

Mini-Matlab Lesson 2: Solving matrix equations

Solving matrix equations is the most important problem you will ever do

Contents

- [Why are matrix solvers important?](#)
- [Bottleneck](#)
- [Solving matrix equations in Matlab](#)
- [How fast can we solve?](#)
- [A plot of speed vs. size](#)
- [The two most important commands you will use in Matlab\(?\)](#)

Why are matrix solvers important?

Over and over again, we have a pattern like this:

Nonlinear problem on continuous interval

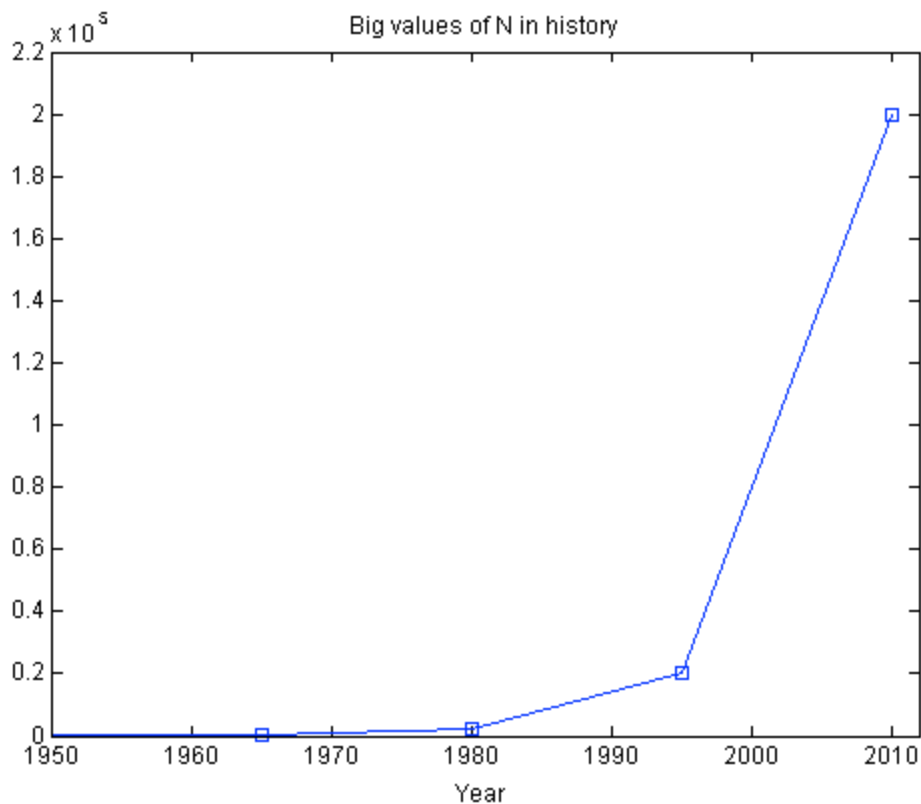
```
linearize and iterate  
----->  
discretize
```

Linear algebra

Bottleneck

Most numerical problems boil down to solving $Ax = b$ where A is $N \times N$ matrix.

```
plot([1950 1965 1980 1995 2010], [20 200 2000 20000 200000], 'bs-');  
title('Big values of N in history');  
xlabel('Year');  
axis([1950 2012 0 2.2e5]);  
  
% Most algorithms are  $O(N^3)$ . If you could find an algorithm for solving  
% matrix equations in  $O(N^2)$ , you would be famous and rich.
```



Solving matrix equations in Matlab

```

% Create random 5x5 matrix
A = rand(5)

% Create vector of ones
b = ones(5,1)

% Solve Ax = b (backslash)
x = A\b

% Check the error
A*x - b

% We can also find A's inverse
invA = inv(A)

% And then solve the matrix equation this way
x = invA*b

% But *DO NOT* use inv(A) unless you need the inverse directly. It's slower
% and less accurate than using the backslash (which uses Gaussian
% elimination)

```

A =

```

0.6403    0.1057    0.0521    0.8604    0.5134

```

```
0.4170 0.1420 0.9312 0.9344 0.1776
0.2060 0.1665 0.7287 0.9844 0.3986
0.9479 0.6210 0.7378 0.8589 0.1339
0.0821 0.5737 0.0634 0.7856 0.0309
```

b =

```
1
1
1
1
1
```

x =

```
0.1172
-0.0496
-0.2314
1.3312
-0.3958
```

ans =

```
1.0e-15 *
-0.1110
0
0
0
0
```

invA =

```
0.8282 0.7860 -1.5584 0.5288 -0.4674
-1.2179 -3.1323 2.3869 1.6601 0.2536
-1.0402 -0.3899 1.3581 0.6491 -0.8084
0.8960 2.4072 -1.8646 -1.3683 1.2609
-0.2305 -4.3303 4.4396 1.2260 -1.5005
```

x =

```
0.1172
-0.0496
-0.2314
1.3312
-0.3958
```

How fast can we solve?

```

N = 100; A = randn(N); b = ones(N,1); tic, A\b; toc
N = 1000; A = randn(N); b = ones(N,1); tic, A\b; toc
N = 2000; A = randn(N); b = ones(N,1); tic, A\b; toc
N = 4000; A = randn(N); b = ones(N,1); tic, A\b; toc

```

```

Elapsed time is 0.001797 seconds.
Elapsed time is 0.046079 seconds.
Elapsed time is 0.290088 seconds.
Elapsed time is 1.693769 seconds.

```

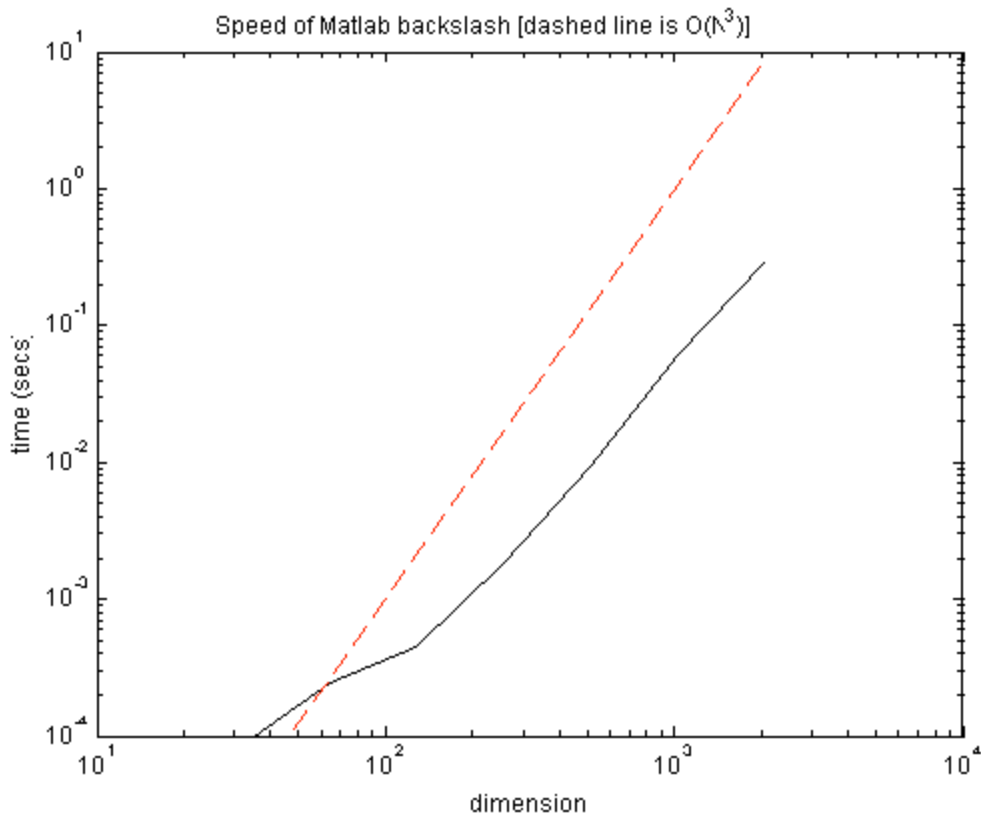
A plot of speed vs. size

```

for i = 1:11, N=2^i;
    A = randn(N);
    tic, A\ones(N,1); t=toc;           % tic and toc are used to time
    NN(i)=N; tt(i)=t;
end

loglog(NN,tt,'.-')                  % Make a loglog plot
hold on
plot(NN, (.001*NN).^3, '--r')       % Make N^3 line
ylim([1e-4 10])
xlabel('dimension')
ylabel('time (secs)')
title('Speed of Matlab backslash [dashed line is O(N^3)]')

```



The two most important commands you will use in Matlab(?)

```
% Get inline help
help inv

% Open document help
doc inv
```

```
INV      Matrix inverse.
        INV(X) is the inverse of the square matrix X.
        A warning message is printed if X is badly scaled or
        nearly singular.

        See also SLASH, PINV, COND, CONDEST, LSQNONNEG, LSCOV.

Overloaded methods:
    scmapinv/inv
    scmap/inv
    moebius/inv
    composite/inv
    chebfun/inv
    gf/inv
    InputOutputModel/inv
    idmodel/inv
    uss/inv
    umat/inv
    ufrd/inv
    ndlft/inv
    atom/inv

Reference page in Help browser
doc inv
```