PROJETO COMPUTACIONAL #1 – MS993/MT404 – 2S2016 – IMECC/UNICAMP

Matemática Aplicada

Atividade	Temas	Disponibilização	Entrega
Projeto #1 (**P1**) **Leitura.** David S. Watkins, Fundamentals of Matrix Computations, New Jersey: John Wiley & Sons (3 ed., 2010).	Fatorações LU, Cholesky, Matrizes esparsas, The Reverse Cuthill-McKee algorithm, “fill” or “fill-in” (i.e, “fenômeno de enchimento” ou “fenômeno de preenchimento”, Discretização de um modelo diferencial de convecção-difusão (em regime estacionário), Pivoteamento, Estrutura de dados.	2/Set	16/Set

OBJETIVO DO PROJETO. Uso de métodos diretos para a resolução de sistemas lineares Ax=b. Será considerado o desempenho das Fatorações LU (matrizes densas) e Cholesky (matrizes definidas-positivas e simétricas, tipicamente esparsas). Em particular, será dado ênfase na parte computacional para o fato de que é possível melhorar o desempenho dos cálculos envolvidos nas fatorações LU e Cholesky por meio da aplicação de um esquema de reordenamento adequado dos elementos das matrizes envolvidas. Além disso, os conceitos de aritimética em precisão finita, número de condição e análise de sensibilidade de sistemas lineares também serão abordados no Projeto 1.

SPARSE POSITIVE DEFINITE SYSTEMS. If one compares a sparse matrix A with its Cholesky factor L ($A=L\,L^T$), one normally finds that R has many more nonzero entries than the upper half of A does. The “new” nonzero entries are called fill or fill-in. How much fill one gets depends on how the equations are ordered. References “*Gilbert, John R., Cleve Moler, and Robert Schreiber, Sparse Matrices in MATLAB: Design and Implementation, SIAM Journal on Matrix Analysis, 1992*”. **Syntax.** “r = symrcm(S)” (Sparse reverse Cuthill-McKee ordering). **Description.** r = symrcm(S) returns the symmetric reverse Cuthill-McKee ordering of S. This is a permutation r such that S(r,r) tends to have its nonzero elements closer to the diagonal. This is a good preordering for LU or Cholesky factorization of matrices that come from long, skinny problems. The ordering works for both symmetric and nonsymmetric S. For a real, symmetric sparse matrix, S, the eigenvalues of S(r,r) are the same as those of S, but eig(S(r,r)) probably takes less time to compute than eig(S).

Additional comments. The Reverse Cuthill-McKee algorithm aims to minimise the bandwidth of the matrix. A Reverse Cuthill–McKee ordering can be computed with the command I =symrcm(A). Apply this permutation, check the resulting sparsity pattern of the permuted matrix and of the LU-factorisation of this matrix. What is the percentage of the nonzeros in the L and U factors ? Moreover, The minimum degree ordering aims to minimise the fill in. This ordering can be computed with the command I = realmmd(A) (see also Matlab’s command amd and symamd). Apply this permutation, check the resulting sparsity pattern of the permuted matrix and of the LU-factorisation of this matrix. Some natural questions: what is the percentage of the nonzeros in the L and U factors? If you have used different permutation strategies. Which one worked best for our problem?

- **FAZER** Exercise 1.6.7, Exercise 1.6.8 e Exercise 1.6.9.

PROGRAMMING GAUSSIAN ELIMINATION. Reliable Fortran routines that perform Gaussian elimination with partial pivoting can be obtained from the public-domain software packages LINPACK and LAPACK. There is also a C wrapper for LAPACK called CLAPACK.

- **LER** apenas o enunciado das atividades seguintes Exercise 1.8.12, Exercise 1.8.15, Exercise 1.8.16 e Exercise 1.8.17.
- **FAZER** Exercise 1.8.20

ON COMPENTING METHODS. A variant of Gaussian elimination that is taught in some elementary linear algebra textbooks is Gauss-Jordan elimination. In this variant the augmented matrix [A | b] is converted to the form [I | x] by elementary row operations. At the kth step the pivot is used to create zeros in column k both above and below the main diagonal. The disadvantage of this method is that it is more costly than the variants that we have discussed.

- **FAZER** Exercise 1.8.28.

SPARSE GAUSSIAN ELIMINATION. If we wish to solve Ax = b, where A is large and sparse, we are well advised to exploit the structure of A.

- **FAZER** Exercise 1.9.2, Exercise 1.9.3 e Exercise 1.9.4.
- **FAZER** No Exercise 1.9.2, Make an LU-decomposition of A using the commands thresh = 0; LU = lu(A,thresh); The parameter thresh determines a pivoting threshold (help lu). Here we only study reordering of the equations to minimise fill in, we suppress reordering of the equation to enhance numerical stability by putting thresh = 0 (in other words: MATLAB offers a threshold pivoting option that can be used instead of partial pivoting. Use of threshold pivoting sometimes reduces fill. Type help lu for details.); Check the nonzero pattern in the LU-decomposition of A. Determine the percentage of nonzero elements.
- **FAZER** Qual é o significado vetor c = [0.2 5] no Exercise 1.9.3 em termos do modelo de convecção-difusão ? Antecipa-se que nesse exercício não estamos resolvendo um problema do tipo Ax=b (estamos explorando alguns recursos e técnicas a partir da matriz A associada a uma discretização de um modelo de convecção-difusão (ver, e.g., Y. Saad, 2nd, Chapter 2); comente um pouco sobre esse modelo !) Experimente c = [1 1] e c = [1 10000] e repita o mesmo exercício. Depois disso, responda qual é a estrutura da matriz em cada um dos três casos para c.

  Obs.: Veja o link http://www.mathematik.uni-dortmund.de/~kuzmin/cfdintro/cfd.html para um tutorial elementar no formato de slides sobre alguns métodos clássicos para aproximação de modelos de Convection-diffusion e modelos de transporte relacionados, incluindo uma lista de livros básicos.
- **FAZER**. Primeiramente, para essa atividade considere o livro "*Timothy A. Davis, Direct methods for sparse linear systems (Fundamentals of algorithms Series), PA, SIAM (2006)*". Este livro fornece uma biblioteca decente de funções especiais para "matrizes esparsas" Ax=b. Este livro considera apenas métodos diretos para Ax=b. O autor T. A. Davis apresenta cuidadosamente ao longo do livro do código que ele desenvolveu, a saber, "bibliografia CSparse", em vários níveis de detalhes. Vale a pena notar que a teoria por trás do algoritmo Cuthill-McKee, "fill" ou "fill-in" (i.e., "Fenômeno de enchimento" ou "Fenômeno de preenchimento" está ligada à teoria dos grafos, passando por algoritmos e estruturas de dados, indo da matemática à ciência da computação e engenharia. Para concretude, você pode focar sua leitura nos capítulos "*3 Solving triangular systems*" e "*4 Cholesky factorization*". Observa-se, ainda, que o texto pode ser um tanto não usual para um bom entendimento dos algoritmos lá reportados. O autor optou por uma exposição mais no estilo "ciência da computação e engenharia". Recomenada-se também a leitura do Prefácio. O autor também faz um "paralelo" entre sua biblioteca "Csparse" e o "Matlab", facilitando um pouco mais a leitura. Com efeito, CSparse é escrita em C, o que garante melhor uso em termos de complexidade de memória, maior eficiência e boa portabilidade. Essa tarefa também ilustra a relevância científica da matemática aplicada no escopo dos cursos MS993/MT404 Métodos computacionais em álgebra linear.

- FAZER. Faça um breve resumo de no máximo uma página sobre "*Rook pivoting*". Como sugestão de leitura como um ponto de partida, consultar os originais:

  (**a**) G. Poole and L. Neal, A geometric analysis of Gaussian elimination, I, Linear Algebra Appl., 149 (1991), pp. 249-272. (**b**) L. Neal and G. Poole, A geometric analysis of Gaussian elimination, II, Linear Algebra Appl., 173 (1992), pp. 239-264. (**c**) G. Poole and L. Neal, The rook's pivoting strategy, J. Comput. Appl. Math., 123 (2000), pp. 353-369. e também alguns trabalhos mais recentes (e.g.) (d) Xiao-Wen Chang, Some features of Gaussian elimination with rook pivoting, BIT 42(1) (2002) 66-83.

- Consulte também link https://www.mathworks.com/matlabcentral/fileexchange/37721-rook-pivoting