# Random Numbers in Matlab, C and Java

Warning: none of these languages provide facilities for choosing truly random numbers. They just provide pseudo-random numbers. But, we'll pretend that they are random for now, and address the details later. In matlab, one can generate a random number chosen uniformly between 0 and 1 by

```
x = rand(1)
```

To obtain a vector of n random numbers, type

```
x = rand(1,n)
```

If you type

```
x = rand(n)
```

you get a n-by-n matrix of random numbers, which could be way too big. Be careful not to confuse rand with randn, which produces Gaussian random variables.

---

In C, the easiest way to produce random numbers is with code like:

```
#include <stdio.h>
#include <stdlib.h>

main(int ac, char **av)
{
  double x;

  x = rand()/(RAND_MAX+1.0);

  printf("%g\n",x);
}
```

You will note that each time you run this code, you get the same answer. You can fix this by changing the seed with which the pseudo-random generator is initialized. An example that takes the seed (a positive integer) on the command line is:

```
#include <stdio.h>
#include <stdlib.h>

main(int ac, char **av)
{
  double x;
  unsigned seed;

  if (ac > 0) {
    sscanf(av[1],"%u", &seed);
    srand(seed);
  }

  x = rand()/(RAND_MAX+1.0);

  printf("%g\n",x);
}
```

Some people prefer to set the seed to something that is always changing, such as the time. If you do this, then each time you run your program you will get different results. While this seems more

random, *it makes debugging very difficult*.