**Hours:** Tue 9:40-12:30 BMB3

**Instructor:** Murat Manguoglu

- Office: A407
- Phone: 210 5551
- Email: manguoglu@ceng.metu.edu.tr
- Web Page: http://www.ceng.metu.edu.tr/~manguoglu
- Office Hours: Thursday 14:00-15:00

**Content:**

Today most real life applications require processing large amounts of data (i.e. "**Big Data**"). The parts of the large data is, however, usually not independent of eachother. Hence, a natural way to represent them is a sparse matrix (i.e. graph). The methods that process these data usually boils down to three fundamental (or related) problems to solve: linear systems, eigenvalue problems or singular value decompositions, whether it is reducing the road noise inside a car’s body or applying the PageRank algorithm when you do a web search. In this course we will get familiar with sparse matrix computations with an emphasis on algorithms for solving large sparse linear systems and eigenvalue problems, and performance implication of related data structures and computer architectures.

**Outline:** (subject to change)

- Review of basic linear algebra, sparsity and implications of sparsity on performance.
- Origin of sparse problems: Finite difference, finite elements and others
- Sparse matrix structures and storage schemes
- Sparse Gaussian elimination and its graph representation
- Various reordering schemes to improve the sparse Gaussian elimination
• Relaxation methods (Jacobi, Gauss-Seidel, SOR,)

• Projection methods and the one dimensional cases Minimum Residual and Steepest Descent

• Krylov Subspace Methods: Arnoldis method and GMRES

• Krylov Subspace Methods: Lanczos tridiagonalization and CG, BCG, BiCGStab and variants

• Preconditioning: left and right preconditioning, and preconditioned Krylov subspace methods

• Preconditioning: the cost and effectiveness of different preconditioners

• Eigenvalue problems: Power method, projection techniques

• Eigenvalue problems: Subspace iterations, Krylov methods, Arnoldis and Lanczos iterations

• Other sparse algorithms: sparse SVD and Linear Least Squares (if time permits)

Prerequisites:

Basic numerical linear algebra, dense matrix algorithms, a programming language (C, C++, ...) and computer architectures.

Midterms:

There will be 2 midterms. One page handwritten A4 size cheatsheet is allowed.

Homeworks:

There will be 4 programming homework assignments. Delayed submissions are accepted with a penalty of $-5 \times d^2$ where $d$ is the number of days in which the solution is submitted late.
**Attendance and Participation:**

Attendance and participation in the class and online discussions and office hours are encouraged. I will check attendance in class (randomly) and in my office hours and I may use it in your favor when assigning letter grades.

**Grading:**

- Homeworks (×4) 60%
- Midterms (×2) 40%
- Total 100%

**References:**

- *Numerical Methods for Large Eigenvalue Problems*, Yousef Saad, 2011
- *Direct Methods for Sparse Linear Systems*, Timothy A. Davis, 2006

**Course Policy and academic honesty:**

All homeworks are expected to be individual work. If you use a source (online or offline), you are expected to cite it. Violation of these general principles will be handled based on the university regulations and may result in disciplinary action.

There will be no make up examinations for this class and late homework assignments will not be accepted. An exception is when there is an official medical or family emergency, in this case you should contact the instructor as soon as possible.