**Test Matrices**

These are descriptions of some constructible families of matrices for use in testing programs designed to solve linear systems.

1. **Hilbert Matrices**: $a_{ij} = 1/(i + j - 1)$.
   
   These matrices are symmetric and positive definite (thus don’t need any pivoting). As $n$ increases they become more and more ill-conditioned. In double precision, you get total garbage around $n = 12$.

2. **Vandermode Matrices**: $a_{ij} = x_{i}^{j-1}$ where $x^{0} = 0$ and $x = (x_{i})$ is a vector of distinct values. Equally spaced values in $[0, 1]$ work fine.
   
   These also become ill-conditioned as $n$ increases. Can be worse than the Hilbert matrices.

3. **Random Matrices**: $a_{ij} = \text{rand}()$, where \text{rand}() is some random number generator. It can be a uniform or normal distribution, scaled or not.
   
   These can become ill-conditioned as $n$ gets large. Nice way to test pivoting strategies, especially if you randomly scale each row.

4. **Random Symmetric Positive Definite Matrices**: $A = R^{T}R$ where $R$ is a random matrix from above.

5. **1D 2nd Order Operator**: $a_{ii} = 2, a_{i-1,i} = a_{i+1,i} = -1$, rest are 0.
   
   Sparse, symmetric, positive definite, tridiagonal.

6. **2D 2nd Order Operator**: $a_{ii} = 4, a_{i-1,i} = a_{i+1,i} = a_{i-M,i} = a_{i+M,i} = -1, 1 \leq i \leq M^{2}$, rest are 0.
   
   Sparse, symmetric, positive definite, only 5 diagonals, but will fill-in using direct methods.


Given an $A$ from above, how do you find a $b$? For testing it is best to choose a vector $x$ and compute $b = Ax$, then you can compare the answer your program gets when solving $Ax = b$ with the answer you started with. Some good choices for $x$: $x = (1, 1, 1, \ldots, 1)$, $x = (1, 2, 3, \ldots, n)$, $x = (1, -1, 1, -1, \ldots, (-1)^{n+1})$, $x =$ random vector.