Matrices, Geometry&*Mathematica*

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

Producer: Bruce Carpenter

Publisher: Math Everywhere, Inc.

MGM.06 Beyond 3D

TUTORIALS

T.1) Quick first impressions based on comparing hitdim and hangdim

If hitdim ≠ hangdim, then A is not invertible

☐T.1.a.i) What to look for when hitdim > hangdim (i.e. fewer equations than unknown x[k]'s)

What are the possibilities when you go with a linear system
A.X = Y and hitdim > hangdim (fewer equations than unknown x[k]'s)?

☐Answer:

In the case of fewer equations than unknown x[k]'s, most folks react initially (because there are fewer constraints than degrees of freedom) with the view that the linear system A.X = Y is likely to have more than one solution (underdetermined). This is a real possibility:

☐Possibility 1: More than one solution if Ytest = Y.

Reason:

If Ytest = Y., then A.Xsol = Y.

Because hitdim > hangdim, hits with A have to squash at least

(hitdim - hangdim) dimensions of hitdimD

onto

{0, 0, . . ., 0}.

This tells you that if you take any nonzero vector XX with

A.XX = {0, 0, . . ., 0},

then

A.(Xsol + XX) = A.Xsol + A.XX = A.Xsol + {0,0,. . .,0} = Y.

This gives you more than one solution.

☐Possibility 2: No solution when Ytest ≠ Y .

There are no other possibilities.

☐T.1.a.ii) What to look for when hitdim = hangdim (same number of equations as unknown x[k]'s)

What are the possibilities when you have a linear system
A.X = Y
with
hitdim = hangdim (i.e. same number of equations as unknown x[k]'s)?

☐Answer:

In the case of the same number equations than unknown x[k]'s, most folks react initially (because there are the same number of constraints as degrees of freedom) with the view that the linear system A.X = Y is likely to have exactly one solution(exactly determined). This is not always correct. In fact, there are three possibilities:

☐Possibility 1: Exactly one solution when Det[A] ≠ 0

This does happen if

Det[A] ≠ 0 (ensuring that A is of full rank).

In the case Det[A] ≠ 0, you don't need to check whether Ytest = Y because the lone solution is

$X = A^{-1}.Y$.

☐Possibility 2: More than one solution when Det[A] = 0

This happens when Ytest = Y but Det[A] = 0 (ensuring that rank of A < hit dim so that A is not of full rank).

☐Possibility 3: No solution when Det[A] = 0

This happens when Ytest ≠ Y. In the case that Det[A] = 0, then you are guaranteed that A is not of full rank so hits with A cannot fill up all of hangdim.

☐T.1.a.iii) What to look for when hitdim < hangdim (i.e. more equations than unknown x[k]'s)

What are the possibilities when you have a linear system
A.X = Y
with
hitdim < hangdim (more equations than unknown x[k]'s)?

☐Answer:

In the case of more equations than unknown x[k]'s, most folks react initially (because there are more constraints than degrees of freedom) with the view that the linear system A.X = Y is likely to have no solutions at all.

This reaction is more-or-less natural because hits on hitdimD cannot fill up all of hangdimD. And this outcome is a possibility:

☐Possibility 1: No solution when Ytest ≠Y.

☐Possibility 2: Exactly one solution when Ytest = Y and rank of A = hitdim (so that A is of full rank).

☐Possibility 3: More than one solution when Ytesy = Y and rank of A < hitdim (so that A is not of full rank).

☐T.1.b.i) If hitdim > hangdim, then A is not invertible.

Here is a random matrix with hitdim > hangdim:

```
hitdim = 5; 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]
```

$$\begin{pmatrix} -0.783793 & 0.47607 & -1.40622 & 1.70517 & -1.87879 \\ -0.437816 & -1.18528 & -1.97546 & -1.33273 & 0.768008 \\ 1.84376 & 0.330224 & 0.0863544 & -1.66092 & -1.75133 \end{pmatrix}$$

Here's *Mathematica*'s attempt at a calculation of A^{-1}:

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

```
Inverse::matsq : Argument {<<1>>} at position 1 is not a square matrix.

Inverse[{{-0.783793, 0.47607, -1.40622, 1.70517, -1.87879},
  {-0.437816, -1.18528, -1.97546, -1.33273, 0.768008},
  {1.84376, 0.330224, 0.0863544, -1.66092, -1.75133}}]
```

Garbage.
Explain how you could have known in advance that A is not invertible.

☐Answer:

This matrix A hits 5D vectors into 3D.

When you hit all of 5D with A, you squash at least two dimensions to {0,0,0}.

You cannot undo the hits on the squashed dimensions.

☐T.1.b.ii) If hitdim < hangdim, then A is not invertible.

Here is a random matrix with hitdim < hangdim:

```
hitdim = 4; hangdim = 7;
Clear[a, i, j];
a[i_, j_] := Random[Real, {-2, 2}];

A = Table[a[i, j], {i, 1, hangdim}, {j, 1, hitdim}];

MatrixForm[A]
```

$$\begin{pmatrix} 0.819222 & 0.235267 & -0.398691 & -1.31124 \\ -1.9197 & -1.74688 & -1.55232 & -1.49804 \\ -1.21589 & 1.03692 & -0.0283876 & 1.90818 \\ -0.921056 & 0.915706 & -1.59057 & 1.09346 \\ -0.945591 & 0.248437 & -0.358579 & 1.2497 \\ 0.724185 & -1.83792 & -0.697654 & 1.00103 \\ 1.90496 & -0.0731845 & 1.70104 & 0.312273 \end{pmatrix}$$

This matrix hits on 4D and hangs in 7D.

Here's *Mathematica*'s attempt at a calculation of A^{-1}:

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

```
Inverse::matsq : Argument {<<1>>} at position 1 is not a square matrix.

Inverse[{{0.819222, 0.235267, -0.398691, -1.31124},
  {-1.9197, -1.74688, -1.55232, -1.49804},
  {-1.21589, 1.03692, -0.0283876, 1.90818},
  {-0.921056, 0.915706, -1.59057, 1.09346},
  {-0.945591, 0.248437, -0.358579, 1.2497},
  {0.724185, -1.83792, -0.697654, 1.00103},
  {1.90496, -0.0731845, 1.70104, 0.312273}}]
```

Useless.
Explain how you could have known in advance that A is not invertible.

☐Answer:

This matrix A hits 4D vectors into 7D.

When you hit all of 4D with A, you cover at most four dimensions of 7D.

That leaves three full dimensions of 7D that are not accessible via hits with A.

When you try to undo A inside these three extra dimensions, you are up the old creek.

And that's why A is not invertible.

T.2) The absolute value of determinant for square matrices A in higher dimensions.

If Det[A] ≠ 0, then the corresponding linear system A.X = Y has exactly one solution (exactly determined).

If Det[A] = 0, then the linear system A.X = Y either has no solution (overdetermined) or has many solutions (underdetermined)

□ T.2.a.i) For a square matrix, |Det[A]| is the product of A's stretch SVD factors

Look at these matrices that hit and hang in the same dimension:

```
Lots of folks call these square matrices
because they have the same number of rows and columns.
This is the same as saying that hitdim = hangdim
```

```
Clear[a, i, j];
dim = 4;
a[i_, j_] := Random[Real, {-7, 7}]

A = Table[a[i, j], {i, 1, dim}, {j, 1, dim}];

MatrixForm[A]
```

$$\begin{pmatrix} 6.3863 & -1.14207 & 4.38674 & -0.663892 \\ 3.64191 & 2.22872 & -2.5139 & -0.342515 \\ -0.134397 & 6.02375 & -3.9469 & 2.83038 \\ -3.82483 & -1.84578 & 4.30812 & 5.45644 \end{pmatrix}$$

The rank of this matrix is:

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

```
4
```

Here is the product of the four stretch factors and a calculation of the absolute value of the determinant of A:

```
Clear[stretch, k];
stretch[k_] := SingularValues[A][[2]][[k]];
```

```
rank
∏   stretch[k]
k=1

Abs[Det[A]]
```

```
489.824
489.824
```

More:

```
Clear[a, i, j];
dim = Random[Integer, {4, 8}];

a[i_, j_] := Random[Real, {-7, 7}]

A = Table[a[i, j], {i, 1, dim}, {j, 1, dim}];

MatrixForm[A]
```

$$\begin{pmatrix} 0.640526 & -2.41306 & -0.250089 & -5.04717 & 0.973158 & 4.84308 & 0.7962 \\ 1.58685 & -1.01485 & 3.40954 & -5.47623 & 4.94495 & 3.75643 & -1.076 \\ -1.92066 & 4.73268 & -4.12966 & 6.0359 & -5.09583 & -0.42155 & -1.437 \\ 1.26364 & -5.00849 & 5.81231 & 5.62663 & -6.70951 & -2.85157 & -1.983 \\ -1.29637 & 5.16328 & 1.60649 & -3.75702 & 0.758687 & -5.59315 & -4.316 \\ -4.32066 & -3.32582 & 6.81271 & 2.34079 & -6.22483 & 4.09573 & 1.2504 \\ -0.488471 & 2.10421 & 2.43818 & 3.1347 & -0.778958 & -2.04422 & -2.577 \\ -6.48259 & -0.207503 & 2.81566 & -4.87503 & -0.241277 & -1.61435 & 0.1326 \end{pmatrix}$$

The rank of this matrix is:

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

```
8
```

Here is the product of the five stretch factors and a calculation of the absolute value of the determinant of A:

```
Clear[stretch, k];
stretch[k_] := SingularValues[A][[2]][[k]];

rank
∏   stretch[k]
k=1

Abs[Det[A]]
```

```
314050.
314050.
```

Another:

```
Clear[a, i, j];
dim = Random[Integer, {4, 7}];

a[i_, j_] := Random[Real, {-7, 7}]

A = Table[a[i, j], {i, 1, dim}, {j, 1, dim}];

MatrixForm[A]
```

$$\begin{pmatrix} -2.92062 & -5.28853 & 0.319905 & 5.40749 \\ -3.69579 & -2.38426 & 6.06942 & -3.35384 \\ 3.79268 & 2.51153 & -3.36876 & 0.511465 \\ -2.42837 & -2.44425 & 6.20909 & -4.85648 \end{pmatrix}$$

The rank of this matrix is:

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

```
4
```

Here is the product of the five stretch factors and a calculation of the absolute value of the determinant of A:

```
Clear[stretch, k];
stretch[k_] := SingularValues[A][[2]][[k]];

rank
∏   stretch[k]
k=1

Abs[Det[A]]
```

```
47.2003
47.2003
```

What's the message?

□ Answer:

The message is that even in higher dimensions, the absolute value of the determinant of a square matrix A is the same as the product of A's stretch factors.

It's quite legal to take this as the definition of the absolute value of the determinant of A.

□ T.2.a.ii) Non-zero determinants and exactly determined linear systems

You are given a linear system with the same number of equations as unknowns. You take the coefficient matrix A and use *Mathematica* to calculate the determinant of A.
It turns out to be non-zero.
You announce that the linear system is exactly determined in the sense that it has exactly one solution.
You are right. Why are you right?

□ Answer:

Just for the sake of explanation, agree that you have 6 equations in 6 unknowns. This means the coefficient matrix A hits on 6D and hangs in 6D.

You are given

$$|\text{Det}[A]| \neq 0$$

and you know that

$|\text{Det}[A]|$ is the product of all of A's 6 stretch factors.

Because

$$|\text{Det}[A]| \neq 0,$$

you are left with the valid conclusion that none of A's stretch factors is 0.

And this tells you that hits with A on 6D fill up all of 6D.

This also tells you that

rank of A = hitdim = 6,

so that A is of full rank.

This means that no matter what 6D Y you go with, you are guaranteed that A.X = Y has at least one solution.

And because A is of full rank, there is exactly one solution.

□ T.2.a.iii) Zero determinants and trouble

You are given a linear system with the same number of equations as unknowns. You take the coefficient matrix A and use *Mathematica* to calculate the determinant of A.
It turns out to be zero.
You announce that the linear system is either under determined (has more than one solution) or is overdetermined (has no solution).
You are right. Why are you right?

□ Answer:

Just for the sake of explanation, agree that you have 7 equations in 7 unknowns. This means the coefficient matrix A hits on 7D and hangs in 7D.

You are given

$$|\text{Det}[A]| = 0$$

and you know that

$|\text{Det}[A]|$ is the product of A's 7 stretch factors.

Because

$$|\text{Det}[A]| = 0,$$

you are left with the valid conclusion that at least one of A's stretch factors is 0.

And this tells you that hits with A on 7D fill up at most six of the seven dimensions of 7D. This allows for plenty of Y's for which A.X = Y has no solution.

This also allows for plenty of Y's for which A.X = Y does have a solution. But in this case, the zero stretch factor tells you that

rank of A < hitdim

so that A is not of full rank. This tells you that any one solution is accompanied by many more solutions.

□T.2.b) Vandermonde matrices

Given distinct (different) numbers t[1], t[2], t[3],t[4] and t[5] make this matrix:

```
dim = 5;
Clear[t, x, y, i, j];
A = Table[t[i]^(j-1), {i, 1, dim}, {j, 1, dim}];

MatrixForm[A]
```

$$\begin{pmatrix} 1 & t[1] & t[1]^2 & t[1]^3 & t[1]^4 \\ 1 & t[2] & t[2]^2 & t[2]^3 & t[2]^4 \\ 1 & t[3] & t[3]^2 & t[3]^3 & t[3]^4 \\ 1 & t[4] & t[4]^2 & t[4]^3 & t[4]^4 \\ 1 & t[5] & t[5]^2 & t[5]^3 & t[5]^4 \end{pmatrix}$$

Some folks like to call this a Vandermonde matrix.

Use this matrix as the coefficient matrix to make this linear system:

```
X = Table[x[k], {k, 1, dim}];
Y = Table[y[k], {k, 1, dim}];

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

$$\begin{aligned} &x[1] + t[1]\,x[2] + t[1]^2\,x[3] + t[1]^3\,x[4] + t[1]^4\,x[5] == y[1] \\ &x[1] + t[2]\,x[2] + t[2]^2\,x[3] + t[2]^3\,x[4] + t[2]^4\,x[5] == y[2] \\ &x[1] + t[3]\,x[2] + t[3]^2\,x[3] + t[3]^3\,x[4] + t[3]^4\,x[5] == y[3] \\ &x[1] + t[4]\,x[2] + t[4]^2\,x[3] + t[4]^3\,x[4] + t[4]^4\,x[5] == y[4] \\ &x[1] + t[5]\,x[2] + t[5]^2\,x[3] + t[5]^3\,x[4] + t[5]^4\,x[5] == y[5] \end{aligned}$$

Explain this:
This linear system is exactly determined in the sense that there is exactly one solution {x[1],y[2],x[3],x[4],x[5]} corresponding to any given {y[1],y[2],y[3],y[4],y[5]}.

□Answer:

Let *Mathematica* calculate the absolute value of the determinant of A in factored form:

```
Abs[Factor[Det[A]]]
```

$$\begin{aligned} &\text{Abs}[(t[1]-t[2])\,(t[1]-t[3])\,(t[2]-t[3])\,(t[1]-t[4])\,(t[2]-t[4]) \\ &\quad (t[3]-t[4])\,(t[1]-t[5])\,(t[2]-t[5])\,(t[3]-t[5])\,(t[4]-t[5])] \end{aligned}$$

Because the t[i]'s are all different, this is the product of non-zero numbers, so there is no way for this to be zero.

This is enough to tell you that the linear system is exactly determined in the sense that there is exactly one solution {x[1],y[2],x[3],x[4],x[5]} corresponding to any given {y[1],y[2],y[3],y[4],y[5]}.

T.3) Columns and rank; rows and rank: How to make matrices that are not of full rank

□T.3.a.i) Columns and rank

Here is a matrix entered via its vertical columns:

```
Clear[column, j];
hitdim = 4;
column[1] = {0.0, 0.0, 3.3, 0.0, 0.2};
column[2] = {0.0, -2.0, -1.3, 1.0, 0.0};
column[3] = {1.4, 0.0, 0.0, 2.0, -3.0};
column[4] = 1.2 column[1] - 3.5 column[2];

A = Transpose[Table[column[j], {j, 1, 4}]];

MatrixForm[A]
```

$$\begin{pmatrix} 0 & 0 & 1.4 & 0 \\ 0 & -2. & 0 & 7. \\ 3.3 & -1.3 & 0 & 8.51 \\ 0 & 1. & 2. & -3.5 \\ 0.2 & 0 & -3. & 0.24 \end{pmatrix}$$

Note that column[4] is a combination of some of other columns.
Now calculate the rank of this matrix:

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

3

This matrix hits on 4D; so

rank < hitdim

and the matrix is not of full rank.
The parts below attempt to give an explanation about why this approach was guaranteed to produce a matrix not of full rank.

Agree that

{col[1], col[2], col[3], . . ., col[hitdim]}

are the vertical columns of a given matrix A.
Explain this statement:
Saying that matrix A is not of full rank is the same as saying that there is a vector

{c[1], c[2], . . ., c[hitdim]}

in hitdimD with at least one non-zero entry such that

$$\sum_{i=1}^{\text{hitdim}} c[i]\,\text{column}[i] = \{0, 0, \ldots, 0\}.$$

□Answer:

Saying that A is not of full rank is the same as saying that a hit with A squashes at least one non-zero vector in hitdimD onto the vector of all zeroes.

This is the same as saying there is a vector

{c[1], c[2], . . ., c[hitdim]}

in hitdimD with at least one non-zero entry such that

A.{c[1], c[2], . . ., c[hitdim]} = {0, 0, ..., 0}.

Because

$$A.\{c[1], c[2], \ldots, c[\text{hitdim}]\} = \sum_{i=1}^{\text{hitdim}} c[i]\,\text{column}[i],$$

this is the same as saying that there is a vector

{c[1], c[2], . . ., c[hitdim]}

in hitdimD with at least one non-zero entry such that

$$\sum_{i=1}^{\text{hitdim}} c[i]\,\text{column}[i] = \{0, 0, \ldots, 0\}.$$

Some folks say that this is the same
as saying that the matrix A has linearly dependent columns.

□T.3.a.ii) More on columns and rank

Expound on this statement:
Saying that matrix A is not of full rank is the same as saying that you can express at least one of the vertical columns of A in terms of some of the others.

□Answer:

Saying that matrix A is not of full rank is the same as saying that there is a vector

{c[1], c[2], . . ., c[hitdim]}

in hitdimD with at least one non-zero entry such that

$$\sum_{i=1}^{\text{hitdim}} c[i]\,\text{column}[i] = \{0, 0, \ldots, 0\}.$$

Saying that matrix A is not of full rank is the same as saying that there is a vector

Take i_0 with $c[i_0] \neq 0$ and note that

$$-c[i_0]\,\text{column}[i_0]$$

$$= \sum_{i=1}^{\text{hitdim}} c[i]\,\text{column}[i] - c[i_0]\,\text{column}[i_0]$$

$$= \sum_{i=1}^{i_0-1} c[i]\,\text{column}[i] + \sum_{i=i_0+1}^{\text{hitdim}} c[i]\,\text{column}[i].$$

This gives

$$-c[i_0]\,\text{column}[i_0] = \sum_{i=1}^{i_0-1} c[i]\,\text{column}[i] + \sum_{i=i_0+1}^{\text{hitdim}} c[i]\,\text{column}[i].$$

so that all the influence of column[i_0] on the right is gone.

Divide through by $-c[i_0]$ to get

$$\text{column}[i_0] = \sum_{i=1}^{i_0-1} \left(-\frac{c[i]}{c[i_0]}\right)\text{column}[i] + \sum_{i=i_0+1}^{\text{hitdim}} \left(-\frac{c[i]}{c[i_0]}\right)\text{column}[i]$$

which exhibits column[i_0] as a combination of the other columns.

□T.3.a.iii) Example: Columns and rank

Here is a matrix A entered via its vertical columns:

```
Clear[column, j];
hitdim = 5;
column[1] = {0.43, -1.32, -2.26, 1.47, 4.13, 2.32};
column[2] = {-0.34, 0.34, 0.0, 7.89, -2.96, 0.89};
column[3] = {2.58, 1.34, 3.82, 2.84, -1.44, 0.78};
column[4] = {0.61, 1.35, 1.91, 17.20, -6.64, 2.17};
column[5] = {1.72, -0.65, -0.35, 2.89, 3.41, 2.71};
```

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

MatrixForm[A]
```

$$\begin{pmatrix} 0.43 & -0.34 & 2.58 & 0.61 & 1.72 \\ -1.32 & 0.34 & 1.34 & 1.35 & -0.65 \\ -2.26 & 0 & 3.82 & 1.91 & -0.35 \\ 1.47 & 7.89 & 2.84 & 17.2 & 2.89 \\ 4.13 & -2.96 & -1.44 & -6.64 & 3.41 \\ 2.32 & 0.89 & 0.78 & 2.17 & 2.71 \end{pmatrix}$$

Calculate the rank of this matrix:

```
rank = Length[SingularValues[A][[2]]]
```
```
3
```

This matrix hits on 5D; its rank is 3.
So this matrix is not of full rank.
Exhibit at least one column that is a combination of at least some of the other columns.

□ **Answer:**

Look at:

```
NullSpace[A]
```
```
{{0.0340453, 0.868496, 0.234147, -0.434248, -0.0340453},
 {0.66757, -0.108036, 0.306776, 0.0540179, -0.66757}}
```

Put:

```
Clear[c];
{c[1], c[2], c[3], c[4], c[5]} = NullSpace[A][[1]]
```
```
{0.0340453, 0.868496, 0.234147, -0.434248, -0.0340453}
```

This tells you:

```
A.{c[1], c[2], c[3], c[4], c[5]}
```
```
{0, 0, 0, 0, 0}
```

This is the same as:

```
Sum[c[i] column[i], {i, 1, 5}]
```
```
{0, 0, 0, 0, 0}
```

This tells you how to exhibit column[3] as a combination of the other columns.

```
column[3]

Sum[-c[i] column[i] / c[3], {i, 1, 2}] + Sum[-c[i] column[i] / c[3], {i, 4, 5}]
```

```
{2.58, 1.34, 3.82, 2.84, -1.44, 0.78}
{2.58, 1.34, 3.82, 2.84, -1.44, 0.78}
```

It also tells you tells you how to exhibit column[1] as a combination of the other columns.

```
column[1]

Sum[-c[i] column[i] / c[1], {i, 2, 5}]
```
```
{0.43, -1.32, -2.26, 1.47, 4.13, 2.32}
{0.43, -1.32, -2.26, 1.47, 4.13, 2.32}
```

□ T.3.b) Rows and rank: How to make matrices that are not of full rank

Here is a matrix entered via its horizontal rows:

```
Clear[row, j];
row[1] = {0.0, 0.0, 3.3, 0.0, 0.2};
row[2] = {0.0, -2.0, -1.3, 1.0, 0.0};
row[3] = {1.4, 0.0, 0.0, 2.0, -3.0};
row[4] = {2.4, -1.0, 1.0, 0.0, -1.0};
row[5] = 1.2 row[1] - 3.5 row[2];

A = Table[row[j], {j, 1, 5}];

MatrixForm[A]
```

$$\begin{pmatrix} 0 & 0 & 3.3 & 0 & 0.2 \\ 0 & -2. & -1.3 & 1. & 0 \\ 1.4 & 0 & 0 & 2. & -3. \\ 2.4 & -1. & 1. & 0 & -1. \\ 0 & 7. & 8.51 & -3.5 & 0.24 \end{pmatrix}$$

Examine the code to see that horizontal
row[5] = 1.2 row[1] − 3.5 row[2]
is a combination of some of other horizontal rows.
Explain why this fact guarantees that A is not of full rank.

□ **Answer:**

Look at A:

```
MatrixForm[A]
```

$$\begin{pmatrix} 0 & 0 & 3.3 & 0 & 0.2 \\ 0 & -2. & -1.3 & 1. & 0 \\ 1.4 & 0 & 0 & 2. & -3. \\ 2.4 & -1. & 1. & 0 & -1. \\ 0 & 7. & 8.51 & -3.5 & 0.24 \end{pmatrix}$$

Look at A^t:

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

$$\begin{pmatrix} 0 & 0 & 1.4 & 2.4 & 0 \\ 0 & -2. & 0 & -1. & 7. \\ 3.3 & -1.3 & 0 & 1. & 8.51 \\ 0 & 1. & 2. & 0 & -3.5 \\ 0.2 & 0 & -3. & -1. & 0.24 \end{pmatrix}$$

The horizontal rows of A^t are the vertical columns of A.

Because

row[5] of A = 1.2 row[1] of A − 3.5 row[2] of A

you know that

column[5] of A^t = 1.2 column[1] of A^t − 3.5 column[2] of A^t

Part a) above tells you that

rank of A^t < 5

and because

rank of A^t = rank of A,

you are guaranteed that

rank of A < 5.

T.4) For under determined linear systems A.X = Y, Xapprox = PseudoInverse[A].Y is the solution of smallest norm

□ T.5.a) Xapprox = PseudoInverse[A].Y is the solution of smallest norm

Go with this linear system:

```
hitdim = 5; hangdim = 3;
Clear[a, i, j];
a[i_, j_] := Random[Real, {-2, 2}];

A = Table[a[i, j], {i, 1, hangdim}, {j, 1, hitdim}];
Y = {1.0, 0.0, 2.0};


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

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

```
-0.84165 x[1] + 1.36093 x[2] - 1.03045 x[3] - 1.9947 x[4] + 1.22729 x[5] == 1
-0.177825 x[1] + 0.931661 x[2] - 0.208496 x[3] + 0.0617493 x[4] - 0.666816 x
-1.15974 x[1] + 0.246507 x[2] - 0.882309 x[3] - 1.9856 x[4] - 0.89386 x[5] =
```

To look for solutions, determine the rank of A:

```
rank = Length[SingularValues[A][[2]]]
```
```
3
```

Compare with:

```
hitdim
```
```
5
```
```
rank < hitdim
```
```
True
```

Bad news. The coefficient matrix A is of not of full rank.
Now build

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

```
Xapprox = PseudoInverse[A].Y
```
```
{-0.337209, -0.276977, -0.200802, -0.62398, -0.292056}
```

Try it out:

```
A.Xapprox
Y
```
```
{1., 0, 2.}
{1., 0, 2.}
```

Xapprox is a solution and because A is of not of full rank, Xapprox **is not the only** solution.

Explain why Xapprox is the solution of smallest norm.
In other words, explain why
‖Xapprox‖ ⩽ ‖any other solution‖.

□ **Answer:**

To get a hold of all the solutions, look at the alignerframe vectors that are squashed to {0,0,0} when they are hit with A:

```
NullSpace[A]
```
```
{{-0.89764, -0.0390258, 0.163434, 0.371139, 0.168122},
 {0, -0.197712, -0.910352, 0.361135, 0.0418463}}
```

All the solutions are parameterized by:

```
Clear[s, t, solution];
solution[s_, t_] = Xapprox + s NullSpace[A][[1]] + t NullSpace[A][[2]]
```

```
{-0.337209 - 0.89764 s, -0.276977 - 0.0390258 s - 0.197712 t,
 -0.200802 + 0.163434 s - 0.910352 t, -0.62398 + 0.371139 s + 0.361135 t,
 -0.292056 + 0.168122 s + 0.0418463 t}
```

Try them out:

```
Expand[A.solution[s, t]]
Y
```

```
{1., 0, 2.}
{1., 0, 2.}
```

Good

Now look at

|| solution[s, t] ||^2 = solution[s, t].solution[s, t]:

```
Expand[solution[s, t].solution[s, t]]
```

```
0.705395 + 1. s^2 + 1. t^2
```

```
Xapprox.Xapprox
```

```
0.705395
```

This tells you that when you go with

solution[s, t] = Xapprox + s NullSpace[A][[1]] + t NullSpace[A][[2]],

you get

|| solution[s, t] ||^2 = Xapprox.Xapprox + $s^2 + t^2$.

So the solution solution[s,t] with smallest norm is

solution[0, 0] = Xapprox .

The upshot:

Xapprox

is the solution of smallest norm.

T.5) Plucking the coefficient matrix off a linear system

Thanks to Horacio Porta of the University of Illinois
for suggesting this technique.

□ T.5.a) Going from linear formulas to matrices

Here's a common way of delivering a linear system:
First, you are given a list of linear formulas:

```
Clear[x];
linearformulas =
  {2.1 x[1] + 2.3 x[2] + 3.4 x[3] - 5.4 x[4] - 0.5 x[5] - 2.5 x[6],
   -1.5 x[1] - 0.4 x[2] + 13.5 x[4] - 0.7 x[5] + 3.7 x[6],
   -0.7 x[1] + 3.4 x[3] + 4.5 x[4] - 2.1 x[5] + 4.2 x[6],
   3.4 x[2] - 0.6 x[3] - 1.2 x[4] - 0.6 x[5],
   1.6 x[1] + 2.8 x[3] + 3.6 x[4] + 2.1 x[5] - 0.7 x[6],
   2.3 x[1] + 3.2 x[3] + 4.9 x[5]};

ColumnForm[linearformulas]
```

```
2.1 x[1] + 2.3 x[2] + 3.4 x[3] - 5.4 x[4] - 0.5 x[5] - 2.5 x[6]
-1.5 x[1] - 0.4 x[2] + 13.5 x[4] - 0.7 x[5] + 3.7 x[6]
-0.7 x[1] + 3.4 x[3] + 4.5 x[4] - 2.1 x[5] + 4.2 x[6]
3.4 x[2] - 0.6 x[3] - 1.2 x[4] - 0.6 x[5]
1.6 x[1] + 2.8 x[3] + 3.6 x[4] + 2.1 x[5] - 0.7 x[6]
2.3 x[1] + 3.2 x[3] + 4.9 x[5]
```

Then you are given values
y[1], . . . ,y[6],
one for each linear formula, and you set
linearformula[k] = y[k]:

```
Clear[y];
Y = {2.4, -0.4, 2.7, -1.9, 0.8, 1.6};
y[k_] := Y[[k]];

linearsystem =
ColumnForm[Thread[Table[
    linearformulas[[k]] == y[k], {k, 1, Length[linearformulas]}]]]
```

```
2.1 x[1] + 2.3 x[2] + 3.4 x[3] - 5.4 x[4] - 0.5 x[5] - 2.5 x[6] == 2.4
-1.5 x[1] - 0.4 x[2] + 13.5 x[4] - 0.7 x[5] + 3.7 x[6] == -0.4
-0.7 x[1] + 3.4 x[3] + 4.5 x[4] - 2.1 x[5] + 4.2 x[6] == 2.7
3.4 x[2] - 0.6 x[3] - 1.2 x[4] - 0.6 x[5] == -1.9
1.6 x[1] + 2.8 x[3] + 3.6 x[4] + 2.1 x[5] - 0.7 x[6] == 0.8
2.3 x[1] + 3.2 x[3] + 4.9 x[5] == 1.6
```

Once you have the coefficient matrix A for this linear system, then going about solving the linear system is not bad at all.
But reading off and manually typing A into *Mathematica* code is a real pain in the butt.
How can you get *Mathematica* pluck out that coefficient matrix for you?

□ Answer:

Here's one way:

```
xvariables = Table[x[k], {k, 1, 6}];
A = Outer[D, {linearformulas}, xvariables][[1]];

MatrixForm[A]
```

$$\begin{pmatrix} 2.1 & 2.3 & 3.4 & -5.4 & -0.5 & -2.5 \\ -1.5 & -0.4 & 0 & 13.5 & -0.7 & 3.7 \\ -0.7 & 0 & 3.4 & 4.5 & -2.1 & 4.2 \\ 0 & 3.4 & -0.6 & -1.2 & -0.6 & 0 \\ 1.6 & 0 & 2.8 & 3.6 & 2.1 & -0.7 \\ 2.3 & 0 & 3.2 & 0 & 4.9 & 0 \end{pmatrix}$$

Check it out:

```
hitdim = 6; hangdim = 6;
X = Table[x[k], {k, 1, hitdim}];
Y = Table[y[k], {k, 1, hangdim}];

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

```
2.1 x[1] + 2.3 x[2] + 3.4 x[3] - 5.4 x[4] - 0.5 x[5] - 2.5 x[6] == 2.4
-1.5 x[1] - 0.4 x[2] + 13.5 x[4] - 0.7 x[5] + 3.7 x[6] == -0.4
-0.7 x[1] + 3.4 x[3] + 4.5 x[4] - 2.1 x[5] + 4.2 x[6] == 2.7
3.4 x[2] - 0.6 x[3] - 1.2 x[4] - 0.6 x[5] == -1.9
1.6 x[1] + 2.8 x[3] + 3.6 x[4] + 2.1 x[5] - 0.7 x[6] == 0.8
2.3 x[1] + 3.2 x[3] + 4.9 x[5] == 1.6
```

Compare:

```
linearsystem
```

```
2.1 x[1] + 2.3 x[2] + 3.4 x[3] - 5.4 x[4] - 0.5 x[5] - 2.5 x[6] == 2.4
-1.5 x[1] - 0.4 x[2] + 13.5 x[4] - 0.7 x[5] + 3.7 x[6] == -0.4
-0.7 x[1] + 3.4 x[3] + 4.5 x[4] - 2.1 x[5] + 4.2 x[6] == 2.7
3.4 x[2] - 0.6 x[3] - 1.2 x[4] - 0.6 x[5] == -1.9
1.6 x[1] + 2.8 x[3] + 3.6 x[4] + 2.1 x[5] - 0.7 x[6] == 0.8
2.3 x[1] + 3.2 x[3] + 4.9 x[5] == 1.6
```

Worked like a charm.

The solution of this linear system is:

```
X = Inverse[A].Y
```

```
{21.937, -2.64839, -6.65207, 0.21869, -5.62622, 6.63661}
```

Check:

```
Y
A.X
```

```
{2.4, -0.4, 2.7, -1.9, 0.8, 1.6}
{2.4, -0.4, 2.7, -1.9, 0.8, 1.6}
```

Had it all the way.

□ T.5.a.ii) The weird instruction Outer[D,{linearformulas}, xvariables][[1]];

Go back to:

```
ColumnForm[linearformulas]
```

```
2.1 x[1] + 2.3 x[2] + 3.4 x[3] - 5.4 x[4] - 0.5 x[5] - 2.5 x[6]
-1.5 x[1] - 0.4 x[2] + 13.5 x[4] - 0.7 x[5] + 3.7 x[6]
-0.7 x[1] + 3.4 x[3] + 4.5 x[4] - 2.1 x[5] + 4.2 x[6]
3.4 x[2] - 0.6 x[3] - 1.2 x[4] - 0.6 x[5]
1.6 x[1] + 2.8 x[3] + 3.6 x[4] + 2.1 x[5] - 0.7 x[6]
2.3 x[1] + 3.2 x[3] + 4.9 x[5]
```

```
xvariables = Table[x[k], {k, 1, 6}];
A = Outer[D, {linearformulas}, xvariables][[1]];

MatrixForm[A]
```

$$\begin{pmatrix} 2.1 & 2.3 & 3.4 & -5.4 & -0.5 & -2.5 \\ -1.5 & -0.4 & 0 & 13.5 & -0.7 & 3.7 \\ -0.7 & 0 & 3.4 & 4.5 & -2.1 & 4.2 \\ 0 & 3.4 & -0.6 & -1.2 & -0.6 & 0 \\ 1.6 & 0 & 2.8 & 3.6 & 2.1 & -0.7 \\ 2.3 & 0 & 3.2 & 0 & 4.9 & 0 \end{pmatrix}$$

How did that weird instruction deliver the goods?

□ Answer:

The instruction

Outer[D,{linearformulas},xvariables][[1]]

just took each linear formula and replaced it by its gradient.

T.6) Tips on entering matrices in Mathematica

□ T.6.a.i) Entering big matrices in *Mathematica* by rows

Here is a matrix

$$A = \begin{pmatrix} 0.43 & -0.34 & 2.58 \\ -1.32 & 0.34 & 1.34 \\ -2.26 & 0 & 3.82 \\ 1.47 & 7.89 & 2.84 \\ 4.13 & -2.96 & -1.44 \\ 2.32 & 0.89 & 0.78 \end{pmatrix}$$

How do you enter this matrix in *Mathematica* by rows?

□ Answer:

Here's one way:

```
Clear[row];
row[1] = {0.43, 0.34, 2.58};
```

```
row[2] = {-1.32, 0.34, 1.34};
row[3] = {-2.26, 0, 3.82};
row[4] = {1.47, 7.89, 2.84};
row[5] = {4.13, -2.96, -1.44};
row[6] = {2.32, 0.89, 0.78};
A = Table[row[k], {k, 1, 6}];
MatrixForm[A]
```

$$\begin{pmatrix} 0.43 & 0.34 & 2.58 \\ -1.32 & 0.34 & 1.34 \\ -2.26 & 0 & 3.82 \\ 1.47 & 7.89 & 2.84 \\ 4.13 & -2.96 & -1.44 \\ 2.32 & 0.89 & 0.78 \end{pmatrix}$$

Compare:

$$A = \begin{pmatrix} 0.43 & -0.34 & 2.58 \\ -1.32 & 0.34 & 1.34 \\ -2.26 & 0 & 3.82 \\ 1.47 & 7.89 & 2.84 \\ 4.13 & -2.96 & -1.44 \\ 2.32 & 0.89 & 0.78 \end{pmatrix}$$

□T.6.a.ii) Entering big matrices in *Mathematica* by columns

Here is a matrix

$$A = \begin{pmatrix} 0.43 & -0.34 & 2.58 \\ -1.32 & 0.34 & 1.34 \\ -2.26 & 0 & 3.82 \\ 1.47 & 7.89 & 2.84 \\ 4.13 & -2.96 & -1.44 \\ 2.32 & 0.89 & 0.78 \end{pmatrix}$$

How do you enter this matrix in *Mathematica* by columns?

□Answer:

Here's one way:

```
Clear[col];
col[1] = {0.43, -1.32, -2.26, 1.47, 4.13, 2.32};
col[2] = {-0.34, 0.34, 0, 7.89, -2.96, 0.89};
col[3] = {2.58, 1.34, 3.82, 2.84, -1.44, 0.78};
```

```
A = Transpose[Table[col[k], {k, 1, 3}]];
MatrixForm[A]
```

$$\begin{pmatrix} 0.43 & -0.34 & 2.58 \\ -1.32 & 0.34 & 1.34 \\ -2.26 & 0 & 3.82 \\ 1.47 & 7.89 & 2.84 \\ 4.13 & -2.96 & -1.44 \\ 2.32 & 0.89 & 0.78 \end{pmatrix}$$

Compare:

$$A = \begin{pmatrix} 0.43 & -0.34 & 2.58 \\ -1.32 & 0.34 & 1.34 \\ -2.26 & 0 & 3.82 \\ 1.47 & 7.89 & 2.84 \\ 4.13 & -2.96 & -1.44 \\ 2.32 & 0.89 & 0.78 \end{pmatrix}$$

□T.6.a.iii) Entering big matrices in *Mathematica* by template

Here is a matrix

$$A = \begin{pmatrix} 0.43 & -0.34 & 2.58 \\ -1.32 & 0.34 & 1.34 \\ -2.26 & 0 & 3.82 \\ 1.47 & 7.89 & 2.84 \\ 4.13 & -2.96 & -1.44 \\ 2.32 & 0.89 & 0.78 \end{pmatrix}$$

How do you enter this matrix in *Mathematica* by template?

□Answer:

Here's one way:

Notice that A hits on 3D and hangs in 6D:

Go to the **Input** menu and select **Create Table/Matrix Palette**.

Check the **Matrix** button and select **6** rows and **3** columns and hit the **OK** button.

The result will be this:

$$\begin{pmatrix} \square & \square & \square \\ \square & \square & \square \\ \square & \square & \square \\ \square & \square & \square \\ \square & \square & \square \\ \square & \square & \square \end{pmatrix}$$

Now type in the entries and you are off and running.

T.7) Square matrices as spreadsheets: Gaussian elimination and reduced row echelon form. How it works and when you might want to use it

□T.7.a.i) The basic idea

What is the basic idea behind Gaussian elimination?

□Answer:

You probably met Gaussian elimination way back in junior high school.

The idea is that when you have a system of linear equations, and you add non-zero multiples of one equation to another, then you change the linear system but do not change its solutions.

For instance if your linear system is

$$\begin{pmatrix} 1 & 1 \\ 2 & 4 \end{pmatrix}.\{x,y\} = \{2, 10\},$$

Add -2 times the first equation to the second equation and keep the first equation the same to get

$$\begin{pmatrix} 1 & 1 \\ 2-2 & 4-2 \end{pmatrix}.\{x,y\} = \{2, 10 - 4\},$$

This is the same as

$$\begin{pmatrix} 1 & 1 \\ 0 & 2 \end{pmatrix}.\{x,y\} = \{2, 6\},$$

And now you add $-\frac{1}{2}$ times the second equation to the first and keep the second equation the same to get

$$\begin{pmatrix} 1 & 1-1 \\ 0 & 2 \end{pmatrix}.\{x,y\} = \{2 -3, 6\},$$

This is the same as

$$\begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}.\{x,y\} = \{-1,6\},$$

Divide 2 through the second equation to get

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.\{x,y\} = \{-1 , 3\},$$

At this point you read off the solution {x,y} = {-1,3}.

Try it out

```
{{1, 1}, {2, 4}}.{-1, 3} == {2, 10}
```

```
True
```

□T.7.a.ii) The augmented matrix for the linear system

$$\begin{pmatrix} \mathbf{1} & \mathbf{1} \\ \mathbf{2} & \mathbf{4} \end{pmatrix}.\mathbf{\{x,y\} = \{2, 10\}} \text{ is } \begin{pmatrix} \mathbf{1} & \mathbf{1} & \mathbf{2} \\ \mathbf{2} & \mathbf{4} & \mathbf{10} \end{pmatrix}.$$

Spread sheet operations on the rows the augmented matrix give you the solution via reduced row echelon form

Stay with the same linear system

$$\begin{pmatrix} 1 & 1 \\ 2 & 4 \end{pmatrix}.\{x,y\} = \{2, 10\},$$

The augmented matrix for this linear system is :$\begin{pmatrix} 1 & 1 & 2 \\ 2 & 4 & 10 \end{pmatrix}$

Use spread sheet operations on the rows of the augmented matrix to come up with the solution to the linear system.

□Answer:

You do the same thing you did above, but this time you do it to the rows of the augmented matrix.

Go with the augmented matrix

$$\begin{pmatrix} 1 & 1 & 2 \\ 2 & 4 & 10 \end{pmatrix}$$

Add -2 times the first row to the second row and keep the first row the same to get

$$\begin{pmatrix} 1 & 1 & 2 \\ 2-2 & 4-2 & 10-4 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 2 \\ 0 & 2 & 6 \end{pmatrix}$$

And now you add $-\frac{1}{2}$ times the second row to the first and keep the second row the same to get

$$\begin{pmatrix} 1-0 & 1-1 & 2-3 \\ 0 & 2 & 6 \end{pmatrix} = \begin{pmatrix} 1 & 0 & -1 \\ 0 & 2 & 6 \end{pmatrix}$$

And then you divide the second row by 2 to get

$$\begin{pmatrix} 1 & 0 & -1 \\ 0 & \frac{2}{2} & \frac{6}{2} \end{pmatrix} = \begin{pmatrix} 1 & 0 & -1 \\ 0 & 1 & 3 \end{pmatrix}$$

And read off the solution {x,y} = {-1,3}.

The beauty of it is that Mathematica will do this for you at a flip of the finger:

```
MatrixForm[RowReduce[ ( 1  1  2
                        2  4  10 )]]
```

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

Some folks call this the reduced row echelon form of the augmented matrix.
Fancy words for such a simple idea,

□T.7.a.iii) A 3D example powered by *Mathematica*

Here's a linear system:

```
A = ( 2.1   2.3   4.2
      4.5  -0.6   3.1
      2.1   1.9  -2.8 );
dim = 3;
Clear[x, j];
X = Table[x[j], {j, 1, dim}];
Y = {1.2, 2.5, 3.0};
linearsystem = ColumnForm[Thread[A.X == Y]]
```

```
2.1 x[1] + 2.3 x[2] + 4.2 x[3] == 1.2
4.5 x[1] - 0.6 x[2] + 3.1 x[3] == 2.5
2.1 x[1] + 1.9 x[2] - 2.8 x[3] == 3.
```

Use the reduced row echelon form of the augmented matrix to solve this system.

□Answer:

The linear system is:

```
linearsystem
```

```
2.1 x[1] + 2.3 x[2] + 4.2 x[3] == 1.2
4.5 x[1] - 0.6 x[2] + 3.1 x[3] == 2.5
2.1 x[1] + 1.9 x[2] - 2.8 x[3] == 3.
```

The augmented matrix is

```
augmentedA = AppendRows[A, Transpose[{Y}]];
MatrixForm[augmentedA]
```

$$\begin{pmatrix} 2.1 & 2.3 & 4.2 & 1.2 \\ 4.5 & -0.6 & 3.1 & 2.5 \\ 2.1 & 1.9 & -2.8 & 3. \end{pmatrix}$$

That's the coefficient matrix A sitting on the left and
Y sitting in the fourth column.

And look at this:

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

$$\begin{pmatrix} 1 & 0 & 0 & 0.785544 \\ 0 & 1 & 0 & 0.305999 \\ 0 & 0 & 1 & -0.274629 \end{pmatrix}$$

Pick out the fourth column and put:

```
X = Table[rowechelon[[j, 4]], {j, 1, 3}]
```

```
{0.785544, 0.305999, -0.274629}
```

Check:

```
A.X
```

```
{1.2, 2.5, 3.}
```

```
Y
```

```
{1.2, 2.5, 3.}
```

Got it.

If you want to see spread sheet operations that do this click on the right.

Go with:

```
augmented = AppendRows[A, Transpose[{Y}]];
MatrixForm[augmented]
```

$$\begin{pmatrix} 2.1 & 2.3 & 4.2 & 1.2 \\ 4.5 & -0.6 & 3.1 & 2.5 \\ 2.1 & 1.9 & -2.8 & 3. \end{pmatrix}$$

Subtract appropriate multiples of the first row from the other rows:

```
pivot1 =
  {augmented[[1]], augmented[[2]] - augmented[[2, 1]]/augmented[[1, 1]] augmented[[1]],
   augmented[[3]] - augmented[[3, 1]]/augmented[[1, 1]] augmented[[1]]};

MatrixForm[pivot1]
```

$$\begin{pmatrix} 2.1 & 2.3 & 4.2 & 1.2 \\ 0 & -5.52857 & -5.9 & -0.0714286 \\ 0 & -0.4 & -7. & 1.8 \end{pmatrix}$$

Subtract the appropriate multiple of the second row from the third row:

```
pivot2 = {pivot1[[1]], pivot1[[2]],
   pivot1[[3]] - pivot1[[3, 2]]/pivot1[[2, 2]] pivot1[[2]]};
MatrixForm[pivot2]
```

$$\begin{pmatrix} 2.1 & 2.3 & 4.2 & 1.2 \\ 0 & -5.52857 & -5.9 & -0.0714286 \\ 0 & 0 & -6.57313 & 1.80517 \end{pmatrix}$$

Subtract the appropriate multiples of the third from the first and second rows:

```
pivot3 = {pivot2[[1]] - pivot2[[1, 3]]/pivot2[[3, 3]] pivot2[[3]],
   pivot2[[2]] - pivot2[[2, 3]]/pivot2[[3, 3]] pivot2[[3]], pivot2[[3]]};
MatrixForm[pivot3]
```

$$\begin{pmatrix} 2.1 & 2.3 & 0 & 2.35344 \\ 0 & -5.52857 & 0 & -1.69174 \\ 0 & 0 & -6.57313 & 1.80517 \end{pmatrix}$$

Now subtract the appropriate multiple of the second from the first row:

```
pivot4 = {pivot3[[1]] - pivot3[[1, 2]]/pivot3[[2, 2]] pivot3[[2]],
   pivot3[[2]], pivot3[[3]]};
MatrixForm[pivot4]
```

$$\begin{pmatrix} 2.1 & 0 & 0 & 1.64964 \\ 0 & -5.52857 & 0 & -1.69174 \\ 0 & 0 & -6.57313 & 1.80517 \end{pmatrix}$$

Clean it up by dividing each row by the diagonal terms:

```
reducedrowecheon = {pivot4[[1]]/pivot4[[1, 1]], pivot4[[2]]/pivot4[[2, 2]], pivot4[[3]]/pivot4[[3, 3]]};
MatrixForm[reducedrowecheon]
```

$$\begin{pmatrix} 1. & 0 & 0 & 0.785544 \\ 0 & 1. & 0 & 0.305999 \\ 0 & 0 & 1. & -0.274629 \end{pmatrix}$$

Compare:

```
MatrixForm[RowReduce[augmented]]
```

$$\begin{pmatrix} 1 & 0 & 0 & 0.785544 \\ 0 & 1 & 0 & 0.305999 \\ 0 & 0 & 1 & -0.274629 \end{pmatrix}$$

On the money.

□T.7.a.iii) A 6D example powered by Mathematica

Here's a linear system:

```
A = ( 3.4   0.0   0.0  -3.3   3.9  -3.5
     -1.2   3.0   1.8  -1.3  -3.9   2.6
      0.9   3.7   2.4  -1.8  -1.8   0.7
     -1.8  -2.4  -2.6  -1.7  -3.0  -1.0
      4.1  -1.6   0.0   2.0   0.0  -3.7
      2.6   3.8  -2.5   2.7  -3.1   0.9 );
dim = 6;
Clear[x, j];
X = Table[x[j], {j, 1, dim}];
Y = {1.9, 0, 0, -2.6, 1.8, 0.8};
linearsystem = ColumnForm[Thread[A.X == Y]]
```

```
3.4 x[1] - 3.3 x[4] + 3.9 x[5] - 3.5 x[6] == 1.9
-1.2 x[1] + 3. x[2] + 1.8 x[3] - 1.3 x[4] - 3.9 x[5] + 2.6 x[6] == 0
0.9 x[1] + 3.7 x[2] + 2.4 x[3] - 1.8 x[4] - 1.8 x[5] + 0.7 x[6] == 0
-1.8 x[1] - 2.4 x[2] - 2.6 x[3] - 1.7 x[4] - 3. x[5] - 1. x[6] == -2.6
4.1 x[1] - 1.6 x[2] + 2. x[4] - 3.7 x[6] == 1.8
2.6 x[1] + 3.8 x[2] - 2.5 x[3] + 2.7 x[4] - 3.1 x[5] + 0.9 x[6] == 0.8
```

Use the reduced row echelon form of the augmented matrix to solve this system.

□Answer:

The linear system is:

```
linearsystem
```

```
3.4 x[1] - 3.3 x[4] + 3.9 x[5] - 3.5 x[6] == 1.9
-1.2 x[1] + 3. x[2] + 1.8 x[3] - 1.3 x[4] - 3.9 x[5] + 2.6 x[6] == 0
0.9 x[1] + 3.7 x[2] + 2.4 x[3] - 1.8 x[4] - 1.8 x[5] + 0.7 x[6] == 0
-1.8 x[1] - 2.4 x[2] - 2.6 x[3] - 1.7 x[4] - 3. x[5] - 1. x[6] == -2.6
4.1 x[1] - 1.6 x[2] + 2. x[4] - 3.7 x[6] == 1.8
2.6 x[1] + 3.8 x[2] - 2.5 x[3] + 2.7 x[4] - 3.1 x[5] + 0.9 x[6] == 0.8
```

The augmented matrix is

```
augmentedA = AppendRows[A, Transpose[{Y}]];
MatrixForm[augmentedA]
```

$$\begin{pmatrix} 3.4 & 0 & 0 & -3.3 & 3.9 & -3.5 & 1.9 \\ -1.2 & 3. & 1.8 & -1.3 & -3.9 & 2.6 & 0 \\ 0.9 & 3.7 & 2.4 & -1.8 & -1.8 & 0.7 & 0 \\ -1.8 & -2.4 & -2.6 & -1.7 & -3. & -1. & -2.6 \\ 4.1 & -1.6 & 0 & 2. & 0 & -3.7 & 1.8 \\ 2.6 & 3.8 & -2.5 & 2.7 & -3.1 & 0.9 & 0.8 \end{pmatrix}$$

That's the coefficient matrix A sitting on the left and Y sitting in the seventh column.

And look at this:

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

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 2.23442 \\ 0 & 1 & 0 & 0 & 0 & 0 & -1.30797 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0.124202 \\ 0 & 0 & 0 & 1 & 0 & 0 & -0.586694 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0.0512225 \\ 0 & 0 & 0 & 0 & 0 & 1 & 2.23796 \end{pmatrix}$$

Pick out the seventh column and put

```
X = Table[rowechelon[[j, 7]], {j, 1, 6}]
```

```
{2.23442, -1.30797, 0.124202, -0.586694, 0.0512225, 2.23796}
```

Check:

```
A.X
```

```
{1.9, 0, 0, -2.6, 1.8, 0.8}
```

```
Y
```

```
{1.9, 0, 0, -2.6, 1.8, 0.8}
```

This is getting to be old hat.

□T.7.a.iv) Failure in 4D

Here's a linear system:

```
A = {{1.2, -0.8, 0.9, 2.3},
     {0.5, -2.4, 0.6, 3.1},
     {2.2, -5.6, 2.1, 8.5},
     {-3.1, 0, -2.1, -3.8}};
Y = {-1.0, 0.4, -1.4, 0.7};

Clear[x, k];
```

```
X = Table[x[k], {k, 1, 4}];

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

```
1.2 x[1] - 0.8 x[2] + 0.9 x[3] + 2.3 x[4] == -1.
0.5 x[1] - 2.4 x[2] + 0.6 x[3] + 3.1 x[4] == 0.4
2.2 x[1] - 5.6 x[2] + 2.1 x[3] + 8.5 x[4] == -1.4
-3.1 x[1] - 2.1 x[3] - 3.8 x[4] == 0.7
```

Use the reduced row echelon form of the augmented matrix to try to solve this system.

□Answer:

The linear system is:

```
linearsystem
```

```
1.2 x[1] - 0.8 x[2] + 0.9 x[3] + 2.3 x[4] == -1.
0.5 x[1] - 2.4 x[2] + 0.6 x[3] + 3.1 x[4] == 0.4
2.2 x[1] - 5.6 x[2] + 2.1 x[3] + 8.5 x[4] == -1.4
-3.1 x[1] - 2.1 x[3] - 3.8 x[4] == 0.7
```

The augmented matrix is

```
augmentedA = AppendRows[A, Transpose[{Y}]];
MatrixForm[augmentedA]
```

$$\begin{pmatrix} 1.2 & -0.8 & 0.9 & 2.3 & -1. \\ 0.5 & -2.4 & 0.6 & 3.1 & 0.4 \\ 2.2 & -5.6 & 2.1 & 8.5 & -1.4 \\ -3.1 & 0 & -2.1 & -3.8 & 0.7 \end{pmatrix}$$

That's the coefficient matrix A sitting on the left and Y sitting in the fifth column.

And look at this:

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

```
RowReduce::luc :
 Result for RowReduce of badly conditioned matrix {{1.2, -0.8, 0.9, 2.3, -1.}, <<2>>,
    {-3.1, 0., -2.1, -3.8, 0.7}} may contain significant numerical errors.
```

$$\begin{pmatrix} 1 & 0 & 0 & 0 & -1.46642 \times 10^{16} \\ 0 & 1 & 0 & 0 & 8.18115 \times 10^{15} \\ 0 & 0 & 1 & 0 & 9.08953 \times 10^{15} \\ 0 & 0 & 0 & 1 & 6.93972 \times 10^{15} \end{pmatrix}$$

Garbage.

Reason: A is not invertible.

```
Inverse[A]
```

```
Inverse::luc : Result for Inverse of badly
   conditioned matrix <<1>> may contain significant numerical errors.
```

$$\begin{pmatrix} -1.11703 \times 10^{17} & -2.37238 \times 10^{16} & 2.6125 \times 10^{16} & -2.85262 \times 10^{16} \\ 2.22855 \times 10^{16} & 6.25939 \times 10^{15} & -5.86623 \times 10^{15} & 5.47307 \times 10^{15} \\ 1.55553 \times 10^{17} & 2.97459 \times 10^{16} & -3.49701 \times 10^{16} & 4.01944 \times 10^{16} \\ 5.16286 \times 10^{15} & 2.91514 \times 10^{15} & -1.9869 \times 10^{15} & 1.05866 \times 10^{15} \end{pmatrix}$$

□T.7.b.i) When you want to use Gaussian elimination (reduced row echelon form)

If you're everything by hand calculations, you really have no choice but to use Gaussian elimination.
And that's why most printed matrix texts emphasize Gaussian elimination so heavily.

But when you work with larger matrices with decimal entries, hand calculations can get out of control very,very quickly. That's why most matrix printed texts usually limit themselves to artificial coefficient matrices with integer entries.

On the other hand, when you deal with huge matrices, calculation of SVD and inverses can get out of hand. In such situations, you have little choice but to go with Gaussian elimination.

But when you do, use professionally written software.

Naively written routines often break down when an appropriate multiple of a row is very small.

For a fairly lucid discussion of this issue, see David Kahaner, Cleve Moler and Stephen Nash - Numerical Methods and Software, Prentice-Hall, 1989: ISBN 0 - 13 - 627258 - 4

If *Mathematica* bails out on your large system, then MatLab or any other product using Linpack make good alternative choices.

□T.7.b.ii) Pseudoinverse

Is Gaussian elimination (reduced row echelon form)useful for getting at the pseudo inverse?

□Answer:

No.