

# Matrices, Geometry & Mathematica

Authors: Bruce Carpenter, Bill Davis and Jerry Uhl ©2001

Producer: Bruce Carpenter

Publisher: Math Everywhere, Inc.

MGM.06 Beyond 3D

GIVE IT A TRY!

## G.1) Linear systems ready for your analysis

### □G.1.a.i) A linear system

Here's a matrix:

```
hitdim = 3;
hangdim = 5;
A = { {0.722, -2.855, 2.841},
      {1.315, -0.229, -1.267},
      {2.884, -1.006, 0.789},
      {0.192, 2.081, -2.131},
      {1.929, -1.171, 2.328} };
MatrixForm[A]
```

```
{0.722, -2.855, 2.841}
{1.315, -0.229, -1.267}
{2.884, -1.006, 0.789}
{0.192, 2.081, -2.131}
{1.929, -1.171, 2.328}
```

And a linear system based on this matrix:

```
Clear[x, y, k];
X = Table[x[k], {k, 1, hitdim}];
Y = {4.7308, -5.5899, -2.3665, -4.5566, 3.1896};
linearsystem = ColumnForm[Thread[A.X == Y]]
0.722 x[1] - 2.855 x[2] + 2.841 x[3] == 4.7308
1.315 x[1] - 0.229 x[2] - 1.267 x[3] == -5.5899
2.884 x[1] - 1.006 x[2] + 0.789 x[3] == -2.3665
0.192 x[1] + 2.081 x[2] - 2.131 x[3] == -4.5566
1.929 x[1] - 1.171 x[2] + 2.328 x[3] == 3.1896
```

Is the coefficient matrix of full rank?  
Determine whether this linear system has

→ no solution,  
→ many solutions,  
→ exactly one solution.  
Say how you came to your conclusion.  
If it has no solution, then give the best approximate solution

### □G.1.a.ii) A related linear system

Stay with the same matrix A as in part i):

```
MatrixForm[A]
{0.722, -2.855, 2.841}
{1.315, -0.229, -1.267}
{2.884, -1.006, 0.789}
{0.192, 2.081, -2.131}
{1.929, -1.171, 2.328}
```

Here's a new linear system based on the same matrix:

```
Y = {3.6, -2.9, 3.55, -2.037, 6.63};
linearsystem = ColumnForm[Thread[A.X == Y]]
0.722 x[1] - 2.855 x[2] + 2.841 x[3] == 3.6
1.315 x[1] - 0.229 x[2] - 1.267 x[3] == -2.9
2.884 x[1] - 1.006 x[2] + 0.789 x[3] == 3.55
0.192 x[1] + 2.081 x[2] - 2.131 x[3] == -2.037
1.929 x[1] - 1.171 x[2] + 2.328 x[3] == 6.63
```

Determine whether this linear system has  
→ no solution,  
→ many solutions,  
→ exactly one solution.  
Say how you came to your conclusion.  
If it has no solution, then give the best approximate solution

### □G.1.b) A new linear system

Here's a matrix:

```
hitdim = 6;
hangdim = 5;
Clear[x, y, k];
A = { {0.722, -2.855, 2.841, 0.341, -2.395, 0.888},
      {1.315, -0.229, -1.267, 0.582, 1.953, -0.011},
      {2.884, -1.006, 0.789, 0, -2.372, 2.222},
      {0.192, 2.081, -2.131, 5.216, -3.333, 0.799},
      {1.929, -1.171, 2.328, 0, 4.998, -2.007} };
MatrixForm[A]
```

```
{0.722, -2.855, 2.841, 0.341, -2.395, 0.888}
{1.315, -0.229, -1.267, 0.582, 1.953, -0.011}
{2.884, -1.006, 0.789, 0, -2.372, 2.222}
{0.192, 2.081, -2.131, 5.216, -3.333, 0.799}
{1.929, -1.171, 2.328, 0, 4.998, -2.007}
```

And a linear system based on this matrix:

```
X = Table[x[k], {k, 1, hitdim}];
Y = {-12.474, 12.6209, 0.1854, 4.2912, 13.2443};
ColumnForm[Thread[A.X == Y]]
```

```
0.722 x[1] - 2.855 x[2] + 2.841 x[3] + 0.341 x[4] - 2.395 x[5] + 0.888 x[6] ==
1.315 x[1] - 0.229 x[2] - 1.267 x[3] + 0.582 x[4] + 1.953 x[5] - 0.011 x[6] ==
2.884 x[1] - 1.006 x[2] + 0.789 x[3] - 2.372 x[5] + 2.222 x[6] == 0.1854
0.192 x[1] + 2.081 x[2] - 2.131 x[3] + 5.216 x[4] - 3.333 x[5] + 0.799 x[6] ==
1.929 x[1] - 1.171 x[2] + 2.328 x[3] + 4.998 x[5] - 2.007 x[6] == 13.2443
```

Is the coefficient matrix of full rank?  
Determine whether this linear system has  
→ no solution,  
→ many solutions,  
→ exactly one solution.  
Say how you came to your conclusion.  
If it has no solution, then give the best approximate solution.

### □G.1.c) A new linear system

Here's a matrix:

```
hitdim = 5;
hangdim = 5;
Clear[x, y, k];
```

```
A = { {1.0, 0, 2.0, 0, 0},
      {0, 1.0, 2.0, 0, 0},
      {0, -1.0, -2.0, 0, 0},
      {0, 0, 0, 1, 0},
      {0, -0, 0, 0, 1} }
```

```
MatrixForm[A]
```

```
{1. 0 2. 0 0}
{0 1. 2. 0 0}
{0 -1. -2. 0 0}
{0 0 0 1 0}
{0 0 0 0 1}
```

And a linear system based on this matrix:

```
Clear[x];
X = Table[x[k], {k, 1, hitdim}];
Y = {2.4, 3.6, -3.7, 0.0, 0.6};
ColumnForm[Thread[A.X == Y]]
1. x[1] + 2. x[3] == 2.4
1. x[2] + 2. x[3] == 3.6
-1. x[2] - 2. x[3] == -3.7
x[4] == 0
x[5] == 0.6
```

Is the coefficient matrix of full rank?  
Determine whether this linear system has  
→ no solution,  
→ many solutions,  
→ exactly one solution.  
Say how you came to your conclusion.  
If it has no solution, then give the best approximate solution.

## G.2) Nuts and bolts

### □G.2.a.i) Hitdim and hangdim

Here is a matrix for your viewing pleasure:

```
A = { {-0.488768, -0.269327, 1.77969, -0.468063},
      {1.61618, -0.675514, -0.236393, -1.55147},
      {1.11176, -0.701408, 1.26663, -0.991858},
      {1.93037, -1.51132, 0.521259, 1.85069},
      {1.13604, -1.2382, 0.532321, -0.412489},
      {-1.46225, -1.12184, -0.0584402, -1.6762} }
```

This matrix hits on hitdimD and hangs its hits in hangdimD.

Fill the blanks: For this matrix

hitdim = .....

hangdim = .....

The SVD hanger frame vectors for this matrix come from .....D.

The SVD aligner frame vectors for this matrix come from .....D

### □ Cols and Ytest

#### □ G.2.a.ii) Stretches

Here's a matrix A which hits on 6D and hangs in 6D:

```
A = {{-3.000, 0.410, -3.390, -0.105, 3.721, 0.000},
      {1.515, 2.948, -2.29, -5.372, -0.496, 1.241},
      {-1.224, -1.130, 3.648, 3.662, 2.022, -1.919},
      {3.071, -3.157, 0.876, 3.446, -1.178, 3.070},
      {-0.697, -0.964, -1.788, -0.824, 2.383, -0.685},
      {0.696, 0.227, 0.673, -1.312, 3.00, 2.985}};
```

MatrixForm[A]

```
{-3.    0.41  -3.39  -0.105  3.721  0
 1.515  2.948  -2.29  -5.372  -0.496  1.241
 -1.224 -1.13  3.648  3.662  2.022  -1.919
 3.071  -3.157 0.876  3.446  -1.178  3.07
 -0.697 -0.964 -1.788 -0.824  2.383  -0.685
 0.696  0.227  0.673  -1.312  3.    2.985}
```

Here is a calculation of the nonzero SVD stretch factors of A:

```
SingularValues[A][[2]]
```

```
{9.64334, 7.29762, 5.44097, 3.82598, 1.65112, 0.591664}
```

What is it about this setup that signals in advance that when you have this matrix A as the coefficient matrix of a linear system,  $AX = Y$ , then you are guaranteed that there is exactly one solution vector X for each given 6D vector Y?

#### □ G.2.a.iii) Rank

Here's a matrix A which hits on 4D and hangs in 5D:

```
A = {{-2.1, 3.8, -2.4, 3.3},
      {6.7, -1.5, 2.4, 3.0},
      {3.6, 3.4, -0.3, -0.7},
      {2, 1.3, 3.9, -3.2},
      {-3.3, 1.2, 3.5, -2.9}};
```

MatrixForm[A]

```
{-2.1  3.8  -2.4  3.3
 6.7  -1.5  2.4  3.
 3.6  3.4  -0.3  -0.7
 2    1.3  3.9  -3.2
 -3.3  1.2  3.5  -2.9}
```

Here is a calculation of the rank of A:

```
rank = Length[SingularValues[A][[2]]]
```

```
4
```

What is it about this setup that signals in advance that when you have this matrix A as the coefficient matrix of a linear system, then the linear system has exactly one solution or has no solution at all.

#### □ G.2.b.i) Determinant of coefficient matrix is 0

Here is a matrix which hits on 5D and hangs in 5D:

```
A = {{2.686, -0.666, -3.0464, -0.1568, -1.881},
      {3.068, 2.667, 1.596, -3.594, -2.986},
      {-1.918, 2.915, -3.093, -0.0829, -0.354},
      {1.15, 5.582, -1.497, -3.6769, -3.34},
      {-0.3820, -3.333, -4.6424, 3.4372, 1.105}};
```

MatrixForm[A]

```
{2.686  -0.666  -3.0464  -0.1568  -1.881
 3.068  2.667  1.596    -3.594  -2.986
 -1.918 2.915  -3.093  -0.0829  -0.354
 1.15   5.582  -1.497  -3.6769  -3.34
 -0.382 -3.333  -4.6424  3.4372  1.105}
```

And the corresponding linear system of equations:

```
hitdim = 5;
hangdim = 5;
```

```
Clear[x, y, k];
```

```
X = Table[x[k], {k, 1, hitdim}];
```

```
Y = Table[y[k], {k, 1, hangdim}];
```

```
linearsystem = ColumnForm[Thread[A.X == Y]]
```

```
2.686 x[1] - 0.666 x[2] - 3.0464 x[3] - 0.1568 x[4] - 1.881 x[5] == y[1]
3.068 x[1] + 2.667 x[2] + 1.596 x[3] - 3.594 x[4] - 2.986 x[5] == y[2]
-1.918 x[1] + 2.915 x[2] - 3.093 x[3] - 0.0829 x[4] - 0.354 x[5] == y[3]
1.15 x[1] + 5.582 x[2] - 1.497 x[3] - 3.6769 x[4] - 3.34 x[5] == y[4]
-0.382 x[1] - 3.333 x[2] - 4.6424 x[3] + 3.4372 x[4] + 1.105 x[5] == y[5]
```

Look at this calculation:

```
Det[A]
```

```
0
```

Use the result to comment on the following statement:

If for a certain given

```
Y = {y[1], y[2], y[3], y[4], y[5]},
```

this linear system has one solution then it has many more.

#### □ G.2.b.ii) Exhibiting a solution and exhibiting a non-solution

Stay with the same linear system as in part i) immediately above:

```
linearsystem
```

```
2.686 x[1] - 0.666 x[2] - 3.0464 x[3] - 0.1568 x[4] - 1.881 x[5] == y[1]
3.068 x[1] + 2.667 x[2] + 1.596 x[3] - 3.594 x[4] - 2.986 x[5] == y[2]
-1.918 x[1] + 2.915 x[2] - 3.093 x[3] - 0.0829 x[4] - 0.354 x[5] == y[3]
1.15 x[1] + 5.582 x[2] - 1.497 x[3] - 3.6769 x[4] - 3.34 x[5] == y[4]
-0.382 x[1] - 3.333 x[2] - 4.6424 x[3] + 3.4372 x[4] + 1.105 x[5] == y[5]
```

Come up with a specific example of a

```
Y = {y[1], y[2], y[3], y[4], y[5]}
```

for which the linear system has a solution.

Put answer here.

Click on the right for a tip

```
A. {0, 0, 0, 0, 0}
```

```
{0, 0, 0, 0, 0}
```

Then come up with a specific example of a

```
Y = {y[1], y[2], y[3], y[4], y[5]}
```

for which the linear system has no solution.

Click on the right for a tip

Try:

```
Y = Table[Random[Real, {-2, 2}], {j, 1, 5}]
```

```
{-0.331711, -0.328928, -0.352985, 1.82774, -0.502711}
```

#### □ G.2.c.i) More than one solution

Here is a matrix which hits on 5D and hangs in 3D:

```
A = {{2.686, -0.666, -3.0464, -0.1568, -1.881},
      {3.068, 2.667, 1.596, -3.594, -2.986},
      {-1.918, 2.915, -3.093, -0.0829, -0.354}};
```

MatrixForm[A]

```
{2.686  -0.666  -3.0464  -0.1568  -1.881
 3.068  2.667  1.596    -3.594  -2.986
 -1.918 2.915  -3.093  -0.0829  -0.354}
```

And the corresponding linear system of equations:

```
hitdim = 5;
hangdim = 3;
```

```
Clear[x, y, k];
```

```
X = Table[x[k], {k, 1, hitdim}];
```

```
Y = Table[y[k], {k, 1, hangdim}];
```

```
linearsystem = ColumnForm[Thread[A.X == Y]]
```

```
2.686 x[1] - 0.666 x[2] - 3.0464 x[3] - 0.1568 x[4] - 1.881 x[5] == y[1]
3.068 x[1] + 2.667 x[2] + 1.596 x[3] - 3.594 x[4] - 2.986 x[5] == y[2]
-1.918 x[1] + 2.915 x[2] - 3.093 x[3] - 0.0829 x[4] - 0.354 x[5] == y[3]
```

The rank of the coefficient matrix is:

```
rank = Length[SingularValues[A][[2]]]
```

```
3
```

You look at this and say that no matter what  $Y = \{y[1], y[2], y[3]\}$  this linear system has more than one solution. You are right. Explain why you are right.

#### □ G.2.c.ii) Coming up with the X of minimum norm that makes A.X as close to Y as it can be

Here is yet another matrix:

```
{2.686  -0.666  -3.046
 3.068  2.667  1.596
 -0.382  -3.333  -4.647
 1.035  -2.421  0.892
 3.817  4.732  -3.987}
```

MatrixForm[A]

```
{2.686  -0.666  -3.046
 3.068  2.667  1.596
 -0.382  -3.333  -4.647
 1.035  -2.421  0.892
 3.817  4.732  -3.987}
```

And a corresponding linear system of equations:

```
hitdim = 3; hangdim = 5;
```

```
Y = {-5.03, 3.023, -7.854, 7.780, -16.905};
```

```
Clear[x, k];
```

```
X = Table[x[k], {k, 1, hitdim}];
```

```
linearsystem = ColumnForm[Thread[A.X == Y]]
```

```
2.686 x[1] - 0.666 x[2] - 3.046 x[3] == -5.03
3.068 x[1] + 2.667 x[2] + 1.596 x[3] == 3.023
```

$$\begin{aligned} -0.382 x[1] - 3.333 x[2] - 4.647 x[3] &= -7.854 \\ 1.035 x[1] - 2.421 x[2] + 0.892 x[3] &= 7.78 \\ 3.817 x[1] + 4.732 x[2] - 3.987 x[3] &= -16.905 \end{aligned}$$

In effort to go after solutions of this system, you do an SVD analysis of the system and put:

```
rank = Length[SingularValues[A][[2]];
Clear[hangerframe, k];
hangerframe[k_] := SingularValues[A][[1]][[k]];
Ytest = Sum[(Y.hangerframe[k]) hangerframe[k], {k, 1, rank}];
{-5.15374, 2.78982, -7.95829, 8.06555, -16.7184}
```

And you compare:

```
Ytest
Y
{-5.15374, 2.78982, -7.95829, 8.06555, -16.7184}
{-5.03, 3.023, -7.854, 7.78, -16.905}
```

Seeing that

$$Y_{test} \neq Y,$$

you announce that the given linear system has no solution. Your job is to do the next best thing:

Come up with the  $X = \{x[1], x[2], x[3]\}$  of minimum norm that makes  $A.X$  as close to  $Y$  as it can be.

### G.2.c.ii) Plotting

Here's a matrix  $A = \begin{pmatrix} 1.5 & -1.1 \\ -0.9 & 0 \\ 0.9 & 0.7 \end{pmatrix}$ :

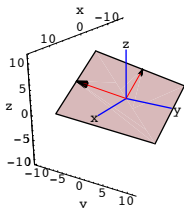
```
A = {{1.5, -1.1}, {-0.9, 0}, {0.9, 0.7}};
MatrixForm[A]
```

$$\begin{pmatrix} 1.5 & -1.1 \\ -0.9 & 0 \\ 0.9 & 0.7 \end{pmatrix}$$

Here's a piece of the plane you get when you hit all of 2D with A:

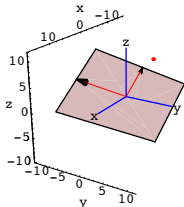
```
A = {{1.5, -1.1}, {-0.9, 0}, {0.9, 0.7}};
{h, k} = {6, 6};
Clear[x, y];
```

```
planepiece = ParametricPlot3D[A.{x, y}, {x, -h, h},
{y, -k, k}, PlotPoints -> {2, 2}, DisplayFunction -> Identity];
scaledAcolumns = {Arrow[A.{h, 0}, Tail -> {0, 0, 0}, VectorColor -> Red],
Arrow[A.{0, k}, Tail -> {0, 0, 0}, VectorColor -> Red]};
Show[planepiece, scaledAcolumns, ThreeAxes[10], PlotRange -> All,
Axes -> True, AxesLabel -> {"x", "y", "z"}, Boxed -> False,
ViewPoint -> CMView, DisplayFunction -> $DisplayFunction];
```



Throw in the plot of the point {1.1, 6.5, 9.1}:

```
point = {1.1, 6.5, 9.1};
pointplot = Graphics3D[{PointSize[0.02], Red, Point[point]}];
Show[planepiece, scaledAcolumns, pointplot, ThreeAxes[10], PlotRange -> All,
Axes -> True, AxesLabel -> {"x", "y", "z"}, Boxed -> False,
ViewPoint -> CMView, DisplayFunction -> $DisplayFunction];
```



The plotted point is not on the plane.

Your job is to come up with the point on the plane closest to the point plotted above.

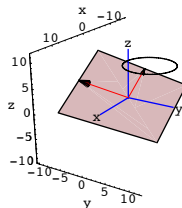
### G.2.c.ii) More plotting

Stay with the same plane as in part i) immediately above and look at this circle which is parallel to the xy-plane:

```
Clear[x, y, t];
{x[t_], y[t_], z[t_]} = {5 Cos[t] + 4, 5 Sin[t] + 7, 9};
{tlow, thigh} = {0, 2 Pi};
```

```
curveplot = ParametricPlot3D[{x[t], y[t], z[t]},
{t, tlow, thigh}, DisplayFunction -> Identity];
```

```
Show[planepiece, scaledAcolumns,
curveplot, ThreeAxes[10], PlotRange -> All,
Axes -> True, AxesLabel -> {"x", "y", "z"}, Boxed -> False,
ViewPoint -> CMView, DisplayFunction -> $DisplayFunction];
```



Your job is to plot the curve on the plane that is composed of the points on the plane closest to corresponding points  $\{x[t], y[t], z[t]\}$  on the circle. Do it.

## G.3) Phenoms asking to be explained

### G.3.a.i) Xapprox and PseudoInverse

Look at this:

```
hitdim = Random[Integer, {3, 7}];
hangdim = Random[Integer, {3, 7}];
A = Table[Random[Real, {-4, 4}], {i, 1, hangdim}, {j, 1, hitdim}];
MatrixForm[A]
```

$$\begin{pmatrix} 2.90709 & 0.361607 & -2.87421 \\ -0.286706 & 1.58047 & 2.30186 \\ -0.129801 & 0.0990801 & 2.37424 \\ 3.7394 & 1.28609 & 1.34483 \\ -3.87578 & -3.56021 & 1.3443 \\ -0.16523 & -0.469878 & -3.73426 \end{pmatrix}$$

Look at this:

```
Y = Table[Random[Real, {-4, 4}], {i, 1, hangdim}];
{-0.275415, -3.50181, -3.81202, 0.971708, 0.0691021, 1.50361}
```

Make Xapprox

```
Clear[alignerframe, stretch, hangerframe, k];
hangerframe[k_] := SingularValues[A][[1]][[k]]
```

```
stretch[k_] := SingularValues[A][[2]][[k]]
alignerframe[k_] := SingularValues[A][[3]][[k]];
rank = Length[SingularValues[A][[2]]];
Xapprox = Expand[Sum[(Y.hangerframe[k]) / stretch[k], {k, 1, rank}]] alignerframe[k]]
{0.342699, -0.676787, -0.447589}
PseudoInverse[A].Y
{0.342699, -0.676787, -0.447589}
```

Run all cells a couple of times and then describe what you see and explain why you see it.

### G.3.a.ii) Ytest and A.(PseudoInverse[A].Y)

Look at this matrix:

```
hitdim = Random[Integer, {3, 7}];
hangdim = Random[Integer, {3, 7}];
A = Table[Random[Real, {-4, 4}], {i, 1, hangdim}, {j, 1, hitdim}];
MatrixForm[A]
```

$$\begin{pmatrix} -2.71912 & -3.3899 & -1.05669 & -2.20968 \\ -0.299583 & -1.69176 & 3.07311 & 1.69124 \\ 1.32617 & -1.43116 & -2.21298 & -3.65359 \\ 1.20196 & -1.87094 & 0.442718 & 0.511637 \\ -2.32817 & -2.13668 & -3.28187 & 0.0134442 \end{pmatrix}$$

Look at this:

```
Y = Table[Random[Real, {-4, 4}], {i, 1, hangdim}];
{-2.51614, 0.891614, 0.64903, 2.50983, -3.79703}
```

Make Ytest:

```
rank = Length[SingularValues[A][[2]]];
Clear[hangerframe];
hangerframe[k_] := SingularValues[A][[1]][[k]];
Ytest = Sum[(Y.hangerframe[k]) hangerframe[k], {k, 1, rank}];
{-2.80635, 1.35538, 0.980126, 2.08134, -3.55038}
```

Compare:

```
A.(PseudoInverse[A].Y)
{-2.80635, 1.35538, 0.980126, 2.08134, -3.55038}
```

Rerun all cells a couple of times and then describe what you see and explain why you see it.

□G.3.a.iii)  $A.X = \{0,0, \dots, 0\}$  has a guaranteed solution

All clued-in matrix folks know that when you go with any coefficient matrix A, the linear system

$$A.X = \{0,0, \dots, 0\} \text{ has a solution.}$$

Write down one.

□G.3.a.iv) When you have a linear system with fewer linear equations than unknowns

All clued-in matrix folks know that when you have a linear system with fewer linear equations than unknown x's, then that linear system has a solution other than  $\{0,0,\dots,0\}$ . Explain how they know this

□G.3.a.v) Calculus Cal on when you have a linear system with the same number of equations as unknowns

Just as you are getting ready to leave your computer and go out and relax, you notice a repulsive figure enter the computer lab. He sits down next to you and your worst fears are confirmed. He is the infamous pest Calculus Cal. Cal is a regular nose-picker; he has foul breath and a strong hint of a garlic/ammonia body odor goes with him at all times.

To make matters worse, Cal thinks he knows everything and is very vocal about it. But his knowledge is almost always lacking an important consideration.

You try to sneak out but Cal catches your eye and says to you through a waft of foul breath.

Cal saw you doing part ii) immediately above. He says to you "When you have a linear system with the same number of equations as unknowns, you can count on an exact solution."

What do you say to Cal? And then tell him where to go.

## G.4) Build an inverse, build a pseudo inverse

□G.4.a.i) Build an inverse

Here's a square matrix A:

```
hitdim = Random[Integer, {4, 6}];
hangdim = hitdim
A = Table[Random[Real, {-2, 2}], {i, 1, hangdim}, {j, 1, hitdim}];
MatrixForm[A]
```

$$\begin{pmatrix} 0.140757 & -1.14714 & 0.359755 & 0.251277 \\ -1.01336 & -0.683693 & 1.51414 & 1.58819 \\ 1.70221 & -1.5772 & 1.34093 & -1.01279 \\ 0.637685 & 0.201439 & -0.914886 & -1.8487 \end{pmatrix}$$

Here is the full SVD of this matrix:

```
rank = Length[SingularValues[A][[2]]];
Clear[alignerframe, k];
alignerframe[k_] := SingularValues[A][[3, k]];
aligner = Table[alignerframe[k], {k, 1, rank}];
MatrixForm[aligner]

Clear[stretch];
stretch[k_] := SingularValues[A][[2, k]];
stretcher = DiagonalMatrix[Table[stretch[k], {k, 1, rank}]];
MatrixForm[stretcher]

Clear[hangerframe, k];
hangerframe[k_] := SingularValues[A][[1, k]];
hanger = Transpose[Table[hangerframe[k], {k, 1, rank}]];
MatrixForm[hanger]
```

$$\begin{pmatrix} 3.34617 & 0 & 0 & 0 \\ 0 & 2.98519 & 0 & 0 \\ 0 & 0 & 0.645049 & 0 \\ 0 & 0 & 0 & 0.473496 \end{pmatrix}$$

The inverse of A is:

```
MatrixForm[Inverse[A]]
```

$$\begin{pmatrix} -0.174613 & -1.00921 & 0.421242 & -1.1215 \\ -1.1166 & -0.0608928 & 0.166758 & -0.295437 \\ -0.894662 & 0.685383 & 0.386251 & 0.255595 \\ 0.260853 & -0.693931 & -0.0276762 & -1.08645 \end{pmatrix}$$

If Mathematica balks on this instruction, go back and run everything again.

Use the hanger matrix, the stretch factors, and the aligner calculated above to duplicate this calculation of the inverse of A

□G.4.b.i) Build a transpose

Here is a matrix A:

```
hitdim = Random[Integer, {4, 7}];
hangdim =
hitdim + (-1)^Random[Integer, {0, 1}] Random[Integer, {1, 3}];
A = Table[Random[Real, {-2, 2}], {i, 1, hangdim}, {j, 1, hitdim}];
MatrixForm[A]
```

$$\begin{pmatrix} -0.293976 & -0.157627 & 1.07839 & 1.40937 \\ 1.26022 & 1.51786 & 1.82348 & 1.30788 \\ -0.880539 & 0.664997 & -0.536279 & -0.943395 \end{pmatrix}$$

Here is the full SVD of this matrix:

```
rank = Length[SingularValues[A][[2]]];
3
Clear[alignerframe, k];
alignerframe[k_] := SingularValues[A][[3, k]];
aligner = Table[alignerframe[k], {k, 1, rank}];
MatrixForm[aligner]

Clear[stretch];
stretch[k_] := SingularValues[A][[2, k]];
stretcher = DiagonalMatrix[Table[stretch[k], {k, 1, rank}]];
MatrixForm[stretcher]

Clear[hangerframe, k];
hangerframe[k_] := SingularValues[A][[1, k]];
hanger = Transpose[Table[hangerframe[k], {k, 1, rank}]];
MatrixForm[hanger]
```

$$\begin{pmatrix} 3.35765 & 0 & 0 \\ 0 & 1.51481 & 0 \\ 0 & 0 & 1.00742 \end{pmatrix}$$

The transpose of A is:

```
MatrixForm[Transpose[A]]
```

$$\begin{pmatrix} -0.293976 & 1.26022 & -0.880539 \\ -0.157627 & 1.51786 & 0.664997 \\ 1.07839 & 1.82348 & -0.536279 \\ 1.40937 & 1.30788 & -0.943395 \end{pmatrix}$$

Use the hanger matrix, the stretch factors, and the aligner calculated above to duplicate this calculation of the transpose of A.

□G.4.b.ii) Rank of  $A^t = \text{rank of } A$

Go back and review what you did in part i).

Then say why it is very vivid that

$$\text{rank of } A^t = \text{rank of } A.$$

for this matrix A or any other matrix A.

□G.4.c.i) Build a pseudoinverse

Here is a matrix A:

```
hitdim = Random[Integer, {4, 5}];
hangdim =
hitdim + (-1)^Random[Integer, {0, 1}] Random[Integer, {1, 2}];
A = Table[Random[Real, {-2, 2}], {i, 1, hangdim}, {j, 1, hitdim}];
MatrixForm[A]
```

$$\begin{pmatrix} -1.86718 & -0.65131 & -0.0504143 & -0.531586 & -1.56939 \\ -1.07411 & 0.608653 & -1.5188 & -0.207075 & 0.724453 \\ -0.476461 & -1.67009 & -1.9131 & -1.11792 & 0.445147 \\ -1.07946 & -1.17332 & -0.635778 & 0.621671 & -0.387344 \\ 1.70722 & 0.699225 & -0.842051 & -1.44395 & 1.5744 \\ -0.649465 & 1.20836 & 1.08764 & 1.14379 & -1.57536 \\ -1.40029 & 0.606436 & -0.649137 & -0.299811 & 1.07617 \end{pmatrix}$$

Here is the full SVD of this matrix:

```
rank = Length[SingularValues[A][[2]]];
5
Clear[alignerframe, k];
alignerframe[k_] := SingularValues[A][[3, k]];
aligner = Table[alignerframe[k], {k, 1, rank}];
MatrixForm[aligner]
```

$$\begin{pmatrix} 0.325639 & -0.0903661 & -0.512715 & -0.447657 & 0.650015 \\ 0.726282 & 0.467849 & 0.468233 & 0.112098 & 0.147725 \\ -0.498142 & 0.792592 & -0.138448 & 0.0878862 & 0.311063 \\ 0.0211278 & -0.274998 & -0.102038 & 0.842667 & 0.451033 \\ -0.343339 & -0.262916 & 0.698785 & -0.263108 & 0.505437 \end{pmatrix}$$

```

Clear[stretch];
stretch[k_] := SingularValues[A][[2, k]];
stretcher = DiagonalMatrix[Table[stretch[k], {k, 1, rank}]];
MatrixForm[stretcher]

```

$$\begin{pmatrix} 4.34685 & 0 & 0 & 0 & 0 \\ 0 & 3.84015 & 0 & 0 & 0 \\ 0 & 0 & 2.42641 & 0 & 0 \\ 0 & 0 & 0 & 1.39275 & 0 \\ 0 & 0 & 0 & 0 & 0.850959 \end{pmatrix}$$

```

Clear[hangerframe, k];
hangerframe[k_] := SingularValues[A][[1, k]];
hanger = Transpose[Table[hangerframe[k], {k, 1, rank}]];
MatrixForm[hanger]

```

$$\begin{pmatrix} -0.300328 & -0.514523 & -0.046993 & -0.725898 & 0.145393 \\ 0.215683 & -0.292356 & 0.591368 & 0.0841221 & -0.507554 \\ 0.406372 & -0.542356 & -0.321988 & -0.0695338 & -0.252698 \\ -0.103429 & -0.421378 & -0.152516 & 0.512571 & -0.146319 \\ 0.596814 & 0.323814 & 0.0754928 & -0.414257 & -0.214738 \\ -0.555429 & 0.129787 & 0.30546 & -0.146261 & -0.507509 \\ 0.150862 & -0.237455 & 0.649718 & 0.0736898 & 0.57646 \end{pmatrix}$$

The pseudoinverse of A is:

```
MatrixForm[PseudoInverse[A]]
```

$$\begin{pmatrix} -0.179836 & 0.0455173 & 0.0948742 & 0.0106805 & 0.17081 & 0.122774 & -0. \\ 0.0266153 & 0.293276 & -0.087898 & -0.155006 & 0.199845 & 0.31282 & -0. \\ 0.147944 & -0.517783 & -0.298104 & -0.188184 & -0.181207 & -0.342128 & 0. \\ -0.469943 & 0.198501 & -0.0332834 & 0.348193 & -0.233522 & 0.140476 & -0. \\ -0.219447 & -0.177406 & -0.173985 & 0.0278564 & -0.15032 & -0.387711 & 0.4 \end{pmatrix}$$

Use the hanger matrix, the stretch factors, and the aligner calculated above to duplicate this calculation of the pseudoinverse of A.

#### □G.4.c.ii) Rank of PseudoInverse[A] = rank of A

Here's a new random matrix:

```

hitdim = Random[Integer, {4, 5}];
hangdim = hitdim + Random[Integer, {1, 2}];
A = Table[Random[Real, {-2, 2}], {i, 1, hangdim}, {j, 1, hitdim}];
MatrixForm[A]

```

$$\begin{pmatrix} 0.27653 & -0.736038 & -1.18189 & -1.36898 \\ -0.644008 & -1.56272 & 1.45389 & 0.00935296 \\ 1.74334 & -1.26994 & -1.24534 & -1.1486 \\ 1.18729 & -0.844342 & 1.40413 & -0.35696 \\ -1.90035 & 0.0118692 & 0.979486 & -0.95667 \end{pmatrix}$$

The pseudo inverse of A is:

```
MatrixForm[PseudoInverse[A]]
```

$$\begin{pmatrix} -0.0818287 & -0.239153 & 0.0987079 & 0.363538 & -0.139399 \\ -0.085227 & -0.610202 & -0.163089 & 0.267588 & 0.211956 \\ -0.148879 & -0.0779602 & -0.110998 & 0.443551 & 0.180047 \\ -0.23012 & 0.370979 & -0.0999108 & -0.34227 & -0.464703 \end{pmatrix}$$

The rank of A is:

```
rankA = Length[SingularValues[A][[2]]]
4
```

The rank of PseudoInverse[A] is:

```
rankpseudoinverseA = Length[SingularValues[A][[2]]]
4
```

Go back and review what you did in part i).

Then say why it is very vivid that

rank of PseudoInverse[A] = rank of A  
for this matrix A and all other matrices A.

### G.5) Agree or disagree

Here are the basic facts:

- The rank of a matrix is the number of non-zero SVD stretch factors for A.

This is also the number of hangerframe vectors for A.

- A given matrix A is of full rank if the rank of A = hitdim.

- Given a linear system  $A.X = Y$ , put

$$Y_{test} = \sum_{k=1}^{\text{rank}} (Y.\text{hangerframe}[k]) \text{hangerframe}[k].$$

- If  $Y_{test} = Y$ , and A is of full rank, then there is exactly one solution of  $A.X = Y$  and that solution is

$$X_{sol} = \sum_{k=1}^{\text{rank}} \left( \frac{Y.\text{hangerframe}[k]}{\text{stretch}[k]} \right) \text{alignerframe}[k] \quad (\text{exactly determined}).$$

- If  $Y_{test} = Y$ , and A is of not of full rank, then

$$X_{sol} = \sum_{k=1}^{\text{rank}} \left( \frac{Y.\text{hangerframe}[k]}{\text{stretch}[k]} \right) \text{alignerframe}[k]$$

- is just one of infinitely many solutions  $A.X = Y$  (under determined).

- If  $Y_{test} \neq Y$ , then

$$X_{sol} = \sum_{k=1}^{\text{rank}} \left( \frac{Y.\text{hangerframe}[k]}{\text{stretch}[k]} \right) \text{alignerframe}[k]$$

- is the solution of  $A.X = Y$  of minimum norm.

- If  $Y_{test} \neq Y$ , then the linear system  $A.X = Y$  has no solutions at all (over determined).

- If A is invertible, then

$$\sum_{k=1}^{\text{rank}} \left( \frac{Y.\text{hangerframe}[k]}{\text{stretch}[k]} \right) \text{alignerframe}[k] = A^{-1}.Y.$$

- For any matrix A (invertible or not)

$$\sum_{k=1}^{\text{rank}} \left( \frac{Y.\text{hangerframe}[k]}{\text{stretch}[k]} \right) \text{alignerframe}[k] = \text{PseudoInverse}[A].Y.$$

- For any matrix A (invertible or not)

$$Y_{test} = \sum_{k=1}^{\text{rank}} (Y.\text{hangerframe}[k]) \text{hangerframe}[k]$$

- is as close to Y as any other possible hit with A.

- For any matrix A (invertible or not)

$$A. \sum_{k=1}^{\text{rank}} \left( \frac{Y.\text{hangerframe}[k]}{\text{stretch}[k]} \right) \text{alignerframe}[k] = Y_{test}$$

- is as close to Y as any other possible hit with A.

Indicate whether you agree or disagree with each of the following statements.

You are not asked to explain your responses, but feel free to throw in an explanation if you want to.

- G.5.a.i) Using  $Y_{test} = \sum_{k=1}^{\text{rank}} (Y.\text{hangerframe}[k]) \text{hangerframe}[k]$

You are given a matrix A and use it to make a linear system  $A.X = Y$ . You check the rank of A and learn A is of full rank. Then you do your SVD analysis of A and calculate

$$Y_{test} = \sum_{k=1}^{\text{rank}} (Y.\text{hangerframe}[k]) \text{hangerframe}[k]$$

and then you hit learn that

$$Y_{test} = Y.$$

You now have enough information to announce that there is exactly one solution of the linear system  $A.X = Y$ .

□G.5.a.ii) Using  $Y_{test} = \sum_{k=1}^{rank} (Y_{hangerframe[k]}) hangerframe[k]$

You are given a matrix A and use it to make a linear system  $A.X = Y$ . You check the rank of A and learn A is not of full rank. Then you do your SVD analysis of A and calculate

$$Y_{test} = \sum_{k=1}^{rank} (Y_{hangerframe[k]}) hangerframe[k]$$

and then you hit learn that

$$Y_{test} = Y.$$

You now have enough information to announce that there is exactly one solution of the linear system  $A.X = Y$ .

□G.5.a.iii) Using  $Y_{test} = \sum_{k=1}^{rank} (Y_{hangerframe[k]}) hangerframe[k]$

You are given a matrix A and use it to make a linear system  $A.X = Y$ . Then you do your SVD analysis of A and calculate

$$Y_{test} = \sum_{k=1}^{rank} (Y_{hangerframe[k]}) hangerframe[k]$$

and learn that

$$Y_{test} \neq Y.$$

You now have enough information to announce that there are no solutions of the linear system  $A.X = Y$ .

□G.5.d.vi) Using  $Y_{test} = \sum_{k=1}^{rank} (Y_{hangerframe[k]}) hangerframe[k]$

You are a given square A (so that  $hitdim = hangdim$ ) and use it to make a linear system  $A.X = Y$ .

You do your SVD analysis of A and calculate

$$Y_{test} = \sum_{k=1}^{rank} (Y_{hangerframe[k]}) hangerframe[k].$$

And then you find that

$$Y_{test} \neq Y.$$

This tells you that

$$Det[A] = 0.$$

□G.5.b.i) Using rank

You are given a matrix A and use it to make a linear system  $A.X = Y$ . You check the rank of A and learn A is of full rank.

You now have enough information to announce that there is exactly one solution of the linear system  $A.X = Y$ .

□G.5.b.ii) Xpprox and Ytest

You are given a matrix A and use it to make a linear system  $A.X = Y$ . You do your SVD analysis of A and calculate

$$X_{approx} = \sum_{k=1}^{rank} \left( \frac{Y_{hangerframe[k]}}{stretch[k]} \right) alignerframe[k]$$

and

$$Y_{test} = \sum_{k=1}^{rank} (Y_{hangerframe[k]}) hangerframe[k]$$

At this stage, you are guaranteed that

$$A.X_{approx} = Y_{test}.$$

□G.5.b.iii) Using  $X_{approx} = \sum_{k=1}^{rank} \left( \frac{Y_{hangerframe[k]}}{stretch[k]} \right) alignerframe[k]$

You are given a matrix A and use it to make a linear system  $A.X = Y$ . You check the rank of A and learn A is not of full rank. Then you do your SVD analysis of A and calculate

$$X_{approx} = \sum_{k=1}^{rank} \left( \frac{Y_{hangerframe[k]}}{stretch[k]} \right) alignerframe[k]$$

and then you hit Xpprox with A and learn that

$$A.X_{approx} = Y.$$

You now have enough information to announce that the linear system  $A.X = Y$  has lots of solutions.

□G.5.b.iv) Using  $X_{approx} = \sum_{k=1}^{rank} \left( \frac{Y_{hangerframe[k]}}{stretch[k]} \right) alignerframe[k]$

You are given a matrix A and use it to make a linear system  $A.X = Y$ . You check the rank of A and learn A is not of full rank. Then you do your SVD analysis of A and calculate

$$X_{approx} = \sum_{k=1}^{rank} \left( \frac{Y_{hangerframe[k]}}{stretch[k]} \right) alignerframe[k]$$

and then you hit Xpprox with A and learn that

$$A.X_{approx} \neq Y.$$

You now have enough information to announce that the linear system  $A.X = Y$  no solution at all.

□G.5.c.i) Full rank

You are given a linear system

$$A.X = Y.$$

You check the rank of A and find that A is of full rank.

This guarantees that the given linear system has exactly one solution (and no more).

□G.5.c.ii) Rank deficient

You are given a linear system

$$A.X = Y.$$

You check the rank of A and find that A is not of full rank.

This guarantees that the given linear system has no solution at all.

□G.5.c.iii) Rows and rank

Saying that a matrix A is of full rank is the same as saying that the rank of A is equal to the number of horizontal rows of A.

□G.5.c.iv) Columns and rank

Saying that a matrix A is of full rank is the same as saying that the rank of A is equal to the number of vertical columns of A.

□G.5.c.v)  $hitdim > hangdim$

You are given a linear system

$$A.X = Y.$$

with  $hitdim > hangdim$

It is automatic that the coefficient matrix A is not of full rank.

□G.5.c.vi) Full rank

You are given a linear system

$$A.X = Y.$$

You check the rank of A and find that A is of full rank. This tells you that the system has exactly one solution or no solutions at all.

□G.5.c.vii) Same number of equations as variable  $x[i]$ 's

You are given a linear system

$$A.X = Y$$

with the same number of equations as variable  $x[i]$ 's

You check the rank of A and find that A is of full rank. This tells you that the system has exactly one solution.

□G.5.c.viii) Rank deficient

You are given a linear system

$$A.X = Y.$$

You check the rank of A and find that A is not of full rank.

This tells you that the system either has no solution or has many solutions.

□G.5.c.ix) Rank and  $hangdim$

Given a matrix hitting on  $hitdimD$  and hanging in  $hangdimD$ , you can be sure that that the rank of A cannot be bigger than  $hangdim$ .

□G.5.c.x) Rank and  $hitdim$

Given a matrix hitting on  $hitdimD$  and hanging in  $hangdimD$ , you can be sure that that the rank of A cannot be bigger than  $hitdim$ .

□G.5.d.i) Many solutions?

You are a given matrix A that hits on 8D and hangs in 5D. If the rank of A is 5, then no matter what Y in 5D you go with, you are guaranteed that the corresponding linear system  $A.X = Y$  has many solutions.

□G.5.d.ii) Rank of A is the same as the rank of  $A^t$ ?

Given any matrix A, you are guaranteed that the rank of A is the same as the rank of  $A^t$ .

□G.5.d.iii)  $Det[A] \neq 0$

Given any square matrix A (so that  $hitdim = hangdim$ ), saying that A is invertible is the same as saying that  $Det[A] \neq 0$ .

□G.5.d.iv) Rank >  $hangdim$ ?

It is possible to encounter a matrix A that hangs its hits in 4D and whose rank is 6.

□G.5.d.v) Stretches

It is possible to encounter a matrix that hits on 6D and hangs its hits in 4D and has five non-zero SVD stretch factors.

□G.5.d.vi) Hanger frame

It is possible to encounter a matrix that hangs its hits in 4D and whose SVD hanger frame consists of five mutually perpendicular unit vectors.

□G.5.d.v) Stretches

It is possible to encounter a matrix that hits on 6D and hangs its hits in 4D and has exactly four non-zero SVD stretch factors.

□G.5.d.vi) Hanger frame

It is possible to encounter a matrix that hangs its hits in 7D whose SVD hanger frame consists of six mutually perpendicular unit vectors.

□G.5.d.vii)  $hitdim < hangdim$

You are given a linear system

$$A.X = Y.$$

with  $hitdim < hangdim$

It is automatic that the coefficient matrix A is of full rank.

□G.5.d.viii) Hitting on 5D and hanging in 8D

You are a given matrix A that hits on 5D and hangs in 8D. Regardless of the rank of A, there are guaranteed to be many Y's in 8D for which the linear system  $A.X = Y$  has no solution.

□G.5.e.i) Stretches and determinants

Here is a 3D matrix shown with its nonzero SVD stretch factors:

$$A = \begin{pmatrix} 0.5 & 0 & 1.1 \\ 1. & -0.2 & 1.5 \\ 3.5 & -0.8 & 4.9 \end{pmatrix};$$

SingularValues [A] [ [ 2 ] ]

$$\{ 6.44761, 0.27987 \}$$

This is enough to tell you that the rank of A is 2 and that the determinant of A is 0



□G.5.e.ii) Stretches and determinants

Here is a 4D matrix shown with its SVD stretch factors:

$$A = \begin{pmatrix} 1. & -0.4 & 0.8 & -1. \\ 2. & -0.4 & 3. & 0 \\ -0.5 & 0 & -1.1 & -0.5 \\ 3.5 & -0.8 & 4.9 & -0.5 \end{pmatrix};$$

**SingularValues [A] [ [2]]**  
 {7.30167, 1.22704}

This is enough to tell you that the rank of A is 2 and that the determinant of A is:

**7.30167 1.22704**  
 8.95944

□G.5.e.iii) Four linear equations in four variables

Here is a linear system consisting of four equations involving four variables x[1],x[2],x[3] and x[4]:

$$\begin{aligned} a_1 x[1] + b_1 x[2] + c_1 x[3] + d_1 x[4] &= y[1] \\ a_2 x[1] + b_2 x[2] + c_2 x[3] + d_2 x[4] &= y[2] \\ a_3 x[1] + b_3 x[2] + c_3 x[3] + d_3 x[4] &= y[3] \\ a_4 x[1] + b_4 x[2] + c_4 x[3] + d_4 x[4] &= y[4] \end{aligned}$$

Given specific values for all the constants

a<sub>1</sub>, a<sub>2</sub>, a<sub>3</sub>, a<sub>4</sub>, b<sub>1</sub>, b<sub>2</sub>, b<sub>3</sub>, b<sub>4</sub>, c<sub>1</sub>, c<sub>2</sub>, c<sub>3</sub>, c<sub>4</sub>, d<sub>1</sub>, d<sub>2</sub>, d<sub>3</sub>, d<sub>4</sub> and y[1], y[2], y[3], y[4],

you are guaranteed that the resulting linear system has exactly one solution for each of x[1], x[2], x[3], x[4] because you have four equations in four unknowns.

□G.5.e.iv) Det[A] ≠ 0

You are given a square matrix A (so that hitdim = hangdim) A and use it to make a linear system A.X = Y. You calculate Det[A] and find that Det[A] ≠ 0. You know have enough information to announce that the linear system has exactly one solution.

□G.5.f.i) Approximate solution

You are given a matrix A and use it to make a linear system A.X = Y. You do your SVD analysis of A and calculate

$$X_{\text{approx}} = \sum_{k=1}^{\text{rank}} \left( \frac{Y_{\text{hangerframe}[k]}}{\text{stretch}[k]} \right) \text{alignerframe}[k]$$

and then you announce that Xapprox the best approximate solution you can get.

□G.5.f.ii) PseudoInverse

You are given a matrix A that hits on 10D and hangs in 13D. Take a Y in 13D, do your SVD analysis of A and you calculate

$$\sum_{k=1}^{\text{rank}} \left( \frac{Y_{\text{hangerframe}[k]}}{\text{stretch}[k]} \right) \text{alignerframe}[k]$$

and then you announce that this is the same as PseudoInverse[A].Y.

□G.5.f.iii) PseudoInverse

You are given an invertible matrix A that hits on 9D and hangs in 9D. You take a Y in 9D, do your SVD analysis of A and you calculate

$$\sum_{k=1}^{\text{rank}} \left( \frac{Y_{\text{hangerframe}[k]}}{\text{stretch}[k]} \right) \text{alignerframe}[k].$$

And then you announce that this is the same as PseudoInverse[A].Y.

□G.5.f.iv) Inverse and PseudoInverse

If A is an invertible matrix, then A<sup>-1</sup> = PseudoInverse[A].

□G.5.f.v) PseudoInverse

A square matrix has an inverse if its determinant is not 0, but all other matrices have a pseudo inverse.

□G.5.f.vi) Ytest and invertibility

You are a given square A (so that hitdim = hangdim) and use it to make a linear system A.X = Y.

You do your SVD analysis of A and calculate

$$Y_{\text{test}} = \sum_{k=1}^{\text{rank}} (Y_{\text{hangerframe}[k]} \text{hangerframe}[k]).$$

And then you find that

Y<sub>test</sub> ≠ Y.

This tells you that A is not invertible.

□G.5.f.vii) Solution of smallest norm

You are given a matrix A and use it to make a linear system A.X = Y. You check the rank of A and learn A is not of full rank. Then you do your SVD analysis of A and calculate

$$X_{\text{approx}} = \sum_{k=1}^{\text{rank}} \left( \frac{Y_{\text{hangerframe}[k]}}{\text{stretch}[k]} \right) \text{alignerframe}[k]$$

and then you hit Xapprox with A and learn that

A.X<sub>approx</sub> = Y.

You now have enough information to announce that the linear system A.X = Y is underdetermined (lots of solutions) and to announce that if X<sub>other</sub> is any solution, then  $\sqrt{X_{\text{test}} \cdot X_{\text{test}}} < \sqrt{X_{\text{other}} \cdot X_{\text{other}}}$

G.6) Measurement errors lead to the need for approximate solutions

□G.6.a) Measurement errors lead to the need for approximate solutions

You have a summer job with C&M Engineering Co in Latrobe, Pennsylvania, home of Rolling Rock.

Today you and your gang are out in the field measuring the heights of four nearby mountains in the Allegheny range.

The gang sets up the transit instrument and goes to work.

You say: "I'll call x[k] the height of the kth mountain."

The gang makes the following measurements (in feet) from afar:

x[1] = 1268.1,  
 x[2] = 1317.4,  
 x[3] = 1723.7,  
 x[4] = 1003.1.

Just to check these measurements, the gang claims to the top of mountain 4 and measures the how much higher than mountain 4 the other mountains are. This leads to the following new measurements:

x[1] - x[4] = 264.1,  
 x[2] - x[4] = 315.0,  
 x[3] - x[4] = 719.8.

You put these equations in matrix form:

$$A = \begin{pmatrix} 1.0 & 0 & 0 & 0 \\ 0 & 1.0 & 0 & 0 \\ 0 & 0 & 1.0 & 0 \\ 0 & 0 & 0 & 1.0 \\ 1.0 & 0 & 0 & -1.0 \\ 0 & 1.0 & 0 & -1.0 \\ 0 & 0 & 1.0 & -1.0 \end{pmatrix};$$

**MatrixForm [A]**

$$\begin{pmatrix} 1. & 0 & 0 & 0 \\ 0 & 1. & 0 & 0 \\ 0 & 0 & 1. & 0 \\ 0 & 0 & 0 & 1. \\ 1. & 0 & 0 & -1. \\ 0 & 1. & 0 & -1. \\ 0 & 0 & 1. & -1. \end{pmatrix}$$

**Y = {1268.1, 1317.4, 1723.7, 1003.1, 264.1, 315.0, 719.8};**

**hitdim = 4; hangdim = 7;**  
**Clear[x, k];**  
**X = Table[x[k], {k, 1, hitdim}];**

**ColumnForm [Thread [A.X == Y]]**

1. x[1] == 1268.1  
 1. x[2] == 1317.4  
 1. x[3] == 1723.7  
 1. x[4] == 1003.1  
 1. x[1] - 1. x[4] == 264.1  
 1. x[2] - 1. x[4] == 315.  
 1. x[3] - 1. x[4] == 719.8

You can see that this system is has no solution because the fifth equation says x[1] - x[4] = 264.1,

But the first equation combined with the fourth equation says

x[1] - x[4] is:

**1268.1 - 1003.1**  
 265.

This tells you that there is no hope using your measurements to find exact solutions for the four heights

{x[1], x[2], x[3], x[4]}.

But you have no way of knowing which of the measurements are most accurate or which are least accurate.

So you decide to use all the measurements and you look at the linear system again:

**linearsystem**  
 2.686 x[1] - 0.666 x[2] - 3.046 x[3] == -5.03  
 3.068 x[1] + 2.667 x[2] + 1.596 x[3] == 3.023  
 -0.382 x[1] - 3.333 x[2] - 4.647 x[3] == -7.854  
 1.035 x[1] - 2.421 x[2] + 0.892 x[3] == 7.78  
 3.817 x[1] + 4.732 x[2] - 3.987 x[3] == -16.905

And you say I'll find the  $x[1]$ ,  $x[2]$ ,  $x[3]$  and  $x[4]$  that make the vector defined by the left sides as close as it can be to the vector defined by the right hand sides.  
Do it.

If you are using a matrix, be sure to hang  $N[]$  around it.  
All matrices should be entered in decimal form.

### □G.6.a.ii) On a different day

On a different day, you and your gang are out in the field measuring the heights of six nearby mountains in the Allegheny range. Again the gang sets up the transit instrument and goes to work. You say: "I'll call  $x[k]$  the height of the  $k$ th mountain."

The gang makes the following measurements (in feet) from afar:

```
x[1] = 968.3
x[2] = 1110.7
x[3] = 1346.8
x[4] = 1452.1
x[5] = 1623.2
x[6] = 1856.8
```

Then the gang claims to the top of mountain 1 and measures the how much higher than mountain 1 the other mountains are. This leads to the following new measurements:

```
x[2] - x[1] = 145.7
x[3] - x[1] = 380.1
x[4] - x[1] = 483.9
x[5] - x[1] = 658.4
x[6] - x[1] = 882.5
```

Using all the measurements made above, give your best shot at the values for

$x[1]$ ,  $x[2]$ ,  $x[3]$ ,  $x[4]$ ,  $x[5]$  and  $x[6]$

that you are going to report.

Give a short justification for your answer.

If you are using a matrix, be sure to hang  $N[]$  around it.  
All matrices should be entered in decimal form.

## G.7) Linear systems in least square line fit.

### Linear systems in Fourier fit with sine and cosine waves: Trig polynomials.

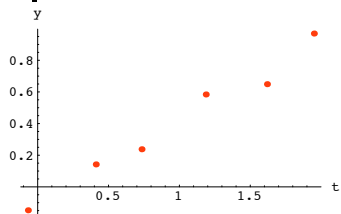
### Linear systems in Polynomial fit: Interpolating polynomials

### □G.7.a) Linear systems in least square line fit

Here is a plot of some points:

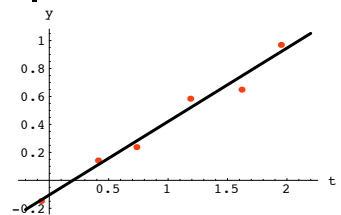
```
points = {{-0.069, -0.148}, {0.410, 0.142}, {0.733, 0.2377},
          {1.186, 0.584}, {1.618, 0.649}, {1.949, 0.969}};
```

```
pointplot = ListPlot[points, PlotStyle -> {PointSize[0.02], GreenishUmbur},
                    AxesOrigin -> {0, 0}, AxesLabel -> {"t", "y"}];
```



As you can see, you can't put a straight line through these points, but you can do the next best thing:

```
Clear[fitter, t];
fitter[t_] = Fit[points, {1, t}, t]
-0.104728 + 0.525496 t
fitplot = Plot[fitter[t], {t, -0.2, 2.2}, PlotStyle ->
             {{Thickness[0.01], Black}}, DisplayFunction -> Identity];
Show[pointplot, fitplot, DisplayFunction -> $DisplayFunction];
```



That's nice.

If you want to know how to come up with the formula for that line, matrix methods can tell you a lot.

Go with this cleared line function:

```
Clear[x, t, line];
line[t_] = x[1] + t x[2]
x[1] + t x[2]
```

Look at the points again:

```
points = {{-0.069, -0.148}, {0.41, 0.142}, {0.733, 0.2377},
          {1.186, 0.584}, {1.618, 0.649}, {1.949, 0.969}};
```

To try to fit a line through all these points you take the first slot of each point, substitute it into the cleared line function and set the result equal to the second slot of the same point:

```
ColumnForm[Table[
  line[points[[k, 1]]] == points[[k, 2]], {k, 1, Length[points]}]]
x[1] - 0.069 x[2] == -0.148
x[1] + 0.41 x[2] == 0.142
x[1] + 0.733 x[2] == 0.2377
x[1] + 1.186 x[2] == 0.584
x[1] + 1.618 x[2] == 0.649
x[1] + 1.949 x[2] == 0.969
```

This linear system has no solution.

Why? Because the points do not line up in a perfect straight line.

So you do the next best thing: You come up with the  $x[1]$  and  $x[2]$  that make the 6D vector defined by the left hand side as close to the 6D vector defined by the right hand side as it can be.

Use matrix methods to do this and compare your resulting  $x[1]$  and  $x[2]$  with:

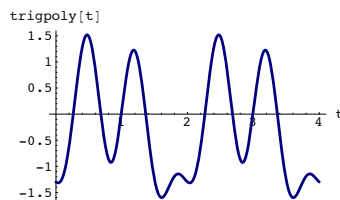
```
fitter[t]
-0.104728 + 0.525496 t
```

### □G.7.b.i) Linear systems in Fourier fit with sine and cosine waves

Trig polynomials are combinations of Sine and Cosine waves.

Here are some samples of trig polynomials and their plots:

```
high = Random[Integer, {1, 5}];
Clear[trigpoly];
trigpoly[t_] =
  Random[Real, {-1, 1}] + Sum[Random[Real, {-1, 1}] Cos[k π t],
    {k, 1, high}] +
  Sum[Random[Real, {-1, 1}] Sin[k π t],
    {k, 1, high}];
Plot[trigpoly[t], {t, 0, 4}, PlotStyle -> {{Navy, Thickness[0.01]}},
     AspectRatio -> 1/GoldenRatio, AxesLabel -> {"t", "trigpoly[t]"}];
-0.253394 - 0.669497 Cos[π t] - 0.360352 Cos[2 π t] - 0.0164205 Cos[3 π t] +
0.608341 Sin[π t] + 0.698522 Sin[2 π t] - 0.769407 Sin[3 π t]
```



Rerun at least five times noting the formulas and the shapes.

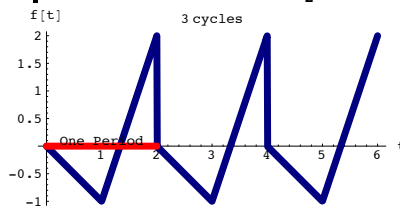
### Historical Marker:

At the December, 1807 meeting of the French Academy of Sciences, mathematician and engineer Joseph Fourier announced that any periodic function can be approximated by sums of sine and cosine waves (trig polynomials). Those in attendance thought he was slightly off his rocker and laughed at him, but eventually they were all forced to change their minds because time has confirmed that Fourier's idea was one of the great scientific break-throughs of all time. In this problem, you get the opportunity to get your hands on Fourier's idea in action.

Here's how it goes.

You start with any periodic function  $f[t]$  and identify its period  $L$ :

```
Clear[f, t];
f[t_] = t - 2 Floor[t/2] - 4 Abs[0.5 t - Round[0.5 t]];
L = 2;
cycles = 3;
Plot[f[t], {t, 0, cycles L}, PlotStyle -> {{Thickness[0.02], Navy}},
     AxesLabel -> {"t", "f[t]"}, PlotLabel -> "cycles" cycles,
     Epilog -> {{Red, Thickness[0.02], Line[{{0, 0}, {L, 0}]},
                {Text["One Period", {L/2, 0.1}]}}, AspectRatio -> Automatic];
```



Plot one period of  $f[t]$  and throw in an odd number of equally spaced points on the plot of  $f[t]$ :

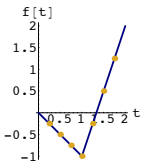
```
periodplot = Plot[f[t], {t, 0, L}, PlotStyle -> {{Thickness[0.02], Navy}},
                  AxesLabel -> {"t", "f[t]"}, DisplayFunction -> Identity];
```



```

jump =  $\frac{L}{8}$ ;
fpoints = Table[N[{t, f[t]}], {t, jump, L - jump, jump}];
pointplot =
ListPlot[fpoints, PlotStyle -> {Goldenrod, PointSize[0.06]},
DisplayFunction -> Identity];
setup =
Show[periodplot, pointplot, DisplayFunction -> $DisplayFunction];

```



Define a trig polynomial whose constituent Cosine and Sine waves are fixed to match the period of f[t] and has one undetermined coefficient x[k] for each point

```

b =  $\frac{1}{2}$  (Length[fpoints] - 1);
Clear[trigpoly, x, t, k];
trigpoly[t_] = x[1] + Sum[x[k] Cos[ $\frac{(k-1)(2\pi)t}{L}$ ], {k, 2, b+1}] +
Sum[x[k] Sin[ $\frac{(k-b-1)(2\pi)t}{L}$ ], {k, b+2, Length[fpoints]}];
x[1] + Cos[ $\pi t$ ] x[2] + Cos[ $2\pi t$ ] x[3] + Cos[ $3\pi t$ ] x[4] +
Sin[ $\pi t$ ] x[5] + Sin[ $2\pi t$ ] x[6] + Sin[ $3\pi t$ ] x[7]

```

To get the approximation, you go after the x[k]'s that make the plot trigpoly[t] go right through the plotted points. This results in the following linear system:

```

linearsystem =
ColumnForm[Table[trigpoly[fpoints[[k, 1]]] == fpoints[[k, 2]],
{k, 1, Length[fpoints]}]]
x[1] + 0.707107 x[2] - 0.707107 x[4] + 0.707107 x[5] + 1. x[6] + 0.707107 x[7]
x[1] - 1. x[3] + 1. x[5] - 1. x[7] == -0.5
x[1] - 0.707107 x[2] + 0.707107 x[4] + 0.707107 x[5] - 1. x[6] + 0.707107 x[7]
x[1] - 1. x[2] + 1. x[3] - 1. x[4] == -1.

```

```

x[1] - 0.707107 x[2] + 0.707107 x[4] - 0.707107 x[5] + 1. x[6] - 0.707107 x[7]
x[1] - 1. x[3] - 1. x[5] + 1. x[7] == 0.5
x[1] + 0.707107 x[2] - 0.707107 x[4] - 0.707107 x[5] - 1. x[6] - 0.707107 x[7]

```

Pluck off the coefficient matrix A:

```

linearformulas =
Table[trigpoly[fpoints[[k, 1]]], {k, 1, Length[fpoints]}];
variables = Table[x[k], {k, 1, Length[fpoints]}];
A = Outer[D, {linearformulas}, variables][[1]];
MatrixForm[A]

```

$$\begin{pmatrix}
1 & 0.707107 & 0 & -0.707107 & 0.707107 & 1. & 0.707107 \\
1 & 0 & -1. & 0 & 1. & 0 & -1. \\
1 & -0.707107 & 0 & 0.707107 & 0.707107 & -1. & 0.707107 \\
1 & -1. & 1. & -1. & 0 & 0 & 0 \\
1 & -0.707107 & 0 & 0.707107 & -0.707107 & 1. & -0.707107 \\
1 & 0 & -1. & 0 & -1. & 0 & 1. \\
1 & 0.707107 & 0 & -0.707107 & -0.707107 & -1. & -0.707107
\end{pmatrix}$$

Check the rank of A:

```
rank = Length[SingularValues[A][[2]]]
7
```

Good.

A is of full rank. This tells you that Xsol is the only solution.

Get the x[k]'s:

```

Y = Table[fpoints[[k, 2]], {k, 1, Length[fpoints]}];
Clear[alignerframe, stretch, hangerframe, k];
alignerframe[k_] := SingularValues[A][[3, k]];
stretch[k_] := SingularValues[A][[2, k]];
hangerframe[k_] := SingularValues[A][[1, k]];
Xsol = Sum[ $\frac{Y.hangerframe[k]}{stretch[k]}$  alignerframe[k], {k, 1, rank}]
{0, 0.853553, 0, 0.146447, -0.603553, -0.25, -0.103553}

```

The trig polynomial you are after is:

```

Clear[trigfitter];
trigfitter[t_] =
trigpoly[t] /. Table[x[k] -> Xsol[[k]], {k, 0, Length[fpoints]}]
0.853553 Cos[ $\pi t$ ] + 0.146447 Cos[ $3\pi t$ ] -
0.603553 Sin[ $\pi t$ ] - 0.25 Sin[ $2\pi t$ ] - 0.103553 Sin[ $3\pi t$ ]

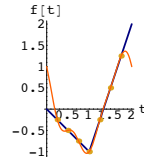
```

See it:

```

fitplot = Plot[trigfitter[t], {t, 0, L},
PlotStyle -> {{CadmiumOrange, Thickness[0.015]}},
DisplayFunction -> Identity];
Show[setup, fitplot, DisplayFunction -> $DisplayFunction];

```



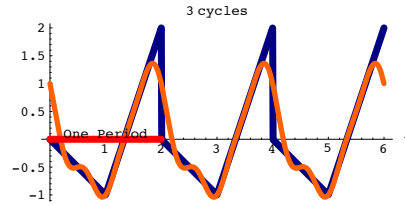
Nailed all the points.

See it over some extra cycles:

```

cycles = 3;
Plot[{f[t], trigfitter[t]}, {t, 0, cycles L}, PlotStyle ->
{{Thickness[0.02], Navy}, {Thickness[0.015], CadmiumOrange}},
AxesLabel -> {"t", ""}, PlotLabel -> "cycles" cycles,
Epilog -> {{Red, Thickness[0.02], Line[{{0, 0}, {L, 0}]}],
{Text["One Period", { $\frac{L}{2}$ , 0.1}]}}, AspectRatio -> Automatic];

```



But the approximation is not all that great.

Copy paste and edit reducing jump from  $\frac{L}{8}$  to  $\frac{L}{16}$  and use the resulting plot to assess how well the resulting new trigfitter[t] approximates the original function f[t].

#### □G.7.b.ii) More Fourier fit

Go with this square wave function f[t]:

```

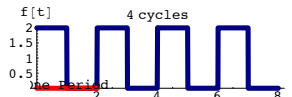
Clear[f, t];
f[t_] = Sign[Sin[ $\pi t$ ]] + 1;
L = 2;

```

```

cycles = 4;
Plot[f[t], {t, 0, cycles L}, PlotStyle -> {{Thickness[0.02], Navy}},
AxesLabel -> {"t", "f[t]"}, PlotLabel -> "cycles" cycles,
Epilog -> {{Red, Thickness[0.02], Line[{{0, 0}, {L, 0}]}],
{Text["One Period", { $\frac{L}{2}$ , 0.1}]}];

```



Come up with a trig polynomial that approximates this function fairly well. Show off your work with winning plots.

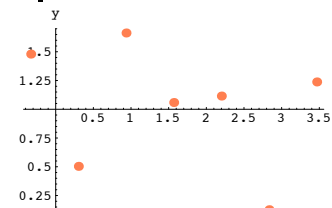
#### □G.7.c.i) How you make an interpolating polynomial

Here are some points:

```

tlow = Random[Real, {-1, 1}];
thigh = Random[Real, {3, 4}];
jump =  $\frac{thigh - tlow}{6}$ ;
points = Table[{t, Random[Real, {0, 2}]}, {t, tlow, thigh, jump}];
pointplot = ListPlot[points,
PlotStyle -> {Coral, PointSize[0.03]}, AxesLabel -> {"t", "y"}];

```



There is an old adage that says there is a unique polynomial of least degree whose plot runs through all these plotted points.

Mathematica can calculate this polynomial it is:

```

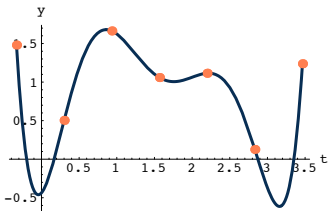
Clear[leastdegree, t];
leastdegree[t_] = Expand[InterpolatingPolynomial[points, t]]
-0.436254 + 1.11977 t + 11.9478 t2 -
21.4475 t3 + 14.0311 t4 - 4.01082 t5 + 0.420252 t6

```

Here are the same points shown together with a plot of this function.

```
curveplot = Plot[leastdegree[t], {t, tlow, thigh}, PlotStyle ->
  {{Thickness[0.01], Indigo}}, DisplayFunction -> Identity];
```

```
Show[pointplot, curveplot, pointplot,
  PlotRange -> All, DisplayFunction -> $DisplayFunction];
```



The polynomial curve runs right through the points.

One way of seeing where this interpolating polynomial comes from involves linear systems.

First you look at the points:

```
points = {{-0.331932, 1.47988}, {0.300918, 0.503649},
  {0.933769, 1.66339}, {1.56662, 1.05868},
  {2.19947, 1.11485}, {2.83232, 0.126318}, {3.46517, 1.23769}};
```

Next you make a cleared polynomial function of t with one undetermined coefficient x[k] for each given point {t[k],y[k]}:

```
Clear[clearpoly, x, k];
clearpoly[t_] = Sum[x[k] t^k-1, {k, 1, Length[points]}];
x[1] + t x[2] + t^2 x[3] + t^3 x[4] + t^4 x[5] + t^5 x[6] + t^6 x[7]
```

To nail down the

```
x[1], x[2], x[3], x[4], x[5], x[6], x[7]
```

that make this into the interpolating polynomial running through all the points, you take the first slot of each point, substitute it into the cleared polynomial function and set the result equal to the second slot of the same point:

```
linearsystem =
  ColumnForm[Table[clearpoly[points[[k, 1]]] == points[[k, 2]],
    {k, 1, Length[points]}]]
```

```
x[1] - 0.331932 x[2] + 0.110179 x[3] - 0.036572 x[4] + 0.0121394 x[5] - 0.00402947 x[6] + 0.00133751 x[7] + 0.300918 x[2] + 0.0905519 x[3] + 0.0272488 x[4] + 0.00819965 x[5] + 0.00246743 x[6] + 0.00074249 x[7] + 0.933769 x[2] + 0.871925 x[3] + 0.814177 x[4] + 0.760253 x[5] + 0.709901 x[6] + 0.662884 x[7] + 1.56662 x[2] + 2.4543 x[3] + 3.84495 x[4] + 6.02358 x[5] + 9.43666 x[6] + 14.7837 x[7] + 2.19947 x[2] + 4.83767 x[3] + 10.6403 x[4] + 23.4031 x[5] + 51.4744 x[6] + 113.216 x[7] + 2.83232 x[2] + 8.02205 x[3] + 22.721 x[4] + 64.3532 x[5] + 182.269 x[6] + 516.244 x[7] + 3.46517 x[2] + 12.0074 x[3] + 41.6078 x[4] + 144.178 x[5] + 499.602 x[6] + 1731.21 x[7]
```

You can see that this is a linear system with the same number of variable x[k]'s as equations.

To save a lot of typing, you can get the coefficient matrix A this way:

If this confuses you, see the tutorial on plucking matrices.

```
linearformulas =
  Table[clearpoly[points[[k, 1]]], {k, 1, Length[points]}];
xvariables = Table[x[k], {k, 1, Length[points]}];
A = Outer[D, {linearformulas}, xvariables][[1]];
MatrixForm[A]
{ 1 -0.331932 0.110179 -0.036572 0.0121394 -0.00402947 0.00133751
  1 0.300918 0.0905519 0.0272488 0.00819965 0.00246743 0.00074249
  1 0.933769 0.871925 0.814177 0.760253 0.709901 0.662884
  1 1.56662 2.4543 3.84495 6.02358 9.43666 14.7837
  1 2.19947 4.83767 10.6403 23.4031 51.4744 113.216
  1 2.83232 8.02205 22.721 64.3532 182.269 516.244
  1 3.46517 12.0074 41.6078 144.178 499.602 1731.21 }
```

Now look at:

```
Det[A]
1671.99
```

How does this signal that the linear system has exactly one solution?

Put answer here.

Use matrix methods to solve the linear system in part i) for

```
x[1],x[2],x[3],x[4],x[5],x[6] and x[7],
```

substitute your x-values into the cleared polynomial and compare the resulting polynomial to:

```
Expand[InterpolatingPolynomial[points, t]]
-0.436254 + 1.11977 t + 11.9478 t^2 -
  21.4475 t^3 + 14.0311 t^4 - 4.01082 t^5 + 0.420252 t^6
```

Describe what you get.

### G.7.c.i) Theory of interpolating polynomials

To make an interpolating polynomial, you start with given points in the form {t,y} all with different t-slots.

This goes with six points, you could do the same thing for any number of points.

```
Clear[t, y, i];
points = Table[{t[i], y[i]}, {i, 1, 6}]
{{t[1], y[1]}, {t[2], y[2]}, {t[3], y[3]},
  {t[4], y[4]}, {t[5], y[5]}, {t[6], y[6]}}
```

Next you make a cleared polynomial function of t with one undetermined coefficient x[i] for each given point {t[i],y[i]}:

```
Clear[clearpoly];
clearpoly[t_] = Sum[x[k] t^k-1, {k, 1, Length[points]}];
x[1] + t x[2] + t^2 x[3] + t^3 x[4] + t^4 x[5] + t^5 x[6]
```

To nail down the

```
x[1], x[2], x[3], x[4], x[6]
```

that make this into the interpolating polynomial running through all the points, you take the first slot of each point, substitute it into the cleared polynomial function and set the result equal to the second slot of the same point:

```
linearsystem =
  ColumnForm[Table[clearpoly[t[k]] == y[k], {k, 1, Length[points]}]]
x[1] + t[1] x[2] + t[1]^2 x[3] + t[1]^3 x[4] + t[1]^4 x[5] + t[1]^5 x[6] == y[1]
x[1] + t[2] x[2] + t[2]^2 x[3] + t[2]^3 x[4] + t[2]^4 x[5] + t[2]^5 x[6] == y[2]
x[1] + t[3] x[2] + t[3]^2 x[3] + t[3]^3 x[4] + t[3]^4 x[5] + t[3]^5 x[6] == y[3]
x[1] + t[4] x[2] + t[4]^2 x[3] + t[4]^3 x[4] + t[4]^4 x[5] + t[4]^5 x[6] == y[4]
x[1] + t[5] x[2] + t[5]^2 x[3] + t[5]^3 x[4] + t[5]^4 x[5] + t[5]^5 x[6] == y[5]
x[1] + t[6] x[2] + t[6]^2 x[3] + t[6]^3 x[4] + t[6]^4 x[5] + t[6]^5 x[6] == y[6]
```

You can see that this is a linear system with the same number of variable x[k]'s as equations.

To save a lot of typing, you can get the coefficient matrix A this way:

```
linearformulas = Table[clearpoly[t[k]], {k, 1, Length[points]}];
xvariables = Table[x[k], {k, 1, Length[points]}];
A = Outer[D, {linearformulas}, xvariables][[1]];
MatrixForm[A]
```

```
{ 1 t[1] t[1]^2 t[1]^3 t[1]^4 t[1]^5
  1 t[2] t[2]^2 t[2]^3 t[2]^4 t[2]^5
  1 t[3] t[3]^2 t[3]^3 t[3]^4 t[3]^5
  1 t[4] t[4]^2 t[4]^3 t[4]^4 t[4]^5
  1 t[5] t[5]^2 t[5]^3 t[5]^4 t[5]^5
  1 t[6] t[6]^2 t[6]^3 t[6]^4 t[6]^5 }
```

That's the right coefficient matrix A and now that you have it, let Mathematica calculate Det[A] in factored form:

```
Factor[Det[A]]
-(t[1] - t[2]) (t[1] - t[3]) (t[2] - t[3]) (t[1] - t[4]) (t[2] - t[4])
(t[3] - t[4]) (t[1] - t[5]) (t[2] - t[5]) (t[3] - t[5]) (t[4] - t[5])
(t[1] - t[6]) (t[2] - t[6]) (t[3] - t[6]) (t[4] - t[6]) (t[5] - t[6])
```

How does this signal that if the given points all have different first slots (t[i]'s), then you are guaranteed that for each given {y[1],y[2],y[3],y[4],y[5],y[6]} this linear system has exactly one solution for {x[1],x[2],x[3],x[4],x[5],x[6]}?

How does signal that there is precisely one polynomial of the form

```
poly[t] = a[1] + a[2] t + a[3] t^2 + a[4] t^3 + a[5] t^4 + a[6] t^5
```

with the property that

```
poly[t[k]] = y[k] for k=1,2,3,4,5,6
```

### G.8) Linear systems in calculus and diffeq

#### G.8.a) The method of undetermined coefficients: Systems of linear equations in integration

When you go with a function f[x] and try to calculate

```
integrate f[x] dx,
```

you look for a function g[x] with

```
g'[x] = f[x].
```

If you find g[x], then you can announce that

```
integrate f[x] dx = g[t] - g[0]
```

In the case

```
f[x] = x^3 e^-x,
```

it's a good idea to look for g[x] like this:

```
Clear[x, a, k];
g[x_] = Sum[a[k + 1] x^k e^-x, {k, 0, 3}]
e^-x a[1] + e^-x x a[2] + e^-x x^2 a[3] + e^-x x^3 a[4]
```

Look at  $g'[x]$  with an eye toward setting the coefficients  $a[1], a[2], a[3]$  and  $a[4]$  so that  $g'[x] = f[x] = x^3 e^{-x}$ :

$$g'[x] = -e^{-x} a[1] + e^{-x} a[2] - e^{-x} x a[2] + 2 e^{-x} x a[3] - e^{-x} x^2 a[3] + 3 e^{-x} x^2 a[4] - e^{-x} x^3 a[4]$$

Collect the terms and factor:

$$\text{Collect}[g'[x], \{1, x, x^2, x^3\}] = -e^{-x} a[1] + e^{-x} a[2] + x(-e^{-x} a[2] + 2 e^{-x} a[3]) - e^{-x} x^2 a[3] + x^2(-e^{-x} a[3] + 3 e^{-x} a[4]) - e^{-x} x^3 a[4]$$

So  $g'[x] = -e^{-x} a[1] + e^{-x} a[2] + x(-e^{-x} a[2] + 2 e^{-x} a[3]) - e^{-x} x^2 a[3] + x^2(-e^{-x} a[3] + 3 e^{-x} a[4])$

You want this to be  $f[x] = x^3 E^{-x}$ , so you write

$$-a[1] + a[2] = 0$$

This makes the coefficient of  $e^{-x}$  in  $g'[x]$  equal to 0.

$$-a[2] + 2 a[3] = 0$$

This makes the coefficient of  $x e^{-x}$  in  $g'[x]$  equal to 0.

$$-a[3] + 3 a[4] = 0$$

This makes the coefficient of  $x^2 e^{-x}$  in  $g'[x]$  equal to 0.

$$-a[4] = 1$$

This makes the coefficient of  $x^3 e^{-x}$  in  $g'[x]$  equal to 1.

Solve these equations by hand and then use matrix methods to confirm your hand calculation. Use your solutions for

$a[1], a[2], a[3], a[4]$  to calculate

$$\int_0^1 x^3 E^{-x} dx$$

For two tips, click on the right.

□ Tip 1:

Solving the equations

$$\begin{aligned} -a[1] + a[2] &= 0 \\ -a[2] + 2 a[3] &= 0 \\ -a[3] + 3 a[4] &= 0 \\ -a[4] &= 1 \end{aligned}$$

by hand is really easy.

You use the last equation to nail down

$$a[4] = -1.$$

Put  $a[4] = -1$ . into the equation

$$-a[3] + 3 a[4] = 0$$

to get

$$-a[3] - 3 = 0.$$

This gives  $a[3] = -1$ .

Now you take over.

Lots of folks call this technique "backwards substitution." It doesn't work for all linear systems, but it works beautifully here.

□ Tip 2:

To do this by matrix methods, use an inverse matrix.

□ G.8.b) Systems of linear equations for coming up with formulas for solutions of linear differential equations

You have the job of coming up with a formula for the solution of the fourth order differential equation

$$y''''[t] + 2.9 y'''[t] + 3.04 y''[t] + 1.356 y'[t] + 0.216 y[t] = 0$$

with

$$y[0] = 2.3, y'[0] = -2.7, y''[0] = -3.7 \text{ and } y'''[0] = 0.8$$

Remembering the characteristic equation, you type:

```
Clear[z];
charequation = z^4 + 2.9 z^3 + 3.04 z^2 + 1.356 z + 0.216 == 0
0.216 + 1.356 z + 3.04 z^2 + 2.9 z^3 + z^4 == 0
```

And solve it:

```
Solve[charequation, z]
{{z -> -1.}, {z -> -0.9}, {z -> -0.6}, {z -> -0.4}}
```

Next you put:

```
Clear[trially, t, x];
trially[t_] = x[1] E^-t + x[2] E^-0.9 t + x[3] E^-0.6 t + x[4] E^-0.4 t
e^-t x[1] + e^-0.9 t x[2] + e^-0.6 t x[3] + e^-0.4 t x[4]
```

Next, you incorporate the starter data

$$y[0] = 2.3, y'[0] = -2.7, y''[0] = -3.7 \text{ and } y'''[0] = 0.8 :$$

```
ColumnForm[Thread[{trially[0] == 2.3,
trially'[0] == -2.7,
trially''[0] == -3.7,
trially'''[0] == 0.8}]]
x[1] + x[2] + x[3] + x[4] == 2.3
-x[1] - 0.9 x[2] - 0.6 x[3] - 0.4 x[4] == -2.7
x[1] + 0.81 x[2] + 0.36 x[3] + 0.16 x[4] == -3.7
-x[1] - 0.729 x[2] - 0.216 x[3] - 0.064 x[4] == 0.8
```

Hey dude! That's a linear system.

To nail down a formula the solution of

$$y''''[t] + 13.4 y'''[t] + 48.41 y''[t] + 51.484 y'[t] + 8.466 y[t] = 0$$

with

$$y[0] = 2.3, y'[0] = -2.7, y''[0] = -3.7 \text{ and } y'''[0] = 0.8,$$

you solve the linear system above for  $x[1], x[2], x[3]$  and  $x[4]$  and insert the results in  $trially[t]$ .

Use matrix methods to do this.

□ G.8.c) Systems of linear equations for the method of undetermined coefficients in diffeq

Advanced folks know that all solutions of the forced, damped oscillator

$$y''[t] + 0.4 y'[t] + 1.3 y[t] = 1.2 \text{ Sin}[1.2 t] + 0.5 \text{ Cos}[2.3 t] + 0.6 \text{ Sin}[4.1 t]$$

have the same long term behavior regardless of what the given initial data on  $y[0]$  and  $y'[0]$  are. So to see what the long term behavior any solution  $y[t]$ , all you need it to come up with any one particular solution of it and plot it.

Glancing at the right side, you get the idea you can make a solution out of:

At this stage, the coefficients  $x[1], x[2], x[3], x[4], x[5]$  and  $x[6]$  are undetermined (cleared).

```
Clear[undetermined, t, x];
undetermined[t_] =
x[1] Sin[1.2 t] + x[2] Cos[1.2 t] +
x[3] Sin[2.3 t] + x[4] Cos[2.3 t] +
x[5] Sin[4.1 t] + x[6] Cos[4.1 t]
Sin[1.2 t] x[1] + Cos[1.2 t] x[2] + Sin[2.3 t] x[3] +
Cos[2.3 t] x[4] + Sin[4.1 t] x[5] + Cos[4.1 t] x[6]
```

To try to determine the  $x[1], x[2], x[3], x[4], x[5]$  and  $x[6]$  that give a solution, you replace  $y[t]$  by  $undetermined[t]$  in

$$y''[t] + 0.4 y'[t] + 1.3 y[t]:$$

```
Clear[y];
left = Expand[y''[t] + 0.4 y'[t] + 1.3 y[t] /. {y[t] -> undetermined[t],
y'[t] -> undetermined'[t], y''[t] -> undetermined''[t]}]
0.48 Cos[1.2 t] x[1] - 0.14 Sin[1.2 t] x[1] - 0.14 Cos[1.2 t] x[2] -
0.48 Sin[1.2 t] x[2] + 0.92 Cos[2.3 t] x[3] - 3.99 Sin[2.3 t] x[3] -
3.99 Cos[2.3 t] x[4] - 0.92 Sin[2.3 t] x[4] + 1.64 Cos[4.1 t] x[5] -
15.51 Sin[4.1 t] x[5] - 15.51 Cos[4.1 t] x[6] - 1.64 Sin[4.1 t] x[6]
```

You want this to be  $1.2 \text{ Sin}[1.2 t] + 0.5 \text{ Cos}[2.3 t] + 0.6 \text{ Sin}[4.1 t]$ , so you type:

```
right = 1.2 Sin[1.2 t] + 0.5 Cos[2.3 t] + 0.6 Sin[4.1 t];
```

```
ColumnForm[Thread[
{Coefficient[left, Sin[1.2 t]] == Coefficient[right, Sin[1.2 t]],
Coefficient[left, Cos[1.2 t]] == Coefficient[right, Cos[1.2 t]],
```

```
Coefficient[left, Sin[2.3 t]] == Coefficient[right, Sin[2.3 t]],
Coefficient[left, Cos[2.3 t]] == Coefficient[right, Cos[2.3 t]],
Coefficient[left, Sin[4.1 t]] == Coefficient[right, Sin[4.1 t]],
Coefficient[left, Cos[4.1 t]] == Coefficient[right, Cos[4.1 t]]]]]
-0.14 x[1] - 0.48 x[2] == 1.2
0.48 x[1] - 0.14 x[2] == 0
-3.99 x[3] - 0.92 x[4] == 0
0.92 x[3] - 3.99 x[4] == 0.5
-15.51 x[5] - 1.64 x[6] == 0.6
1.64 x[5] - 15.51 x[6] == 0
```

That's a 6D linear system.

Solve by hand or use matrix methods to solve it (your choice).

Then substitute your solutions for  $x[1], x[2], x[3], x[4], x[5]$  and  $x[6]$  into  $trially[t]$ , thereby obtaining a formula for one solution  $y[t]$  of

$$y''[t] + 0.4 y'[t] + 1.3 y[t] = 1.2 \text{ Sin}[1.2 t] + 0.5 \text{ Cos}[2.3 t] + 0.6 \text{ Sin}[4.1 t]$$

Exhibit the long term behavior of this oscillator by plotting your formula over a healthy range of  $t$ 's.

G.9) Null space and rank

□ G.9.a.i) Null frames

Here's a random matrix that hits on 5D and hangs in 3D

```
hitdim = 6; hangdim = 3;
Clear[a, i, j];
a[i_, j_] := Random[Real, {-2, 2}];
A = Table[a[i, j], {i, 1, hangdim}, {j, 1, hitdim}];
MatrixForm[A]
```

$$\begin{pmatrix} 0.930068 & -0.925949 & -1.15149 & -1.16766 & 0.83042 & 1.06218 \\ -0.130978 & 1.78901 & -0.662792 & 0.401176 & -1.41027 & -0.17815 \\ 0.120527 & 1.00413 & -1.87146 & -1.51429 & -1.74016 & -1.95563 \end{pmatrix}$$

```
Clear[nullframe];
nullframe[k_] := nullframe[k] = NullSpace[A][[k]];
zeroes = Table[nullframe[k], {k, 1, Length[NullSpace[A]]}]
{{-0.916324, -0.043484, -0.25933, -0.171766, 0.0729626, 0.237445},
{0, 0.564734, -0.0843279, -0.0606855, 0.77288, -0.27007},
{0, 0.300257, 0.593389, -0.696553, -0.126191, 0.237962}}
```

You can think of these as SVD aligner frames for A that correspond to zero stretch factors.

```
Table[A.nullframe[k], {k, 1, Length[NullSpace[A]]}]
{{0, 0, 0}, {0, 0, 0}, {0, 0, 0}}
```

Here are the SVD aligner frames for A that correspond to positive stretch factors:

```
rank = Length[SingularValues[A][[2]]];
Clear[alignerframe, k];
alignerframe[k_] := SingularValues[A][[3, k]];
nonzeroes = Table[alignerframe[k], {k, 1, rank}]
{{0.0267016, -0.457015, 0.429521, 0.230508, 0.566971, 0.480985},
{-0.328402, 0.405837, 0.522984, 0.627401, -0.224775, -0.0989017},
{-0.227573, -0.464182, 0.339892, -0.186716, 0.0971032, -0.75692}}
```

Notice that:

```
Length[zeroes] + Length[nonzeroes] == hitdim
True
```

Is this just an isolated accident or will it happen in every set up? Explain.

□G.9.a.ii) No null frames

Here's a random matrix of full rank:

```
hitdim = 4; hangdim = 6;
Clear[a, i, j];
a[i_, j_] := Random[Real, {-2, 2}];
A = Table[a[i, j], {i, 1, hangdim}, {j, 1, hitdim}];
MatrixForm[A]
```

```
{1.12124 -0.841068 0.142485 -0.185341
0.868604 0.683553 1.21242 -1.25939
0.020096 -0.148786 -1.618 -0.321574
-1.84893 0.0622057 1.04479 1.27725
1.56135 -1.75964 -1.07574 -1.72688
1.43281 1.75464 -1.33558 -1.77125}
```

```
rank = Length[SingularValues[A][[2]]]
```

4

If by some chance, the rank turns out to be less than hitdim, rerun until it is.

Now look at this:

```
NullSpace[A]
```

{}

Nothing there. Why was this guaranteed to happen?

□G.9.b.i) No null frames

Here's a linear system:

```
A = {{2.686, -0.666, -3.0464, -2.1568, -1.881},
{3.068, 2.667, 4.596, -3.594, -2.986},
```

```
{-1.918, 2.915, -3.093, -0.0829, -0.354},
{1.15, 5.582, -1.497, -6.6769, -3.34},
{-0.3820, -5.333, -4.6424, 3.4372, 1.105}};
```

```
hitdim = 5; hangdim = 5;
Clear[x, k];
X = Table[x[k], {k, 1, hitdim}];
Y = {-2.34, 1.84, -2.98, 7.04, -6.70};
```

```
ColumnForm[Thread[A.X == Y]]
2.686 x[1] - 0.666 x[2] - 3.0464 x[3] - 2.1568 x[4] - 1.881 x[5] == -2.34
3.068 x[1] + 2.667 x[2] + 4.596 x[3] - 3.594 x[4] - 2.986 x[5] == 1.84
-1.918 x[1] + 2.915 x[2] - 3.093 x[3] - 0.0829 x[4] - 0.354 x[5] == -2.98
1.15 x[1] + 5.582 x[2] - 1.497 x[3] - 6.6769 x[4] - 3.34 x[5] == 7.04
-0.382 x[1] - 5.333 x[2] - 4.6424 x[3] + 3.4372 x[4] + 1.105 x[5] == -6.7
```

Intent on solving it quickly, you look at:

```
Xapprox = PseudoInverse[A].Y;
A.Xapprox
{-2.34, 1.84, -2.98, 7.04, -6.7}
Y
{-2.34, 1.84, -2.98, 7.04, -6.7}
NullSpace[A]
```

And announce that this system has exactly one solution. And its solution is:

```
Xapprox
{-1.5685, -1.08823, 0.756877, -3.48354, 2.15807}
```

Are you right?

□G.9.b.ii) Some null frames

Here's another linear system:

```
A = {{2.686, -0.666, -3.0464, -0.1568, -1.881},
{3.068, 2.667, 1.596, -3.594, -2.986},
{-1.918, 2.915, -3.093, -0.0829, -0.354},
{1.15, 5.582, -1.497, -3.6769, -3.34},
{-0.3820, -3.333, -4.6424, 3.4372, 1.105}};
```

```
hitdim = 5; hangdim = 5;
Clear[x, k];
X = Table[x[k], {k, 1, hitdim}];
Y = {1.71986, 14.9293, 23.1721, 38.1014, -13.2094};
```

```
linearsystem = ColumnForm[Thread[A.X == Y]]
```

```
2.686 x[1] - 0.666 x[2] - 3.0464 x[3] - 0.1568 x[4] - 1.881 x[5] == 1.71986
3.068 x[1] + 2.667 x[2] + 1.596 x[3] - 3.594 x[4] - 2.986 x[5] == 14.9293
-1.918 x[1] + 2.915 x[2] - 3.093 x[3] - 0.0829 x[4] - 0.354 x[5] == 23.1721
1.15 x[1] + 5.582 x[2] - 1.497 x[3] - 3.6769 x[4] - 3.34 x[5] == 38.1014
-0.382 x[1] - 3.333 x[2] - 4.6424 x[3] + 3.4372 x[4] + 1.105 x[5] == -13.2094
```

Intent on solving it quickly, you look at:

```
Xapprox = PseudoInverse[A].Y;
A.Xapprox
{1.71988, 14.9293, 23.1721, 38.1014, -13.2094}
Y
{1.71986, 14.9293, 23.1721, 38.1014, -13.2094}
NullSpace[A]
{{-0.540576, -0.504089, -0.0773624, -0.516719, -0.425074},
{-0.160092, 0.0882502, 0.247105, 0.630627, -0.712623}}
```

And announce that this system is underdetermined (lots of solutions) and the solution of minimum norm is:

```
Xsol
{0, 0.853553, 0, 0.146447, -0.603553, -0.25, -0.103553}
```

Are you right?

□G.5.b.iii)

Here's another linear system:

```
A = {{2.686, -0.666, -3.0464, -0.1568, -1.881},
{3.068, 2.667, 1.596, -3.594, -2.986},
{-1.918, 2.915, -3.093, -0.0829, -0.354},
{1.15, 5.582, -1.497, -3.6769, -3.34},
{-0.3820, -3.333, -4.6424, 3.4372, 1.105}};
```

```
hitdim = 5; hangdim = 5;
Clear[x, k];
X = Table[x[k], {k, 1, hitdim}];
Y = {0.12, 3.14, 2.72, -5.06, -2.43};
```

```
linearsystem = ColumnForm[Thread[A.X == Y]]
2.686 x[1] - 0.666 x[2] - 3.0464 x[3] - 0.1568 x[4] - 1.881 x[5] == 0.12
3.068 x[1] + 2.667 x[2] + 1.596 x[3] - 3.594 x[4] - 2.986 x[5] == 3.14
-1.918 x[1] + 2.915 x[2] - 3.093 x[3] - 0.0829 x[4] - 0.354 x[5] == 2.72
1.15 x[1] + 5.582 x[2] - 1.497 x[3] - 3.6769 x[4] - 3.34 x[5] == -5.06
-0.382 x[1] - 3.333 x[2] - 4.6424 x[3] + 3.4372 x[4] + 1.105 x[5] == -2.43
```

Intent on solving it quickly, you look at:

```
Xapprox = PseudoInverse[A].Y;
A.Xapprox
{-1.02375, 0.2625, -1.30125, -1.03875, -1.28625}
Y
{0.12, 3.14, 2.72, -5.06, -2.43}
```

And announce that this system is overdetermined. Are you right?

G.10) Columns and rank, rows and rank

□G.10.a) Columns and rank

Here is a matrix entered via its vertical columns:

```
Clear[column, j];
column[1] = {0.96, 1.58, 0.91, -0.22, 0.97};
column[2] = {0.57, 1.55, -1.42, -1.42, -1.45};
column[3] = {2.07, -2.91, -2.74, -1.40, -2.87};
column[4] = {-0.76, 0.74, -2.56, -0.61, 0.78};
column[5] = {1.36, -0.93, 2.73, 2.38, -2.49};
```

```
A = Transpose[Table[column[j], {j, 1, 5}]];
MatrixForm[A]
```

```
{0.96 0.57 2.07 -0.76 1.36
1.58 1.55 -2.91 0.74 -0.93
0.91 -1.42 -2.74 -2.56 2.73
-0.22 -1.42 -1.4 -0.61 2.38
0.97 -1.45 -2.87 0.78 -2.49}
```

This matrix is of full rank:

```
rank = Length[SingularValues[A][[2]]]
5
```

Your job is to come up with a nonzero 5D vector

```
Z = {z[1],z[2],z[3],z[4],z[5]}
```

so that when you reset

```
column[3] = Z,
```

then the resulting matrix is not of full rank.

Check yourself by calculating the rank of the resulting matrix.

□G.10.b) Rows and rank

Here is a matrix entered via its horizontal rows:

```
Clear[row, i];
row[1] = {1.96, 1.78, 2.91, -3.22};
row[2] = {3.69, 2.55, -0.42, -2.27};
row[3] = {2.17, -3.01, 4.74, -2.30};
row[4] = {-1.76, 1.74, -0.96, 0.97};
```

```
A = Table[row[i], {i, 1, 4}];
MatrixForm[A]
```

$$\begin{pmatrix} 1.96 & 1.78 & 2.91 & -3.22 \\ 3.69 & 2.55 & -0.42 & -2.27 \\ 2.17 & -3.01 & 4.74 & -2.3 \\ -1.76 & 1.74 & -0.96 & 0.97 \end{pmatrix}$$

This matrix is of full rank:

```
rank = Length[SingularValues[A][[2]]]
4
```

Your job is to come up with a nonzero 4D vector

$$Z = \{z[1], z[2], z[3], z[4]\}$$

so that when you reset

$$\text{row}[2] = Z,$$

then the resulting matrix is not of full rank.

Check yourself by calculating the rank of the resulting matrix.

## G.11) Inverses and eigenvalues via row reduction

### G.11.a.i) Inverses via row reduction

Here's random 4D matrix A

```
dim = 4;
A = Table[Random[Real, {-5, 5}], {i, 1, dim}, {j, 1, dim}];
MatrixForm[A]
```

$$\begin{pmatrix} -4.22107 & 1.48928 & 1.30485 & 1.03523 \\ -1.39258 & 4.78039 & 3.2738 & -0.816292 \\ 3.55718 & 0.152357 & 2.31881 & 4.98764 \\ 3.17949 & 4.99684 & 4.70684 & -3.20548 \end{pmatrix}$$

Augment A with this right column:

```
Clear[y];
Y = Table[y[i], {i, 1, dim}];
augmentedA = AppendRows[A, Transpose[{Y}]];
MatrixForm[augmentedA]
```

$$\begin{pmatrix} -4.22107 & 1.48928 & 1.30485 & 1.03523 & y[1] \\ -1.39258 & 4.78039 & 3.2738 & -0.816292 & y[2] \\ 3.55718 & 0.152357 & 2.31881 & 4.98764 & y[3] \\ 3.17949 & 4.99684 & 4.70684 & -3.20548 & y[4] \end{pmatrix}$$

Look at this:

```
rowechelon = Expand[RowReduce[augmentedA]];
MatrixForm[rowechelon]
```

$$\begin{pmatrix} 1 & 0 & 0 & 0 & -0.253883 y[1] + 0.109985 y[2] + 0.0507094 y[3] - 0.0310984 y[4] \\ 0 & 1 & 0 & 0 & -0.568176 y[1] + 0.801117 y[2] - 0.00595275 y[3] - 0.396766 y[4] \\ 0 & 0 & 1 & 0 & 0.691024 y[1] - 0.755616 y[2] + 0.0638772 y[3] + 0.514982 y[4] \\ 0 & 0 & 0 & 1 & -0.12284 y[1] + 0.248382 y[2] + 0.134814 y[3] - 0.205122 y[4] \end{pmatrix}$$

And look at this:

```
Inverse[A].Y
{-0.253883 y[1] + 0.109985 y[2] + 0.0507094 y[3] - 0.0310984 y[4],
-0.568176 y[1] + 0.801117 y[2] - 0.00595275 y[3] - 0.396766 y[4],
0.691024 y[1] - 0.755616 y[2] + 0.0638772 y[3] + 0.514982 y[4],
-0.12284 y[1] + 0.248382 y[2] + 0.134814 y[3] - 0.205122 y[4]}
```

Describe what you see and explain why you see it.

### G.11.a.ii) Eigenvalues via row reduction

Here's random 2D matrix A :

```
dim = 2;
A = Table[N[Random[Real, {-5, 5}]], {i, 1, dim}, {j, 1, dim}];
MatrixForm[A]
```

$$\begin{pmatrix} 4.27612 & 4.39595 \\ 2.39619 & -3.88828 \end{pmatrix}$$

Integers are not necessary for the concept, but they keep everything on the screen.

The eigenvalues of A are the  $\lambda$ 's for which the matrix

$$A\lambda = A - \lambda \text{IdentityMatrix}[\text{dim}]$$

is not invertible:

Eigenvalues will become a big issue in a later lesson.

```
Aλ = A - λ IdentityMatrix[dim];
MatrixForm[Aλ]
```

$$\begin{pmatrix} 4.27612 - \lambda & 4.39595 \\ 2.39619 & -3.88828 - \lambda \end{pmatrix}$$

Augment  $A\lambda$  with this right column:

```
Clear[y];
Y = Table[y[i], {i, 1, dim}];
augmentedAλ = AppendRows[Aλ, Transpose[{Y}]];
MatrixForm[augmentedAλ]
```

$$\begin{pmatrix} 4.27612 - \lambda & 4.39595 & y[1] \\ 2.39619 & -3.88828 - \lambda & y[2] \end{pmatrix}$$

Look at this:

```
rowechelon = Together[RowReduce[augmentedAλ]];
MatrixForm[rowechelon]
```

$$\begin{pmatrix} 1 & 0 & \frac{3.88828 y[1] + 1. \lambda y[1] - 4.39595 y[2]}{27.1603 + 0.387841 \lambda - \lambda^2} \\ 0 & 1 & \frac{2.39619 y[1] - 4.27612 y[2] + \lambda y[2]}{27.1603 + 0.387841 \lambda - \lambda^2} \end{pmatrix}$$

Remembering that the eigenvalues of A are the  $\lambda$ 's for which  $A\lambda$  is not invertible, give a quadratic equation you can solve to come up with the eigenvalues of A. Explain why your equation is correct.

Put answer here.

Compare your equation with:

```
Det[Aλ] == 0
-27.1603 - 0.387841 λ + λ^2 == 0
```

Say why you are not surprised.

Put answer here.

Here's random 4D matrix A :

```
dim = 4;
A = Table[Random[Integer, {-5, 5}], {i, 1, dim}, {j, 1, dim}];
MatrixForm[A]
```

$$\begin{pmatrix} -5 & -4 & 1 & 2 \\ -1 & -3 & -3 & 0 \\ -2 & 2 & -4 & -1 \\ -2 & 1 & 4 & -1 \end{pmatrix}$$

Integers are not necessary for the concept, but they keep everything on the screen.

The eigenvalues of A are the  $\lambda$ 's for which the matrix

$$A\lambda = A - \lambda \text{IdentityMatrix}[\text{dim}]$$

is not invertible:

```
Aλ = A - λ IdentityMatrix[dim];
MatrixForm[Aλ]
```

$$\begin{pmatrix} -5 - \lambda & -4 & 1 & 2 \\ -1 & -3 - \lambda & -3 & 0 \\ -2 & 2 & -4 - \lambda & -1 \\ -2 & 1 & 4 & -1 - \lambda \end{pmatrix}$$

Augment  $A\lambda$  with this right column:

```
Clear[y];
Y = Table[y[i], {i, 1, dim}];
augmentedAλ = AppendRows[Aλ, Transpose[{Y}]];
MatrixForm[augmentedAλ]
```

$$\begin{pmatrix} -5 - \lambda & -4 & 1 & 2 & y[1] \\ -1 & -3 - \lambda & -3 & 0 & y[2] \\ -2 & 2 & -4 - \lambda & -1 & y[3] \\ -2 & 1 & 4 & -1 - \lambda & y[4] \end{pmatrix}$$

Look at this:

```
rowechelon = Together[RowReduce[augmentedAλ]];
MatrixForm[rowechelon]
```

$$\begin{pmatrix} 1 & 0 & 0 & 0 & \frac{-27 y[1] - 29 \lambda y[1] - 8 \lambda^2 y[1] - \lambda^3 y[1] + 7 y[2] + 16 \lambda y[2] + 4 \lambda^2 y[2] - 33 y[3] - 24 \lambda y[3] - \lambda^2 y[3] + 236 + 230 \lambda + 71 \lambda^2 + 13 \lambda^3 + \lambda^4}{236 + 230 \lambda + 71 \lambda^2 + 13 \lambda^3 + \lambda^4} \\ 0 & 1 & 0 & 0 & \frac{8 y[1] - \lambda y[1] + \lambda^2 y[1] - 72 y[2] - 39 \lambda y[2] - 10 \lambda^2 y[2] - \lambda^3 y[2] + 36 y[3] + 19 \lambda y[3]}{236 + 230 \lambda + 71 \lambda^2 + 13 \lambda^3 + \lambda^4} \\ 0 & 0 & 1 & 0 & \frac{y[1] + 8 \lambda y[1] + 2 \lambda^2 y[1] - 9 y[2] - 19 \lambda y[2] - 2 \lambda^2 y[2] - 25 y[3] - 23 \lambda y[3] - 9 \lambda^2 y[3] - \lambda^3 y[3]}{236 + 230 \lambda + 71 \lambda^2 + 13 \lambda^3 + \lambda^4} \\ 0 & 0 & 0 & 1 & \frac{66 y[1] + 23 \lambda y[1] + 2 \lambda^2 y[1] - 122 y[2] - 25 \lambda y[2] - \lambda^2 y[2] + 2 y[3] - 27 \lambda y[3] - 4 \lambda^2 y[3] - 106}{236 + 230 \lambda + 71 \lambda^2 + 13 \lambda^3 + \lambda^4} \end{pmatrix}$$

Remembering that the eigenvalues of A are the  $\lambda$ 's for which  $A\lambda$  is not invertible, give a fourth degree equation you can solve to come up with the eigenvalues of A. Explain why your equation is correct.

Put answer here.

Compare your equation with:

```
Det[Aλ] == 0
236 + 230 λ + 71 λ^2 + 13 λ^3 + λ^4 == 0
```

Say why you are not surprised.

Put answer here.