

You may work on the projects by yourself or with a group of 2 (maximum allowed). More is expected from a two-student paper: it is a good idea to team up with someone else if the project you have in mind requires a big effort (e.g. substantial coding). In this situation, the report should clarify who did what in a separate section [call it e.g. “Tasks”]

This is a preliminary version!

1 Project themes.

You essentially have three or four options.

1. You can do a project on a topic related to sparse matrix computations in your own research. [for example : test / development of a solver for fluid flow problems, or sparse matrix methods for information retrieval]
2. Write a theoretical paper or a survey paper related to sparse matrix computation [for example a survey ‘on the use of sparse matrix techniques in genomics’].
3. You can select among a few topics suggested in the next section. These are just a few examples.

2 A few suggestions

The following is a list of possible projects. This list is by no means exhaustive. The ordering is not significant. The list may be updated a couple of times during the semester.

Project#1 Explore the performance of sparse matrix operations on various platforms. By this I mean various sparse vector operations such as sparse dot products, sparse saxpy, etc. as well as matvecs, triangular solves, etc. In matlab we saw a peculiar behavior once. It would be interesting to explore this issue in detail from a standard language (fortran or C).

Project#2 Sparse matrix computations on GPUs – either an overview of what people have done or an implementation / comparison of your own of some techniques [in the latter case you will need to have access to a GPU - e.g. at MSI] – see, e.g., [7].

Project#3 Comparison of reordering techniques for direct solvers. Nested Dissection ordering, Minimal Degree algorithm, etc. You can obtain the ‘meshpart’ matlab toolbox from

<http://www.cerfacs.fr/algor/Softs/MESHPART/>

Then test various ordering strategies available (see the parameter ‘method’ in ndpart). Compare with any other reorderings you can get from matlab (those available from matlab

are are colamd and symrcm). If you prefer you can do this project in C or Fortran and download a few reordering techniques available.

Project#4 Generate linear systems from IFISS CDF code and integrate different solvers into ifiss. Ifiss is a matlab code for finite element discretization and solution of various fluid dynamics problems. The code can be downloaded from here:

<http://www.maths.manchester.ac.uk/~djs/ifiss/>

Project#5 Explore the use of sparse matrices in signal processing, e.g., the problem of Dictionary learning, see for example [9].

Project#6 There are very interesting applications of sparse matrix techniques in computer graphics. One such application (or a class of applications actually) is *Poisson Image Editing*. This consists of modifying images in a seamless way. Modifications may include importing a piece from another image ('cloning'), illumination changes, background color modifications, seamless tiling, etc.. This leads to solving sparse linear systems, specifically after discretizing a Poisson equation. The main reference is [13].

Project#7 Present an overview of methods used in image segmentation. A few good starting points: [17, 12].

Project#8 Sparse tensors and applications. Tensors have become very important due to their use in image/video processing. More recently, *Sparse* tensors have appeared as an important tool in text processing for example. Here are a couple of references: [16, 1, 15].

Project#9 Explore sparsity on Deep Neural networks.. It is known that the types of nonlinear functions used in neural networks tend to introduce sparsity. It is not clear how/whether this has been effectively exploited by practitioners so far. This could be an exciting study on a subject of great current interest. A few recent references are [8, 10, 11, 5].

Project#10 Sparse matrix techniques are also finding their way in bio-informatics. The Non-Negative Matrix Factorization (NMF) becomes an attractive alternative to the SVD. A sparse version of the technique seems to be getting very popular in bio-informatics. The starting paper here is [4] but there are others.

Project#11 Along the same lines an important recent area of research centers around "matrix completion", i.e., the problem of finding missing entries in a given matrix. The best illustration of this is in recommender system: you are given a matrix of ratings of (e.g.) movies by individuals and you would like to guess some of those entries which are not filled up for the purpose of recommending movies (or books to by as is done in Amazon).. There are many references here - a good starting point for recommender systems is [14]. To study the problem in-depth see for example [6, 2, 3]. An idea that is key in this topic is the judicious use of norms to control sparsity and rank.

References

- [1] M. Baskaran, B. Meister, N. Vasilache, and R. Lethin. Efficient and scalable computations with sparse tensors. In *2012 IEEE Conference on High Performance Extreme Computing*, pages 1–6, Sep. 2012.
- [2] Emmanuel Cands and Benjamin Recht. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 9:717–772, 2009. 10.1007/s10208-009-9045-5.
- [3] Emmanuel J. Cands and Terence Tao. The power of convex relaxation: Near-optimal matrix completion. *IEEE Transactions on Information Theory*, 56(5):2053–2080, 2010.
- [4] Yuan Gao and George Church. Improving molecular cancer class discovery through sparse non-negative matrix factorization. *Bioinformatics*, 21:3970–3975, 2005.
- [5] Benjamin Graham. Spatially-sparse convolutional neural networks. *CoRR*, abs/1409.6070, 2014. available from arXiv:<https://arxiv.org/abs/1409.6070v1>.
- [6] R.H. Keshavan, A. Montanari, and Sewoong Oh. Matrix completion from a few entries. *IEEE Transactions on Information Theory*, 56(6):2980–2998, june 2010.
- [7] Ruipeng Li and Yousef Saad. Gpu-accelerated preconditioned iterative linear solvers. *The Journal of Supercomputing*, 63(2):443–466, 2013.
- [8] Baoyuan Liu, Min Wang, Hassan Foroosh, Marshall Tappen, and Marianna Pensky. Sparse convolutional neural networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.
- [9] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online learning for matrix factorization and sparse coding. *JMLR*, 11:1960, 2010.
- [10] A. Parashar, M. Rhu, A. Mukkara, A. Puglielli, R. Venkatesan, B. Khailany, J. Emer, S. W. Keckler, and W. J. Dally. Scnn: An accelerator for compressed-sparse convolutional neural networks. In *2017 ACM/IEEE 44th Annual International Symposium on Computer Architecture (ISCA)*, pages 27–40, June 2017.
- [11] JongSoo Park, Sheng R. Li, Wei Wen, Ping Tak Peter Tang, Hai Li, Yiran Chen, and Pradeep Dubey. Faster cnns with direct sparse convolutions and guided pruning. In *ICLR 2017*, 2017. Available from arXiv: <https://arxiv.org/abs/1608.01409>.
- [12] Bo Peng, Lei Zhang, and David Zhang. A survey of graph theoretical approaches to image segmentation. *Pattern Recognition*, 46(3):1020 – 1038, 2013.
- [13] P. Pérez, P. Gangnet, and A. Blake. Poisson image editing. *ACM Trans. Graphics*, 22(3):313–318, 2003.
- [14] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Application of dimensionality reduction in recommender systems – a case study. In *ACM WebKDD, Web Mining for E-Commerce Workshop*, 2000.

- [15] S. Smith and G. Karypis. A medium-grained algorithm for sparse tensor factorization. In *2016 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, pages 902–911, May 2016.
- [16] S. Smith, N. Ravindran, N. D. Sidiropoulos, and G. Karypis. Splatt: Efficient and parallel sparse tensor-matrix multiplication. In *2015 IEEE International Parallel and Distributed Processing Symposium*, pages 61–70, May 2015.
- [17] Nida M. Zaitoun and Musbah J. Aqel. Survey on image segmentation techniques. *Procedia Computer Science*, 65:797 – 806, 2015. International Conference on Communications, management, and Information technology (ICCMIT’2015).