

Matrices, Geometry & Mathematica

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

Producer: Bruce Carpenter

Publisher: Math Everywhere, Inc.

MGM.08 Subspaces, Spans, Dimension, Linear Independence, Basis, Orthonormal Bases BASICS

B.1) Subspaces, spanning sets.

The **SpannerMatrix** for a given spanning set {spanner[1], spanner[2], ..., spanner[n]} is

$$\text{SpannerMatrix} = \begin{pmatrix} \text{spanner}[1] & \text{spanner}[2] & \dots & \text{spanner}[n] \\ \downarrow & \downarrow & \dots & \downarrow \end{pmatrix}.$$

The subspace spanned by a given spanning set consists exactly of the vectors that are hits with the **SpannerMatrix**.

The **dimension** of a subspace spanned by a given spanning set is the **rank** of the **SpannerMatrix**.

□ B.1.a.i) Subspaces defined by spanning sets

To get a line through (0, 0, 0) in 3D, you go with one direction vector such as:

```
Clear[s, spanner];
spanner[1] = {1.43, -0.5, 1.2}
{1.43, -0.5, 1.2}
```

And then you make the line by taking all multiples of this vector.
Here's a piece of it:

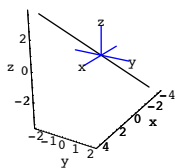
```
Clear[s, spanner];
spanner[1] = {1.43, -0.5, 1.2};
h = 3;

Clear[s];
lineplot = ParametricPlot3D[s spanner[1],
  {s, -h, h}, PlotPoints -> 2, DisplayFunction -> Identity];

linepiece =
```

```
Show[lineplot, ThreeAxes[2], ViewPoint -> CMView, PlotRange -> All,
  Axes -> True, AxesLabel -> {"x", "y", "z"}, Boxed -> False,
  PlotLabel -> "Piece of line", DisplayFunction -> $DisplayFunction];
```

Piece of line



Folks say that the specified vector spans the line because you can get to any point on the line by advancing on a multiple of the given vector:

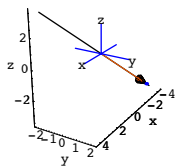
```
s = Random[Real, {-h, h}];
linepoint = s spanner[1];

pointplot = Graphics3D[{Blue, PointSize[0.025], Point[linepoint]}];

pointspannerplot =
  Arrow[s spanner[1], Tail -> {0, 0, 0}, VectorColor -> CadmiumOrange];

Show[linepiece, pointplot, pointspannerplot,
  ViewPoint -> CMView, PlotRange -> All, Axes -> True,
  AxesLabel -> {"x", "y", "z"}, Boxed -> False,
  PlotLabel -> "Piece of line", DisplayFunction -> $DisplayFunction];
```

Piece of line



Rerun several times.

To get a plane through {0,0,0} in 3D, you just go with two vectors such as:

```
Clear[s, spanner, i];
spanner[1] = {1.43, -0.5, 1.2};
spanner[2] = {-0.9, 0.0, 1.5};
spanners = Table[spanner[i], {i, 1, 2}];

ColumnForm[spanners]
```

```
{1.43, -0.5, 1.2}
{-0.9, 0, 1.5}
```

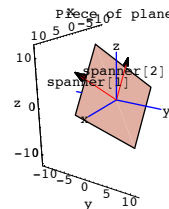
And then you make the plane by taking all combinations
s spanner[1] + t spanner[2]
of multiples of these two vectors:

```
h = 5;
k = 4;
Clear[s, t];
planeplot = ParametricPlot3D[s spanner[1] + t spanner[2], {s, -h, h},
  {t, -k, k}, PlotPoints -> {2, 2}, DisplayFunction -> Identity];

spannerplot =
  {Arrow[1.3 h spanner[1], Tail -> {0, 0, 0}, VectorColor -> Red],
  Arrow[1.3 k spanner[2], Tail -> {0, 0, 0}, VectorColor -> Red]};

spannerlabels = {Graphics3D[Text["spanner[1]", 0.9 h spanner[1]]],
  Graphics3D[Text["spanner[2]", 0.9 k spanner[2]]]};

planepiece = Show[planeplot, spannerplot, spannerlabels,
  ThreeAxes[10], ViewPoint -> CMView, PlotRange -> All,
  Axes -> True, AxesLabel -> {"x", "y", "z"}, Boxed -> False,
  PlotLabel -> "Piece of plane", DisplayFunction -> $DisplayFunction];
```



Folks say that the two specified vectors span the plane because you can get to any point on the plane by advancing on a multiple of spanner[1] and then advancing on a multiple of spanner[2].

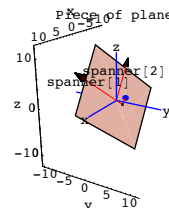
See it happen for some random points on this plane:

```
s = Random[Real, {-h, h}];
t = Random[Real, {-k, k}];
planepoint = s spanner[1] + t spanner[2];

pointplot = Graphics3D[{Blue, PointSize[0.035], Point[planepoint]}];

pointspannerplot =
  {Arrow[s spanner[1], Tail -> {0, 0, 0}, VectorColor -> Red],
  Arrow[t spanner[2], Tail -> s spanner[1], VectorColor -> Red]};
```

```
Show[planepiece, pointplot, pointspannerplot,
  ViewPoint -> CMView, PlotRange -> All, Axes -> True,
  AxesLabel -> {"x", "y", "z"}, Boxed -> False,
  PlotLabel -> "Piece of plane", DisplayFunction -> $DisplayFunction];
```



Rerun at least seven times.

Once you see how to make lines and planes with spanning vectors in 3D, you can make the jump to working with the analogs of planes and lines in higher dimensions. Most folks call these higher dimensional analogs by the name "subspaces."

Here are three vectors in 4D:

```
Clear[s, i, spanner];
spanner[1] = {3.43, -0.57, 8.19, -0.94};
spanner[2] = {0.97, 1.29, 0.83, 5.23};
spanner[3] = {-6.14, 3.08, -5.00, -1.97};
spanners = Table[spanner[i], {i, 1, 3}]

{{3.43, -0.57, 8.19, -0.94},
 {0.97, 1.29, 0.83, 5.23}, {-6.14, 3.08, -5., -1.97}}
```

When folks talk about the subspace of 4D spanned by these three vectors, they are talking about the whole collection of all vectors expressible as a combination

$$x[1] \text{ spanner}[1] + x[2] \text{ spanner}[2] + x[3] \text{ spanner}[3]$$

where {x[1], x[2], x[3]} range over all real numbers:

```
Clear[x, k];
Length[spanners]
Sum[x[k] spanner[k], {k, 1, Length[spanners]}]
{3.43 x[1] + 0.97 x[2] - 6.14 x[3], -0.57 x[1] + 1.29 x[2] + 3.08 x[3],
 8.19 x[1] + 0.83 x[2] - 5. x[3], -0.94 x[1] + 5.23 x[2] - 1.97 x[3]}
```

Here are some random members of this subspace:

```
a = 8;
Clear[x];
x[i_] := Random[Real, {-a, a}];
Length[spanners]
Sum[x[k] spanner[k], {k, 1, Length[spanners]}]
{-15.0145, 2.27767, -8.94446, -22.6734}
```

Rerun a couple of times.

What's a good way of describing the vectors in this subspace?

□ Answer:

The vectors in this subspace of 4D look like this

$$x[1] \text{spanner}[1] + x[2] \text{spanner}[2] + x[3] \text{spanner}[3].$$

These can be described as the 4D destinations of all three-stage trips made by

- 1) going at first on a straight line segment from {0, 0, 0, 0} in the direction of spanner[1],
- 2) then stopping and switching directions by going on a straight line segment in the direction of spanner[2],
- 3) then stopping and switching directions by going on a straight line segment in the direction of spanner[3].

The size of the coefficients $x[i]$ indicate the length of the trip at the i th stage.

□ B.1.a.ii) The spanner matrix for a given spanning set {spanner[1],spanner[2], , , ,spanner[n]}

$$\text{is } \begin{pmatrix} \text{spanner}[1] & \text{spanner}[2] & \dots & \text{spanner}[n] \\ \downarrow & \downarrow & \dots & \downarrow \end{pmatrix}.$$

The subspace spanned by {spanner[1],spanner[2], , , ,spanner[n]} consists exactly of the vectors that are hits with this matrix

Stay with the same subspace as in part i) immediately above spanned by the vectors:

```
Spanners = {spanner[1], spanner[2], spanner[3]}
{{3.43, -0.57, 8.19, -0.94},
 {0.97, 1.29, 0.83, 5.23}, {-6.14, 3.08, -5., -1.97}}
```

and look at this associated spanner matrix

$$\text{SpannerMatrix} = \begin{pmatrix} \text{spanner}[1] & \text{spanner}[2] & \text{spanner}[3] \\ \downarrow & \downarrow & \downarrow \end{pmatrix}.$$

```
SpannerMatrix = Transpose[spanners];
MatrixForm[SpannerMatrix]
```

$$\begin{pmatrix} 3.43 & 0.97 & -6.14 \\ -0.57 & 1.29 & 3.08 \\ 8.19 & 0.83 & -5. \\ -0.94 & 5.23 & -1.97 \end{pmatrix}$$

spanner[1] is in the first vertical column of SpannerMatrix.
spanner[2] is in the second vertical column of SpannerMatrix.
spanner[3] is in the third vertical column of SpannerMatrix.

How are the vectors in the subspace S of 4D spanned by

spanner[1], spanner[2] and spanner[3]

related to hits with the associated spanner matrix?

□ Answer:

The subspace S of 4D spanned by spanner[1], spanner[2] and spanner[3] consists exactly of the vectors that are hits with the SpannerMatrix.

Reason:

Hits with SpannerMatrix look like this:

$$\begin{aligned} & \text{SpannerMatrix} \cdot \{x[1], x[2], x[3]\} \\ &= \begin{pmatrix} \text{spanner}[1] & \text{spanner}[2] & \text{spanner}[3] \\ \downarrow & \downarrow & \downarrow \end{pmatrix} \cdot \{x[1], x[2], x[3]\} \\ &= x[1] \text{spanner}[1] + x[2] \text{spanner}[2] + x[3] \text{spanner}[3] \end{aligned}$$

where $x[1]$, $x[2]$ and $x[3]$ are real numbers.

The vectors in the subspace of 4D spanned by spanner[1], spanner[2] and spanner[3] look like this:

$$x[1] \text{spanner}[1] + x[2] \text{spanner}[2] + x[3] \text{spanner}[3]$$

where $x[1]$, $x[2]$ and $x[3]$ are real numbers.

This is a big break in your favor because the spanner matrix is the key to working with subspaces.

□ B.1.b.i) The rank of the spanner matrix is the dimension of the corresponding subspace

Here are five vectors in 5D:

```
Clear[i, spanner];
spanner[1] = {3.43, -0.57, 2.19, -0.94, 1.28};
spanner[2] = {0.97, 1.29, 0.83, 5.23, 1.26};
spanner[3] = {-6.14, 3.08, -5.00, -1.97, -0.96};
spanner[4] = {-3.68, 1.22, -3.64, -8.14, -0.94};
spanner[5] = {2.71, -2.51, 2.81, 2.91, -0.32};
spanners = Table[spanner[i], {i, 1, 5}]
{{3.43, -0.57, 2.19, -0.94, 1.28},
 {0.97, 1.29, 0.83, 5.23, 1.26}, {-6.14, 3.08, -5., -1.97, -0.96},
 {-3.68, 1.22, -3.64, -8.14, -0.94}, {2.71, -2.51, 2.81, 2.91, -0.32}}
```

This matrix hits on 5D and hangs in 5D.

These vectors span a subspace S of 5D.

Make the associated spanner matrix for this subspace S:

```
SpannerMatrix = Transpose[spanners];
MatrixForm[SpannerMatrix]
```

$$\begin{pmatrix} 3.43 & 0.97 & -6.14 & -3.68 & 2.71 \\ -0.57 & 1.29 & 3.08 & 1.22 & -2.51 \\ 2.19 & 0.83 & -5. & -3.64 & 2.81 \\ -0.94 & 5.23 & -1.97 & -8.14 & 2.91 \\ 1.28 & 1.26 & -0.96 & -0.94 & -0.32 \end{pmatrix}$$

The subspace S is the same as all possible hits with this spanner matrix.

Check the rank of this spanner matrix:

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

Why do folks say that because the rank of the SpannerMatrix is 3, the dimension of the subspace S is 3?

□ Answer:

The rank of the spanner matrix is 3.

This means the Spanner Matrix needs a perpendicular frame consisting of three 5D mutually perpendicular unit vectors to hang its hits.

Here is one such perpendicular frame:

```
Clear[k, hangerframe];
hangerframe[k_] := SingularValues[SpannerMatrix][[1, k]];
Table[hangerframe[k], {k, 1, rank}]
{{-0.550517, 0.225396, -0.486973, -0.628875, -0.116196},
 {-0.458778, 0.348461, -0.301171, 0.759864, -0.000785019},
 {0.334709, 0.705309, -0.0275415, -0.131643, 0.610263}}
```

And because the subspace S is the same as the collection of all possible hits with

SpannerMatrix, this tells you that everything in S is framed up with the same three 5D mutually perpendicular unit vectors.

That's why folks say that the dimension of S is 3.

□ B.1.c.i) Calculating the dimension of a subspace of 6D

A subspace S of 6D is defined by the following given spanning set:

```
Clear[s, i, spanner];
spanner[1] = {0.829, 1.202, 0.539, -0.925, 1.289, -0.397};
spanner[2] = {0.073, 1.463, 0.312, 1.622, 1.841, 2.916};
spanner[3] = {0.902, 2.665, 0.851, 0.697, 3.130, 2.519};
spanner[4] = {2.335, -1.908, 0.925, -1.391, -1.820, -0.154};
spanner[5] = {0.390, -4.513, 0.0, 1.316, -1.968, -0.037};
spanners = Table[spanner[i], {i, 1, 5}]
{{0.829, 1.202, 0.539, -0.925, 1.289, -0.397},
 {0.073, 1.463, 0.312, 1.622, 1.841, 2.916},
 {0.902, 2.665, 0.851, 0.697, 3.13, 2.519},
 {2.335, -1.908, 0.925, -1.391, -1.82, -0.154},
 {0.39, -4.513, 0, 1.316, -1.968, -0.037}}
```

Calculate the dimension of S.

□ Answer:

The spanner matrix is:

```
SpannerMatrix = Transpose[spanners];
MatrixForm[SpannerMatrix]
```

$$\begin{pmatrix} 0.829 & 0.073 & 0.902 & 2.335 & 0.39 & 0 \\ 1.202 & 1.463 & 2.665 & -1.908 & -4.513 & 0 \\ 0.539 & 0.312 & 0.851 & 0.925 & 0 & 0 \\ -0.925 & 1.622 & 0.697 & -1.391 & 1.316 & 0 \\ 1.289 & 1.841 & 3.13 & -1.82 & -1.968 & 0 \\ -0.397 & 2.916 & 2.519 & -0.154 & -0.037 & 0 \end{pmatrix}$$

This matrix hits on 5D and hangs in 6D.

The dimension of S is the same as the rank of the spanner matrix:

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

The dimension of S is 4.

□ B.1.c.ii) A perpendicular frame spanning S

Stay with the same four dimensional subspace S of 6D as in part i) immediately above. Exhibit a spanning set for S that consists of 4 mutually perpendicular 6D unit vectors.

□ Answer:

Just go with the SVD hangerframe for the spanner matrix.

```
hangerframe = SingularValues[SpannerMatrix][[1]]
{{0.0362149, -0.715869, -0.0551429, -0.089052, -0.591056, -0.354827},
 {-0.126229, 0.457445, -0.0907112, -0.571628, -0.0713706, -0.65934},
 {-0.795902, 0.0216159, -0.390903, 0.434967, -0.0111113, -0.15475},
 {0.088828, -0.386841, 0.130041, 0.138401, 0.743399, -0.503743}}
```

Done.

B.2) Using the **Ytest** idea to determine whether a given vector is in a given subspace.

Using the **Sprojection** matrix to find the vector in a subspace that is closest to a given vector.

Using the fact that **Sprojection.Y = Ytest** to build Sprojection as the sum of rank one matrices or through SVD of the SpannerMatrix.

Using the Sprojection matrix to determine whether a given vector is in a subspace.

Using the Sprojection matrix to determine whether two given subspaces are the same or different

□ B.2.a.i) Using **Ytest** to determine whether a given vector is in a given subspace

Here is a spanning set for a subspace S of 4D:

```
Clear[k, spanner];
spanner[1] = {1.0, 0, 2.6, -4.1};
spanner[2] = {0.3, 1.2, 0.5, 1.3};
spanner[3] = {-0.6, 1.8, -1.2, 0.2};

spanners = Table[spanner[k], {k, 1, 3}]
{{1., 0, 2.6, -4.1}, {0.3, 1.2, 0.5, 1.3}, {-0.6, 1.8, -1.2, 0.2}}
```

Here are three 4D vectors:

```
Y1 = {0.76, -0.96, 2.2, -5.14}
Y2 = {0.48, -2.13, 0.66, 1.66}
Y3 = {-0.62, 1.38, -0.92, -2.42}
```

Which of these vectors are in S?

□ Answer:

Make the spanner matrix

```
SpannerMatrix = (spanner[1] spanner[2] spanner[3])
                (   ↓           ↓           ↓ )
SpannerMatrix = Transpose[spanners];
MatrixForm[SpannerMatrix]
```

$$\begin{pmatrix} 1. & 0.3 & -0.6 \\ 0 & 1.2 & 1.8 \\ 2.6 & 0.5 & -1.2 \\ -4.1 & 1.3 & 0.2 \end{pmatrix}$$

Calculate the dimension of S:

```
dim = Length[SingularValues[SpannerMatrix][[2]]]
3
The dimension of S is the same as the rank of the SpannerMatrix.
```

Calculate a hangerframe for SpannerMatrix:

```
Clear[hangerframe];
hangerframe[k_] := SingularValues[SpannerMatrix][[1, k]];
Table[hangerframe[k], {k, 1, dim}]
{{-0.20286, 0.11737, -0.521491, 0.820439},
 {0.0580724, -0.978078, 0.0582512, 0.191307},
 {-0.383639, -0.171209, -0.73236, -0.535871}}
```

Because S consists exactly of all hits with the SpannerMatrix, saying that a 4D vector Y is in S is the same as saying that the linear system

$$\text{SpannerMatrix} \cdot X = Y$$

has a solution for X.

Consequently, saying that Y is in S is the same as saying that Y is the same as

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

Try it out for Y1:

```
Y1 = {0.76, -0.96, 2.2, -5.14}
{0.76, -0.96, 2.2, -5.14}

Y1test = Sum[Y1.hangerframe[k] hangerframe[k], {k, 1, dim}]
{0.76, -0.96, 2.2, -5.14}
```

The call:

Y1 is in S because Y1test = Y1.

Try it out for Y2:

```
Y2 = {0.48, -2.13, 0.66, 1.66}
{0.48, -2.13, 0.66, 1.66}
```

```
Y2test = Sum[Y2.hangerframe[k] hangerframe[k], {k, 1, dim}]
{0.464726, -2.13028, 0.667372, 1.66095}
```

The call:

Y2 is not in S because Y2test ≠ Y2.

Try it out for Y3:

```
Y3 = {-0.62, 1.38, -0.92, -2.42}
{-0.62, 1.38, -0.92, -2.42}

Y3test = Sum[Y3.hangerframe[k] hangerframe[k], {k, 1, dim}]
{-0.62, 1.38, -0.92, -2.42}
```

The call:

Y3 is in S because Y3test = Y3.

□ B.2.a.ii) Using a hit with **Sprojection = SpannerMatrix**

PseudoInverse[SpannerMatrix] to get as close as you can get

Here's a spanning set for a random subspace S of 5D:

```
Clear[k, spanner];
spanner[1] = Table[Random[Real, {-2, 2}], {k, 1, 5}];
spanner[2] = Table[Random[Real, {-2, 2}], {k, 1, 5}];
spanner[3] = Table[Random[Real, {-2, 2}], {k, 1, 5}];
spanner[4] = Table[Random[Real, {-2, 2}], {k, 1, 5}];

spanners = Table[spanner[k], {k, 1, 4}]
{{1.52787, 0.792859, -1.92951, -1.46637, -1.38905},
 {0.630179, -1.6928, -1.2541, -1.75133, -0.878534},
 {-1.72373, -1.19392, 0.466473, 0.373048, -1.24189},
 {1.39958, 1.68209, 0.711557, -1.36852, -0.120829}}
```

Here is a random 5D vector:

```
Y = Table[Random[Real, {-7, 7}], {k, 1, 5}]
{-2.65998, -5.06362, 5.15674, 4.59338, -1.00753}
```

Come up with a matrix Sprojection so that

Sprojection.Y is the member of S that is **closest** to Y.

□ Answer:

Just do this:

Make the spanner matrix

```
SpannerMatrix = (spanner[1] spanner[2] spanner[3] spanner[4])
                (   ↓           ↓           ↓           ↓ )
SpannerMatrix = Transpose[spanners];
MatrixForm[SpannerMatrix]
```

$$\begin{pmatrix} 1.52787 & 0.630179 & -1.72373 & 1.39958 \\ 0.792859 & -1.6928 & -1.19392 & 1.68209 \\ -1.92951 & -1.2541 & 0.466473 & 0.711557 \\ -1.46637 & -1.75133 & 0.373048 & -1.36852 \\ -1.38905 & -0.878534 & -1.24189 & -0.120829 \end{pmatrix}$$

Define the Sprojection matrix:

```
Sprojection = SpannerMatrix.PseudoInverse[SpannerMatrix];
MatrixForm[Sprojection]
```

$$\begin{pmatrix} 0.562928 & 0.194149 & -0.193421 & -0.330821 & 0.247977 \\ 0.194149 & 0.913759 & 0.0859179 & 0.146952 & -0.110152 \\ -0.193421 & 0.0859179 & 0.914404 & -0.1464 & 0.109739 \\ -0.330821 & 0.146952 & -0.1464 & 0.749601 & 0.187694 \\ 0.247977 & -0.110152 & 0.109739 & 0.187694 & 0.859308 \end{pmatrix}$$

The member of S closest to Y is:

```
Sprojection.Y
{-5.24732, -3.91431, 4.01174, 2.63502, 0.460417}
```

Reason:

When you go with any matrix A, then

$$A \cdot (\text{PseudoInverse}[A] \cdot Y)$$

is as close as you can get to Y with a hit with A.

So when you have a spanning set for a subspace S of kD and make the corresponding SpannerMatrix, you know that for any Y in kD

$$\text{SpannerMatrix} \cdot (\text{PseudoInverse}[\text{SpannerMatrix}] \cdot Y)$$

is as close as you can get to Y with a hit with SpannerMatrix.

And because S consists precisely of all possible hits with SpannerMatrix, it is guaranteed that for any kD Y,

$$\text{Sprojection} \cdot Y = \text{SpannerMatrix} \cdot \text{PseudoInverse}[\text{SpannerMatrix}] \cdot Y$$

is the member of S closest to Y.

□ B.2.a.iii) **Sprojection.Y = Ytest**, so **Ytest** is as close as you can get to Y.

Here's a spanning set for a random subspace S of 6D:

```

Clear[k, spanner];
spanner[1] = Table[Random[Real, {-2, 2}], {k, 1, 6}];
spanner[2] = Table[Random[Real, {-2, 2}], {k, 1, 6}];
spanner[3] = Table[Random[Real, {-2, 2}], {k, 1, 6}];
spanner[4] = Table[Random[Real, {-2, 2}], {k, 1, 6}];

spanners = Table[spanner[k], {k, 1, 4}]
{{-0.239607, 1.40286, 0.77876, -0.898815, 1.13021, 1.09566},
{0.0328652, -1.14749, 0.00874763, 0.819392, -0.773217, 0.386038},
{1.6357, 0.0612834, -0.172799, 0.703949, -1.07586, -0.570195},
{1.94803, -0.536057, -1.62911, -0.043548, -1.36436, 1.75181}}

```

Here is a random 6D vector:

```

Clear[y];
Y = Table[Random[Real, {-8, 8}], {k, 1, 6}]
{2.44199, 2.21437, -0.572501, 2.60249, 5.92113, 5.83172}

```

The vector in S closest to Y is:

```

SpannerMatrix = Transpose[spanners];
Projection = SpannerMatrix.PseudoInverse[SpannerMatrix];
closest = Projection.Y
{1.13124, 2.39271, 1.47556, -1.41373, 1.18052, 5.45649}

```

Now look at this:

```

dim = Length[SingularValues[SpannerMatrix][[2]]];
Clear[hangerframe];
hangerframe[k_] := SingularValues[SpannerMatrix][[1, k]];
Ytest = Sum[(Y.hangerframe[k]) hangerframe[k], {k, 1, dim}]
{1.13124, 2.39271, 1.47556, -1.41373, 1.18052, 5.45649}

```

Ytest came out to be equal to the vector in S closest to Y.

Is this some sort of accident?

If not explain why it is not an accident.

□ Answer:

Criminee.

In mathematics, there are no accidents.

To explain why it is not an accident and will happen every time, remember

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

where $\text{stretch}[k]$, $\text{hangerframe}[k]$ and $\text{alignerframe}[k]$ come from SVD analysis of SpannerMatrix so that

$$\text{SpannerMatrix}.\text{alignerframe}[k] = \text{stretch}[k] \text{hangerframe}[k].$$

Now rock on:

$$\begin{aligned}
\text{Projection.Y} &= \text{SpannerMatrix}.\text{PseudoInverse}[\text{SpannerMatrix}].Y \\
&= \text{SpannerMatrix}.\sum_{k=1}^{\dim} \left(\frac{Y.\text{hangerframe}[k]}{\text{stretch}[k]} \right) \text{alignerframe}[k] \\
&= \sum_{k=1}^{\dim} \left(\frac{Y.\text{hangerframe}[k]}{\text{stretch}[k]} \right) \text{SpannerMatrix}.\text{alignerframe}[k] \\
&= \sum_{k=1}^{\dim} \left(\frac{Y.\text{hangerframe}[k]}{\text{stretch}[k]} \right) \text{stretch}[k] \text{hangerframe}[k] \\
&= \sum_{k=1}^{\dim} (Y.\text{hangerframe}[k]) \text{hangerframe}[k] \\
&= .Ytest.
\end{aligned}$$

That's it.

□ B.2.a.iv) Using the fact that $\text{Projection.Y} = Ytest$ to build Projection as the sum of rank one matrices

Here's a spanning set for a subspace S of 6D:

```

Clear[k, spanner];
spanner[1] = {-0.36, -1.77, -1.55, 1.24, 0.26, -0.57};
spanner[2] = {-1.76, -1.34, 1.39, 0.73, 1.85, 0.87};
spanner[3] = {1.41, 0.40, 1.94, -1.48, 1.52, 1.57};
spanner[4] = {3.16, 0.91, -4.33, -0.22, -3.44, -2.31};
spanner[5] = {0.21, -1.32, -1.77, -1.39, 0.67, -0.29};
spanner[6] = {-3.73, -1.36, 4.55, 2.85, 3.03, 2.03};
spanners = Table[spanner[k], {k, 1, 6}]
{{-0.36, -1.77, -1.55, 1.24, 0.26, -0.57},
{-1.76, -1.34, 1.39, 0.73, 1.85, 0.87},
{1.41, 0.4, 1.94, -1.48, 1.52, 1.57},
{3.16, 0.91, -4.33, -0.22, -3.44, -2.31},
{0.21, -1.32, -1.77, -1.39, 0.67, -0.29},
{-3.73, -1.36, 4.55, 2.85, 3.03, 2.03}}

```

The Projection matrix for this subspace is:

```

SpannerMatrix = Transpose[spanners];
Projection = SpannerMatrix.PseudoInverse[SpannerMatrix];
MatrixForm[Projection]

```

```

0.97499 0.0340183 -0.0702951 0.0179532 -0.0160007 0.133071
0.0340183 0.61284 0.268585 -0.143764 -0.376996 -0.0343272
-0.0702951 0.268585 0.714658 0.111017 0.157364 0.299591
0.0179532 -0.143764 0.111017 0.94533 -0.128119 -0.0441726
-0.0160007 -0.376996 0.157364 -0.128119 0.52331 0.256704
0.133071 -0.0343272 0.299591 -0.0441726 0.256704 0.228871

```

Build this matrix as the sum of well-chosen rank one matrices.

□ Answer:

In part iii) immediately above, you saw that

$$\text{Sprojection.Y} = \sum_{k=1}^{\dim} (Y.\text{hangerframe}[k]) \text{hangerframe}[k].$$

This tells you how to make rank one matrices whose sum is Sprojection.

Just do this:

```

dim = Length[SingularValues[SpannerMatrix][[2]]];
Clear[hangerframe, rankOne];
hangerframe[k_] := SingularValues[SpannerMatrix][[1, k]];
rankOne[k_] := Transpose[{hangerframe[k]}.{hangerframe[k]}];
sum = Sum[rankOne[k], {k, 1, dim}]
MatrixForm[sum]

```

```

0.97499 0.0340183 -0.0702951 0.0179532 -0.0160007 0.133071
0.0340183 0.61284 0.268585 -0.143764 -0.376996 -0.0343272
-0.0702951 0.268585 0.714658 0.111017 0.157364 0.299591
0.0179532 -0.143764 0.111017 0.94533 -0.128119 -0.0441726
-0.0160007 -0.376996 0.157364 -0.128119 0.52331 0.256704
0.133071 -0.0343272 0.299591 -0.0441726 0.256704 0.228871

```

Compare:

```
MatrixForm[Sprojection]
```

```

0.97499 0.0340183 -0.0702951 0.0179532 -0.0160007 0.133071
0.0340183 0.61284 0.268585 -0.143764 -0.376996 -0.0343272
-0.0702951 0.268585 0.714658 0.111017 0.157364 0.299591
0.0179532 -0.143764 0.111017 0.94533 -0.128119 -0.0441726
-0.0160007 -0.376996 0.157364 -0.128119 0.52331 0.256704
0.133071 -0.0343272 0.299591 -0.0441726 0.256704 0.228871

```

Nailed it clean.

This will work every time because

$$\text{Sprojection.Y} = \sum_{k=1}^{\dim} (Y.\text{hangerframe}[k]) \text{hangerframe}[k].$$

□ B.2.a.v) Building Projection directly from the SVD hanger frames of the

SpannerMatrix

Here's a spanning set for a subspace S of 6D:

```

Clear[k, spanner];
spanner[1] = {-0.36, -1.77, -1.55, 1.24, 0.26, -0.57};
spanner[2] = {-1.76, -1.34, 1.39, 0.73, 1.85, 0.87};
spanner[3] = {1.41, 0.40, 1.94, -1.48, 1.52, 1.57};
spanner[4] = {3.16, 0.91, -4.33, -0.22, -3.44, -2.31};
spanner[5] = {0.21, -1.32, -1.77, -1.39, 0.67, -0.29};
spanner[6] = {-3.73, -1.36, 4.55, 2.85, 3.03, 2.03};
spanners = Table[spanner[k], {k, 1, 6}]
{{-0.36, -1.77, -1.55, 1.24, 0.26, -0.57},
{-1.76, -1.34, 1.39, 0.73, 1.85, 0.87},
{1.41, 0.4, 1.94, -1.48, 1.52, 1.57},
{3.16, 0.91, -4.33, -0.22, -3.44, -2.31},
{0.21, -1.32, -1.77, -1.39, 0.67, -0.29},
{-3.73, -1.36, 4.55, 2.85, 3.03, 2.03}}

```

The Projection matrix for this subspace is:

```

SpannerMatrix = Transpose[spanners];
Projection = SpannerMatrix.PseudoInverse[SpannerMatrix];
MatrixForm[Projection]

```

```

0.97499 0.0340183 -0.0702951 0.0179532 -0.0160007 0.133071
0.0340183 0.61284 0.268585 -0.143764 -0.376996 -0.0343272
-0.0702951 0.268585 0.714658 0.111017 0.157364 0.299591
0.0179532 -0.143764 0.111017 0.94533 -0.128119 -0.0441726
-0.0160007 -0.376996 0.157364 -0.128119 0.52331 0.256704
0.133071 -0.0343272 0.299591 -0.0441726 0.256704 0.228871

```

Build this matrix directly from the SVD of the SpannerMatrix.

□ Answer:

In part iii) immediately above, you saw that

$$\text{Sprojection.Y} = \sum_{k=1}^{\dim} (Y.\text{hangerframe}[k]) \text{hangerframe}[k].$$

This tells you how to make Sprojection directly from the SVD hangerframe of the Spanner Matrix

You go with the SVD hangerframe for both the alignerframe and the hangerframe of Sprojection with all stretch factors equal to 1.

```
dim = Length[SingularValues[SpannerMatrix][[2]]];
Clear[hangerframe, rankOne];
aligner = SingularValues[SpannerMatrix][[1]];
stretcher = IdentityMatrix[dim];
hanger = Transpose[SingularValues[SpannerMatrix][[1]]];
```

```
Sprojection = hanger.stretcher.aligner;
MatrixForm[Sprojection]
```

$$\begin{pmatrix} 0.97499 & 0.0340183 & -0.0702951 & 0.0179532 & -0.0160007 & 0.133071 \\ 0.0340183 & 0.61284 & 0.268585 & -0.143764 & -0.376996 & -0.0343272 \\ -0.0702951 & 0.268585 & 0.714658 & 0.111017 & 0.157364 & 0.299591 \\ 0.0179532 & -0.143764 & 0.111017 & 0.94533 & -0.128119 & -0.0441726 \\ -0.0160007 & -0.376996 & 0.157364 & -0.128119 & 0.52331 & 0.256704 \\ 0.133071 & -0.0343272 & 0.299591 & -0.0441726 & 0.256704 & 0.228871 \end{pmatrix}$$

Compare:

```
MatrixForm[SpannerMatrix.PseudoInverse[SpannerMatrix]]
```

$$\begin{pmatrix} 0.97499 & 0.0340183 & -0.0702951 & 0.0179532 & -0.0160007 & 0.133071 \\ 0.0340183 & 0.61284 & 0.268585 & -0.143764 & -0.376996 & -0.0343272 \\ -0.0702951 & 0.268585 & 0.714658 & 0.111017 & 0.157364 & 0.299591 \\ 0.0179532 & -0.143764 & 0.111017 & 0.94533 & -0.128119 & -0.0441726 \\ -0.0160007 & -0.376996 & 0.157364 & -0.128119 & 0.52331 & 0.256704 \\ 0.133071 & -0.0343272 & 0.299591 & -0.0441726 & 0.256704 & 0.228871 \end{pmatrix}$$

Nailed it clean.

This will work every time because

$$\text{Sprojection.Y} = \sum_{k=1}^{\text{dim}} (\text{Y.hangerframe}[k]) \text{hangerframe}[k].$$

□B.2.b) Using the Sprojection matrix to determine whether a given vector is in a given subspace

Here is a spanning set for a subspace S of 5D:

```
Clear[k, spanner];
spanner[1] = {-1.4, 1.5, 2.9, 0.6, -2.4};
spanner[2] = {-1.7, 0.0, 2.9, 0.0, -3.6};
spanner[3] = {0.2, 1.0, 0.0, 0.4, 0.8};
spanner[4] = {-0.3, 1.7, -2.1, -0.7, 0.1};
spanners = Table[spanner[k], {k, 1, 4}];
```

Here are two 5D vectors:

```
x = {-5.08, 4.15, 6.89, 0.28, -9.1}
{-5.08, 4.15, 6.89, 0.28, -9.1}
```

```
y = {-5.57, 4.15, 6.89, 0.28, -9.1}
{-5.57, 4.15, 6.89, 0.28, -9.1}
```

One of these vectors is in S.

The other is not.

Your job is to determine which one is in S and which one is not.

□Answer:

Go directly to the spanner matrix for S:

```
SpannerMatrix = Transpose[spanners];
MatrixForm[SpannerMatrix]
```

$$\begin{pmatrix} -1.4 & -1.7 & 0.2 & -0.3 \\ 1.5 & 0 & 1. & 1.7 \\ 2.9 & 2.9 & 0 & -2.1 \\ 0.6 & 0 & 0.4 & -0.7 \\ -2.4 & -3.6 & 0.8 & 0.1 \end{pmatrix}$$

Calculate Sprojection matrix

```
Sprojection = SpannerMatrix.PseudoInverse[SpannerMatrix];
MatrixForm[Sprojection]
```

$$\begin{pmatrix} 0.200471 & -0.160379 & -0.0302252 & 0.0942973 & 0.353207 \\ -0.160379 & 0.958022 & -0.0440102 & 0.104572 & 0.0402819 \\ -0.0302252 & -0.0440102 & 0.852033 & 0.334983 & -0.104922 \\ 0.0942973 & 0.104572 & 0.334983 & 0.240787 & 0.225318 \\ 0.353207 & 0.0402819 & -0.104922 & 0.225318 & 0.748687 \end{pmatrix}$$

The first given vector is:

```
x
{-5.08, 4.15, 6.89, 0.28, -9.1}
```

The vector in S that is closest to X is:

```
Sprojection.x
{-5.08, 4.15, 6.89, 0.28, -9.1}
```

The result:

$$\text{Sprojection.X} = X$$

The vector in S that is closest to X is X itself.

This tells you that X is in S.

The second given vector is:

```
y
{-5.57, 4.15, 6.89, 0.28, -9.1}
```

The vector in S that is closest to Y is:

```
Sprojection.y
{-5.17823, 4.22859, 6.90481, 0.233794, -9.27307}
```

Close, but no cigar.

$$\text{Sprojection.Y} \neq Y$$

The vector in S that is closest to Y is not the same as Y.

This tells you that Y is NOT in S.

□B.2.c) Different spanning sets for the same subspace

Here's a set of spanning vectors which define a subspace S1 of 5D:

```
Clear[s, i, spanner1];
spanner1[1] = {3.43, -0.57, 8.19, -0.94, 0.48};
spanner1[2] = {0.97, 1.29, 0.83, 5.23, -2.31};
spanner1[3] = {-6.14, 3.08, -5.00, -1.97, 0.78};
spanners1 = Table[spanner1[i], {i, 1, 3}]
{{3.43, -0.57, 8.19, -0.94, 0.48},
{0.97, 1.29, 0.83, 5.23, -2.31}, {-6.14, 3.08, -5., -1.97, 0.78}}
```

Here's a another set of spanning vectors which define a subspace S2 of 5D:

```
Clear[s, i, spanner2];
spanner2[1] = {4.40, 0.72, 9.02, 4.29, -1.83};
spanner2[2] = {-5.17, 4.37, -4.17, 3.26, -1.53};
spanner2[3] = {-5.17, 4.37, -4.17, 3.26, -1.53};
spanner2[4] = {1.94, 2.58, 1.66, 10.46, -4.62};
spanners2 = Table[spanner2[i], {i, 1, 4}]
{{4.4, 0.72, 9.02, 4.29, -1.83}, {-5.17, 4.37, -4.17, 3.26, -1.53},
{-5.17, 4.37, -4.17, 3.26, -1.53}, {1.94, 2.58, 1.66, 10.46, -4.62}}
```

At first glance, it would appear that S1 and S2 are in fact different subspaces of 5D.

Is this true?

□Answer:

Go directly to the spanner matrices for each subspace.

```
SpannerMatrix1 = Transpose[spanners1];
MatrixForm[SpannerMatrix1]

SpannerMatrix2 = Transpose[spanners2];
MatrixForm[SpannerMatrix2]
```

$$\begin{pmatrix} 3.43 & 0.97 & -6.14 \\ -0.57 & 1.29 & 3.08 \\ 8.19 & 0.83 & -5. \\ -0.94 & 5.23 & -1.97 \\ 0.48 & -2.31 & 0.78 \end{pmatrix}$$

$$\begin{pmatrix} 4.4 & -5.17 & -5.17 & 1.94 \\ 0.72 & 4.37 & 4.37 & 2.58 \\ 9.02 & -4.17 & -4.17 & 1.66 \\ 4.29 & 3.26 & 3.26 & 10.46 \\ -1.83 & -1.53 & -1.53 & -4.62 \end{pmatrix}$$

Calculate the S1projection matrix and the S2projection matrix and look at them:

```
S1projection = SpannerMatrix1.PseudoInverse[SpannerMatrix1];
MatrixForm[S1projection]
S2projection = SpannerMatrix2.PseudoInverse[SpannerMatrix2];
MatrixForm[S2projection]
```

$$\begin{pmatrix} 0.603393 & -0.441223 & 0.153928 & 0.135312 & -0.0512741 \\ -0.441223 & 0.50893 & 0.171339 & 0.144684 & -0.070372 \\ 0.153928 & 0.171339 & 0.940216 & -0.0498938 & 0.0258753 \\ 0.135312 & 0.144684 & -0.0498938 & 0.792362 & -0.350417 \\ -0.0512741 & -0.070372 & 0.0258753 & -0.350417 & 0.155099 \end{pmatrix}$$

$$\begin{pmatrix} 0.603393 & -0.441223 & 0.153928 & 0.135312 & -0.0512741 \\ -0.441223 & 0.50893 & 0.171339 & 0.144684 & -0.070372 \\ 0.153928 & 0.171339 & 0.940216 & -0.0498938 & 0.0258753 \\ 0.135312 & 0.144684 & -0.0498938 & 0.792362 & -0.350417 \\ -0.0512741 & -0.070372 & 0.0258753 & -0.350417 & 0.155099 \end{pmatrix}$$

The result:

$$\text{S1projection} = \text{S2projection}$$

The call::

The two subspaces are the same.

Reason:

Saying a vector Y is in S1 is the same as saying S1projection.Y = Y.

Saying a vector Y is in S2 is the same as saying S2projection.Y = Y.

And because

$$\text{S1projection} = \text{S2projection},$$

saying that a vector Y is in S1 is the same as saying Y is in S2.

□B.2.c.ii) Different spanning sets for different subspaces

Here's a set of spanning vectors which define a subspace S1 of 6D:

```
Clear[s, i, spanner1];
spanner1[1] = {3.43, -0.57, 8.19, -0.94, 0.48, 0.78};
spanner1[2] = {0.97, 1.29, 0.83, 5.23, -2.31, 7.80};
spanner1[3] = {-6.14, 3.08, -5.00, -1.97, 0.78, 0.85};
spanner1[4] = {0.0, -1.00, 0.0, -2.97, 0.69, 1.75};
spanners1 = Table[spanner1[i], {i, 1, 4}]
{{3.43, -0.57, 8.19, -0.94, 0.48, 0.78},
{0.97, 1.29, 0.83, 5.23, -2.31, 7.8},
{-6.14, 3.08, -5., -1.97, 0.78, 0.85}, {0, -1., 0, -2.97, 0.69, 1.75}}
```

Here's a another set of spanning vectors which define a subspace S2 of 6D:

```
Clear[s, i, spanner2];
spanner2[1] = {1.39, -0.49, -2.96, -4.13, 0.74, 4.19};
spanner2[2] = {-0.68, 1.23, 0.62, 5.09, 2.06, -2.37};
spanner2[3] = {1.16, 1.18, 1.20, -1.30, 2.42, 2.85};
spanner2[4] = {0.48, 2.41, 1.82, 3.79, 4.48, 0.48};
spanners2 = Table[spanner2[i], {i, 1, 4}]
{{1.39, -0.49, -2.96, -4.13, 0.74, 4.19},
{-0.68, 1.23, 0.62, 5.09, 2.06, -2.37},
{1.16, 1.18, 1.2, -1.3, 2.42, 2.85},
{0.48, 2.41, 1.82, 3.79, 4.48, 0.48}}
```

Are these two subspaces the same?

□Answer:

Go directly to the spanner matrices for each subspace.

```
SpannerMatrix1 = Transpose[spanners1];
MatrixForm[SpannerMatrix1]

SpannerMatrix2 = Transpose[spanners2];
MatrixForm[SpannerMatrix2]
```

$$\begin{pmatrix} 3.43 & 0.97 & -6.14 & 0 \\ -0.57 & 1.29 & 3.08 & -1. \\ 8.19 & 0.83 & -5. & 0 \\ -0.94 & 5.23 & -1.97 & -2.97 \\ 0.48 & -2.31 & 0.78 & 0.69 \\ 0.78 & 7.8 & 0.85 & 1.75 \end{pmatrix}$$

$$\begin{pmatrix} 1.39 & -0.68 & 1.16 & 0.48 \\ -0.49 & 1.23 & 1.18 & 2.41 \\ -2.96 & 0.62 & 1.2 & 1.82 \\ -4.13 & 5.09 & -1.3 & 3.79 \\ 0.74 & 2.06 & 2.42 & 4.48 \\ 4.19 & -2.37 & 2.85 & 0.48 \end{pmatrix}$$

Calculate the S1projection matrix and the S2projection matrix and look at them:

```
S1projection = SpannerMatrix1.PseudoInverse[SpannerMatrix1];
MatrixForm[S1projection]
S2projection = SpannerMatrix2.PseudoInverse[SpannerMatrix2];
MatrixForm[S2projection]

{ 0.601023, -0.43777, 0.156589, 0.123484, -0.0914484, -0.0045266,
-0.43777, 0.504919, 0.167416, 0.173455, 0.0134314, 0.0061786,
0.156589, 0.167416, 0.937231, -0.0371411, 0.0698263, 0.0051009,
0.123484, 0.173455, -0.0371411, 0.864044, -0.2646, -0.027292,
-0.0914484, 0.0134314, 0.0698263, -0.2646, 0.101259, -0.087027,
-0.00452663, 0.00617861, 0.00510096, -0.0272926, -0.0870274, 0.991525}

{ 0.0737793, 0.0522096, 0.00451895, -0.102958, 0.135945, 0.191071 }
{ 0.0522096, 0.150556, 0.181612, 0.127324, 0.260096, 0.0912049 }
{ 0.00451895, 0.181612, 0.954586, -0.0666378, -0.0152872, -0.0753269 }
{-0.102958, 0.127324, -0.0666378, 0.704669, 0.255263, -0.334214 }
{ 0.135945, 0.260096, -0.0152872, 0.255263, 0.604059, 0.296054 }
{ 0.191071, 0.0912049, -0.0753269, -0.334214, 0.296054, 0.512351 }
```

The result: S1projection ≠ S2projection

The call: The two subspaces are the not same.

Reason: Saying a vector Y is in S1 is the same as saying S1projection.Y = Y.
Saying a vector Y is in S2 is the same as saying S2projection.Y = Y.
And because S1projection ≠ S2projection, saying that a vector Y is in S1 is not the same as saying Y is in S2.

subspace S of nD is linearly independent is the same as saying the dimension of S is k.

Redundant spanning sets and linear dependence

Most folks like to say: A given spanning set for a subspace S is linearly independent if the dimension of S is the same as the number of vectors in the given spanning set.

A given spanning set for a subspace S is linearly dependent if the dimension of S is less than the number of vectors in the given spanning set.

□B.3.a.i) A linearly independent set in 6D

Here is a spanning set for a subspace S of 6D:

```
Clear[spanner, i];
spanner[1] = {1.0, -2.3, 5.0, 6.7, -4.9, -2.0};
spanner[2] = {0.3, 2.3, -4.6, -1.7, 5.8, 0.0};
spanner[3] = {1.0, 7.1, 5.0, 3.7, 0.9, -2.0};
spanners = Table[spanner[i], {i, 1, 3}]
{{1., -2.3, 5., 6.7, -4.9, -2.},
{0.3, 2.3, -4.6, -1.7, 5.8, 0}, {1., 7.1, 5., 3.7, 0.9, -2.}}
```

Determine whether this spanning set is linearly independent or linearly dependent.

□Answer:

The spanner matrix for the subspace spanned by the given spanning set is:

```
SpannerMatrix = Transpose[spanners];
MatrixForm[SpannerMatrix]
```

$$\begin{pmatrix} 1. & 0.3 & 1. \\ -2.3 & 2.3 & 7.1 \\ 5. & -4.6 & 5. \\ 6.7 & -1.7 & 3.7 \\ -4.9 & 5.8 & 0.9 \\ -2. & 0 & -2. \end{pmatrix}$$

The dimension of the subspace spanned by these vectors is the rank of SpannerMatrix:

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

The number of vectors in this given spanning set is:

```
Length[spanners]
3
```

The result: The dimension of the subspace spanned by this spanning set is the same as the number of vectors in the given spanning set.

The call: This spanning set is LINEARLY INDEPENDENT.

□B.3.a.ii) A linearly dependent set in 7D

Here is a spanning set of made with four vectors in 7D:

```
Clear[spanner, i];
spanner[1] = {1.0, -2.3, 5.0, 6.7, -4.9, -2.0, -2.3};
spanner[2] = {0.3, 2.3, -4.6, -1.7, 5.8, 0.0, 3.5};
spanner[3] = {1.0, 7.1, 5.0, 3.7, 0.9, -2.0, 4.9};
spanner[4] = {1.6, 11.7, -4.2, 0.3, 12.5, -2.0, 11.9};
spanners = Table[spanner[i], {i, 1, 4}]
{{1., -2.3, 5., 6.7, -4.9, -2., -2.3},
{0.3, 2.3, -4.6, -1.7, 5.8, 0, 3.5}, {1., 7.1, 5., 3.7, 0.9, -2., 4.9},
{1.6, 11.7, -4.2, 0.3, 12.5, -2., 11.9}}
```

Determine whether this spanning set is linearly independent or linearly dependent.

□Answer:

The spanner matrix for the subspace spanned by the given spanning set is:

```
SpannerMatrix = Transpose[spanners];
MatrixForm[SpannerMatrix]
```

$$\begin{pmatrix} 1. & 0.3 & 1. & 1.6 \\ -2.3 & 2.3 & 7.1 & 11.7 \\ 5. & -4.6 & 5. & -4.2 \\ 6.7 & -1.7 & 3.7 & 0.3 \\ -4.9 & 5.8 & 0.9 & 12.5 \\ -2. & 0 & -2. & -2. \\ -2.3 & 3.5 & 4.9 & 11.9 \end{pmatrix}$$

The dimension of the subspace spanned by these vectors is the rank of SpannerMatrix:

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

The number of vectors in this given spanning set is:

```
Length[spanners]
4
```

The result: The dimension of the subspace spanned by this spanning set is less than the number of vectors in the given spanning set.

B.3) Linear independence and dimension:

Saying that a spanning set {spanner[1],spanner[2],...spanner[k]} for a

The call:
This spanning set is **LINEARLY DEPENDENT**.

□ **B.3.a.iii) A set of four 3D vectors cannot be linearly independent**

You are given a set of four vectors in 3D.
How do you know in advance that the given set of four given vectors is linearly dependent?

□ **Answer:**

If the four vectors are linearly independent, then they span a **four** dimensional subspace of 3D.
But 3D can accommodate at most 3 dimensions and this rules out the possibility of any four dimensional subspace of 3D.

□ **B.3.a.iv) A set of twelve 8D vectors cannot be linearly independent**

You are given a set of 12 vectors in 8D.
How do you know in advance that the given set of 12 given vectors is linearly dependent?

□ **Answer:**

If the 12 vectors are linearly independent, then they span an **12** dimensional subspace of 8D.
But 8D can accommodate at most 8 dimensions and this rules out the possibility of any 12 dimensional subspace of 8D.

□ **B.3.b.i) Finding redundant vectors in linearly dependent sets**

Here's a spanning set for a subspace S of 5D:

```
Clear[s, i, spanner];
spanner[1] = {1.43, 2.57, 6.19, -3.94, 0.0};
spanner[2] = {0.95, 1.50, 1.85, 1.20, 0.50};
spanner[3] = {0.67, 1.37, 4.71, -4.9, -0.4};
spanner[4] = {2.1, 3.94, 10.9, -8.84, -0.4};
spanner[5] = {1.10, 2.00, -2.1, -1.90, 0.46};
spanners = Table[spanner[i], {i, 1, 5}]
{{1.43, 2.57, 6.19, -3.94, 0},
 {0.95, 1.5, 1.85, 1.2, 0.5}, {0.67, 1.37, 4.71, -4.9, -0.4},
 {2.1, 3.94, 10.9, -8.84, -0.4}, {1.1, 2., -2.1, -1.9, 0.46}}
```

When you look at:

```
SpannerMatrix = Transpose[spanners];
MatrixForm[SpannerMatrix];
Sdim = Length[SingularValues[SpannerMatrix][[2]]]
3
```

And:

```
Length[spanners]
5
```

You see:
The original spanning set is linearly dependent.

And after a little more thought, you realize the $5 - 3 = 2$ of the given spanners are redundant in the sense when you throw them out of the spanning set, the resulting spanning set still spans the same subspace S.
Find the redundant vectors in the given spanning set.

□ **Answer:**

To identify the redundant spanners, look at

```
spanners12 = {spanner[1], spanner[2]};
SpannerMatrix12 = Transpose[spanners12];
rank = Length[SingularValues[SpannerMatrix12][[2]]];
S12dim = rank
2
```

This tells you that {spanner[1], spanner[2]} span a two dimensional subspace S12 of 5D.

The call: Neither spanner[1] nor spanner[2] are redundant.

Now look at:

```
spanners123 = {spanner[1], spanner[2], spanner[3]};
SpannerMatrix123 = Transpose[spanners123];
rank = Length[SingularValues[SpannerMatrix123][[2]]];
S123dim = rank
2
```

The upshot: Including spanner[3] does not raise the dimension.

So spanner[3] belongs to the subspace S12 spanned by {spanner[1], spanner[2]}.

The call: spanner[3] is redundant.

This means it is possible express spanner[3] in terms of spanner[1] and spanner[2] via

$$\text{spanner}[3] = x[1] \text{spanner}[1] + x[2] \text{spanner}[2]$$
for some choice of numbers $x[1]$ and $x[2]$.

Now look at:

```
spanners1234 = {spanner[1], spanner[2], spanner[3], spanner[4]};
SpannerMatrix1234 = Transpose[spanners1234];
rank = Length[SingularValues[SpannerMatrix1234][[2]]];
S1234dim = rank
2
```

The upshot: Including spanner[4] does not raise the dimension.

So spanner[3] belongs to the subspace S12 spanned by {spanner[1], spanner[2]}.

The call: spanner[4] is redundant.

Now look at:

```
spanners12345 =
{spanner[1], spanner[2], spanner[3], spanner[4], spanner[5]};
SpannerMatrix12345 = Transpose[spanners12345];
rank = Length[SingularValues[SpannerMatrix12345][[2]]];
S12345dim = rank
3
```

The upshot: Including spanner[5] does raise the dimension.

The call: spanner[5] is not redundant.

The original (linearly dependent) spanning set for S is:

```
originalspanners =
{spanner[1], spanner[2], spanner[3], spanner[4], spanner[5]}
{{1.43, 2.57, 6.19, -3.94, 0},
 {0.95, 1.5, 1.85, 1.2, 0.5}, {0.67, 1.37, 4.71, -4.9, -0.4},
 {2.1, 3.94, 10.9, -8.84, -0.4}, {1.1, 2., -2.1, -1.9, 0.46}}
```

The redundant vectors are spanner[3] and spanner[4]. Throwing them out gives a linearly independent spanning set for S (with no redundant vectors):

```
reducedspanners = {spanner[1], spanner[2], spanner[5]}
{{1.43, 2.57, 6.19, -3.94, 0},
 {0.95, 1.5, 1.85, 1.2, 0.5}, {1.1, 2., -2.1, -1.9, 0.46}}
```

Confirm by looking at the Sprojection matrices coming from the original spanning set and the reduced spanning set:

```
SpannerMatrix1 = Transpose[originalspanners];
SpannerMatrix2 = Transpose[reducedspanners];

S1projection = SpannerMatrix1.PseudoInverse[SpannerMatrix1];
MatrixForm[S1projection]

S2projection = SpannerMatrix2.PseudoInverse[SpannerMatrix2];
MatrixForm[S2projection]
```

$\left(\begin{array}{ccccc} 0.245193 & 0.411998 & 0.0117952 & 0.0133179 & 0.122535 \\ 0.411998 & 0.696312 & 0.00594912 & -0.0392122 & 0.200366 \\ 0.0117952 & 0.00594912 & 0.997869 & 0.00481288 & -0.0439228 \\ 0.0133179 & -0.0392122 & 0.00481288 & 0.986817 & 0.106163 \\ 0.122535 & 0.200366 & -0.0439228 & 0.106163 & 0.0738089 \end{array} \right)$
$\left(\begin{array}{ccccc} 0.245193 & 0.411998 & 0.0117952 & 0.0133179 & 0.122535 \\ 0.411998 & 0.696312 & 0.00594912 & -0.0392122 & 0.200366 \\ 0.0117952 & 0.00594912 & 0.997869 & 0.00481288 & -0.0439228 \\ 0.0133179 & -0.0392122 & 0.00481288 & 0.986817 & 0.106163 \\ 0.122535 & 0.200366 & -0.0439228 & 0.106163 & 0.0738089 \end{array} \right)$

Right on target.

□ **B.3.b.ii) Linearly independent sets have no redundant members**

Given a spanning set, agree that one of the spanners is redundant if you remove it from the spanning set and the remaining spanners span the same subspace as the original spanning set.

Explain why it is IMPOSSIBLE for a linearly independent spanning set to contain a redundant vector.

□ **Answer:**

If a given spanning set

```
{spanner[1], spanner[2], ..., spanner[k]}
```

is linearly independent, then the dimension of the subspace S spanned by this set is k.

If you remove any one of the spanners from this set, then you are left with (k-1) vectors.

These (k - 1) vectors can span at most a (k-1) dimensional subspace. This (k-1) dimensional subspace cannot be all of S (because S is k-dimensional).

B.4) A basis for S is any linearly independent spanning set for S.

Using calculations of dimension to help to pare a linearly dependent spanning set down to a basis

Most folks say that a basis for a subspace S is any linearly independent spanning set for S.

□B.4.a.i) A basis for a subspace S is any spanning set for S not containing a redundant vector

Is it true that a basis for a subspace S is any spanning set for S not containing a redundant vector?

□Answer:

In one word:

Yes!

□B.4.a.ii) Saying that a spanning set is a basis is the same as saying that the associated spanner matrix is of full rank

Is it true that saying that a spanning set is a basis is the same as saying that the associated spanner matrix is of full rank?

□Answer:

In one word:

Yes!

□B.4.b) Basis expansion

Given a basis {spanner[1], spanner[2], ..., spanner[k]} of a subspace S, explain why X in S determines unique coefficients x[1], x[2], ..., x[k] so that

$$X = \sum_{i=1}^k x[i] \text{spanner}[i]$$

□Answer:

You are given that

$$\{\text{spanner}[1], \text{spanner}[2], \dots, \text{spanner}[k]\}$$

is linearly independent.

This tells you that the associated spanner matrix is of full rank.

Now saying

$$X = \sum_{i=1}^k x[i] \text{spanner}[i]$$

is the same as saying

$$X = \text{SpannerMatrix}.\{x[1], x[2], \dots, x[k]\}$$

This is the same as saying

$$XX = \{x[1], x[2], \dots, x[k]\}$$

is a solution of

$$\text{SpannerMatrix}.XX = X.$$

And because SpannerMatrix is of full rank,

the solution

$$XX = \{x[1], x[2], \dots, x[k]\}$$

is the only solution.

So the coefficients {x[1], x[2], ..., x[k]}

that make

$$X = \sum_{i=1}^k x[i] \text{spanner}[i]$$

are uniquely determined.

□B.4.c.i) Is it a basis?

Here's a spanning set for a subspace S of 4D:

```
Clear[s, i, spanner];
spanner[1] = {3.43, -0.57, 8.19, -0.94};
spanner[2] = {0.97, 1.29, 0.83, 5.23};
spanner[3] = {-6.14, 3.08, -5.00, -1.97};
spanners = Table[spanner[i], {i, 1, 3}];
```

```
ColumnForm[spanners]
{3.43, -0.57, 8.19, -0.94}
{0.97, 1.29, 0.83, 5.23}
{-6.14, 3.08, -5., -1.97}
```

Determine whether this spanning set is a basis for S.

□Answer:

This is the same as checking whether this spanning set is linearly independent.

To do this, check the rank of the associated spanner matrix:

```
SpannerMatrix = Transpose[spanners];
rank = Length[SingularValues[SpannerMatrix][[2]]];
Sdim = rank
3
```

This tells you that these 3 spanning vectors span a 3-dimensional subspace of S of 4D, so the given spanner set is linearly independent.

This confirms that the given spanning set is a basis of S.

□B.4.c.ii) Calculating a basis expansion

Stay with the same spanning set and subspace S as in part i) and look at the projection of a random 4D vector onto S:

```
random4D = {Random[Real, {-4, 4}], Random[Real, {-4, 4}],
            Random[Real, {-4, 4}], Random[Real, {-4, 4}]};

Sprojection = SpannerMatrix.PseudoInverse[SpannerMatrix];

Svector = Sprojection.random4D
{1.80159, -2.45412, -0.34201, -1.9882}
```

This vector is guaranteed to be in S.

Your job is to come up with the basis expansion of Svector by coming up with the unique numbers

x[1], x[2] and x[3]

that make

$$Svector = \sum_{i=1}^3 x[i] \text{spanner}[i].$$

□Answer:

The unique x[1], x[2] and x[3] that make

$$Svector = \sum_{i=1}^3 x[i] \text{spanner}[i]$$

solve the linear system

$$\text{SpannerMatrix}.\{x[1], x[2], x[3]\} = Svector.$$

The solution is:

```
{x[1], x[2], x[3]} = PseudoInverse[SpannerMatrix].Svector
{-0.330321, -0.658753, -0.582017}
```

Check:

```
\sum_{i=1}^3 x[i] spanner[i]
Svector
{1.80159, -2.45412, -0.34201, -1.9882}
{1.80159, -2.45412, -0.34201, -1.9882}
```

Nailed it.

□B.4.d.i) Is it a basis?

Here's a spanning set for a subspace S of 5D:

```
Clear[s, i, spanner];
spanner[1] = {1.43, 2.57, 6.19, -3.94, 0.0};
spanner[2] = {0.95, 1.50, 1.85, 1.20, 0.50};
spanner[3] = {0.67, 1.37, 4.71, -4.9, -0.4};
spanner[4] = {2.1, 3.94, 10.9, -8.84, -0.4};
spanner[5] = {1.10, 2.00, -2.1, -1.90, 0.46};
spanners = Table[spanner[i], {i, 1, 5}];
```

```
ColumnForm[spanners]
{1.43, 2.57, 6.19, -3.94, 0}
{0.95, 1.5, 1.85, 1.2, 0.5}
{0.67, 1.37, 4.71, -4.9, -0.4}
{2.1, 3.94, 10.9, -8.84, -0.4}
{1.1, 2., -2.1, -1.9, 0.46}
```

Determine whether this spanning set is a basis for S.

□Answer:

This is the same as checking whether this spanning set is linearly independent.

To do this, check the rank of the associated spanner matrix:

```
SpannerMatrix = Transpose[spanners];
rank = Length[SingularValues[SpannerMatrix][[2]]];
Sdim = rank
3
```

This tells you that these 5 spanning vectors span a 3-dimensional subspace of S of 5D, so the given spanner set is linearly dependent.

This confirms that the given spanning set is NOT a basis of S.

□B.4.d.ii) Paring a linearly dependent spanning set down to a basis

Stay with the same spanning set for a subspace S of 5D as in part i).

```
Clear[s, i, spanner];
spanner[1] = {1.43, 2.57, 6.19, -3.94, 0.0};
spanner[2] = {0.95, 1.50, 1.85, 1.20, 0.50};
spanner[3] = {0.67, 1.37, 4.71, -4.9, -0.4};
spanner[4] = {2.1, 3.94, 10.9, -8.84, -0.4};
spanner[5] = {1.10, 2.00, -2.1, -1.90, 0.46};
spanners = Table[spanner[i], {i, 1, 5}];
```

```
ColumnForm[spanners]
```



```
{1.43, 2.57, 6.19, -3.94, 0}
{0.95, 1.5, 1.85, 1.2, 0.5}
{0.67, 1.37, 4.71, -4.9, -0.4}
{2.1, 3.94, 10.9, -8.84, -0.4}
{1.1, 2., -2.1, -1.9, 0.46}
```

Sitting in this spanning set is a basis for S.

Come up with this basis for S by identifying and throwing out the redundant spanning vectors.

□ Answer:

In part i), You saw that the dimension of subspace S spanned by the original spanning set is 3. Because there are 5 vectors in the spanning set you are sure that the original spanning set has two redundant vectors.

To identify them, look at

```
spanners12 = {spanner[1], spanner[2]};
SpannerMatrix12 = Transpose[spanners12];
rank = Length[SingularValues[SpannerMatrix12][[2]]];
S12dim = rank
2
```

This tells you that {spanner[1], spanner[2]} span a two dimensional subspace S12 of 5D.

The call: Neither spanner[1] nor spanner[2] are redundant.

Now look at:

```
spanners123 = {spanner[1], spanner[2], spanner[3]};
SpannerMatrix123 = Transpose[spanners123];
rank = Length[SingularValues[SpannerMatrix123][[2]]];
S123dim = rank
2
```

The upshot: Including spanner[3] does not raise the dimension.

So spanner[3] belongs to the subspace S12 spanned by {spanner[1], spanner[2]}.

The call: spanner[3] is redundant.

Now look at:

```
spanners1234 = {spanner[1], spanner[2], spanner[3], spanner[4]};
SpannerMatrix1234 = Transpose[spanners1234];
rank = Length[SingularValues[SpannerMatrix1234][[2]]];
S1234dim = rank
2
```

The upshot: Including spanner[4] does not raise the dimension.

So spanner[3] belongs to the subspace S12 spanned by {spanner[1], spanner[2]}.

The call: spanner[4] is redundant.

Now look at:

```
spanners12345 =
{spanner[1], spanner[2], spanner[3], spanner[4], spanner[5]};
SpannerMatrix12345 = Transpose[spanners12345];
rank = Length[SingularValues[SpannerMatrix12345][[2]]];
S12345dim = rank
3
```

The upshot: Including spanner[5] does raise the dimension.

The call: spanner[5] is not redundant.

The original (linearly dependent) spanning set for S is:

```
{spanner[1], spanner[2], spanner[3], spanner[4], spanner[5]}
{{1.43, 2.57, 6.19, -3.94, 0},
{0.95, 1.5, 1.85, 1.2, 0.5}, {0.67, 1.37, 4.71, -4.9, -0.4},
{2.1, 3.94, 10.9, -8.84, -0.4}, {1.1, 2., -2.1, -1.9, 0.46}}
```

The redundant vectors are spanner[3] and spanner[4]. Throwing them out gives a (linearly independent) basis for S:

```
{spanner[1], spanner[2], spanner[5]}
{{1.43, 2.57, 6.19, -3.94, 0},
{0.95, 1.5, 1.85, 1.2, 0.5}, {1.1, 2., -2.1, -1.9, 0.46}}
```

B.5) Traditional book definitions of linear independence and linear dependence

□ B.5.a.i) Saying a spanning set is linearly independent is the same as saying the SpannerMatrix is of full rank

Explain this:

Saying that a given spanning set

```
{spanner[1], spanner[2], ..., spanner[k]}
```

is linearly independent is the same as saying that the associated spanner matrix is of full rank.

□ Answer:

Call S the subspace of nD spanned by the given spanning set

```
{spanner[1], spanner[2], ..., spanner[k]}.
```

The associated spanner matrix (whose ith column is spanner[i]) hits on kD.

Saying that the spanner matrix is of full rank is the same as saying that the rank of the spanner matrix is k.

Saying that {spanner[1], spanner[2], ..., spanner[k]} is linearly independent is the same as saying that the rank of the spanner matrix is k.

So saying that a given spanning set

```
{spanner[1], spanner[2], ..., spanner[k]}
```

is linearly independent is the same as saying that the associated spanner matrix is of full rank.

□ B.5.a.ii) Traditional book definition of linear independence

What you see below is the definition of linear independence that you will see in most traditional linear algebra or matrix theory texts. Many students have struggled with this definition.

Explain this arcane statement:

Saying that a given spanning set

```
{spanner[1], spanner[2], ..., spanner[k]}
```

is linearly independent the same as saying that it is **IMPOSSIBLE** to come up with numbers

```
{x[1], x[2], ..., x[k]}
```

not all of which are zero, so that

$$\sum_{i=1}^k x[i] \text{spanner}[i] = \{0, 0, \dots, 0\}.$$

□ Answer:

Saying that a given spanning set

```
{spanner[1], spanner[2], ..., spanner[k]}
```

is linearly independent is the same as saying that the associated spanner matrix is of full rank.

This is the same as saying that all the SVD stretch factors for the spanner matrix are positive.

This is the same as saying that the only solution of

$$\text{SpannerMatrix}.\{x[1], x[2], \dots, x[k]\} = \{0, 0, \dots, 0\}$$

is

$$\{x[1], x[2], \dots, x[k]\} = \{0, 0, \dots, 0\}.$$

Because

$$\text{SpannerMatrix}.\{x[1], x[2], \dots, x[k]\} = \sum_{i=1}^k x[i] \text{spanner}[i],$$

this is the same as saying that the only solution of

$$\sum_{i=1}^k x[i] \text{spanner}[i] = \{0, 0, \dots, 0\}$$

is

$$\{x[1], x[2], \dots, x[k]\} = \{0, 0, \dots, 0\}.$$

And this is the same as saying it is impossible to come up with numbers

```
{x[1], x[2], ..., x[k]}
```

not all of which are zero, so that

$$\sum_{i=1}^k x[i] \text{spanner}[i] = \{0, 0, \dots, 0\}.$$

□ B.5.b.i) Saying a spanning set is linearly dependent is the same as saying the SpannerMatrix is not of full rank

Explain this:

Saying that a given spanning set

```
{spanner[1], spanner[2], ..., spanner[k]}
```

is linearly dependent is the same as saying that the associated spanner matrix is not of full rank.

□ Answer:

Call S the subspace spanned by the given spanning set

```
{spanner[1], spanner[2], ..., spanner[k]}.
```

The associated spanner matrix (whose ith column is spanner[i]) hits on kD.

Saying that the spanner matrix is not of full rank is the same as saying that the rank of the spanner matrix is less than k.

Saying that {spanner[1], spanner[2], ..., spanner[k]} is linearly dependent is the also same as saying that the rank of the spanner matrix is less than k.

So saying that a given spanning set

$$\{\text{spanner}[1], \text{spanner}[2], \dots, \text{spanner}[k]\}$$

is linearly dependent is the same as saying that the associated spanner matrix is of not of full rank.

□ B.5.b.ii) Traditional book definition of linear dependence

What you see below is the definition of linear dependence that you will see in most traditional linear algebra or matrix theory texts. Many students have also struggled with this definition.

Explain this arcane statement:

Saying that a given spanning set

$$\{\text{spanner}[1], \text{spanner}[2], \dots, \text{spanner}[k]\}$$

is linearly dependent is the same as saying that it is POSSIBLE to come up with numbers

$$\{x[1], x[2], \dots, x[k]\}$$

not all of which are zero, so that

$$\sum_{i=1}^k x[i] \text{spanner}[i] = \{0, 0, \dots, 0\}.$$

□ Answer:

Saying that a given spanning set

$$\{\text{spanner}[1], \text{spanner}[2], \dots, \text{spanner}[k]\}$$

is linearly dependent is the same as saying that the associated spanner matrix is of not of full rank.

This is the same as saying that at least one SVD stretch factors for the spanner matrix is 0.

This is the same as saying there is at least one vector

$$\{x[1], x[2], \dots, x[k]\} \neq \{0, 0, \dots, 0\}$$

so that

$$\text{SpannerMatrix} \cdot \{x[1], x[2], \dots, x[k]\} = \{0, 0, \dots, 0\}.$$

Because

$$\text{SpannerMatrix} \cdot \{x[1], x[2], \dots, x[k]\} = \sum_{i=1}^k x[i] \text{spanner}[i],$$

this is the same as saying there is at least one vector

$$\{x[1], x[2], \dots, x[k]\} \neq \{0, 0, \dots, 0\}$$

so that

$$\sum_{i=1}^k x[i] \text{spanner}[i] = \{0, 0, \dots, 0\}.$$

And this is the same as saying it is possible to come up with numbers

$$\{x[1], x[2], \dots, x[k]\}$$

not all of which are zero, so that

$$\sum_{i=1}^k x[i] \text{spanner}[i] = \{0, 0, \dots, 0\}.$$

B.6) All bases for a given subspace of nD have the same number of vectors

Implicit in the definition of dimension as the rank of the Spanner Matrix is the fact that if a subspace S of nD is spanned by a perpendicular frame $\{B_1, B_2, \dots, B_k\}$, then any basis of S also consists of exactly k vectors.

The purpose of this problem is to explain this in detail.

Mathematics majors should fully engage. Others might want to as well.

If so, then continue on for a dose of some advanced mathematical reasoning.

□ B.6.a.i) If a subspace S of nD is spanned by perpendicular frame (orthonormal basis) consisting of exactly k vectors, then all other bases of S also consist of exactly k vectors

This problem was adapted from the classic book by Nicolaas H. Kuiper, Linear Algebra and Geometry, North Holland, Amsterdam, 1965.

Explain this statement:

If a subspace S of nD is spanned by perpendicular frame (orthonormal basis)

$$\{B_1, B_2, \dots, B_k\}$$

then all other bases (linearly independent spanning sets) of S consist of exactly k vectors.

□ Answer:

This is what mathematicians call a proof by induction.

The statement is true for k=1.

To prove this by induction, you take a positive integer k.

You get to assume that for any positive integer m < k that

if a subspace S of nD is spanned by perpendicular frame (orthonormal basis)

$$\{B_1, B_2, \dots, B_m\}$$

then all other bases of S consist of exactly m vectors.

The goal is to use this assumption to argue why the statement is true when you replace m with k.

To this end, go with a subspace S of nD spanned by a perpendicular frame

$$\{B_1, B_2, \dots, B_k\}$$

Put

SS = the subspace spanned by

$$\{B_1, B_2, \dots, B_{k-1}\}$$

Legal Assumption:

Because k - 1 < k, you get to assume that every basis for SS consists of exactly k-1 vectors.

Notice that SS = everything in S that is perpendicular to B_k.

Now take another basis

$$\{A_1, A_2, \dots, A_r\}$$

of S.

The goal is to explain why r = k.

First note that it must be true that B_k • A_j ≠ 0 for some j.

If r > j, swap the labels so that the old A_j becomes the new A_r.

This guarantees that B_k • A_r ≠ 0.

Now look at the vectors

$$C_i = A_i - \left(\frac{B_k \cdot A_i}{B_k \cdot A_r} \right) A_r \quad \text{for } i=1, 2, \dots, r-1.$$

Fact 1: The set {C₁, C₂, ..., C_{r-1}} is linearly independent.

To see a reason, click on the right.

Reason: Because

$$\{A_1, A_2, \dots, A_r\}$$

is linearly independent,

and

$$C_i = A_i - \left(\frac{B_k \cdot A_i}{B_k \cdot A_r} \right) A_r \quad \text{for } i=1, 2, \dots, r-1.$$

you know that

$$\sum_{i=1}^{r-1} \lambda_i C_i = \sum_{i=1}^{r-1} \lambda_i A_i + (\text{messy constant}) A_r$$

cannot be {0, 0, ..., 0} unless λ₁ = λ₂ = ... = λ_{r-1} = 0.

This explains (via the traditional definition of linear independence) why the set {C₁, C₂, ..., C_{r-1}} is linearly independent.

Fact 2: The set {C₁, C₂, ..., C_{r-1}} spans SS and is therefore a basis of SS.

To see a reason, click on the right.

Because S consists of all sums

$$\sum_{i=1}^r \lambda_i A_i$$

and because SS is everything in S perpendicular to B_k, you know that SS is all sums

$$A = \sum_{i=1}^r \lambda_i A_i \quad \text{with} \quad \sum_{i=1}^r \lambda_i A_i \cdot B_k = 0.$$

Remember that

$$C_i = A_i - \left(\frac{B_k \cdot A_i}{B_k \cdot A_r} \right) A_r \quad \text{for } i=1, 2, \dots, r-1.$$

So SS is all sums

$$\begin{aligned} A &= \sum_{i=1}^{r-1} \lambda_i \left(C_i + \left(\frac{B_k \cdot A_i}{B_k \cdot A_r} \right) A_r \right) + \lambda_r A_r \quad \text{with} \quad \sum_{i=1}^r \lambda_i A_i \cdot B_k = 0 \\ &= \sum_{i=1}^{r-1} \lambda_i C_i + \sum_{i=1}^{r-1} \lambda_i \left(\frac{B_k \cdot A_i}{B_k \cdot A_r} \right) A_r + \lambda_r A_r \quad \text{with} \quad \sum_{i=1}^r \lambda_i A_i \cdot B_k = 0. \\ &= \sum_{i=1}^{r-1} \lambda_i C_i + \sum_{i=1}^{r-1} \lambda_i \left(\frac{B_k \cdot A_i}{B_k \cdot A_r} \right) A_r + \lambda_r \left(\frac{B_k \cdot A_r}{B_k \cdot A_r} \right) A_r \quad \text{with} \quad \sum_{i=1}^r \lambda_i A_i \cdot B_k = 0. \\ &= \sum_{i=1}^{r-1} \lambda_i C_i + \sum_{i=1}^r \lambda_i \left(\frac{B_k \cdot A_i}{B_k \cdot A_r} \right) A_r \quad \text{with} \quad \sum_{i=1}^r \lambda_i A_i \cdot B_k = 0. \\ &= \sum_{i=1}^{r-1} \lambda_i C_i + \frac{1}{B_k \cdot A_r} A_r \sum_{i=1}^r \lambda_i (B_k \cdot A_i) \quad \text{with} \quad \sum_{i=1}^r \lambda_i A_i \cdot B_k = 0. \\ &= \sum_{i=1}^{r-1} \lambda_i C_i + 0 \quad \text{with} \quad \sum_{i=1}^r \lambda_i A_i \cdot B_k = 0. \\ &= \sum_{i=1}^{r-1} \lambda_i C_i \end{aligned}$$

This shows that the set $\{C_1, C_2, \dots, C_{r-1}\}$ spans SS.

And because the set $\{C_1, C_2, \dots, C_{r-1}\}$ is also linearly independent, the set $\{C_1, C_2, \dots, C_{r-1}\}$ is a basis of SS.

Fact 3: $r = k$.

To see a reason, click on the right.

$\{C_1, C_2, \dots, C_{r-1}\}$ is a basis of SS.

The legal assumption is that **every basis for SS consists of exactly $k-1$ vectors.**

So $r - 1 = k - 1$. This yields the equality $r = k$.

Summary:

You started with a subspace S of nD spanned by a perpendicular frame

$$\{B_1, B_2, \dots, B_k\}$$

And then you took any old basis

$$\{A_1, A_2, \dots, A_r\}$$

of S and argued why $r = k$.

This shows that all other bases (linearly independent spanning sets) of S consist of exactly k vectors.

□ **B.6.a.ii) If a subspace S of nD is spanned by basis (linearly independent spanning set) consisting of exactly k vectors, then all other bases of S also consist of exactly k vectors**

Explain this statement:

If a subspace S of nD is spanned by a basis (linearly independent spanning set but not necessarily orthonormal)

$$\{X_1, X_2, \dots, X_k\}$$

then all other bases (linearly independent spanning sets) of S consist of exactly k vectors.

□ **Answer:**

Apply the Gram-Schmidt (See Tutorials) process to

$$\{X_1, X_2, \dots, X_k\}$$

to get an orthonormal basis

$$\{G_1, G_2, \dots, G_k\}$$

of S. According to the problem immediately above, all other bases of S consist of exactly k vectors.