

Sequential sparse matrix–vector multiplication (PSC §4.1)

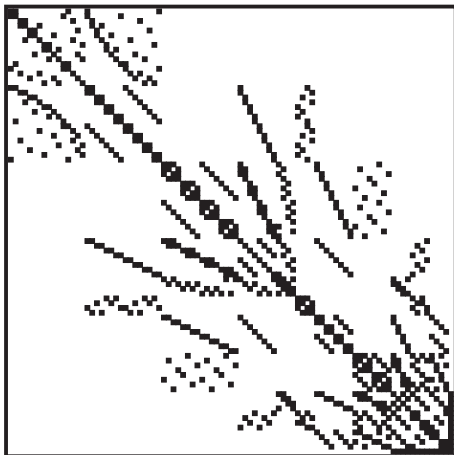


Sparse and dense matrices

- ▶ **Sparse matrices** are sparsely populated by nonzero elements.
- ▶ **Dense matrices** have mostly nonzeros.
- ▶ Sparse matrix computations **save time**: operations with zeros can be skipped or simplified; only the nonzeros must be handled.
- ▶ Sparse matrix computations also **save memory**: only the nonzero elements need to be stored (together with their location).



Sparse matrix cage6



$n = 93$, $nz = 785$ nonzeros, $c = 8.4$ nonzeros per row,
 $d = 9.1\%$ density

Sequential sparse matrix-vector multiplication



Matrix statistics

- ▶ Number of nonzeros is

$$nz = nz(A) = |\{a_{ij} : 0 \leq i, j < n \wedge a_{ij} \neq 0\}|.$$

- ▶ Average number of nonzeros per row or column is

$$c = c(A) = \frac{nz(A)}{n}.$$

- ▶ Density is

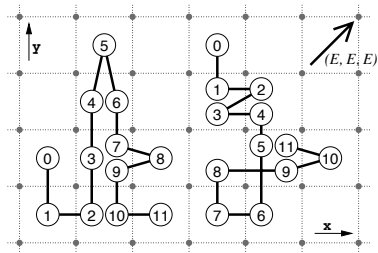
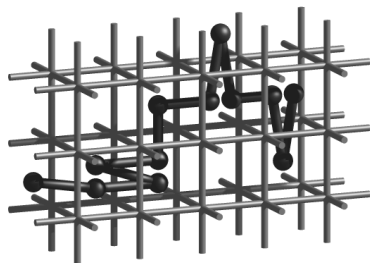
$$d = d(A) = \frac{nz(A)}{n^2}.$$

- ▶ Matrix is **sparse** if $nz(A) \ll n^2$, or $c(A) \ll n$, or $d(A) \ll 1$.



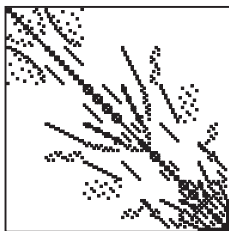
Application: cage model for DNA electrophoresis

(A. van Heukelum, G. T. Barkema, R. H. Bisseling,
Journal of Computational Physics **180** (2002) pp. 313–326.)



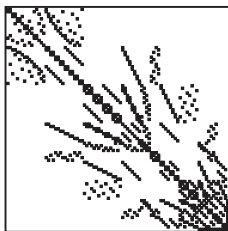
- ▶ 3D cubic lattice models a gel
- ▶ DNA polymer reptates (moves like a snake):
kinks and end points move
- ▶ DNA sequencing machines: electric field E .
Aim: study drift velocity $v(E)$.

Transition matrix for cage model



- ▶ Matrix element a_{ij} is the **probability** that a polymer in state j moves to a state i . Hence, $0 \leq a_{ij} \leq 1$.
- ▶ Polymer has 6 **monomers** for cage6. We can move only one monomer at a time. Hence, each state has only a few connected states and the matrix is sparse.

Sparsity structure of cage6



- ▶ Each move can be reversed, hence $a_{ij} \neq 0 \iff a_{ji} \neq 0$, i.e., the matrix is **structurally symmetric**.
- ▶ Move against the electric field has different probability than move with the field. Hence $a_{ij} \neq a_{ji}$, so that the matrix is **unsymmetric**.

Power method

- ▶ Let \mathbf{x} be the vector of state frequencies: component x_i represents the relative **frequency** of state i , with $0 \leq x_i \leq 1$ and $\sum_i x_i = 1$.
- ▶ The **power method** computes $A\mathbf{x}, A^2\mathbf{x}, A^3\mathbf{x}, \dots$, until convergence.
- ▶ Final component x_i represents the frequency of state i in the **steady-state** situation, where $A\mathbf{x} = \mathbf{x}$.
- ▶ Main operation: multiplication of sparse matrix A and dense vector \mathbf{x} .



Sparse matrix–vector multiplication

- ▶ Let A be a sparse $n \times n$ matrix and \mathbf{v} a dense input vector of length n .
- ▶ We consider the problem of computing the dense output vector \mathbf{u} ,

$$\mathbf{u} := A\mathbf{v}.$$

- ▶ The components of \mathbf{u} are

$$u_i = \sum_{j=0}^{n-1} a_{ij}v_j, \quad \text{for } 0 \leq i < n.$$

Sparse matrix–vector multiplication algorithm

input: A : sparse $n \times n$ matrix,
 \mathbf{v} : dense vector of length n .
output: \mathbf{u} : dense vector of length n , $\mathbf{u} = A\mathbf{v}$.

for $i := 0$ **to** $n - 1$ **do**

$u_i := 0$;

for all $(i, j) : 0 \leq i, j < n \wedge a_{ij} \neq 0$ **do**

$u_i := u_i + a_{ij}v_j$;

The sparsity of A is expressed by the test $a_{ij} \neq 0$. Such a test is never executed in practice, and instead a **sparse data structure** is used.



Iterative solution methods

- ▶ Sparse matrix–vector multiplication is the main computation step in iterative solution methods for linear systems or eigensystems.
- ▶ Iterative methods start with an **initial guess** \mathbf{x}^0 and then successively improve the solution by finding **better approximations** \mathbf{x}^k , $k = 1, 2, \dots$, until the error is tolerable.
- ▶ Examples:
 - ▶ Linear systems $A\mathbf{x} = \mathbf{b}$, solved by the conjugate gradient (CG) method or MINRES, GMRES, QMR, BiCG, Bi-CGSTAB, IDR, SOR, FOM, ...
 - ▶ Eigensystems $A\mathbf{x} = \lambda\mathbf{x}$ solved by the Lanczos method, Jacobi–Davidson, ...

Web searching: which page ranks first?

A screenshot of a Google search results page for the query "parallel scientific computation". The browser address bar shows the URL "http://www.google.co.uk/#hl=en&client=psy-ab&q=parallel+scientific+computation". The search bar contains the text "parallel scientific computation". Below the search bar, it indicates "About 53,900,000 results (0.32 seconds)".

The left sidebar shows navigation options: Web, Images, Maps, Videos, News, Shopping, and More. Below these is a "Search near..." section with a location input field and a "Set" button. At the bottom of the sidebar are "The web Pages from the UK" and "More search tools".

The main search results are as follows:

- Scholarly articles for parallel scientific computation**
 - [Parallel scientific computation](#) - Bisseling - Cited by 214
 - [Parallel scientific computation](#) - Hillis - Cited by 30
 - ...: [A parallel workstation for scientific computation](#) - Becker - Cited by 764
- Parallel Scientific Computation - Oxford University Press**
ukcatalogue.oup.com/product/9780198529392.do
This is the first text explaining how to use the bulk synchronous **parallel** (BSP) model and the freely available BSPlib communication library in **parallel** algorithm ...
- 18.337 Parallel Scientific Computing**
web.mit.edu/18.337/
8 Feb 1996 - 18.337 **Parallel Scientific Computing**. Spring, 1996. Tuesday/Thursday 3:00-4:30 / 1-390. Prof. Alan Edelman. This is an old page. Go to the ...
- Parallel Scientific Computing**
www.sandia.gov/~bahendr/parallel.html
With a number of different collaborators, I have devised new **parallel** algorithms for a variety of problems in **scientific computing**. These include many-body ...
- Parallel Scientific Computing and Optimization**
www.springer.com/mathematics/applications/.../978-0-387-09706-0
Parallel Scientific Computing and Optimization introduces new developments in the construction, analysis, and implementation of parallel computing algorithms.
- Parallel Scientific Computation: A Structured Approach using BSP ...**
books.google.co.uk > ... > Distributed Systems & Computing
Based on the author's extensive development, this is the first text explaining how to use BSPlib, the bulk synchronous **parallel** library, which is freely available for ...
- Parallel Scientific Computing in C++ and MPI: A ... - Amazon.co.uk**
www.amazon.co.uk > ... > Infinite Series
Parallel Scientific Computing in C++ and MPI: A Seamless Approach to Parallel Algorithms and their Implementation: Amazon.co.uk: George Em Karniadakis, ...

Sequential sparse matrix-vector multiplication



The link matrix A

- ▶ Given n web pages with links between them. We can define the sparse $n \times n$ **link matrix** A by

$$a_{ij} = \begin{cases} 1 & \text{if there is a link from page } j \text{ to page } i \\ 0 & \text{otherwise.} \end{cases}$$

- ▶ Let $\mathbf{e} = (1, 1, \dots, 1)^T$, representing an initial uniform importance (rank) of all web pages. Then

$$(\mathbf{A}\mathbf{e})_i = \sum_j a_{ij}e_j = \sum_j a_{ij}$$

is the **total number of links pointing to page i** .

- ▶ The vector $\mathbf{A}\mathbf{e}$ represents the importance of the pages; $\mathbf{A}^2\mathbf{e}$ takes the importance of the pointing pages into account as well; and so on.



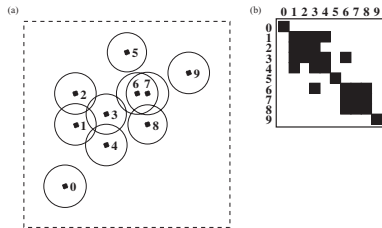
The Google matrix

- ▶ A **web surfer** chooses each of the **outgoing N_j links** from page j with equal probability. Define the $n \times n$ diagonal matrix D with $d_{jj} = 1/N_j$.
- ▶ Let α be the probability that a surfer follows an outlink of the current page. Typically $\alpha = 0.85$. The surfer **jumps to a random page** with probability $1 - \alpha$.
- ▶ The **Google matrix** is defined by (Brin and Page 1998)

$$G = \alpha AD + (1 - \alpha)\mathbf{e}\mathbf{e}^T/n.$$

- ▶ The PageRank of a set of web pages is obtained by repeated multiplication by G , involving sparse matrix–vector multiplication by A , and some vector operations.

Insight into other applications



- ▶ (a) A 2D molecular dynamics domain of size 1.0×1.0 with 10 particles.
- ▶ The **cut-off radius** for the interaction between particles is $r_c = 0.2$. The circles shown have radius $r_c/2 = 0.1$.
- ▶ (b) The corresponding sparse 10×10 force matrix F . If the circles of radius $r_c/2$ of particles i and j overlap, then i and j interact, so that nonzero forces f_{ij} and f_{ji} appear in F .

Sequential sparse matrix-vector multiplication

Summary

- ▶ Sparse matrices are the **rule, rather than the exception**. In many applications, variables are connected to only a few others, leading to sparse matrices.
- ▶ Sparse matrices occur in various application areas:
 - ▶ transition matrices in Markov models;
 - ▶ finite-element matrices in engineering;
 - ▶ linear programming matrices in optimisation;
 - ▶ weblink matrices in Google PageRank computation.
- ▶ We often express computation costs in the matrix size n and the average number of nonzeros per row c .
- ▶ Sparse matrix–vector multiplication is important for **iterative solvers**. It can also capture **other applications** such as molecular dynamics.
- ▶ The sequential computation is simple, but its parallelisation is a big challenge.