

Se Wan de Campos 7991722
Yoshio Mori 6432393

1 Parte C

Arquivos	Coluna	Linhas
a1.dat	0.001275s	0.001432s
a2.dat	0.010070s	0.010404s
a3.dat	0.033716s	0.035451s
a4.dat	0.076737s	0.083558s
a5.dat	0.152260s	0.165075s
a6.dat	0.263685s	0.287239s
a7.dat	0.421572s	0.449313s

Essa tabela foi gerada a partir de programas implementados na linguagem c, a coluna rotulada Coluna mostra o tempo gasto para processar a matriz armazenada nos arquivos correspondente com algoritmos orientado a coluna, assim como a coluna das Linhas mostra o tempo gasto para algoritmos orientado a linhas. O programa foi executado numa máquina de processador i5-3470 com 6 MB de cache. Em todos os processos, o algoritmo orientado a colunas foi mais rápido. Algo curioso uma vez que como as matrizes em c são orientadas à linhas esperava que os algoritmos orientados a linha fossem mais rápidos. Para rodar o program c_ep1 é necessário ter todos os arquivos a1.dat a2.dat ... a7.dat no mesmo diretório do programa.

2 Parte em Fortran e f2c

Fortran

Orientado a Linha

m1.dat - real 0m0.050s user 0m0.040s sys 0m0.004s
m2.dat - real 0m0.093s user 0m0.080s sys 0m0.008s
m3.dat - real 0m0.225s user 0m0.220s sys 0m0.004s
m4.dat - real 0m0.431s user 0m0.424s sys 0m0.004s
m5.dat - real 0m0.708s user 0m0.704s sys 0m0.004s
m6.dat - real 0m1.125s user 0m1.112s sys 0m0.000s
m7.dat - real 0m1.691s user 0m1.664s sys 0m0.012s

Orientado a Coluna

m1.dat - real 0m0.030s user 0m0.028s sys 0m0.000s
m2.dat - real 0m0.089s user 0m0.084s sys 0m0.004s
m3.dat - real 0m0.219s user 0m0.220s sys 0m0.000s
m4.dat - real 0m0.415s user 0m0.412s sys 0m0.000s
m5.dat - real 0m0.704s user 0m0.692s sys 0m0.008s
m6.dat - real 0m1.120s user 0m1.100s sys 0m0.008s
m7.dat - real 0m1.687s user 0m1.660s sys 0m0.012s

F2C

Orientado a Linha

m1.dat - real 0m0.030s user 0m0.020s sys 0m0.008s

```

m2.dat - real 0m0.096s user 0m0.092s sys 0m0.004s
m3.dat - real 0m0.223s user 0m0.216s sys 0m0.004s
m4.dat - real 0m0.448s user 0m0.444s sys 0m0.000s
m5.dat - real 0m0.733s user 0m0.728s sys 0m0.004s
m6.dat - real 0m1.125s user 0m1.112s sys 0m0.008s
m7.dat - real 0m1.683s user 0m1.664s sys 0m0.012s

```

Orientada a Coluna

```

m1.dat - real 0m0.028s user 0m0.020s sys 0m0.004s
m2.dat - real 0m0.099s user 0m0.096s sys 0m0.004s
m3.dat - real 0m0.220s user 0m0.212s sys 0m0.004s
m4.dat - real 0m0.436s user 0m0.420s sys 0m0.012s
m5.dat - real 0m0.760s user 0m0.712s sys 0m0.036s
m6.dat - real 0m1.162s user 0m1.152s sys 0m0.004s
m7.dat - real 0m1.704s user 0m1.688s sys 0m0.008s

```

A saída do arquivo convertido de Fortran para C, quando usado o tipo de variável "double precision", estava dando errado, algo como: 1. 2. 1. 3. ... Saía o resultado arredondado e sem as casas decimais. Quando o tipo da variável foi mudada para "real", a saída deu certo. Então o arquivo em f_ep1.f esta com o tipo de variável double precision e o arquivo f_ep1.c esta com o tipo de variável real.

O programa em f_ep1 foi mais rápido usando o algoritmo orientado a colunas do que orientado a linhas o que já era esperado. E o programa f_ep1 apresentou uma redução na velocidade de processamento comparado com o programa c_ep1.

Para se rodar o programa f_ep1 é necessário passar um arquivo contendo as informações da matriz como argumento e o tipo de orientação. Por exemplo: ./f_ep1 m1.dat linha