

Lab 1 CCI22 ITA 2018

Prof. Luiz Mirisola

March 1, 2018

Instruções (válidas para os outros labs!)

- DC: Questões de implementação devem ser tratadas como laboratórios. Questões teóricas devem ser tratadas como questões de prova em casa.
- Faça um relatório sucinto: basta responder às questões, mostrando códigos, resultados e conclusões.
- Se precisar apresentar vários resultados, tabelas são melhores do que números jogados em texto corrido.
- Gráficos muito porcos perderão ponto. Pequenos erros não serão penalizados, não é artigo científico nem relatório de física, mas ninguém é obrigado a adivinhar que está certo para corrigir, nem a supor que está certo porque parece certo mas não está escrito, nem a dar 3 zooms no pdf pra conseguir ler uma letra (ou não conseguir...).
- Submeta na Atividade do Tidia:
 - pdf do seu relatório.
 - Inclua arquivos de código escritos (.cpp e .m) na sua entrega, *inclusive* de questões *sem* correção automática.
 - Código não precisa ser repetido completamente no texto do relatório. Faça copy-paste de porções de código apenas se achar pertinente para ilustrar alguma explicação ou clarificar alguma resposta.
- Evite usar o Octave, pois ele pode alterar as conclusões em relação aos tempos de execução. Não há problema em usar o octave se a questão não envolve medida de tempo. Não sei se o Octave inclui a ferramenta de testes do matlab.
- As questões marcadas com AUTOTEST serão corrigidas automaticamente:
 - Complete as funções com os cabeçalhos fornecidos
 - submeter diretamente os arquivos .m correspondentes a estas funções, fora de zips, diretorios, etc. O propósito é que o auto-corretor enxergue os .m corretos no diretório raiz da sua submissão, sem precisar abrir zips, procurar em diretórios, etc.
 - * se usar funções auxiliares, elas também devem ser submetidas no mesmo diretório.
 - Os testes foram fornecidos, e podem ser usados e modificados. Note:
 - * o campo 'nota' é apenas #testes que passam / #testes existentes. Ou seja, podem não corresponder aos pesos das nossas questões.
 - * O professor pode executar outros testes se necessário. O que vale é resolver o problema, não passar no teste.
 - * O professor mantém a sua própria cópia dos testes e *não* aplicará testes presentes no próprio diretório do aluno. Portanto, qualquer modificação (inclusive erros e bugs) que o aluno introduza nos testes não será aplicada pelo corretor automático.
 - * Se o aluno encontrar erros no código do curso (testes ou '.p'), haverá bônus sempre que o professor precisar corrigir o código do curso.
 - Para as funções com correção automática, pode ser fornecida uma solução pronta na forma de arquivos com extensão '.p'. Estes arquivos podem ser usados da mesma forma que arquivos .m, mas o código está criptografado e portanto ilegível. Obviamente, não pode usar a solução pronta para a sua própria solução.
- O lab1 vale 11 pontos, explicado no ex. 5

1 Mais um ϵ !

Uma definição de ϵ de máquina (para o número 1) é de que se trata da distância entre 1 e o próximo número representável no sistema de ponto flutuante em questão. O seguinte algoritmo determina o ϵ de máquina:

```
1 ep = 1
2 while 1 + ep/2 > 1 do
3     ep = ep/2
4 endwhile
```

Implemente um script para determinar o ϵ de máquina do MATLAB, e um programa para determinar o ϵ em C. Em C, para double e float. Em matlab para double e single. Compare o valor encontrado com o valor retornado pela função matlab `eps` e `eps('single')`. (dica: procure o comando `single` para criar variáveis de precisão single em matlab). Explique os resultados encontrados.

Resumo: Deve encontrar 6 valores de ϵ : para double e float (single) (encontrar valores para 2 tipos), usando um programa em C, seu próprio programa MATLAB, e o comando `eps` do matlab (encontrar valores por 3 meios). A tabela abaixo serve de modelo para a sua resposta. A seguir compare e comente sobre os resultados. Entenda e explique: porque são iguais ou diferentes?

	MATLAB eps	programa MATLAB	programa C
single			não existe
float	não existe	não existe	
double			

Não testarei o seu programa, apenas lerei o comentário.

1.1 Dica: como saber o tamanho dos tipos em C

Não há problema para esse trabalho se o seu compilador define os tamanhos dos tipos de forma diferente. Mas isso afeta o resultado, portanto se encontrar um resultado estranho, descubra as definições do seu compilador, e inclua estes fatos na sua explicação. A própria definição da linguagem C admite alternativas para os tamanhos dos tipos de ponto flutuante. Mas, para ter certeza que a diferença vem do compilador e não é erro do seu programa, pode-se checar o tamanho dos tipos com o seguinte código C:

```
printf("size float %ld\n", sizeof(float));
// some compilers have 8-byte float
printf("size double %ld\n", sizeof(double));
// some compilers have 16-byte double
printf("size long double %ld\n", sizeof(long double));
```

No gcc do linux, os 3 tipos acima tem tamanhos diferentes!

A definição dos tipos do matlab está aqui: https://www.mathworks.com/help/matlab/matlab_prog/floating-point-numbers.html

2 Operador ou loop?

Três formas de se implementar o produto interno entre dois vetores coluna de mesma dimensão x e y no MATLAB são:

1. Utilizando o operador de multiplicação de matrizes: `pi = x' * y;`
2. Utilizando a função dot: `pi = dot(x,y);`
3. Utilizando um laço de repetição explícito (comando `for`): $\sum_{i=1}^n x(i) * y(i)$ onde n é a dimensão de cada vetor.

Utilizando as funções `tic` e `toc`, pode-se comparar os tempos de execução. Portanto, meça os tempos de 1000 operações de produto interno para cada maneira armazene os resultados nos vetores `t1,t2,t3`, respectivamente. Utilize vetores x e y com 10^6 elementos cada. Meça o tempo apenas para realizar a operação, sem considerar o tempo para alocar e inicializar os vetores. Então, plote os vetores `t1,t2,t3` (dica: utilize a função `plot`). Comente os gráficos obtidos.

Resultados de medida de tempo são tipicamente ruidosos. Para reduzir o efeito do ruído, pode-se realizar uma média aritmética com as medidas. (função `mean` do MATLAB calcula a média). Para verificar quantas

medidas são necessárias até a média convergir, plote um gráfico onde nas abcissas, está o numero da medida i, de 1 a 1000, e no eixo y, a media de todas as medidas da medida 1 até a medida i.

Por exemplo, o ponto $x = 100$ representa a media das medidas 1,2,3,...,100. Deverá verificar a convergência da média.

- Inclua no relatório imagens dos gráficos
- submeta o código para gerar os dados e gráficos.
- Com base nos resultados obtidos, diga qual método de cálculo de produto interno é preferível em termos de tempo de execução. Note que os resultados deste experimento indicam uma maneira de programar em MATLAB que é consideravelmente mais eficiente.
- Atenção: Ao terminar a soma dos termos com o comando `for`, temos o resultado final. Portanto, para a comparação ser justa, é preciso realizar a atribuição do resultado final. Portanto, meça o tempo para executar `pi = x' * y`; e não apenas `x' * y`;

2.1 Medindo tempo de execução

Imprimir na tela gasta tempo. Certifique-se ao comparar os tempos, que nenhuma mensagem é impressa na tela. Claro, fora do trecho de código cujo tempo de execução é medido, não há problema em imprimir mensagens.

- Todo comando matlab terminado com `';` não imprime o seu resultado na tela. Sem o `';`, o resultado será impresso.

Exemplo uso das funções `tic` e `toc` para medir tempo, e mostrando o que quer dizer 'sem considerar o tempo de alocação de variáveis':

```
1 M = zeros(10,20); // creates and allocates M as a 10x20 matrix with all elements = 0.
2 tic; % starts clock
3 % note: M was allocated before the tic.
4 M = my_calculation(); % this is what you want to measure. DO NOT PRINT MESSAGES HERE
5 wtime = toc; % toc returns the time elapsed since the previous tic
6 sprintf('_my_calculation_took_%f_seconds_to_run.\n', wtime)
7 % last line has no ';' as it should print the result
```

3 exp() com precisão (AUTOTEST)

Use a definição da função exponencial em forma de série para calcular $\exp(x)$ exato até o 500º (quingentésimo) termo da série.

$$\exp(x) = \sum_{k=0}^{\infty} \frac{x^k}{k!}$$

- A sua função também deve retornar o erro relativo, comparando com o valor retornado pela função `exp()` do matlab (sim, pode chamar o `exp()` do matlab dentro da própria função para comparar os valores. Não estamos medindo o tempo de execução agora).
 - a comparacao eh feita usando a fórmula vista em aula, com o resultado calculado no denominador e não o valor real de `exp()`: `relerr = abs(resultado - exp(x))/resultado;`
- O seu trabalho deve implementar uma função com cabeçalho já fornecido.

Dicas:

- Não é possível calcular fatorial de 500 explicitamente, com a função `factorial`. Porque? Como o método de horn evitou calcular todas as potências de x explicitamente? Da mesma forma, o fatorial não deve ser calculado explicitamente.
- Foi fornecida a função `expseries(x,n_termos)` que resolve o problema, calculando `n_termos` termos para e^x . Vocês não tem acesso ao arquivo de código .m desta solução, apenas ao arquivo .p, que pode ser chamado como um .m, mas o conteúdo é criptografado. Claro que **não** pode usar a solução completa dada na sua própria solução!

- Para verificar os resultados, o comando `format long` faz o matlab imprimir valores com 15 casas de precisão. O comando não afeta os cálculos, nem a correção automática, apenas como o MATLAB imprime na tela. Também existe o comando `sprintf('%0.15f', value)` com sintaxe similar ao equivalente C.
- Para mais detalhes sobre um comando, use o comando `help` ou `doc`. O último abre uma janela de help. Por exemplo: `doc format` ou `help sprintf`

4 Somas não causam erro, certo?

4.1 [15pt] Calcular um somatório.

Se não computamos nada além de somas e subtrações, podem haver erros numéricos? Escreva um programa para calcular:

$$D = 10000 - \sum_{k=1}^n x$$

para:

- $n = 100.000$ (cem mil) e $x = 0.1$
- $n = 80.000$ (oitenta mil) e $x = 0.125$

Compare o resultado obtido nos dois casos com o resultado esperado e comente o *porque* deste comportamento.

- Dica: deve calcular o somatório com um loop explícito (`for`). Se, ao invés disso, utilizar $n * x$, terá o resultado exato.
- Dica: para ajudar a pensar no “porque” com um problema similar, mas de tamanho menor, compare:
 - a) $0.1 + 0.1 + 0.1 - 0.3$
 - b) $0.125 + 0.125 + 0.125 + 0.125 - 0.5$
 - Use o programa `flut.c` para verificar a representação em ponto flutuante dos números envolvidos nestes 2 exemplos menores.

4.2 [5 pt] Aplicação: comparação de double

Baseado neste exercício, explique porque não podemos comparar variáveis de ponto flutuante com o operador `==`, mesmo que as contas pareçam simples. E o que devemos fazer, então, se quisermos comparar valores `double`, por exemplo, para decidir se dois valores de ponto flutuante são iguais?

5 [2pt + 1pt] Ordenação (AUTOTEST)

Implementações de algoritmos $O(n^2)$ serão desconsideradas. O tamanho da entrada é grande justamente para impedir que algoritmos $O(n^2)$ terminem em tempo razoável.

Implemente um algoritmo de ordenação $O(n \log n)$ para ordenar as linhas de uma matriz. Não deve trocar valores entre colunas. Apenas troque a ordem das linhas. A comparação entre as duas linhas deve comparar os elementos da primeira coluna. Caso sejam iguais, compare os elementos da segunda coluna; etc. Caso duas linhas sejam iguais em todas as colunas, não importa a ordem.

Avaliação:

- 1.5 pt com ordenação realizada para matrizes grandes (o teste com matrizes pequenas não vale nada, é apenas uma primeira validação).
- 0.5 pt com ordenação realizada incluindo matrizes grandes inteiras (há um teste separado para este caso).
- 1 pt pelo tempo de execução. Agruparei os alunos pelo tempo de execução e para os grupos correspondendo aproximadamente aos 50% melhores da turma será atribuída uma nota para cada grupo entre 0 e 1.
- Portanto a lista vale 11 pontos (e não será arredondada para 10 depois)

1. Similar à função `sortrows`, que não pode ser utilizada. Da mesma forma, não pode ser utilizada a função `sort`, ou equivalente. Não é necessário ordenar numeros complexos, retornar vetor de índices, ou quaisquer outras funcionalidades da função `sortrows`.
2. O cabeçalho também foi dado e deve ser completado.

3. O proposito do exercicio, especialmente do bônus por velocidade, é treinar programação MATLAB com um algoritmo similar aos de CES10 e CES11. Ordenar as linhas obrigará os alunos a utilizarem a sintaxe do matlab para operações com matrizes, e a otimizar a alocação de variaveis e as atribuições! Usem a sintaxe do matlab para evitar loops desnecessários a todo custo!
4. A única complicação das matrizes inteiras: provavelmente haverão elementos repetidos na mesma coluna. Assim, a comparação das linhas não pode olhar apenas para a primeira coluna. Recomendo que implementem uma função auxiliar para comparar linhas, algo como `comparaLinhas(i,j)` que compara a linha `i` e a linha `j`. O código da função auxiliar pode estar no mesmo `mysortrows.m` (nesse caso deve estar *depois* da função `mysortrows`) ou pode estar em um arquivo `.m` separado.