Return m-files before 31.5.2011
Email the solution to :
`fysp120(at)gmail.com` Subject-line: `FNM exercise4`

1. **Get a copy of** `hydrogen.m`.
   Let's solve the radial wavefunction of the Hydrogen atom in matrix form. The differential equation is

$$-\frac{d^2}{dr^2}\phi(r) - \frac{2}{r}\phi(r) = E\phi(r),$$

which after discretization becomes the matrix equation

$$\left(\frac{1}{h^2}\begin{pmatrix} 2 & -1 & 0 & 0 & \dots & 0 \\ -1 & 2 & -1 & 0 & \dots & 0 \\ 0 & -1 & 2 & -1 & \dots & 0 \\ & & \vdots & & & \\ 0 & 0 & 0 & 0 & \dots & 2 \end{pmatrix} - \alpha \begin{pmatrix} 1/r_1 & 0 & 0 & 0 & \dots & 0 \\ 0 & 1/r_2 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1/r_3 & 0 & \dots & 0 \\ & & \vdots & & & \\ 0 & 0 & 0 & 0 & \dots & 1/r_N \end{pmatrix}\right)\begin{pmatrix} \phi(r_1) \\ \phi(r_2) \\ \phi(r_3) \\ \vdots \\ \phi(r_N) \end{pmatrix} = E \begin{pmatrix} \phi(r_1) \\ \phi(r_2) \\ \phi(r_3) \\ \vdots \\ \phi(r_N) \end{pmatrix}.$$

The second derivative discretization is in the lecture notes. Here $h = r_k - r_{k-1}$ is the step size and for Hydrogen the factor $\alpha = 2$. The accuracy of the eigenvalues depends on the discretization $r_1, ..., r_N$; the exact values are $E = -1/n^2$, where $n = 1, 2...$ is the principal quantum number, hence $E = -1, -0.25, ...$
a) Modify `hydrogen.m` to use sparse matrices, using `spdiags` and `eigs` (also `gallery` may be usefull).
Examine how the potential affects the ground state wave function $\phi_1(r)$. Solve $\phi_1(r)$ for $\alpha = \{0, 0.1, ..., 2\}$, that is, from a free particle to full Coulomb potential. Note: it's a good idea to turn all solution eigenvectors on the positive side; *i.e.* some solutions multiplied with -1.
b) Compute the expansion of the function

$$f(r) = re^{-0.6r}\sin(r)$$

in the basis of Hydrogen eigenstates, *i.e.*, compute the coefficients a_i in the sum

$$f(x) = \sum_{i=1}^{N} a_i\phi_i(r).$$

Draw the approximating sum functions in the cases of 1,5,20 or all N terms in the sum.

CONTINUES ON THE NEXT PAGE!

HINTS

Filling tridiagonal matrices

Matrix elements can be set in a loop, or using built-in help functions. Check out what matrices these produce:

```
N = 4;
A =  spdiags(ones(N,1),-1,N,N) + spdiags((2:5)',0,N,N) ...
   + spdiags(ones(N,1),1,N,N)
full(A)
```

and

```
N = 4;
A = gallery('tridiag',N,1,0,1) + spdiags((2:5)',0,N,N)
full(A)
```

Finding a few lowest eigenvalues of a sparse matrix

```
E = eigs(A,3,'sa'); % 3 lowest eigenvalues of A
```

You can get all eigenvalues of a sparse matrix `A` taking first a `full` form of it, and using `eig`:

```
E = eig(full(A)); % E is a vector
```

Getting also eigenvectors, Matlab returns eigenvalues in a diagonal matrix:

```
[V,E] = eigs(A);        % E is a diagonal matrix
[V,E] = eig(full(A)) ; % E is a diagonal matrix
E     = diag(E)      ; % diagonal values to a vector E
```

Finally, the eigenvalues returned by `eig` or `eigs` are not necessarily in ascending order. To be sure, sort them yourself:

```
[E,ind] = sort(diag(E)); % if E is a matrix, else sort(E)
V = V(:,ind);
```

Here `ind` is the sorting index. Now `V(:,1)` is the eigenvector corresponding to the lowest eigenvalue.