado e Dr. Guilherme Alex Derenievicz.

CURITIBA

2022

RESUMO

Após adaptação do código relativo ao trabalho 1, Método de Newton Padrão e Método de Newton Inexato, para que o mesmo aceitasse os algoritmos direcionados a rosenbrock, foram feitas melhorias para otimização de tempo e acesso a memória, e usando a biblioteca likwid direcionada a linguagem de programação C, foram obtidas medidas relacionadas a tempo de execução, banda de memória, miss ratio e operações de ponto flutuante, e com essas informações, foram gerados gráficos com a ferramenta pyplot, utilizando a linguagem de programação python, esses gráficos são utilizados para compara os códigos originais do trabalho 1 com a versão deles otimizada.

Palavras-chave: Otimização, Método de Newton, likwid.

LISTA DE ILUSTRAÇÕES

Figura 1 – FLOPS_AVX Newton Padrão não otimizado.....	11
Figura 2 – FLOPS_AVX Newton Padrão otimizado.....	11
Figura 3 – FLOPS_DP Newton Padrão não otimizado.....	11
Figura 4 – FLOPS_DP Newton Padrão otimizado.....	11
Figura 5 – L2CACHE Newton Padrão não otimizado.....	11
Figura 6 – L2CACHE Newton Padrão otimizado.....	11
Figura 7 – Memory bandwidth Newton Padrão não otimizado.....	12
Figura 8 – Memory bandwidth Newton Padrão otimizado.....	12
Figura 9 – Newton Padrão não otimizado.....	12
Figura 10 – Newton Padrão otimizado.....	12
Figura 11 – FLOPS_AVX Newton Inexato não otimizado.....	13
Figura 12 – FLOPS_AVX Newton Inexato otimizado	13
Figura 13 – FLOPS_DP Newton Inexato não otimizado	13
Figura 14 – FLOPS_DP Newton Inexato otimizado.....	13
Figura 15 – L2CACHE Newton Inexato não otimizado.....	13
Figura 16 – L2CACHE Newton Inexato otimizado.....	13
Figura 17 – Memory bandwidth Newton Inexato não otimizado.....	14
Figura 18 – Memory bandwidth Newton Inexato otimizado.....	14
Figura 19 – Tempo Newton Inexato não otimizado.....	14
Figura 20 – Tempo Newton Inexato otimizado.....	14

SUMÁRIO

1	INTRODUÇÃO.....	04
2	COMPUTADOR UTILIZADO.....	06
3	COMO RODAR OS EXPERIMENTOS.....	09
4	OTIMIZAÇÕES EFETUADAS.....	10
5	GRÁFICOS.....	11
5.1	Método de Newton Padrão.....	11
5.2	Método de Newton Inexato.....	13
6	CONSIDERAÇÕES FINAIS.....	15

INTRODUÇÃO

Aqui sera demonstrado o resultado da otimização de um código feito para resolução de pontos criticos de funções, mais especificamente as funções Rosenbrock, utilizando os métodos de Newton padrão e Newton Inexato.

2 COMPUTADOR UTILIZADO

CPU name: Intel(R) Core(TM) i3-7100U CPU @ 2.40GHz

CPU type: Intel Kabylake processor

CPU stepping: 9

Hardware Thread Topology

Sockets: 1

Cores per socket: 2

Threads per core: 2

HWThread	Thread	Core	Die	Socket	Available
0	0	0	0	0	*
1	0	1	0	0	*
2	1	0	0	0	*
3	1	1	0	0	*

Socket 0: (0 2 1 3)

Cache Topology

Level: 1

Size: 32 kB

Type: Data cache

Associativity: 8

Number of sets: 64

Cache line size: 64

Cache type: Non Inclusive

Shared by threads: 2

Cache groups: (0 2) (1 3)

Level: 2
Size: 256 kB
Type: Unified cache
Associativity: 4
Number of sets: 1024
Cache line size: 64
Cache type: Non Inclusive
Shared by threads: 2
Cache groups: (0 2) (1 3)

Level: 3
Size: 3 MB
Type: Unified cache
Associativity: 12
Number of sets: 4096
Cache line size: 64
Cache type: Inclusive
Shared by threads: 4
Cache groups: (0 2 1 3)

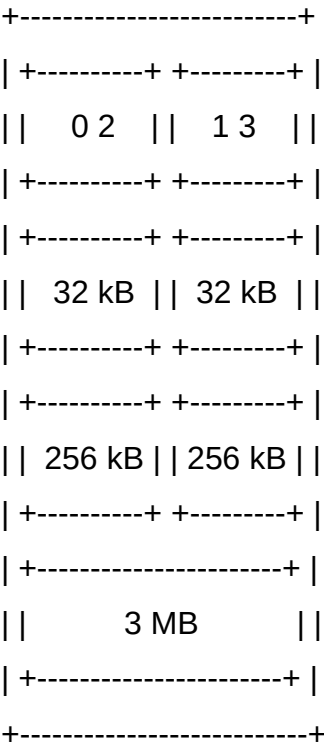
NUMA Topology

NUMA domains: 1

Domain: 0
Processors: (0 2 1 3)
Distances: 10
Free memory: 5370.61 MB
Total memory: 11853.4 MB

Graphical Topology

Socket 0:



Memória RAM:

12 GB DDR4 2400 Mhz

3 COMO RODAR OS EXPERIMENTOS

Basta dar o seguinte comando no diretório aacf20-eec18 :

```
sh geral.sh
```

O script “geral.sh” recompila o código original, gera um arquivo “funcoes.dat”, com as funções rosenbrock, executa o código com as flags necessárias, indicando uma saída apropriada. Após isso ele repete o processo, porém com o código otimizado.

Em seguida é chamado dois scripts python para tratar as informações e gerar os gráficos.

4 OTIMIZAÇÕES EFETUADAS

- Divisão da estrutura de dados SistLinear para torná-la mais coesa.
- Diminuição de parâmetros passado de forma desnecessária
- Otimização na alocação de matrizes
- Melhoria de laço de repetição
- Função triang (Newton padrão)
- Função calcula independente (Newton inexato)
- Minimização de cache thrashing

5 GRÁFICOS

5.1 Método de Newton Padrão

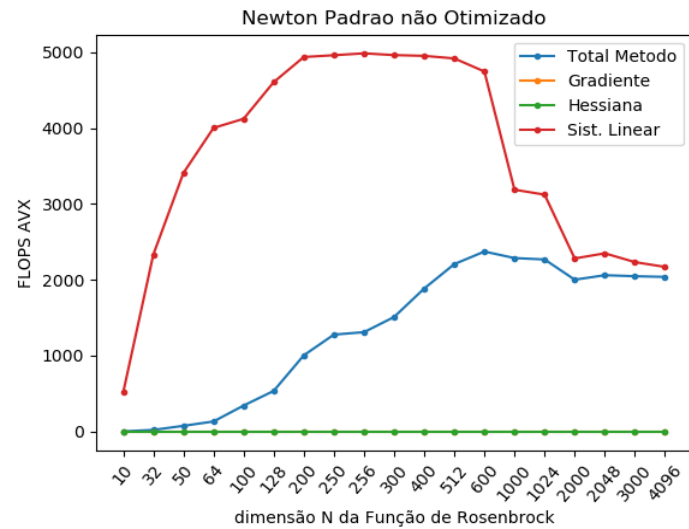


Figura 1

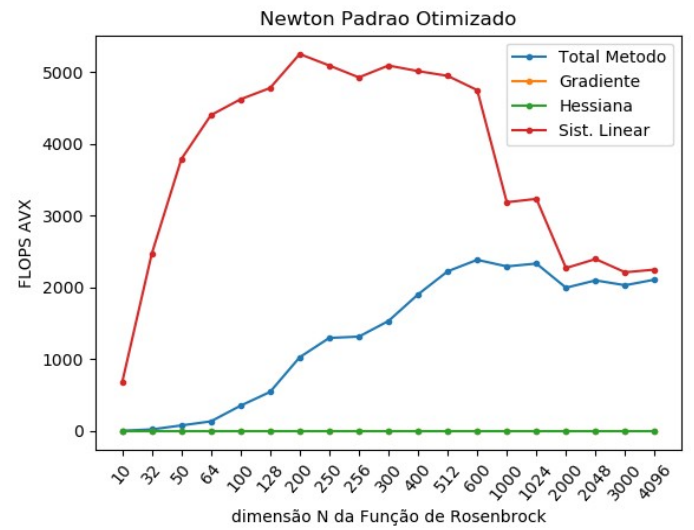


Figura 2

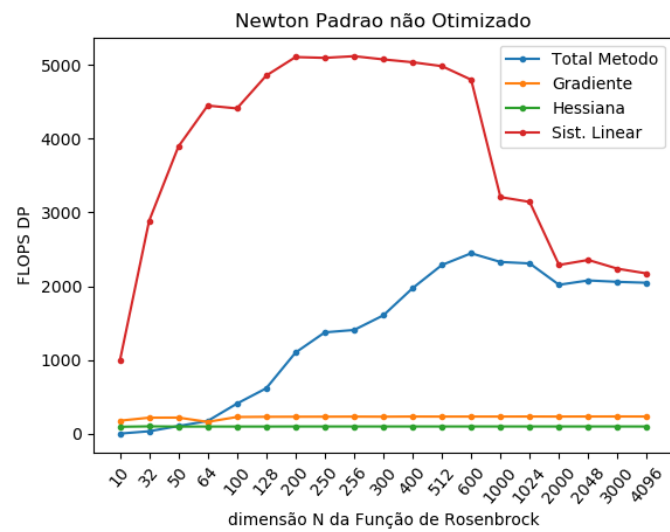


Figura 3

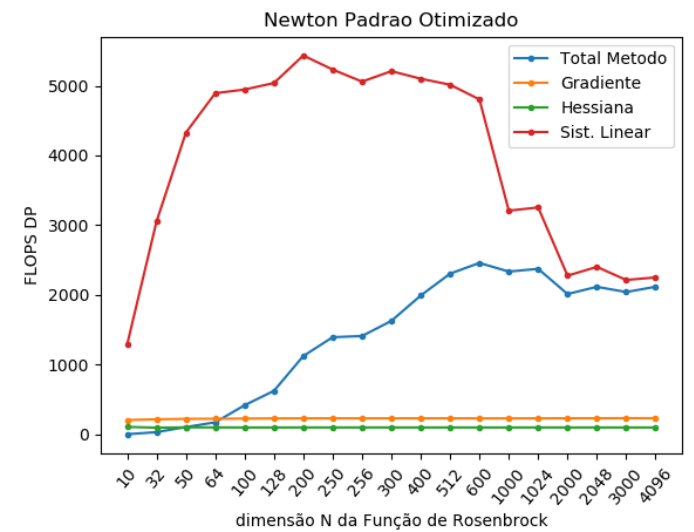


Figura 4

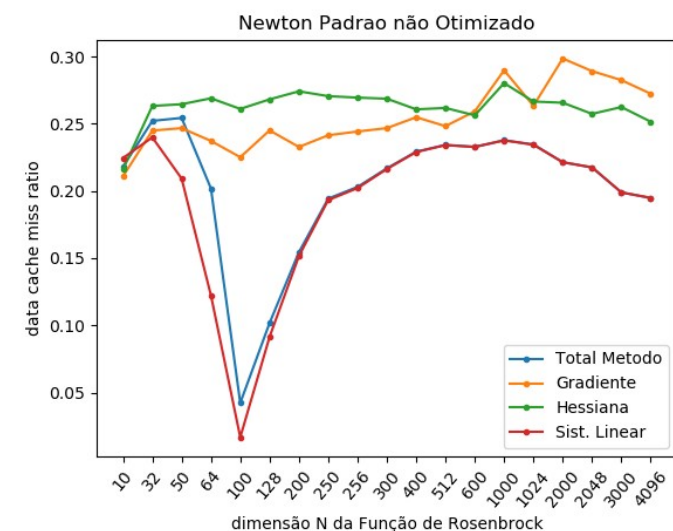


Figura 5

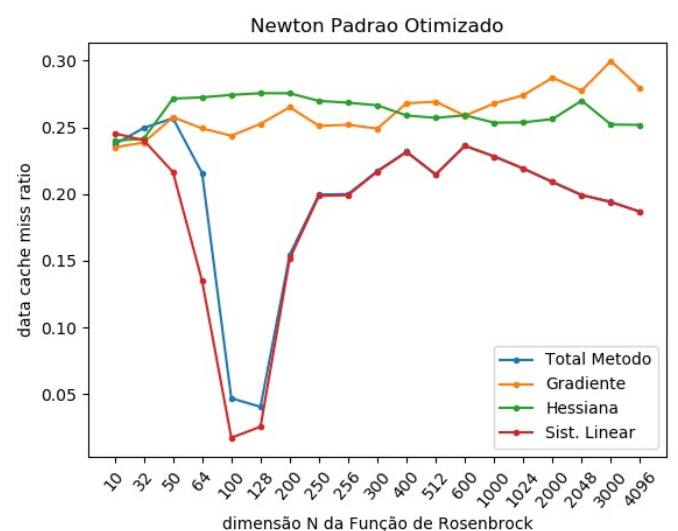


Figura 6

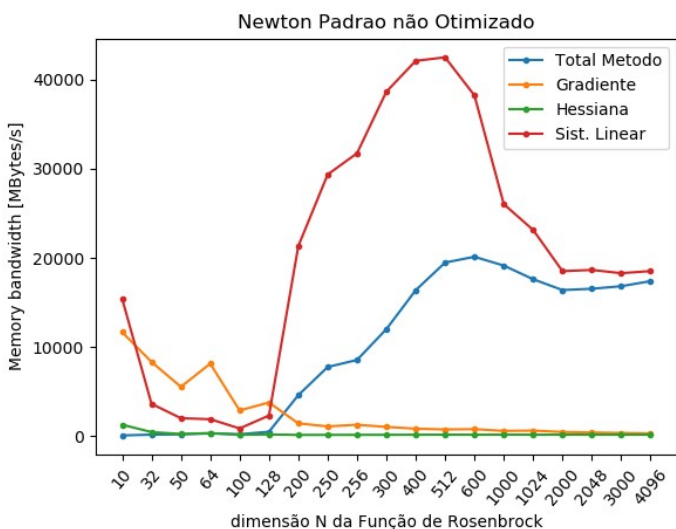


Figura 7

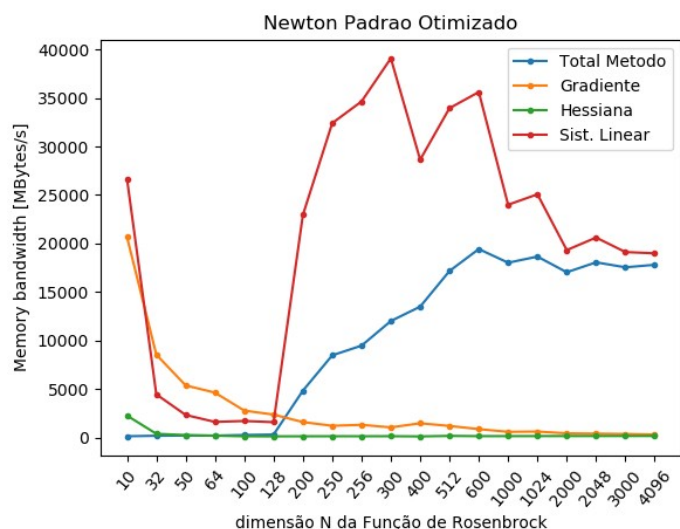


Figura 8

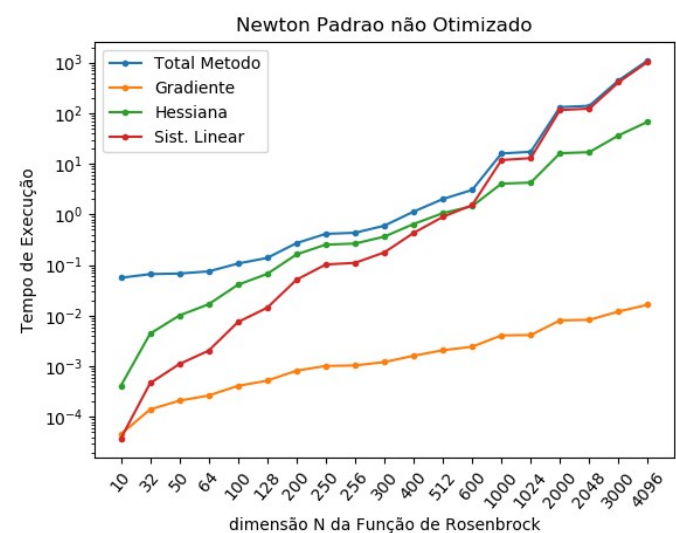


Figura 9

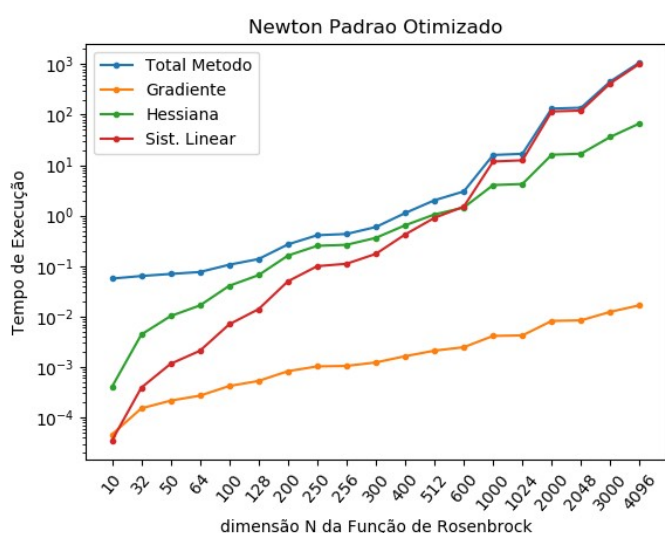


Figura 10

Os tempos de execução se mantiveram parecidos pois não houveram mudanças necessárias para esse requisito.

Houve aumento no numero de operações em diferentes dimensões da função de rosenbrock, devido a melhoria em alguns laços de repetição.

Tiveram-se expressiva queda em miss ratio, pelo fato de mudar o método de alocação de memória.

A banda de memória manteve maior constância, e em vários momentos se manteve mais alta com a otimização, graças ao formato das matrizes alocadas.

Algumas variações se deram pela troca das funções de criação de derivação das funções.

5.2 Método de Newton Inexato

Newton Inexato não Otimizado

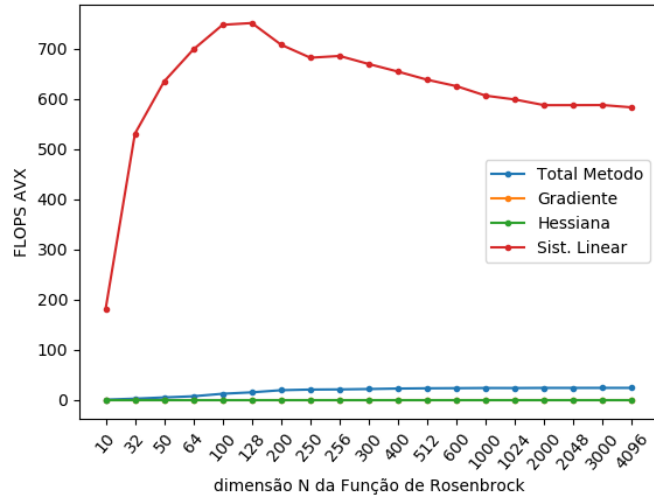


Figura 11

Newton Inexato Otimizado

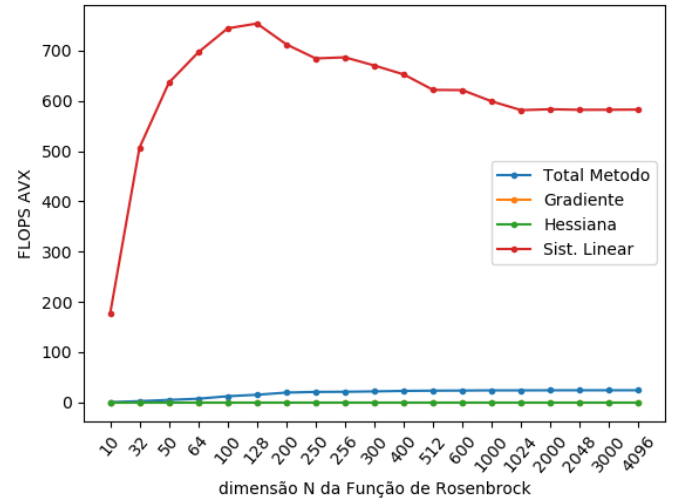


Figura 12

Newton Inexato não Otimizado

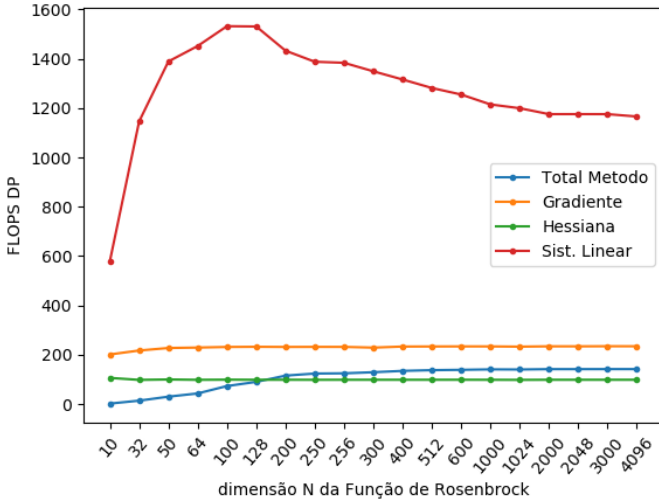


Figura 13

Newton Inexato Otimizado

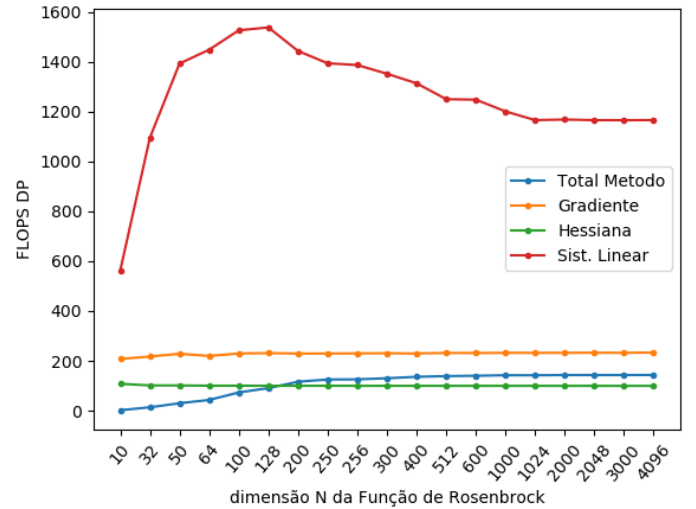


Figura 14

Newton Inexato não Otimizado

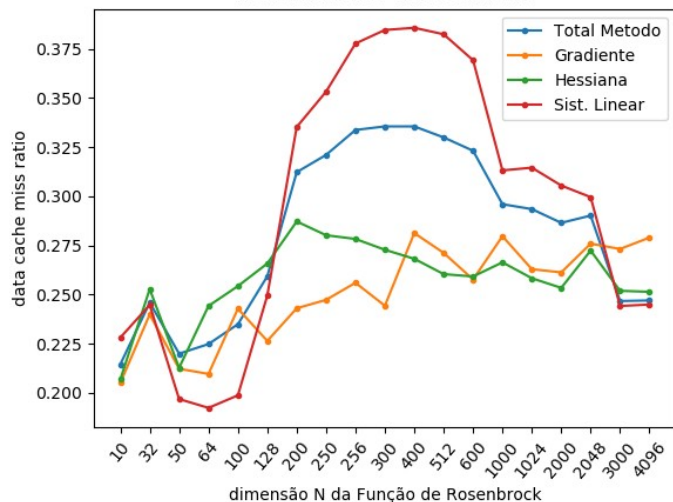


Figura 15

Newton Inexato Otimizado

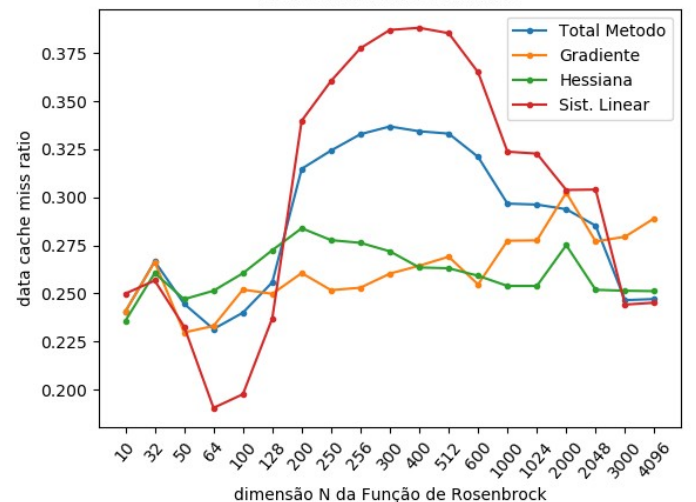


Figura 16

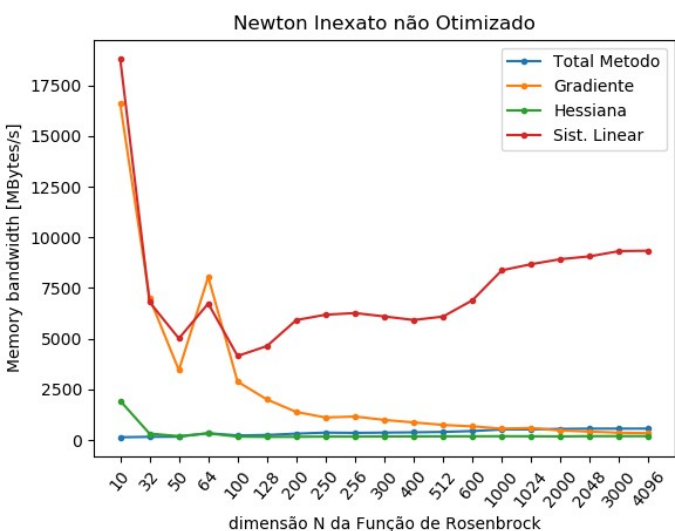


Figura 17

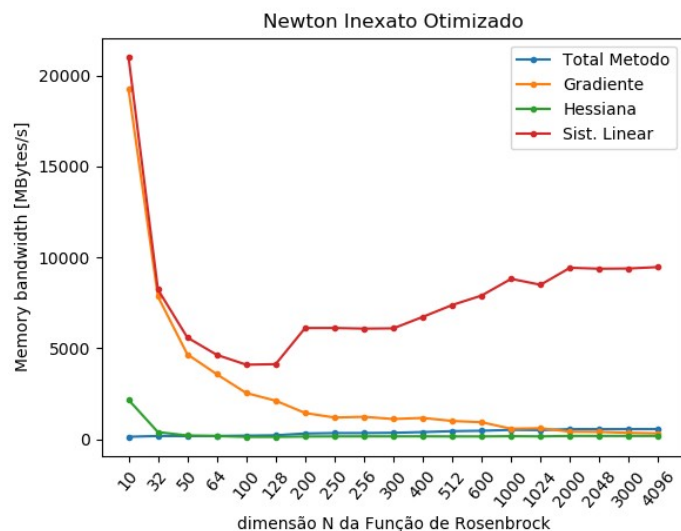


Figura 18

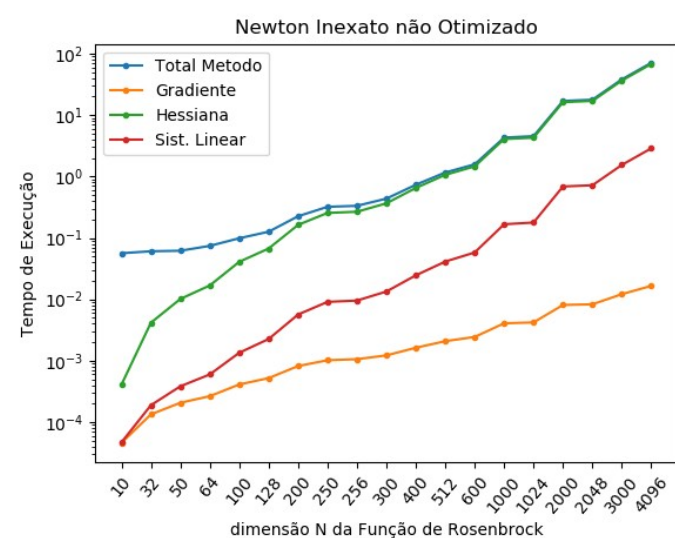


Figura 19

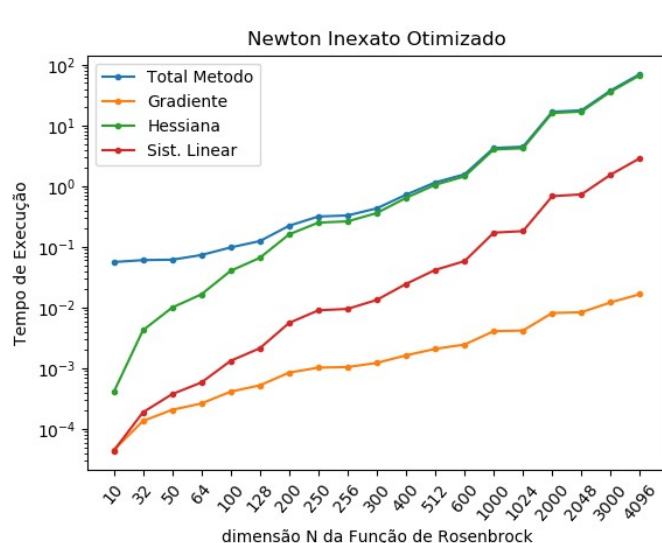


Figura 20

Os tempos de execução se mantiveram parecidos pois não houveram mudanças necessárias para esse requisito.

O numero de operações de ponto flutuante se manteve constante pois na região onde ocorre o maior numero de operações tem muita dependência de dados tornando inviável mudanças.

A taxa de miss ratio também se manteve próxima devido ao mesmo motivo anterior.

A banda de memória manteve maior constância, e em boa parte dos testes se manteve mais alta com a otimização, graças ao formato das matrizes alocadas.

Algumas variações se deram pela troca das funções de criação de derivação das funções.