Numerical Methods 35006
Computer Lab 9: Sparse Linear Systems

For this Lab we will be getting using the `scipy` sparse package, as well as the sparse linear algebra package. These can be imported using the command

```
from scipy import sparse as sp
from scipy.sparse import linalg as sla
```

---

1. Use python to create the following sparse matrices in CSR format:

   a) A 5×5 matrix with the integers 1..5 numbered down the diagonal

   b) A matrix the same as (a), but with number 1 in the diagonal immediately above the main diagonal.

   Convert both these matrices to dense format and print them.

2. a) Create a random dense matrix using the command

   `A = (np.random.randn(20,20)-0.5)**5`

   Set each entry of `A` which has an absolute value less than 0.1 to zero, then use this to define a new sparse matrix `A2`. View A2 using the command `plt.spy(A2)`.

   b) Create a 5000×5000 sparse matrix similar to that of part (a). By using the `.nbytes` method (for the dense matrix), and the `.size` method for the sparse matrix, compare the sizes of the two matrices in MB.

3. Use the `sp.random` command to create a 5000×5000 sparse matrix A filled with random entries with a density of 0.1, as well as 5000×1 random sparse vector with the same density. Solve the sparse linear system

   $$Ax = b$$

   and check that the solution is correct, using the L2 norm of the vector $c = Ax - b$.

4. Create, and visualise using `spy`, the following sparse matrices:

   a) A 500×500 sparse identity matrix

   b) A sparse matrix with the number 1 down the diagonal and the number 2 down the upper diagonal

   c) A tri-diagonal sparse matrix with ones down the main diagonal and the number 0.5 down the upper and lower diagonal.

5. Create a 500×500 random sparse diagonal matrix of the type done in Question 3. Use the function `sla.splu` to form the LU decomposition of this matrix, and use `spy` to check that the matrices have the correct form.

6. Create a random 1000×1000 sparse matrix of the type shown in Question 3. Modify your code from Lab 8 to use power iteration to compute the largest eigenvalue of this matrix. Check this eigenvalue and your eigenfunction using the in-built sparse eigenvalue solver `sla.eigs`.

7. Use your code from the previous section to create a function

   `lam,vec = speig(A,tol)`

   That computes the largest eigenvalue and the corresponding eigenvector of a sparse matrix `A` to a given tolerance. Save this to your `myeigs.py` module.

8. A Hermitian matrix, which has real eigenvalues, can be created by forming the sum

   $$A^H = A + (A^*)^T \ .$$

   Build an Arnoldi-iteration algorithm, as covered in lectures, to find the eigenvalues and eigenvectors of a sparse Hermitian matrix A. Test it, and save it to your `myeigs.py` module.