Computational Microelectronics
Lecture 11 Sparse Matrix

Sung-Min Hong (smhong@gist.ac.kr)
Semiconductor Device Simulation Laboratory
School of Electrical Engineering and Computer Science
Gwangju Institute of Science and Technology

Summary of diffusion

Now we can

- Solve a set of equations.
  - Transient simulation
  - Boundary conditions (Dirichlet or Neumann)
  - Multiple unknown variables
  - Nonlinearity (Drift)

- Essentially,
  - It is $Ax = b$.

Sparse matrix

A simple example of `Ax = b`

- (Steady-state) diffusion equation with N points.
  - Let us assume that $C(0) = 0$ and $C(L) = 1$.
  - For example, when $N = 6$,

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 & 0 \\ 0 & 0 & 1 & -2 & 1 & 0 \\ 0 & 0 & 0 & 1 & -2 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C(x_0) \\ C(x_1) \\ C(x_2) \\ C(x_3) \\ C(x_4) \\ C(x_5) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

Number of nonzero entries

- In A, how many nonzero entries are there?
  - Three entries in a row, $3 \times (N-2)$
  - Boundary rows, 2
  - Total number is $3N-4$.
  - Therefore, the number of zero-valued elements divided by the total number of elements (sparsity) is

$$1 - \frac{3N-4}{N^2} \rightarrow 1 - \frac{3}{N}$$

  - Of course, for a large N, the sparsity approaches 1. (Almost completely empty!)

HW#10

- Due: AM08:00, October 11
- Problem#1
  - Solve the Laplace equation with a dense matrix. Draw the elapsed time as a function of N.

Sparse matrix

- Storage
  - For a dense matrix, how many double-precision numbers are needed? We need N^2 entries.
  - For a sparse matrix, how many double-precision numbers are needed? It is the number of nonzero entries, N_{nz}.
  - Of course, it is not sufficient. We must specify (row, column) pairs. For example, we may need $2N_{nz}$ integer numbers.
- Compressed sparse row (CSR) format

```
V = [ 10 20 30 40 50 60 70 80 ]
COL_INDEX = [ 0 1 1 3 2 3 4 5 ]
ROW_INDEX = [ 0 2 4 7 8 ]
```

Example taken from Wikipedia

Sparse matrix solver

- Special code for solving `Ax = b`, when `A` is sparse
- Direct solvers
  - Representative direct sparse matrix solvers:
  - UMFPACK
  - PARDISO
  - There are many other solvers, too.
- Iterative solvers
  - Conjugate gradient (CG)
  - Generalized minimum residual method (GMRES)

MATLAB

- MATLAB
  - Just say that `A` is a sparse matrix by (instead of `zeros`)

```
A = sparse(N,N)
```

  - You may not modify other lines.

Python

- Python
  - Scipy package

```
import math
import numpy as np
from scipy import sparse
import matplotlib.pyplot as plt

N = 10

#A = np.zeros( (N+1, N+1) )
A = sparse.lil_matrix( (N+1, N+1) )
b = np.zeros( (N+1, 1  ) )

for ii in range(1,N):
    A[ii,ii+1] = 1.0
    A[ii,ii  ] = -2.0
    A[ii,ii-1] = 1.0

A[0,0] = 1.0
A[N,N] = 1.0
b[N] = 1.0

A = A.tocsr()

#update = np.linalg.solve(A, b)
update = sparse.linalg.spsolve(A,b)
update = update[:,None]
```

C++

- There are many options.

HW#10

- Problem#2
  - Solve the Laplace equation with a sparse matrix. Draw the elapsed time as a function of N.

Thank you!