Goals:
Vector normalization; Participation Ratio; Matrix Multiplication; Localized Modes.

[1] In Lab 13 we used some linear algebra software ('Numerical Recipes') to diagonalize a matrix. Let's learn a bit about what more simple linear algebra codes look like. Here's a code which computes the length of a vector and 'normalizes' it:

```c
#include <stdio.h>
#include <math.h>
int main(void)
{
    int i,N;
    double patronus[1024],length;

    printf("Enter the length N of the vector ");
    scanf("%d",&N);

    for (i=0; i<N; i=i+1)
    {
        printf("Enter vector component %4d ",i);
        scanf("%lf",&patronus[i]);
    }

    length=0.0;
    for (i=0; i<N; i=i+1)
    {
        length=length+patronus[i]*patronus[i];
    }
    length=sqrt(length);
    printf("Your vector has length %12.6lf 
",length);

    for (i=0; i<N; i=i+1)
    {
        patronus[i]=patronus[i]/length;
        printf("Component %d of your normalized vector: %12.6lf \n",i,patronus[i]);
    }

    return 0;
}
```
[HW7-5] Explain what it means to ‘normalize’ a vector.

[HW7-6] Modify the preceding code to compute the ‘participation ratio’ $P$ of a vector. You should keep all of the code: the input of the vector, the calculation of its length, its normalization, and then finally add one more loop which gets $P$. What do you get for $P$ for the vectors:

$$
\vec{v}_1 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \quad \vec{v}_2 = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \quad \vec{v}_3 = \begin{bmatrix} 2.5 \\ 3.0 \\ 3.1 \\ 2.6 \\ 2.8 \end{bmatrix}, \quad \vec{v}_4 = \begin{bmatrix} 4 \\ 5 \\ 0 \\ 0 \\ 4.5 \end{bmatrix}, \quad \vec{v}_5 = \begin{bmatrix} -0.1 \\ 0.2 \\ 0.1 \\ -0.3 \\ 3 \end{bmatrix}
$$

Discuss why these values ‘make sense’.

[HW7-7] One of the most basic linear algebra procedures is multiplying a vector by a matrix. In the ‘BLAS’ package (http://www.netlib.org/blas/) the function is called ‘dgemm’. We’ll see here how dgemm works. Write a code which
(a) initializes the elements of a 5x5 matrix $\mathcal{M}$ to have the values

$$
\begin{bmatrix}
3.0 & 1.0 & -0.5 & 0.4 & 1.2 \\
1.0 & 4.0 & 0.7 & 0.2 & -0.9 \\
-0.5 & 0.7 & -3.2 & 0.3 & -0.1 \\
0.4 & 0.2 & 0.3 & 3.5 & 0.8 \\
1.2 & -0.9 & -0.1 & 0.8 & -3.7
\end{bmatrix}
$$

(b) reads in a five component vector $\vec{v}$; and
(c) Computes $\vec{w} = \mathcal{M} \vec{v}$. Then use your code to get $\vec{w}_1 = \mathcal{M} \vec{v}_1$ and $\vec{w}_2 = \mathcal{M} \vec{v}_2$ for

$$
\vec{v}_1 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \quad \vec{v}_2 = \begin{bmatrix} 1 \\ 2 \\ 1 \\ -2 \end{bmatrix}
$$

[2] Preview of coming attractions: We may delay python one more lab and use this machinery in combination with Lab 13 to learn about localized and extended modes!