L1 Linear Equations

January 2, 2015

1 Linear Equations

In [101]:
```
%matplotlib inline

# import libraries
import numpy as np
import matplotlib as mp
import pandas as pd
import matplotlib.pyplot as plt
import laUtilities as ut
import slideUtilities as sl
from IPython.display import Image
from IPython.display import display_html
from IPython.display import display
reload(ut)
```

Out[101]:
```
<module 'laUtilities' from 'laUtilities.pyc'>
```

In [102]:
```
%%html
<style>
 .container.slides .celltoolbar, .container.slides .hide-in-slideshow {
    display: None ! important;
}
</style>
```

```
<IPython.core.display.HTML at 0x10de56d90>
```

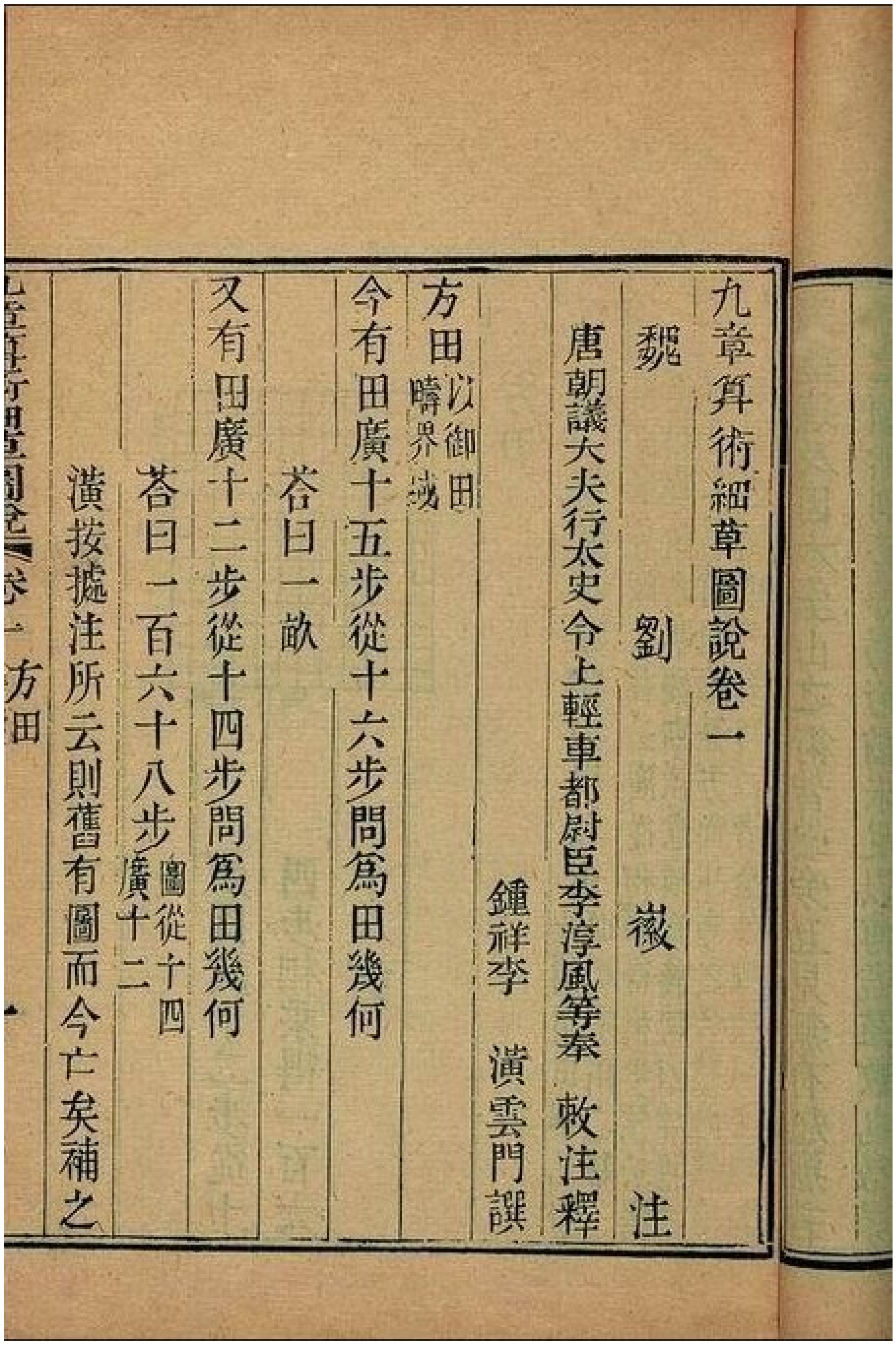
> Traditionally, algebra was the art of solving equations and systems of equations. The word *algebra* comes form the Arabic *al-jabr* which means *restoration* (of broken parts). The term was first used in a mathematical sense by Mohammed al-Khowarizmi (c. 780-850) who worked at the House of Wisdom, an academy established by Caliph al Ma'mum in Baghdad. Linear algebra, then, is the art of solving systems of linear equations.

Linear Algebra with Applications, Bretscher
Al-Khowarizmi gave his name to the *algorithm*.

> The yield of one bundle of inferior rice, two bundles of medium grade rice, and three bundles of superior rice is 39 *dou* of grain. The yield of one bundle of inferior rice, three bundles of medium grade rice, and two bundles of superior rice is 34 *dou*. The yield of three bundles of inferior rice, two bundles of medium grain rice, and one bundle of superior rice is 26 *dou*. What is the yield of one bundle of each grade of rice?

Nine Chapters on the Mathematical Art, c 200 BCE, China

```
In [103]: # image credit: http://en.wikipedia.org/wiki/The_Nine_Chapters_on_the_Mathematical_Art
          sl.hide_code_in_slideshow()
          display(Image("images/nine-chapters-mathematical-art.jpg"))
```



九章算術細草圖說卷一

魏 劉 徽 注

唐朝議大夫行太史令上輕車都尉臣李淳風等奉 敕注釋

鍾祥李 潢雲門譔

方田以御田疇界域

今有田廣十五步從十六步問爲田幾何

荅曰一畝

又有田廣十二步從十四步問爲田幾何

荅曰一百六十八步圖從十四廣十二

潢按攄注所云則舊有圖而今亡矣補之

九章算術細草圖說卷一 方田

Let's denote the unknown quantities as x_1, x_2, and x_3. These are the yields of one bundle of inferior, medium grade, and superior rice, respectively. We can then write the problem as:

$$
\begin{aligned}
x_1 + 2x_2 + 3x_3 &= 39 \\
x_1 + 3x_2 + 2x_3 &= 34 \\
3x_1 + 2x_2 + x_3 &= 26
\end{aligned}
$$

The problem then is to determine the values of x_1, x_2, and x_3.
These are *linear* equations. No term has power other than 1.
For example, there are no terms involving x_1^2, or $x_1 x_2$, or $\sqrt{x_3}$.

1.1 Basic Definitions

- A *linear equation* in the variables $x_1, \ldots, x_n$ is an equation that can be written in the form

$$a_1 x_1 + a_2 x_2 + \cdots + a_n x_n = b$$

  where b and the coefficients $a_1, \ldots, a_n$ are real or complex numbers that are usually known in advance.
- A *system of linear equations* (or *linear system*) is a collection of one or more linear equations involving the same variables - say $x_1, \ldots, x_n$.
- A *solution* of the system is a list of numbers $(s_1, s_2, \ldots, s_n)$ that makes each equation a true statement when the values $s_1, s_2, \ldots, s_n$ are substituted for $x_1, x_2, \ldots, x_n$, respectively.
- The set of all possible solutions is called the *solution set* of the linear system.
- Two linear systems are called *equivalent* if they have the same solution set.
- A system of linear equations has
  1. no solution, or
  2. exactly one solution, or
  3. infinitely many solutions.
- A system of linear equations is said to be *consistent* if it has either one solution or infinitely many solutions.
- A system of linear equations is said to be *inconsistent* if it has no solution.

1.2 Interpreting Things Geometrically

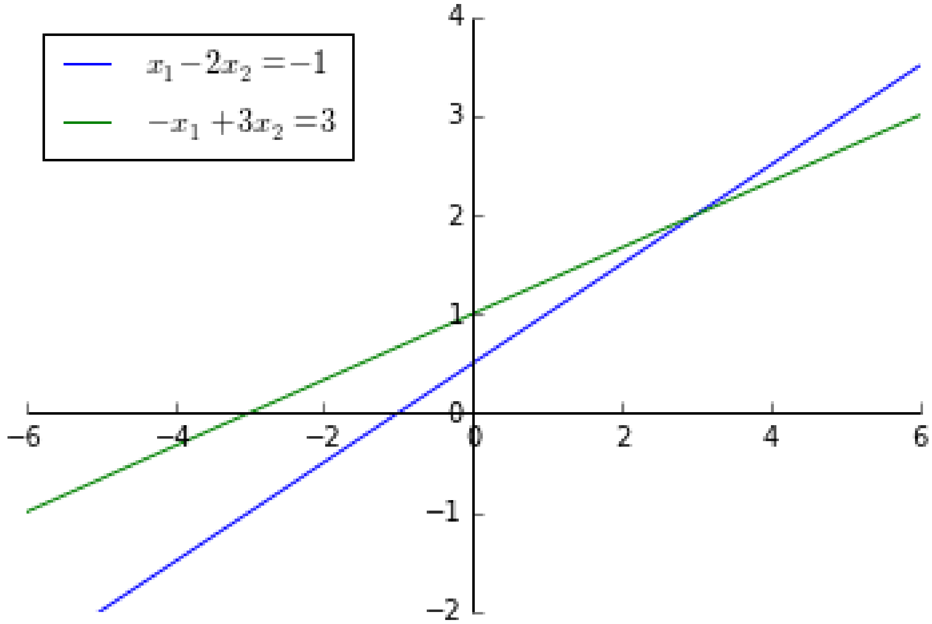
Any list of numbers $(s_1, s_2, \ldots, s_n)$ can be thought of as a point in n-dimensional space, called a *vector space.*
We call that vector space $\mathbb{R}^n$.
So if we are considering linear equations with n unknowns, the solutions are points in $\mathbb{R}^n$.
Now, any linear equation defines a point set with dimension one less than the space. For example:

- if we are in 2-space (2 unknowns), a linear equation defines a line.
- if we are in 3-space (3 unknowns), a linear equation defines a plane.
- in higher dimensions, we refer to all such sets as *hyperplanes.*

Question: why does a linear equation define a point-set of dimension one less than the space?

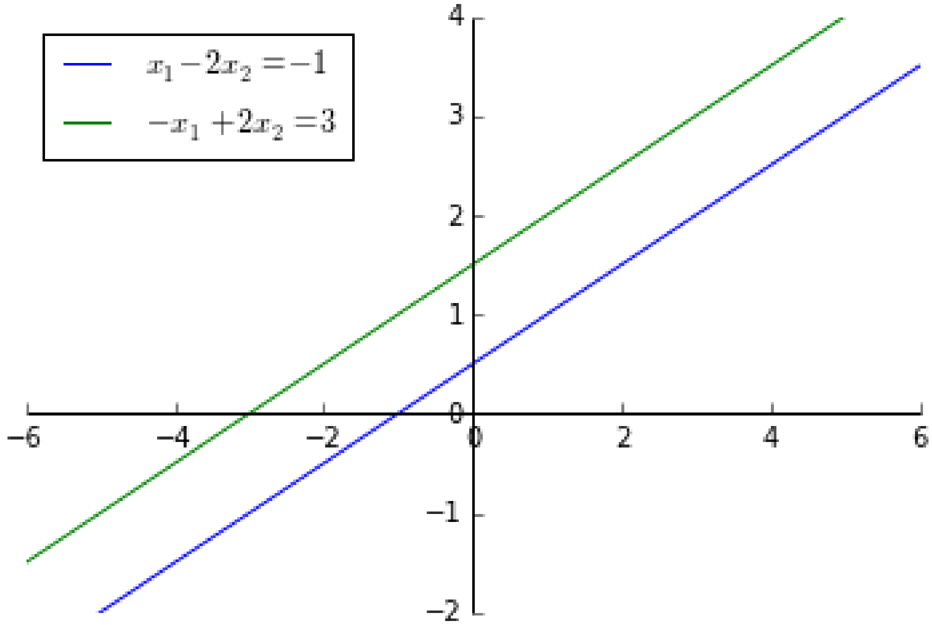
1.3 Some Examples in $\mathbb{R}^2$

In [104]:
```
# No need to study this code unless you want to.
sl.hide_code_in_slideshow()
ax = ut.plotSetup()
ut.centerAxes(ax)
ut.plotLinEqn(1, -2, -1)
ut.plotLinEqn(-1, 3, 3)
plt.legend(loc='best')
print ''
```



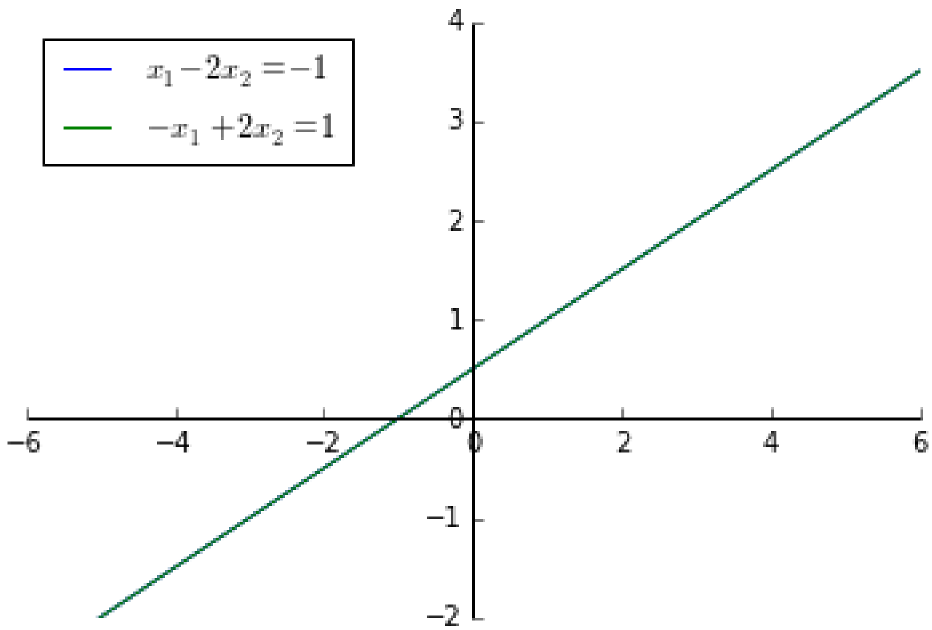
This system of two equations has **exactly one** solution.

```
In [105]: # No need to study this code unless you want to.
          sl.hide_code_in_slideshow()
          ax = ut.plotSetup()
          ut.centerAxes(ax)
          ut.plotLinEqn(1, -2, -1)
          ut.plotLinEqn(-1, 2, 3)
          plt.legend(loc='best')
          print ''
```



This system of two equations has **no** solutions.

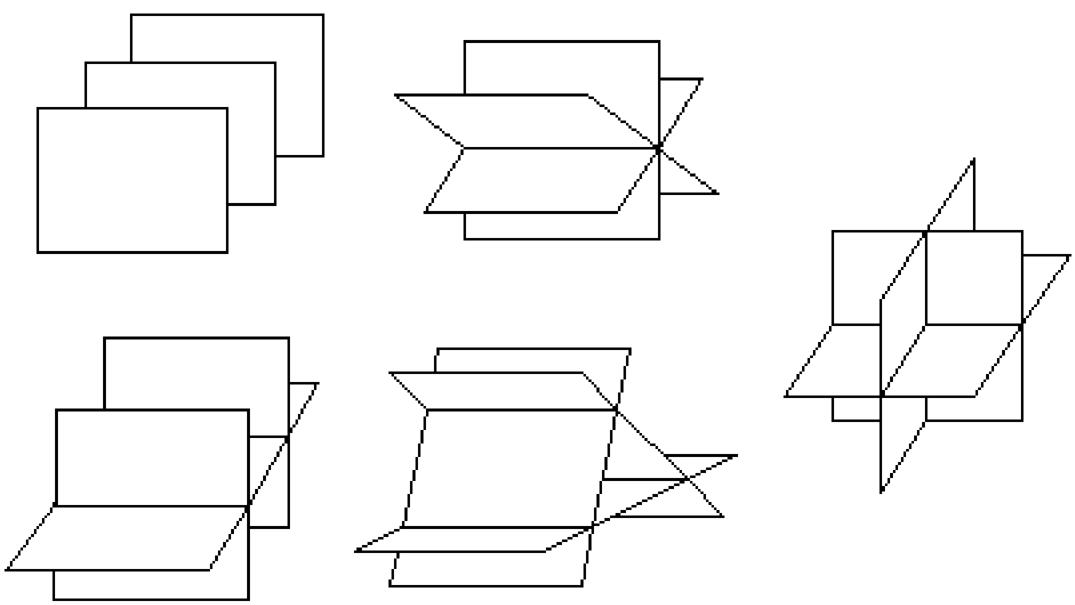
```
In [106]: # No need to study this code unless you want to.
          sl.hide_code_in_slideshow()
          ax = ut.plotSetup()
          ut.centerAxes(ax)
          ut.plotLinEqn(1, -2, -1)
          ut.plotLinEqn(-1, 2, 1)
          plt.legend(loc='best')
          print ''
```



This system of equations has **infinitely many** solutions.

1.4 Some Examples in $\mathbb{R}^3$

```
In [107]: sl.hide_code_in_slideshow()
          display(Image("images/Pic_3-planes.png"))
```



How many solutions are there in each of these cases?

The essential information of a linear system can be recorded compactly in a rectangular array called a matrix. For the following system of equations,

$$
\begin{aligned}
x_1 - 2x_2 + x_3 &= 5 \\
2x_2 - 8x_3 &= -4 \\
6x_1 + 5x_2 + 9x_3 &= -4,
\end{aligned}
$$

the matrix

$$
\begin{bmatrix} 1 & -2 & 1 \\ 0 & 2 & -8 \\ 6 & 5 & 9 \end{bmatrix}
$$

is called the *coefficient matrix* of the system.

An augmented matrix of a system consists of the coefficient matrix with an added column containing the constants from the right sides of the equations.

For the same system of equations,

$$
\begin{aligned}
x_1 - 2x_2 + x_3 &= 5 \\
2x_2 - 8x_3 &= -4 \\
6x_1 + 5x_2 + 9x_3 &= -4,
\end{aligned}
$$

the matrix

$$
\begin{bmatrix} 1 & -2 & 1 & 5 \\ 0 & 2 & -8 & -4 \\ 6 & 5 & 9 & -4 \end{bmatrix}
$$

is called the *augmented matrix* of the system.

A matrix with m rows and n columns is referred to as ''an $m \times n$ matrix" and is an element of the set $\mathbb{R}^{m \times n}$. (Note that we always list the number of rows first, then the number of columns.)

1.5 Solving Linear Systems

To solve a linear system, we transform it into a *new* system which is equivalent to the old system, meaning it has the same solution set. However the new system is easier to solve.

We make the following observation: given a set of linear equations, we can **add one equation to another** without changing the solution set. By definition, any solution of the old system makes each old equation true; therefore any solution of the old system makes each new equation true.

Another, more obvious fact is that we can multiply any equation by a constant without changing its meaning (and therefore the solution set).

And an even more obvious fact is that we can change the order of the equations without changing anything.

Together, these three rules form a set of tools we can use to solve linear systems. Here is an example.

We'll do this with the equations and the matrix side-by-side. The basic operation we will repeatedly apply is to *add a multiple of one equation (row) to another.* The goal is to eliminate terms to create a *triangular* matrix (or system).

Here is the original system:

$$
\begin{aligned}
x_1 - 2x_2 + x_3 &= 5 \\
2x_2 - 8x_3 &= -4 \\
6x_1 + 5x_2 + 9x_3 &= -4,
\end{aligned}
\qquad
\begin{bmatrix} 1 & -2 & 1 & 5 \\ 0 & 2 & -8 & -4 \\ 6 & 5 & 9 & -4 \end{bmatrix}
$$

To begin: we add -6 times the first equation to the third equation:

$$
\begin{array}{rrrrcr}
 & 6x_1 & +5x_2 & +9x_3 & = & -4 \\
+ & -6x_1 & +12x_2 & -6x_3 & = & -30 \\
\hline
 & & 17x_2 & +3x_3 & = & -34
\end{array}
$$

This gives us a new system.

$$\begin{array}{rcl} x_1 - 2x_2 + x_3 & = & 5 \\ 2x_2 - 8x_3 & = & -4 \\ +17x_2 + 3x_3 & = & -34, \end{array} \qquad \left[\begin{array}{rrrr} 1 & -2 & 1 & 5 \\ 0 & 2 & -8 & -4 \\ 0 & 17 & 3 & -34 \end{array}\right]$$

Note that this is not the *same* system of equations, but it is *equivalent* – it has the same solution set.
Next, we multiply the second equation by $1/2$ to get its leading coefficient to be 1:

$$\begin{array}{rcl} x_1 - 2x_2 + x_3 & = & 5 \\ x_2 - 4x_3 & = & -2 \\ +17x_2 + 3x_3 & = & -34, \end{array} \qquad \left[\begin{array}{rrrr} 1 & -2 & 1 & 5 \\ 0 & 1 & -4 & -2 \\ 0 & 17 & 3 & -34 \end{array}\right]$$

Next, we multiply the second equation by -17 and add it to the third equation:

$$\begin{array}{rcl} x_1 - 2x_2 + x_3 & = & 5 \\ x_2 - 4x_3 & = & -2 \\ 72x_3 & = & 0, \end{array} \qquad \left[\begin{array}{rrrr} 1 & -2 & 1 & 5 \\ 0 & 1 & -4 & -2 \\ 0 & 0 & 72 & 0 \end{array}\right]$$

And next we can divide the third equation by 72 to get its leading coefficient equal to 1:

$$\begin{array}{rcl} x_1 - 2x_2 + x_3 & = & 5 \\ x_2 - 4x_3 & = & -2 \\ x_3 & = & 0, \end{array} \qquad \left[\begin{array}{rrrr} 1 & -2 & 1 & 5 \\ 0 & 1 & -4 & -2 \\ 0 & 0 & 1 & 0 \end{array}\right]$$

We have now put the system and matrix into *triangular* form. In the matrix, all values below the diagonal are zero.

At this point, the process shifts a bit to *backsubstitution*. We now have the value for one variable, and we will substitute it into other equations to simply them and get values for the other variables.

Although we think of its as a somewhat different stage, in reality it still comes down to applying the three rules. First, we substitute the value of x_3 into the equations above it. This is actually multiplying equation 3 by the proper value and adding it to equations above it.

$$\begin{array}{rcl} x_1 - 2x_2 & = & 5 \\ x_2 & = & -2 \\ x_3 & = & 0, \end{array} \qquad \left[\begin{array}{rrrr} 1 & -2 & 0 & 5 \\ 0 & 1 & 0 & -2 \\ 0 & 0 & 1 & 0 \end{array}\right]$$

Next, we do the same thing with equation 2, substituting it into equation 1 above it:

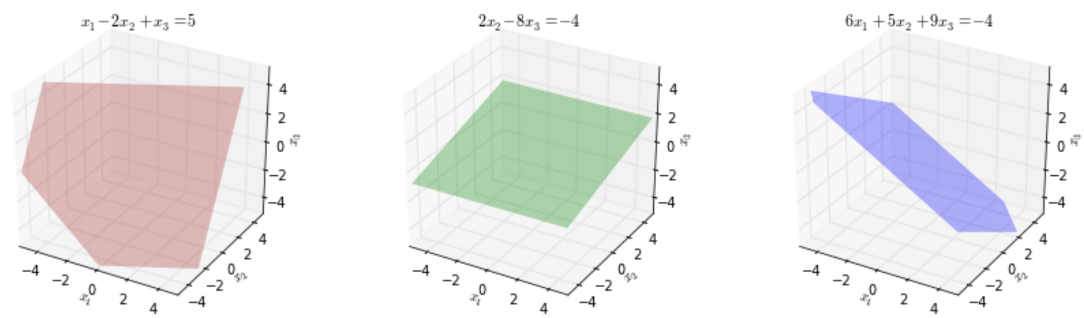
$$\begin{array}{rcl} x_1 & = & 1 \\ x_2 & = & -2 \\ x_3 & = & 0, \end{array} \qquad \left[\begin{array}{rrrr} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & -2 \\ 0 & 0 & 1 & 0 \end{array}\right]$$

Now we can read off the solution: it is $x_1 = 1$, $x_2 = -2$, $x_3 = 0$.

Let's get a sense of this process geometrically. Here are the three starting equations:

```
In [108]: sl.hide_code_in_slideshow()
          fig = plt.figure()
          xmin = ymin = zmin = -5
          xmax = ymax = zmax = 5
          axs=[1,2,3]
          axs[0] = fig.add_subplot(131, projection='3d')
          axs[1] = fig.add_subplot(132, projection='3d')
          axs[2] = fig.add_subplot(133, projection='3d')
          for ax in axs:
              ax.axes.set_xlim([xmin, xmax])
              ax.axes.set_ylim([ymin, ymax])
              ax.axes.set_zlim([zmin, zmax])
              ax.axes.set_xlabel('$x_1$')
```

```
    ax.axes.set_ylabel('$x_2$')
    ax.axes.set_zlabel('$x_3$')
#
ax = axs[0]
eq1 = [1,-2,1,5]
ut.plotLinEqn3d(ax, eq1, 'Brown')
ax.set_title('${}$'.format(ut.formatEqn(eq1[0:3], eq1[3])))
#
ax = axs[1]
eq2 = [0,2,-8,-4]
ut.plotLinEqn3d(ax, eq2, 'Green')
ax.set_title('${}$'.format(ut.formatEqn(eq2[0:3], eq2[3])))
#
ax = axs[2]
eq3 = [6,5,9,-4]
ut.plotLinEqn3d(ax, eq3, 'Blue')
ax.set_title('${}$'.format(ut.formatEqn(eq3[0:3], eq3[3])))
#
plt.subplots_adjust(right = 2.0)
```

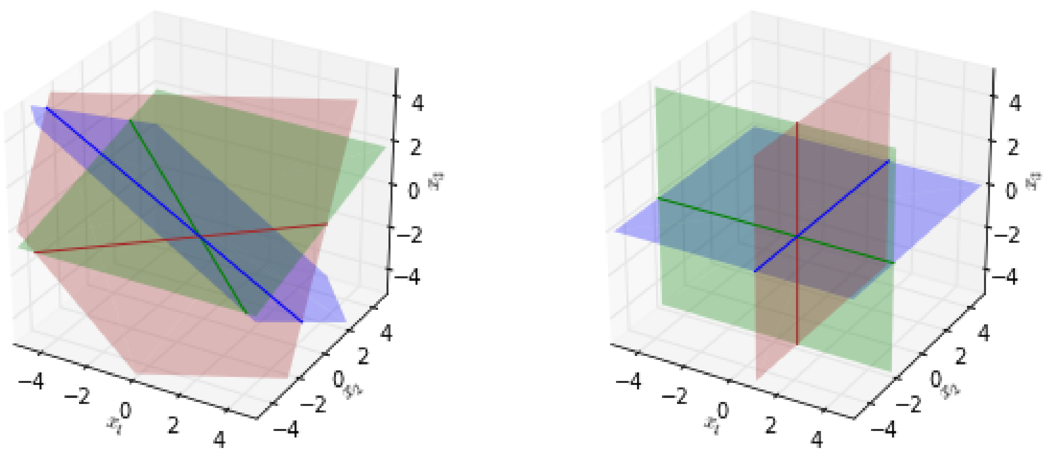


The starting point and the finishing point:

$$\begin{aligned} x_1 - 2x_2 + x_3 &= 5 \\ 2x_2 - 8x_3 &= -4 \\ 6x_1 + 5x_2 + 9x_3 &= -4 \end{aligned} \qquad \begin{aligned} x_1 &= 1 \\ x_2 &= -2 \\ x_3 &= 0 \end{aligned}$$

```
In [109]: sl.hide_code_in_slideshow()
          fig = plt.figure()
          xmin = ymin = zmin = -5
          xmax = ymax = zmax = 5
          axs=[1,2]
          axs[0] = fig.add_subplot(131, projection='3d')
          axs[1] = fig.add_subplot(132, projection='3d')
          for ax in axs:
              ax.axes.set_xlim([xmin, xmax])
              ax.axes.set_ylim([ymin, ymax])
              ax.axes.set_zlim([zmin, zmax])
              ax.axes.set_xlabel('$x_1$')
              ax.axes.set_ylabel('$x_2$')
              ax.axes.set_zlabel('$x_3$')
          ax = axs[0]
          eq1 = [1,-2,1,5]
```

```
eq2 = [0,2,-8,-4]
eq3 = [6,5,9,-4]
ut.plotLinEqn3d(ax, eq1, 'Brown')
ut.plotLinEqn3d(ax, eq2, 'Green')
ut.plotLinEqn3d(ax, eq3, 'Blue')
ut.plotIntersection3d(ax, eq1, eq2, 'Brown')
ut.plotIntersection3d(ax, eq2, eq3, 'Green')
ut.plotIntersection3d(ax, eq1, eq3, 'Blue')
#ax.mouse_init()
#
ax = axs[1]
eq1 = [1, 0, 0, 1]
eq2 = [0, 1, 0, -2]
eq3 = [0, 0, 1, 0]
ut.plotLinEqn3d(ax, eq1, 'Brown')
ut.plotLinEqn3d(ax, eq2, 'Green')
ut.plotLinEqn3d(ax, eq3, 'Blue')
ut.plotIntersection3d(ax, eq1, eq2, 'Brown')
ut.plotIntersection3d(ax, eq2, eq3, 'Green')
ut.plotIntersection3d(ax, eq1, eq3, 'Blue')
#
plt.subplots_adjust(right = 2.0)
```



$$\begin{bmatrix} 1 & -2 & 1 & 5 \\ 0 & 2 & -8 & -4 \\ 6 & 5 & 9 & -4 \end{bmatrix} \qquad \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & -2 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

1.6 Verifying the Solution

TBD

1.7 Row Equivalence

Elementary Row Operations include the following: 1. (Replacement) Replace one row by the sum of itself and a multiple of another row. 2. (Interchange) Interchange two rows. 3. (Scaling) Multiply all entries in a row by a nonzero constant.

Two matrices are called *row equivalent* if there is a sequence of elementary row operations that transforms one matrix into the other.

If the augmented matrices of two linear systems are row equivalent, then the two systems have the same solution set.

1.8 Fundamental Questions

Two fundamental questions about a linear system are as follows: 1. Is the system *consistent;* that is, does at least one solution exist? 2. If a solution exists, is it the only one; that is, is the solution *unique?*

Example of an inconsistent system.

Example of a non-unique system.

```
In [ ]:
```