

Aberto: terça, 28 jun 2022, 13:00

Vencimento: quinta, 14 jul 2022, 23:59

Exercício sobre Sistemas Lineares

(Este exercício não vale nota)

Objetivo

O objetivo deste exercício é implementar uma biblioteca para a resolução de sistemas lineares.

Especificação

As funções a serem implementadas estão declaradas no arquivo **Metodos.h**.

- Método da Eliminação de Gauss, com **pivoteamento parcial (EGP)**
- Método de Gauss Seidel (**GS**):
 - **Valor inicial:** o vetor **x** deve ter valor inicial nulo
 - **Critério de parada:** máximo de iterações (vide arquivo **Metodos.h**) e a norma máxima do erro absoluto aproximado em **x**:

$$\|\vec{X}^{(k+1)} - \vec{X}^{(k)}\|_{\infty} = \max_{1 \leq i \leq n} |x_i^{(k+1)} - x_i^{(k)}|$$

- Método de Refinamento (**REF**):
 - **Critério de Parada:** máximo de iterações (vide arquivo **Metodos.h**) e a norma euclidiana do resíduo:

$$\|\vec{r}\|_2 = \sqrt{\sum_{1 \leq i \leq n} r_i^2} \text{ onde } \vec{r} = \vec{b} - A\vec{x}$$

Uso da biblioteca

Todas funções da biblioteca devem "**falhar graciosamente**". Em caso de erro (alocação, geração de valores **inf** ou **nan**, etc) , o retorno das funções deve ser negativo, e uma mensagem de erro deve ser impressa na saída de erros (**stderr**), sem no entanto encerrar a execução do programa. **CUIDADO:** Nenhuma função deve exibir resultados na saída padrão, apenas seu programa principal.

Testes a serem efetuados

O objetivo é comparar o comportamento dos métodos em diferentes sistemas lineares. Para tanto, faça o programa **labSisLin** que deve:

1. Gerar SL do tipo **diagDominante** com os seguintes tamanhos: 10, 30, 50, 128, 256, 512, 1000, 2000, 3000
Para cada tamanho, resolver o SL pelos métodos **EGP**, **GS** e **REF**.
2. Para cada caso acima, medir o tempo de cada método até a solução utilizando a função **timestamp()** (vide **Funções utilitárias**). No caso dos métodos iterativos, obter também o número de iterações ao finalizar o método.
3. Ao aplicar o método **REF**, deve-se usar como valor inicial do vetor **x** a solução final obtida no método **GS**. Deve-se também exibir a **norma do resíduo** para este valor inicial de **x** e para o valor final calculado pelo método **REF**.
4. Gerar tabela com as colunas **n**, **t_egg**, **t_gs**, **it_gs**, **normaResiduo_gs**, **t_ref**, **it_ref**, **normaResiduo_ref**



5. Os passos **1 a 4** devem ser feitos com duas formas de alocação para as matrizes de coeficientes (via função **alocaSisLin()**):
 - **pontPont** (vetor de **n** ponteiros, cada um apontando para outro vetor com **n** elementos)
 - **pontVet** (vetor com **n** ponteiros, cada um apontando para um trecho de um único vetor de **n*n** elementos, como visto na disciplina CI1001.
6. Repita os passos de **1 a 5** com matrizes do tipo **hilbert**.
7. Procure usar uma forma eficiente de cálculo das normas usadas nos critérios de parada dos métodos **GS** e **REF**.

A partir dos resultados obtidos (tempos, número de iterações, tamanho do SL):

1. É possível estimar as condições em que o método **EGP** é mais vantajoso que o método **GS**?
2. Qual forma de alocação de matriz é mais eficiente: **pontPont** ou **pontVet** ?
DICA: procure usar o gnuplot para gerar gráficos a partir das tabelas de tempo.

Funções utilitárias

São fornecidas para uso livre neste exercício um conjunto de funções utilitárias , disponíveis via Gitlab C3SL no link <https://gitlab.c3sl.ufpr.br/nicolui/ci1164-utils> ou via linha de comando: **git clone git@gitlab.c3sl.ufpr.br:nicolui/ci1164-utils.git**

O módulo **sislin.*** contém funções para definir um sistema linear. O módulo **utils.*** contém a definição da função **timestamp()** e outras funções que serão úteis no decorrer da disciplina CI1164.

 [labSisLin.c](#)

28 junho 2022, 10:37

 [Makefile](#)

27 junho 2022, 17:42

 [Metodos.c](#)

28 junho 2022, 14:06

 [Metodos.h](#)

28 junho 2022, 10:37