Lesson 12: Linear Algebra I

Linear algebra is one of the most important subjects in mathematics, with applications found in almost every branch of physics. The techniques of linear algebra also form the basis of many numerical algorithms used in mathematics, science and engineering. In this lesson, you will translate linear algebraic systems into matrix–vector notation and solve these problems using built–in python commands.

1 Linear systems

One of the basic problems of linear algebra is the solution of n linear algebraic equations for n unknowns. Consider the system

$$x + 3y = -3 \, , \tag{1a}$$
$$2x - 2y = 10 \, , \tag{1b}$$

of two equations for two unknowns. You can easily verify that the solution is $x = 3$, $y = -2$.

The problem (1) can be written in matrix notation as

$$\begin{pmatrix} 1 & 3 \\ 2 & -2 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} -3 \\ 10 \end{pmatrix} . \tag{2}$$

Here and below, we assume that you are familiar with matrix multiplication. More compactly, the problem (2) can be written as $Av = r$ where

$$A = \begin{pmatrix} 1 & 3 \\ 2 & -2 \end{pmatrix} \tag{3}$$

is the coefficient matrix, $v = (x, y)^T$ is the vector of unknowns, and $r = (-3, 10)^T$ is the vector of right–hand side values.[1] The problem of solving for x and y can now be expressed as the problem of inverting the matrix A. If A^{-1} denotes the inverse of A, then the solution is $v = A^{-1}r$.

It is straightforward to verify that

$$A^{-1} = \begin{pmatrix} 2/8 & 3/8 \\ 2/8 & -1/8 \end{pmatrix} \tag{4}$$

[1]The superscript T denotes transpose. The transpose of a row vector is a column vector.

is the inverse of A. That is, using matrix multiplication, $A^{-1}A = I$ where I is the identity matrix. Thus, the result $v = A^{-1}r$ can be written explicitly as

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 2/8 & 3/8 \\ 2/8 & -1/8 \end{pmatrix} \begin{pmatrix} -3 \\ 10 \end{pmatrix} \tag{5}$$

By carrying out the multiplication of A^{-1} and r, we find the solution

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 3 \\ -2 \end{pmatrix} . \tag{6}$$

That is, $x = 3$ and $y = -2$.

These calculations are easy to carry out in python. Let's assume you have the numpy functions loaded using

```
import numpy as n
```

The syntax for defining the matrix A and computing its inverse A^{-1} is

```
A = n.matrix([[1,3],[2,-2]])
Ainv = n.linalg.inv(A)
```

(Note that the `inv` command is contained in the linear algebra subpackage `linalg` within numpy. The inverse can be computed using the shorter command `Ainv = A.I`.) We can now define the right–hand side column vector r and compute the answer $A^{-1}r$ as follows:

```
r = n.matrix([[-3],[10]])
v = Ainv*r
```

The result is $v = (3, -2)^T$.

Exercise 1: Solve the linear system

$$\begin{aligned} 2x + 5y - z &= -8.5 && \text{(7a)} \\ -x + 3y + 4z &= 3 && \text{(7b)} \\ 4x - 6z &= -10 && \text{(7c)} \end{aligned}$$

by computing the inverse of the coefficient matrix.

In general, solving a system of linear equations by inverting the coefficient matrix is a bad idea. The subject of linear algebra is largely concerned with finding alternative methods for solving systems of linear equations. Finding the numerical inverse of a matrix is:

- Very time consuming. Other methods of solving linear systems are faster.
- Very inaccurate. Inverting a matrix can lead to very large floating–point round–off errors.

As an example, consider the matrix

$$A = \begin{pmatrix} 4001 & -2001 \\ -8000 & 4001 \end{pmatrix} . \tag{8}$$

You can easily verify that the inverse is

$$A^{-1} = \begin{pmatrix} 4001 & 2001 \\ 8000 & 4001 \end{pmatrix} . \tag{9}$$

However, A is an example of an "ill–conditioned" matrix that is difficult to invert numerically.

> Exercise 2: Use python to compute A^{-1} and compare the result with the exact answer. Compute $A^{-1}A$ and AA^{-1}, and compare to the identity matrix.

Numpy includes the function `solve` in the subpackage `linalg` for solving the linear system $Av = r$:

```
v = n.linalg.solve(A,r)
```

The algorithm used by `solve` can be faster and more accurate than explicit matrix inversion. From now on, you should use `solve` rather than matrix inversion to solve linear systems of equations.

> Exercise 3: Use the `solve` command to solve the linear system from Exercise I.

2 Example: Statics

A beam of length ℓ and mass m is connected to a wall at one end, and supported by a cable attached to the other end. See Fig. 1. The beam is in static equilibrium, so the sum of forces must vanish, $\sum \vec{F} = 0$, and the sum of torques about any point $\mathcal{P}$ must vanish, $\sum \vec{\tau}_{\mathcal{P}} = 0$. All forces lie in a common plane, namely, the x–y plane. Thus, $\sum \vec{F} = 0$ reduces to two equations, one for the x–component and one for the y–component. Likewise, $\sum \vec{\tau}_{\mathcal{P}} = 0$ reduces to a single equation for the z–component. If we use the middle of the beam as the point $\mathcal{P}$, the three equations are

$$F_x - T\cos\theta = 0 \, , \tag{10a}$$
$$F_y + T\sin\theta - mg = 0 \, , \tag{10b}$$
$$(T\sin\theta)(\ell/2) - F_y(\ell/2) = 0 \, . \tag{10c}$$

Assuming m, ℓ and θ are known, these three equations can be solved for the three unknowns F_x, F_y and T.

> Homework: Write a code to solve Eqs. (10) for F_x, F_y and T. Use SI units and let $m = 14$, $\ell = 1.2$, $g = 9.8$ and $\theta = 35°$.

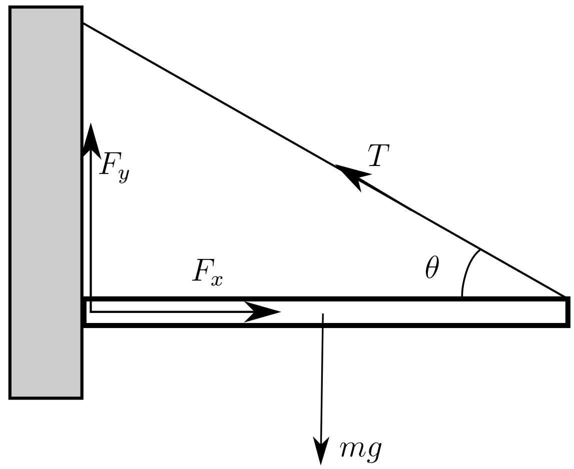

Figure 1: A static beam connected to a wall and a cord. The cord makes an angle θ with the beam, and has tension T. The wall exerts a force on the beam with components F_x, F_y.

3 Example: Kirchoff's laws

Kirchoff's laws apply to electrical circuits with steady currents. Kirchoff's first law states that the total current entering (or leaving) a node is zero. Kirchoff's second law states that the sum of voltages around any closed loop must vanish. For example, Kirchoff's second law applied to the left–side loop of Fig. 2 gives $\mathcal{E}_1 - I_1R_1 - I_3R_3 = 0$.

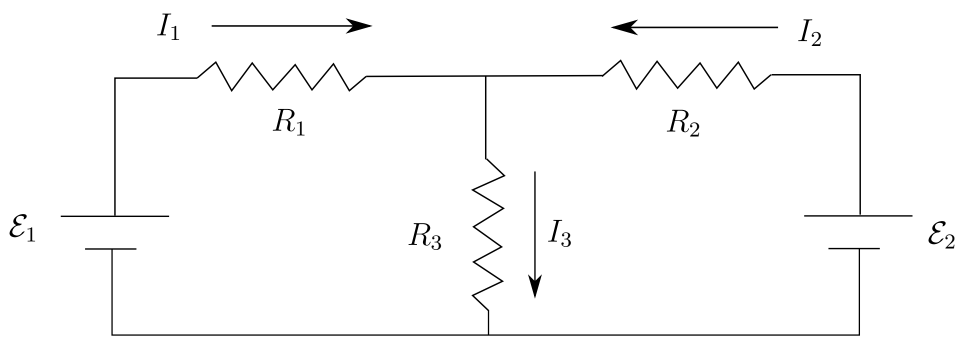

Figure 2: A simple circuit. The currents are determined by Kirchoff's laws.

Homework: Use Kirchoff's laws to write a system of equations for the circuit shown in Fig. 2. Express the equations in matrix form, with the currents as unknowns. Write a python code to solve for the currents in term of voltages and resistances. Find the currents for the data $\mathcal{E}_1 = 12\,\mathrm{V}$, $\mathcal{E}_2 = 9\,\mathrm{V}$, $R_1 = 100\,\Omega$, $R_2 = 120\,\Omega$, $R_3 = 65\,\Omega$.