• CSCI 8314 • Spring 2014 •

SPARSE MATRIX COMPUTATIONS

Class time	:	MW 9:45 – 11:00 AM
Room	:	STSS 117
Instructor	:	Yousef Saad
URL	:	www-users.cselabs.umn.edu/classes/Spring-2014/csci8314/

Before we begin....

➤ Lecture notes will be posted on the class web-site – usually before the lecture.

➤ Review them and try to get some understanding (help: text) if possible before class.

➤ Lecture note packets are grouped by topics rather than by lecture.

➤ In the notes the symbol indicates suggested exercises – often [not always] done in class.

➤ I will often post the matlab diaries used for the demos (if any). [with the help of matlab's `diary` utility].

➤ Do not hesitate to contact me for any questions...

CSCI 8314: SPARSE MATRIX COMPUTATIONS

GENERAL INTRODUCTION

- **General introduction - a little history**
- **Motivation**
- **Resources**
- **What will this course cover**

What this course is about

➤ Solving linear systems and (to a lesser extent) eigenvalue problems with matrices that are sparse.

➤ Sparse matrices : matrices with mostly zero entries [details later]

➤ Many applications of sparse matrices...

➤ ... and we are seing more with new applications everywhere

A brief history

Sparse matrices have been identified as important early on – origins of terminology is quite old. Gauss defined the first method for such systems in 1823 (now the Gauss-Seidel iteration). Varga used explicitly the term 'sparse' in his 1962 book on iterative methods.

- Special techniques used for sparse problems coming from Partial Differential Equations
- One has to wait until to the 1960s to see the birth of the general technology available today
- Graphs introduced as tools for Sparse Gaussian matrices in 1961 [Seymour Parter]

- Early work on reordering for banded systems, envelope methods
- Various reordering techniques for general sparse matrices introduced.
- Minimal degree ordering [Markowitz - 1957] ...
- ... later used in Harwell MA28 code [Duff] - released in 1977.
- Tinney-Walker Minimal degree ordering for power systems [1967]
- Nested Dissection [A. George, 1973]
- SPARSPAK [commercial code, Univ. Waterloo]
- Elimination trees, symbolic factorization, ...

History: development of iterative methods

- 1950s up to 1970s : focus on "relaxation" methods
- Development of 'modern' iterative methods took off in the mid-70s. but...
- ... The main ingredients were in place earlier [late 40s, early 50s: Lanczos; Arnoldi ; Hestenes (a local!) and Stiefel; ....]
- The next big advance was the push of 'preconditioning': in effect a way of combining iterative and (approximate) direct methods – [Meijerink and Van der Vorst, 1977]

History: eigenvalue problems

- Another parallel branch was followed in sparse techniques for large eigenvalue problems.
- A big problem in 1950s and 1960s : flutter of airplane wings.. This leads to a large (sparse) eigenvalue problem
- Overlap between methods for linear systems and eigenvalue problems [Lanczos, Arnoldi]

Resources

[See the "links" page in the course web-site]

➤ Matrix market

`http://math.nist.gov/MatrixMarket/`

➤ Florida collection

`http://www.cise.ufl.edu/research/sparse/matrices/`

➤ SPARSKIT, etc.

`http://www.cs.umn.edu/~saad/software`

Resources – continued

Books: on sparse direct methods.

➤ Book by Tim Davis [SIAM, 2006] see class web-site.

➤ Best reference [old, out-of print, but still the best]:

• Alan George and Joseph W-H Liu, Computer Solution of Large Sparse Positive Definite Systems, Prentice-Hall, 1981. Englewood Cliffs, NJ.

➤ Of interest mostly for references:

• I. S. Duff and A. M. Erisman and J. K. Reid, Direct Methods for Sparse Matrices, Clarendon press, Oxford, 1986.

➤ References to articles will be posted on a regular basis.

Overall plan for the class

➤ We will begin by sparse matrices in general, their origin, storage, manipulation, etc..

➤ We will then spend about 1/3 of the class on sparse direct methods

➤ .. and about 1/3 on iterative methods

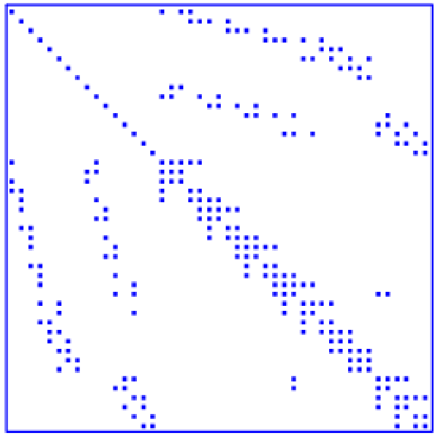
➤ ... rest on eigenvalue problems and applications..

➤ Plan is not rigid!

SPARSE MATRICES

- **See Chap. 3 of text**
- **See the "links" page on the class web-site**
- **See also the various sparse matrix sites.**
- **Introduction to sparse matrices**
- **Sparse matrices in matlab –**

What are sparse matrices?



Pattern of a small sparse matrix

➤ Vague definition: matrix with few nonzero entries

➤ For all practical purposes: an $m \times n$ matrix is sparse if it has $O(\min(m, n))$ nonzero entries.

➤ This means roughly a constant number of nonzero entries per row and column -

➤ This definition excludes a large class of matrices that have $O(\log(n))$ nonzero entries per row.

➤ Other definitions use a slow growth of nonzero entries with respect to n or m.

"..matrices that allow special techniques to take advantage of the large number of zero elements." (J. Wilkinson)

A few applications which lead to sparse matrices:

Structural Engineering, Computational Fluid Dynamics, Reservoir simulation, Electrical Networks, optimization, Google Page rank, information retrieval (LSI), circuit similation, device simulation, ...

Goal of Sparse Matrix Techniques

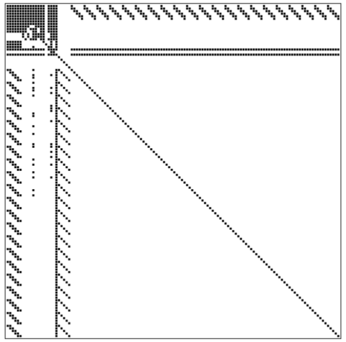
➤ To perform standard matrix computations economically i.e., without storing the zeros of the matrix.

Example: To add two square dense matrices of size n requires $O(n^2)$ operations. To add two sparse matrices A and B requires $O(nnz(A) + nnz(B))$ where $nnz(X)$ = number of nonzero elements of a matrix X.

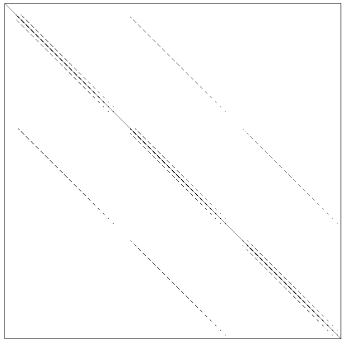
➤ For typical Finite Element /Finite difference matrices, number of nonzero elements is $O(n)$.

Remark: A^{-1} is usually dense, but L and U in the LU factorization may be reasonably sparse (if a good technique is used).

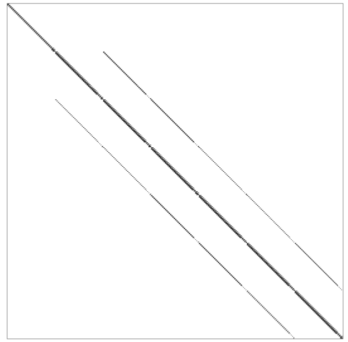
Nonzero patterns of a few sparse matrices



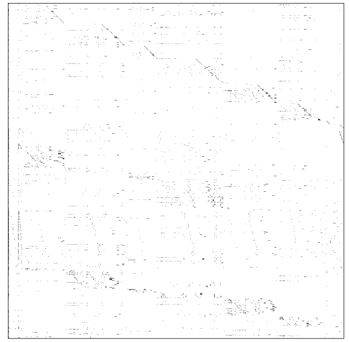
ARC130: Unsymmetric matrix from laser problem. a.r.curtis, oct 1974



SHERMAN5: fully implicit black oil simulator 16 by 23 by 3 grid, 3 unk



PORES3: Unsymmetric MATRIX FROM PORES



BP_1000: UNSYMMETRIC BASIS FROM LP PROBLEM BP

Types of sparse matrices

➤ Two types of matrices: **structured** (e.g. Sherman5) and **unstructured** (e.g. BP_1000)

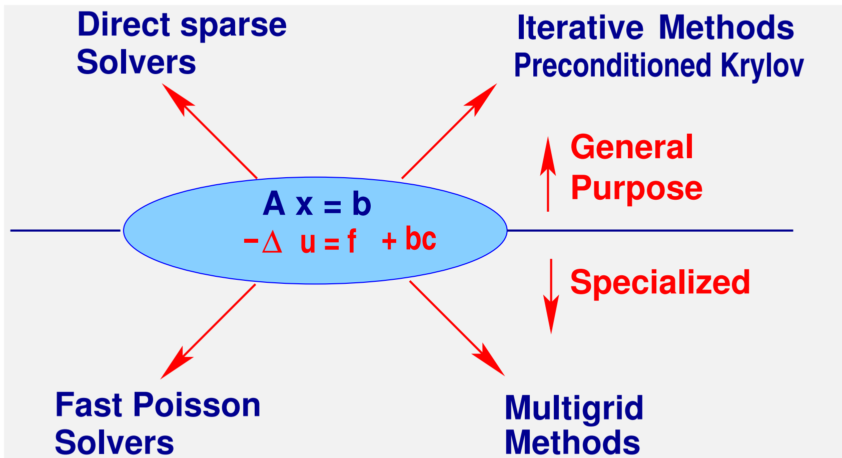
➤ The matrices PORES3 and SHERMAN5 are from Oil Reservoir Simulation. Often: 3 unknowns per mesh point (Oil , Water saturations, Pressure). Structured matrices.

➤ 40 years ago reservoir simulators used rectangular grids.

➤ Modern simulators: Finer, more complex physics ➤ harder and larger systems. Also: unstructured matrices

➤ A naive but representative challenge problem: $100 \times 100 \times 100$ grid + about 10 unknowns per grid point ➤ $N \approx 10^7$, and $nnz \approx 7 \times 10^8$.

Solving sparse linear systems: existing methods



Two types of methods for general systems:

➤ Direct methods : based on sparse Gaussian eimination, sparse Cholesky,..

➤ Iterative methods: compute a sequence of iterates which converge to the solution - preconditioned Krylov methods..

Remark: These two classes of methods have always been in competition.

➤ 40 years ago solving a system with $n = 10,000$ was a challenge

➤ Now you can solve this in a fraction of a second on a laptop.

➤ Sparse direct methods made huge gains in efficiency. As a result they are very competitive for 2-D problems.

➤ 3-D problems lead to more challenging systems [inherent to the underlying graph]

Difficulty:

- No robust 'black-box' iterative solvers.
- At issue: Robustness in conflict with efficiency.

➤ Iterative methods are starting to use some of the tools of direct solvers to gain 'robustness'

Consensus:

1. Direct solvers are often preferred for two-dimensional problems (robust and not too expensive).

2. Direct methods loose ground to iterative techniques for three-dimensional problems, and problems with a large degree of freedom per grid point,

Sparse matrices in matlab

➤ Matlab supports sparse matrices to some extent.

➤ Can define sparse objects by conversion

```
A = sparse(X) ;  X = full(A)
```

➤ Can show pattern

```
spy(X)
```

➤ Define the analogues of `ones`, `eye`:

```
speye(n,m),   spones(pattern)
```

➤ A few reorderings functions provided..

```
symrcm, symamd, colamd, colperm
```

➤ Random sparse matrix generator:

`sprand(S)` or `sprand(m,n, density)`

➤ To read if you are interested in sparse matrices in matlab: • John R. Gilbert, Cleve, Moler and Robert Schreiber, *"Sparse Matrices in MATLAB: Design and Implementation"*, SIAM Journal on Matrix Analysis and Applications, volume 13, number 1, pages 333–356 (1992).

Graph Representations of Sparse Matrices

➤ Graph theory is a fundamental tool in sparse matrix techniques.

DEFINITION. A graph G is defined as a pair of sets $G = (V, E)$ with $E \subset G \times G$. So G represents a binary relation. The graph is undirected if the binary relation is reflexive. It is directed otherwise. V is the vertex set and E is the edge set.

$\boxed{\textit{Example:}}$ Given the numbers 5, 3, 9, 15, 16, show the two graphs representing the relations

R1: Either $x < y$ or y divides x.

R2: x and y are congruent modulo 3. [mod(x,3) = mod(y,3)]

➤ Graph $G = (V, E)$ of an $n \times n$ matrix A defined by

➤ Vertices $V = \{1, 2, ..., N\}$.

➤ Edges $E = \{(i, j) | a_{ij} \neq 0\}$.

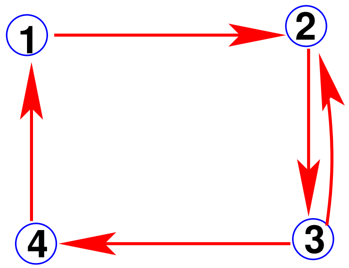
➤ Often self-loops (i, i) are not represented [because they are always there]

➤ Graph is undirected if the matrix has a symmetric structure:

$$a_{ij} \neq 0 \quad \text{iff} \quad a_{ji} \neq 0.$$

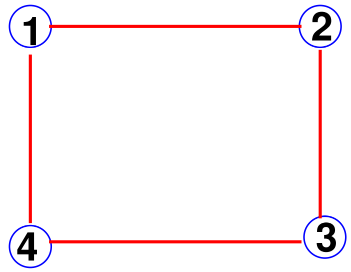
Example: (directed graph)

$$\begin{bmatrix} & \star & & \\ & & \star & \\ & \star & & \star \\ \star & & & \end{bmatrix}$$



Example: (undirected graph)

$$\begin{bmatrix} & \star & & \star \\ \star & & \star & \\ & \star & & \star \\ \star & & \star & \end{bmatrix}$$



Example: Adjacency graph of:

$$A = \begin{bmatrix} \star & \star & & & \star & \\ \star & \star & \star & & & \star \\ & \star & \star & & & \\ & & & \star & \star & \\ \star & & & \star & \star & \star \\ & \star & & & \star & \star \end{bmatrix}.$$

Example: What is the graph of a tridiagonal matrix? Of a dense matrix?

➤ We will see much on graphs and their use for sparse matrices later.