Homogeneous-Transform-Based Optical Analysis

Abbreviated as:

"H-Optics"

Posted on : www.setterholm.com /Optics

Author: Jeffrey M. Setterholm
Systems Engineer
Lakeville, Minnesota, USA
(952) 461-3445

Version: 0.8
Date: January 8, 2010

Image #1: A mirrored figurine:

Image #2: ~ A unity lens-volume-mapped figurine. Analysis herein addresses the phenomena.

Executive Summary:

Cameras capture images of scenes by projecting a scene volume onto an image plane.

But in general – lenses map 3-D volumes into other 3-D volumes, and homogeneous transforms (4x4 matrices) can be used to characterize the general behavior.

The H-optics approach can be used to deepen the understanding of "What lenses do."

Local "History":

A competent present-day optical system designer recently told me that homogeneous transforms are not part of his design tools environment; beyond that, I don't know what is known or published about (what I call) "H-optics".

My analytical approach is a merging of traditional (freshman) Physics-I introductory optics with the math of homogeneous transforms – given the additional insight that: "… lenses map volumes into volumes. ", which was taught to me by Marvin L. Pund in the latter 1970's during the time that we both worked at McDonnell Aircraft in St. Louis. Marv prefaced the communication with: "You know…", which, when spoken by technologists, is often followed by a novel idea. At the time Marv was creating spectacularly innovative optical paths that I'd seen with my own eyes, so his observation was instantly & completely credible.

In the early 1980's I spent a few weeks in beginning to understand the mathematics of "lens volume mapping", using 1960's freshman Physics concepts.

Single lenses significantly distort the volume surrounding an "Object" in creating the corresponding "Image" volume. In fact, when the Object depth is at the lens focal length, the Image depth is discontinuous between infinity on the far side of the lens and infinity on the near side of the lens.

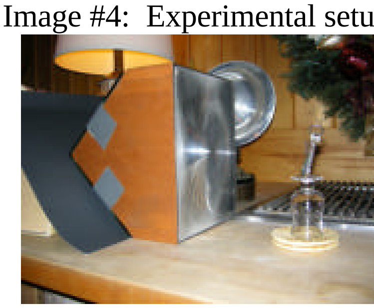
Using algebra, a second lens (of the same focal length) placed two focal lengths beyond the first lens was shown to remove the distortions and discontinuities of the first lens resulting in a net translation of the Object volume to an image volume four focal lengths down the optical path, along with a 180 degree rotation around the optical path – so that the image volume is undistorted but upside down.

In fairly short order I built the "optical demonstrator" shown at the right: →
A Fresnel lens combines with two first-surface mirrors set one focal length behind the lens and inclined at +- 45. degrees. The lens does double-duty in the optical path as the first and last element.

Image #3: Optical demonstrator

The experimental setup: →
As shown in image #2 on the first page, the real and real-image figurines are both about one focal length in front of the lens, In contrast, in image #1, the mirrored figurine is much smaller and more distant than the real figurine.

Image #4: Experimental setup



Herein "**H**" is my abbreviation for a single square matrix having four rows and four columns (16 elements total) which is interpreted as a **homogeneous matrix** (e.g. the bottom row of the matrix operates as a division factor). Plural: " **H's** ".

The remainder of this paper will introduce the analytics of "H-Optics" in the course of explaining how the behavior of my optical demonstrator (images #3) can be predicted.

Elementary Optics:

My reference of choice for optics is my freshman Physics book. The present-day version is: "Halliday/Resnick Fundamentals of Physics, 8^{th} edition", Jearl Walker, 2008, ISBN978-0-470-04472-8, chapter 34.

Two equations form the basis of elementary optics of a "thin lens":

Let

O = the distance of an object from a lens, and
I = the distance of an image from a lens, and
F = the focal length of a lens

Then:

1./**O** - 1./***I*** = 1./**F** which describes distance relationships along the optical axis, and — Equation #1

Magnification= **I**/**O** which describes displacements transverse to the optical axis. — Equation #2:

In analyzing optics, plus and minus signs vary, depending on whether reflective or refractive optics are being considered, and also how the coordinates of image space differ from the coordinates of object space. I've made the object space and the image space a common coordinate frame in equation #1, which calls for subtracting 1./I from 1./**O** .

Equations #1 and #2 will be combined into a single H, and the utility of the equations will be greatly expanded in the process.

An Introduction to H's :

My introduction to H's came while programming computer graphics using OpenGL which was created by Silicon Graphics. Reference e.g.: "OpenGL Programming Guide, Second Edition, the Official guide to OpenGL, Version 1.1", Woo, Neider, & Davis, 1997, ISBN 0-201-46138-2, including appendix F. ("Lens volume mapping" is outside the scope of the appendix.) OpenGL analytics extensively use H's.

H's use the bottom row of the matrix to perform division of the upper three rows. Homogeneous (four-element) vectors use the fourth element as the division factor of the upper three elements. Despite the unusual division interpretation, H-matrices are multiplied and inverted in the same way as ordinary matrices.

Multiplication of two or more matrices into a single matrix is called "**concatenation**". When matrices are concatenated, the intermetidate states (between the input and the output) vanish. *Concatenating* H-matrices transforms entire <u>volume</u> relationships, whereas *multiplying* an H-vector times an H-matrix transforms a single <u>point</u> from the input volume to a corresponding point in the output volume.

If you can "**invert**" a matrix, you can force information flow "backwards" from "the outputs" through the inverse to "the inputs" ***<u>without any loss of information</u>*** …philosophically speaking – way too cool!

Appendix "A" provides an introduction to matrix multiplication and inversion.

So, what follows are details about how to gracefully navigate the successive volumetric transformations of multiple optical elements while *pure mathematics* does the legwork.

<u>Re-expressing Refractive Optics in Homogeneous matrix (4x4) form:</u>

Suppose an object point of interest is the 3-D point (X,Y,Z) and the corresponding image 3-D point is (R,S,T).

Equation#1 is a relationship between X and R.

Equation#2 relates Y -to- S, and also Z -to- T

[The ***common coordinate system*** shared by the *object volume* and the *image volume* has its origin at the center of the first lens with +X & +R "forward" along the optical axis in the direction of the object location, +Y & +S "right", and +Z & +T "down", forming a right-handed coordinate system. Drawing #1 illustrates the object volume.]

From equation#1: $1./X - 1./R = 1./F$

$R = -F*X \ / (X-F)$,or equivalently…

so: $R = F*X \ / (F-X)$,and Equation#3

from equation#2: Magnification = R/X and so

= F / (F-X) and note that

Magnification = R/X = S/Y = T/Z because the ratio of transverse displacements is proportional to the ratio of depths.

(Rays pass ~ unbent through the center of lenses, creating two similar triangles.)

Hence: $S = Y *$ magnification

$= Y * F / (F-X)$ and likewise Equation#4

$T = Z * F / (F-X)$ Equation#5

Equations 3,4, & 5 share <u>a common division factor</u> (F-X).

Homogenize the vectors as follows:

The object vector (X,Y,Z) becomes (x,y,z,w1); initially w1=1.0

The image vector (R,S,T) becomes (r,s,t,w2)

Then we seek an H-matrix “H”:

```
    |a,e,i,m|
H = |b,f,j,n| such that:
    |c,g,k,o|
    |d,h,l,p|

R=r/w2   | r  |       | x |
S=s/w2 & | s  | = H * | y |
T=t/w2   | t  |       | z |
         | w2 |       | w1|
```

Considering the multiplication of the 4^{th} row times the vector:

```
W2= d*x + h*y + l*z + p*w1
```

& remembering that the 4^{th} row does division, yields:

```
    d=-1., h=0., l=0., p=+F   which takes care of the (F-X)
                               term in equations 3,4,& 5.
t = c*x + q*y + k*z + o*w1   so for equation#5
    c=0., q=0., k=+F, o=0.
s = b*x + f*y + j*z + n*w1    & for equation#4
    b=0., f=+F, j=0., n=0.
r = a*x + e*y + i*z + m*w1    & for equation#3
    a=+F, e=0., i=0., m=0.
```

[Note: All the elements of an *invertible* H can be multiplied by a constant without altering the underlying transformation – because the 4^{th} row divides the upper three rows . For the final result - divide all the component by F...]

Hence:

```
    | 1.   , 0., 0., 0.|
H = | 0.   , 1., 0., 0.| = ”H_Refraction” henceforth
    | 0.   , 0., 1., 0.|             for the refractive case.
    |-1./F, 0., 0., 1.|       … A simple result!
```

Proof that H_{lens} is correct - by substitution:

```
r=     x    =  X*w1
s=     y    =  Y*w1
t=     z    =  Z*w1
w2 =-x/F+w1 =  w1-X*w1/F =w1*(F-X)/F
```

Since every element on the right side has a w1 term, and since the 4^{th} component divides the upper 3 components, the w1 term cancels, leaving:

```
r =  X    and hence: R=r/w2= F*X/(F-X)
s =  Y               S=s/w2= F*Y/(F-X)
t =  Z               T=t/w2= F*Z/(F-X)
w2= (F-X)/F             which are the desired results,
                and relate back to the original equations:

R= F*X/(F-X) → 1/R = +1./X -1./F
             → 1./X -1./R = 1./F     which is equation#1

S= F*Y/(F-X) → S/Y = F/(F-X) = R/X = magnification
                               which satisfies equation#2

T= F*Z/(F-X) → T/Z = F/(F-X) = R/X = magnification
                          which also satisfies equation#2
```

$H_{Refraction}$ is now defined.

I examined the behavior of reflective lenses (see Appendix "B") and have tentatively concluded that:

$$\mathbf{H_{Reflection}} = \begin{vmatrix} -1. & 0. & 0. & 0. \\ 0. & 1. & 0. & 0. \\ 0. & 0. & 1. & 0. \\ -1./F & 0. & 0. & 1. \end{vmatrix}$$

for the reflective case.

Treating mirrors as reflective lenses with infinite focal lengths yields:

$$\mathbf{H_{Mirror}} = \begin{vmatrix} -1. & 0. & 0. & 0. \\ 0. & 1. & 0. & 0. \\ 0. & 0. & 1. & 0. \\ 0. & 0. & 0. & 1. \end{vmatrix}$$

for the mirror case.

But the first mirror is also first rotated by +45. degrees:

```
              | cos(45.), 0.,  sin(45.), 0.|
HRotate     = |     0. , 1.,       0. , 0.|
              |-sin(45.), 0.,  cos(45.), 0.|
              |     0. , 0.,       0. , 1.|
```

and then translated by -1. focal length:

```
              | 1., 0., 0.,-1.*F|
HTranslate  = | 0., 1., 0., 0.  |
              | 0., 0., 1., 0.  |
              | 1., 0., 0., 1.  |
```

In a similar manner, the second mirror is rotated by -45. degrees and then translated by -1. focal length.

And finally the lens transform is yawed by 180. degrees for the second-use of the lens. (This is an experimental result at this point; the resulting transform matches the inverse of the first-use transform, which seems appropriate .)

Properly sequencing H-transforms requires attention to detail. In the two pages of numbers that follow in computing the quantitative result for each optical element:

The "optical bench" (i.e. global) coordinate frame is **de-translated** and then **de-rotated** into optical element coordinates (`6DOF: Lens<-Bench`), the elements optical effect occurs(`Optical Effect: Image<-Object`), and then the result is **re-rotated** and **re-translated** back into "optical bench" coordinates" (`6DOF: Bench<-Lens`). Overall yielding: `Lens Net: Image<-Object`.
The cumulative element concatenations yeild: `Bench Net: Image<-Object`.

Appendix "A" includes the H matrix for 6DOF'ing that accomplishes the "re-rotation" and the "re-translation" simultaneously. The *inverse* H performs the "de-translation" followed-by "de-rotation" sequence.

In the numerical example F=1.0, whereas my optical demonstrator F =~ .21 meters.

```
LensNumber,iType,FocalL,ApRad,  Xc,  Yc,  Zc, Roll,Pitch,  Yaw,iColor
    1       1     1.00    1.00 0.00 0.00 0.00  0.00  0.00  0.00   11
    Forward Transforms                  Reverse (i.e. Inverse) Transforms
6DOF: Bench<-Lens                       Lens<-Bench   First use of the lens
 1.000000 0.000000 0.000000    0.000   1.000000 0.000000 0.000000    0.000
 0.000000 1.000000 0.000000    0.000   0.000000 1.000000 0.000000    0.000
 0.000000 0.000000 1.000000    0.000   0.000000 0.000000 1.000000    0.000
 0.000000 0.000000 0.000000    1.000   0.000000 0.000000 0.000000    1.000
Optical Effect: Image<-Object     not= Object<-Image
 1.000000 0.000000 0.000000    0.000   1.000000 0.000000 0.000000    0.000
 0.000000 1.000000 0.000000    0.000   0.000000 1.000000 0.000000    0.000
 0.000000 0.000000 1.000000    0.000   0.000000 0.000000 1.000000    0.000
-1.000000 0.000000 0.000000    1.000   1.000000 0.000000 0.000000    1.000
Lens Net: Image<-Object                 Object<-Image
 1.000000 0.000000 0.000000    0.000   1.000000 0.000000 0.000000    0.000
 0.000000 1.000000 0.000000    0.000   0.000000 1.000000 0.000000    0.000
 0.000000 0.000000 1.000000    0.000   0.000000 0.000000 1.000000    0.000
-1.000000 0.000000 0.000000    1.000   1.000000 0.000000 0.000000    1.000
Bench Net: Image<-Object                Object<-Image
 1.000000 0.000000 0.000000    0.000   1.000000 0.000000 0.000000    0.000
 0.000000 1.000000 0.000000    0.000   0.000000 1.000000 0.000000    0.000
 0.000000 0.000000 1.000000    0.000   0.000000 0.000000 1.000000    0.000
-1.000000 0.000000 0.000000    1.000   1.000000 0.000000 0.000000    1.000
 2  0  0.00  1.00 -1.00  0.00  0.00  0.00 45.00  0.00  12     First Mirror
6DOF: Bench<-Lens                       Lens<-Bench
 0.707107 0.000000 0.707107   -1.000   0.707107 0.000000-0.707107    0.707
 0.000000 1.000000 0.000000    0.000   0.000000 1.000000 0.000000    0.000
-0.707107 0.000000 0.707107    0.000   0.707107 0.000000 0.707107    0.707
 0.000000 0.000000 0.000000    1.000   0.000000 0.000000 0.000000    1.000
Optical Effect: Image<-Object        = Object<-Image
-1.000000 0.000000 0.000000    0.000  -1.000000 0.000000 0.000000    0.000
 0.000000 1.000000 0.000000    0.000   0.000000 1.000000 0.000000    0.000
 0.000000 0.000000 1.000000    0.000   0.000000 0.000000 1.000000    0.000
 0.000000 0.000000 0.000000    1.000   0.000000 0.000000 0.000000    1.000
Lens Net: Image<-Object              = Object<-Image
 0.000000 0.000000 1.000000   -1.000   0.000000 0.000000 1.000000   -1.000
 0.000000 1.000000 0.000000    0.000   0.000000 1.000000 0.000000    0.000
 1.000000 0.000000 0.000000    1.000   1.000000 0.000000 0.000000    1.000
 0.000000 0.000000 0.000000    1.000   0.000000 0.000000 0.000000    1.000
Bench Net: Object<-Image                Object<-Image
 1.000000 0.000000 1.000000   -1.000   0.000000 0.000000 1.000000   -1.000
 0.000000 1.000000 0.000000    0.000   0.000000 1.000000 0.000000    0.000
 0.000000 0.000000 0.000000    1.000   1.000000 0.000000 0.000000    1.000
-1.000000 0.000000 0.000000    1.000   0.000000 0.000000 1.000000    0.000
 3  0  0.00  1.00 -1.00  0.00  0.00  0.00-45.00  0.00  13    Second Mirror
6DOF: Bench<-Lens                       Lens<-Bench
 0.707107 0.000000-0.707107   -1.000   0.707107 0.000000 0.707107    0.707
 0.000000 1.000000 0.000000    0.000   0.000000 1.000000 0.000000    0.000
 0.707107 0.000000 0.707107    0.000  -0.707107 0.000000 0.707107   -0.707
 0.000000 0.000000 0.000000    1.000   0.000000 0.000000 0.000000    1.000
Optical Effect: Image<-Object        = Object<-Image
-1.000000 0.000000 0.000000    0.000  -1.000000 0.000000 0.000000    0.000
 0.000000 1.000000 0.000000    0.000   0.000000 1.000000 0.000000    0.000
 0.000000 0.000000 1.000000    0.000   0.000000 0.000000 1.000000    0.000
 0.000000 0.000000 0.000000    1.000   0.000000 0.000000 0.000000    1.000
Lens Net: Image<-Object              = Object<-Image
 0.000000 0.000000-1.000000   -1.000   0.000000 0.000000-1.000000   -1.000
 0.000000 1.000000 0.000000    0.000   0.000000 1.000000 0.000000    0.000
-1.000000 0.000000 0.000000   -1.000  -1.000000 0.000000 0.000000   -1.000
 0.000000 0.000000 0.000000    1.000   0.000000 0.000000 0.000000    1.000
```

```
Bench Net: Object<-Image                Object<-Image
 1.000000 0.000000 0.000000   -2.000  -1.000000 0.000000 0.000000   -2.000
 0.000000 1.000000 0.000000    0.000   0.000000 1.000000 0.000000    0.000
 0.000000 0.000000-1.000000    0.000   0.000000 0.000000-1.000000    0.000
-1.000000 0.000000 0.000000    1.000  -1.000000 0.000000 0.000000   -1.000
 4  1  1.00  1.00  0.00  0.00  0.00  0.00180.00  0.00  11
6DOF: Bench<-Lens                       Lens<-Bench       Second use of Lens
-1.000000 0.000000 0.000000    0.000  -1.000000 0.000000 0.000000    0.000
 0.000000 1.000000 0.000000    0.000   0.000000 1.000000 0.000000    0.000
 0.000000 0.000000-1.000000    0.000   0.000000 0.000000-1.000000    0.000
 0.000000 0.000000 0.000000    1.000   0.000000 0.000000 0.000000    1.000
Optical Effect: Image<-Object      not= Object<-Image
 1.000000 0.000000 0.000000    0.000   1.000000 0.000000 0.000000    0.000
 0.000000 1.000000 0.000000    0.000   0.000000 1.000000 0.000000    0.000
 0.000000 0.000000 1.000000    0.000   0.000000 0.000000 1.000000    0.000
-1.000000 0.000000 0.000000    1.000   1.000000 0.000000 0.000000    1.000
Lens Net: Image<-Object            not= Object<-Image
 1.000000 0.000000 0.000000    0.000   1.000000 0.000000 0.000000    0.000
 0.000000 1.000000 0.000000    0.000   0.000000 1.000000 0.000000    0.000
 0.000000 0.000000 1.000000    0.000   0.000000 0.000000 1.000000    0.000
 1.000000 0.000000 0.000000    1.000  -1.000000 0.000000 0.000000    1.000
******* The Total Transform *******   (This volume map is its own inverse.)
Bench Net: Image<-Object          =  Object<-Image
 1.000000 0.000000 0.000000   -2.000   1.000000 0.000000 0.000000   -2.000
 0.000000 1.000000 0.000000    0.000   0.000000 1.000000 0.000000    0.000
 0.000000 0.000000-1.000000    0.000   0.000000 0.000000-1.000000    0.000
 0.000000 0.000000 0.000000   -1.000   0.000000 0.000000 0.000000   -1.000
```

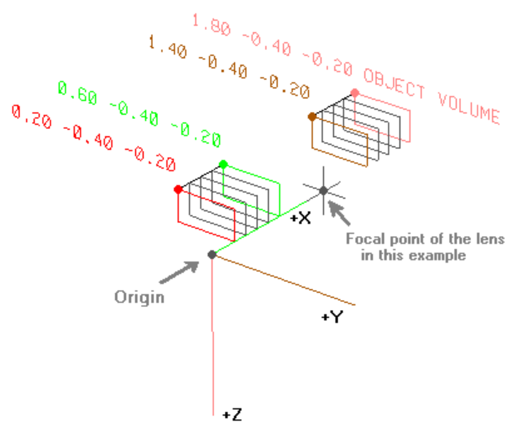
So – let's visualize these numbers.

The "Object" Volume:

The volumetric "object" is a stack of rectangles with a gap in the middle to avoid the area near the focal point of the first lens. One corner of each "object" rectangle is on the lens optical axis.

This 'object" stack will be redrawn after each successive optical element transforms the volume. Overall, the stack halves will be turned inside out, reversed in depth, and the far objects will be turned upside-down by the first lens.

You "observe" the object's image from the non-object side of a *refractive (transparent))* lens, but from the object side of a *reflecting (mirrored)* lens.



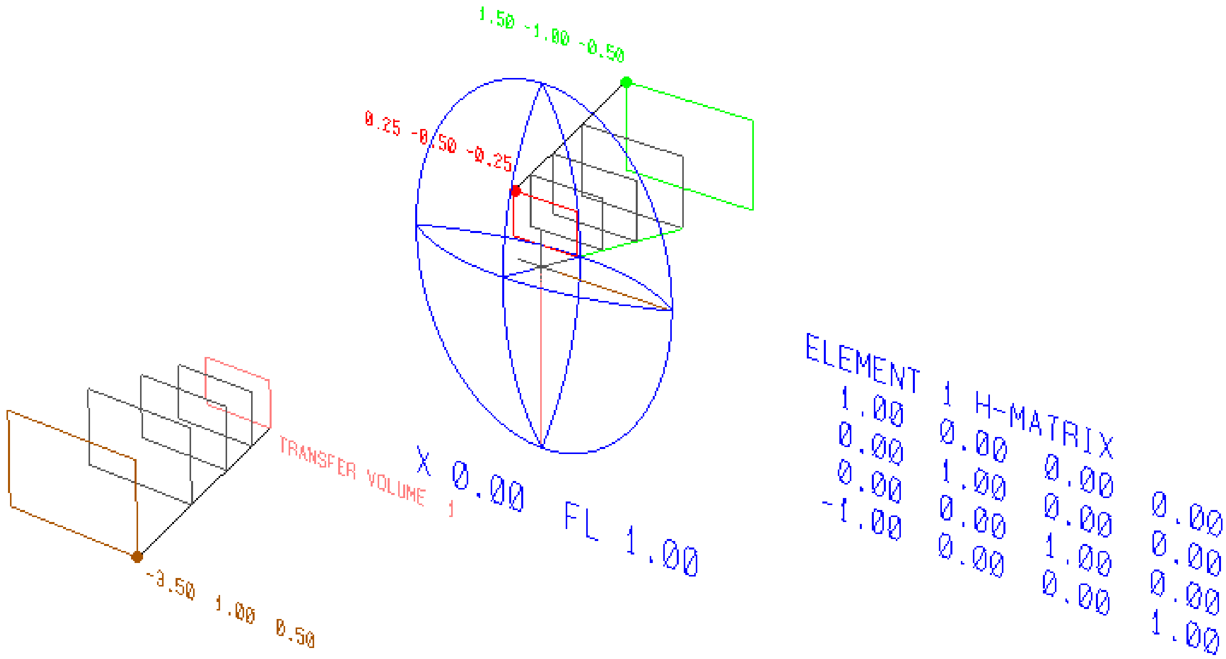
Drawing #1

The first lens would map *a rectangle at the depth of the focal point* to two infinitely large rectangles infinitely far away in opposite directions. Far from useless - such "infinity collimated displays" have been used in the "Heads-Up Displays" – HUD's – of fighter aircraft for many decades; telescopic sights place crosshairs near an optical focal point.

<u>The First use of the Lens</u>:

Drawing #2: **"FL" = F = Focal Length =1.0**

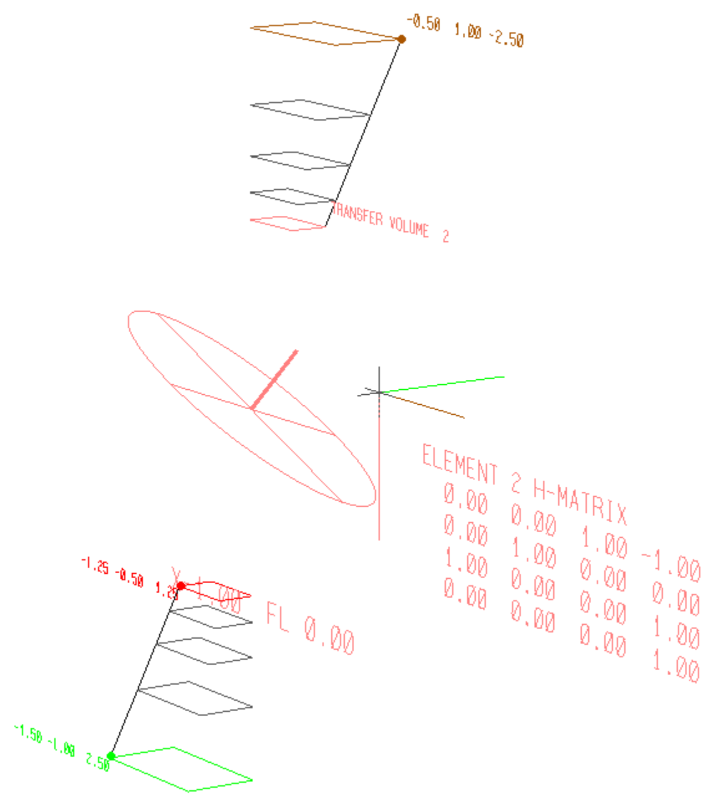
The lens is drawn with a radius = F



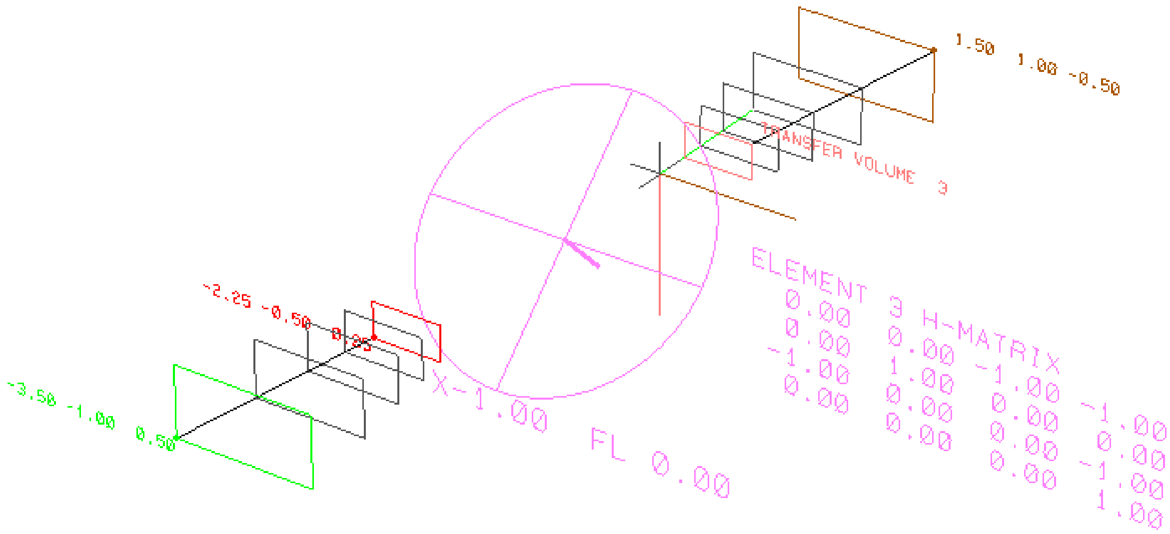
<u>The First Mirror:</u>

Drawing #3:

For the mirrors:
FL = F = infinity, not 0.00

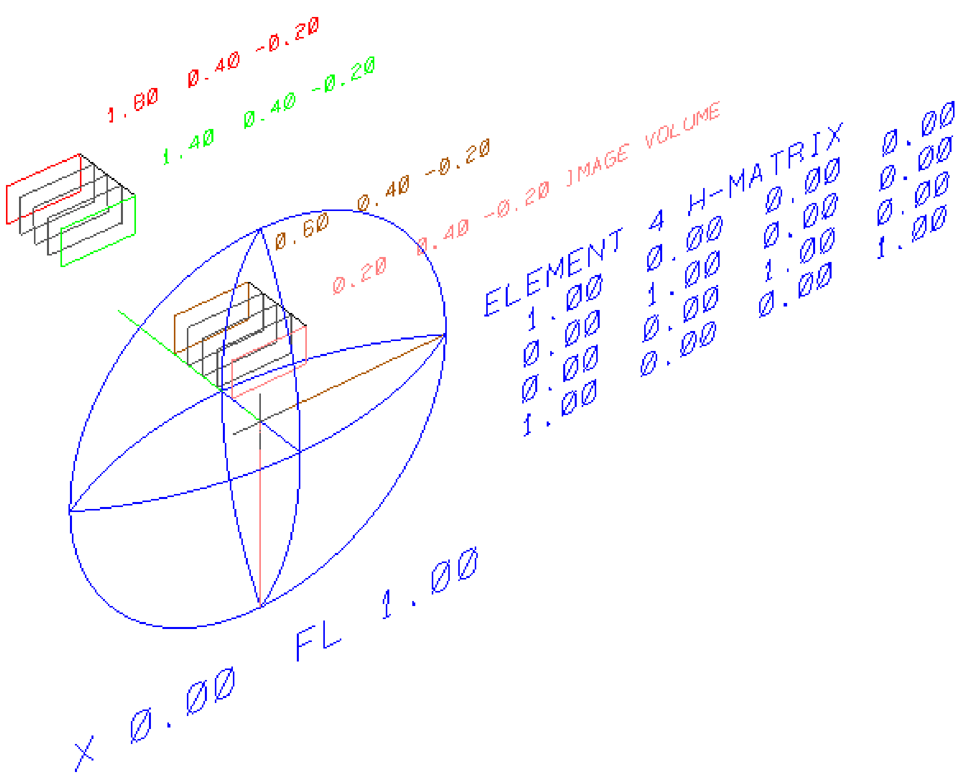


The Second Mirror: **Drawing #4:**

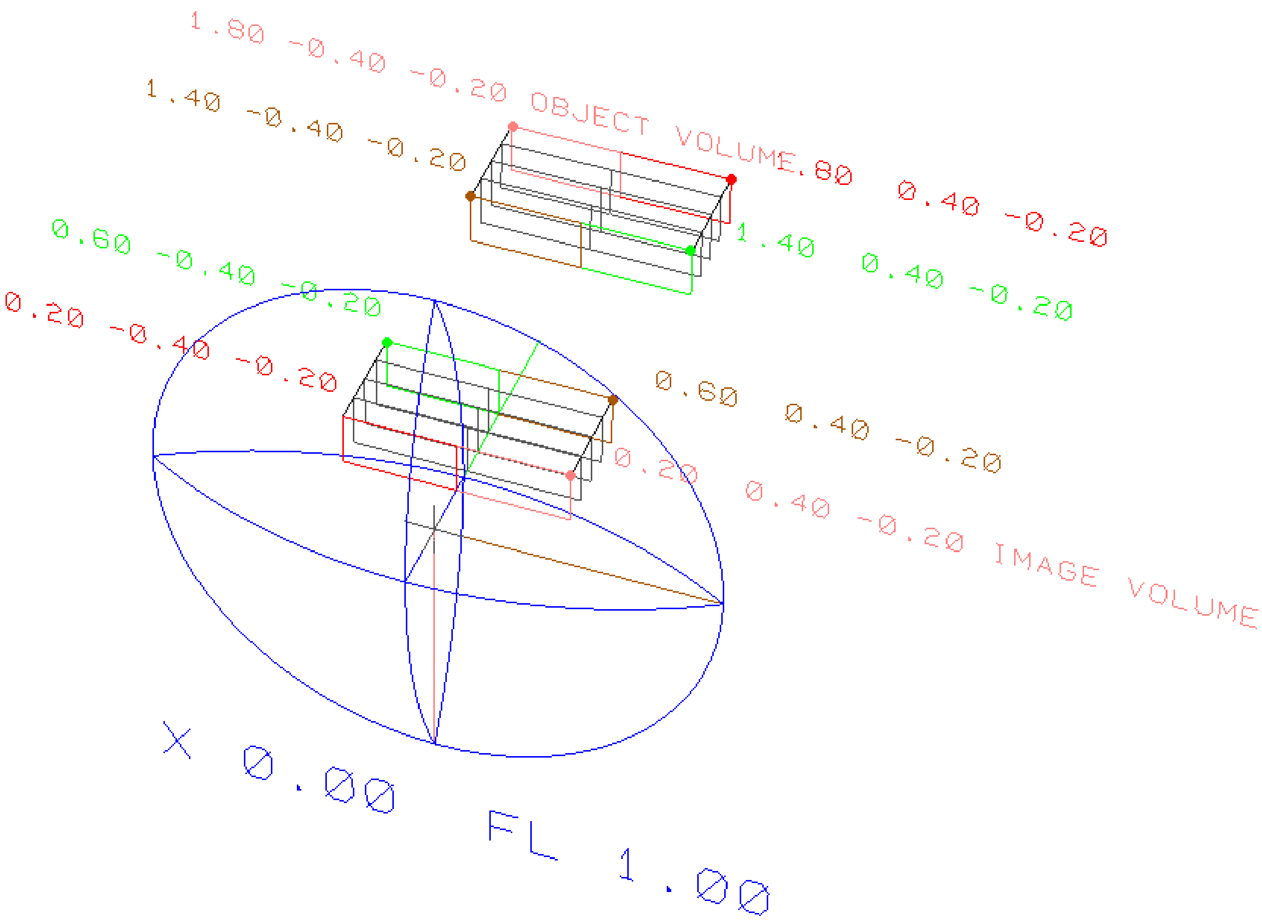


The second use of the lens – yielding the “Image” volume:

Drawing #5:



The “Object” & “Image” together: **Drawing #6:**



Bench Net: Image<-Object

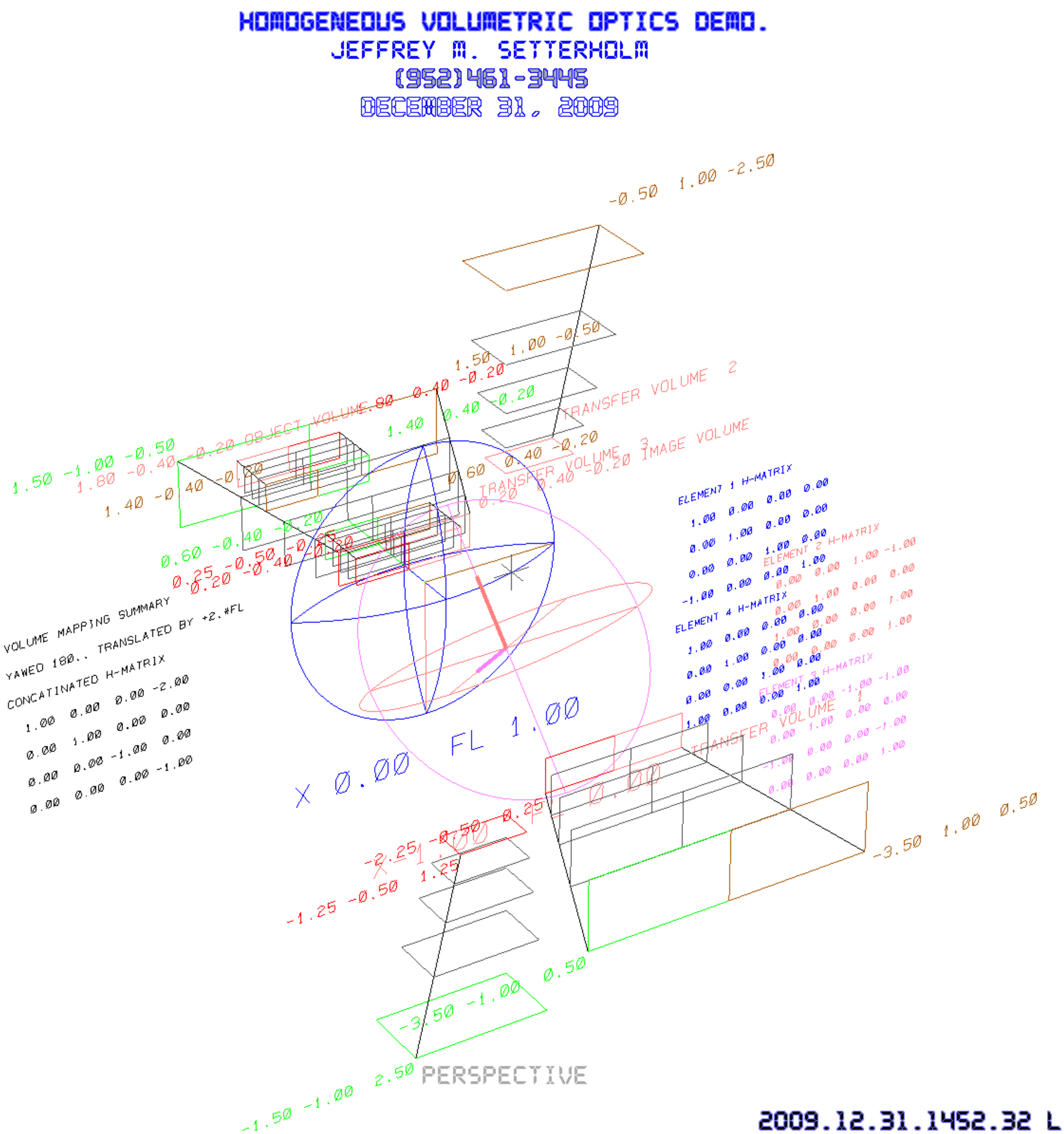
$$\begin{vmatrix} r \\ s \\ t \\ w2 \end{vmatrix} = \begin{vmatrix} -1. & 0. & 0. & 2. \\ 0. & -1. & 0. & 0. \\ 0. & 0. & 1. & 0. \\ 0. & 0. & 0. & 1. \end{vmatrix} * \begin{vmatrix} x \\ y \\ z \\ w1 \end{vmatrix}$$

An “H” overall-effects interpretation:

The object volume is yawed 180. degrees around the origin (at the lens center) and moved +2.0*F in the +R (= +X) direction to form the image volume .

(The “angular” interpretation, not inherent in H matrices, is with respect to “Standard Flight Simulation Coordinates”.)

<u>All the elements together:</u> **Drawing #7:**



Appendix "A" Matrix multiplication, inversion, & 6DOF'ing

Symbolic multiplication of a 2x2 matrix:

```
| a*e+c*f, a*g+c*h | = | a, c | * | e, g |
| b*e+d*f, b*g+d*h |   | b, d |   | f, h |
```

In ~Fortran, multiplying A*B in the general (N,N) case where
C(N,N)=A(N,N)*B(N,N), the nested do-loops are:

```
real*8   ::A(N,N),B(N,N),C(N,N)
integer*4::i,j,k
do i=1,N
  do j=1,N
    C(i,j)=0.d0
    do k=1,N
      C(i,j)=C(i,j)+A(i,k)*B(k,j)
    enddo !k
  enddo !j
enddo !i
```

Symbolic inversion of a "well conditioned" 2x2 matrix :

```
Given: | a, c |
       | b, d |
Append an identity matrix:
       | a, c | 1., 0. | <- row one      and row-reduce the left half
       | b, d | 0., 1. | <- row two         to an identity matrix.
```

[If the absolute value of a
is less than the absolute value of b,
then swap rows one and two.
The computer code below manages the nuances.
The matrix would be "ill conditioned" if, for example, both a and b were zero.]

```
Divide row one by a:
       | 1., c/a | 1./a, 0. |
       | b , d   | 0.  , 1. |
Subtract b*row 1 from row 2.
       | 1.,     c/a | 1./a, 0. |
       | 0., d-b*c/a | -b/a, 1. |
Divide row 2 by d-b*c/a.
       | 1., c/a |  1./a       , 0.          |
       | 0.,  1. | -b/(a*d-b*c), a/(a*d-b*c) |
Subtract c/a*row 2 from row 1.
       | 1., 0. | d/( a*d-b*c), -c/( a*d-b*c) |
       | 0., 1. |-b/( a*d-b*c),  a/( a*d-b*c) |
                |       ^ The inverse  ^      |
```

In the process the right half became the inverse!

```
 In ~Fortran, the do-loops for inverting A(N,N) are:

real*8   ::A(N,N),B(N,2*N),Temp,AbsValMax  use 64-bit reals - minimum
real*8   ::Ainverse(N,N)
integer*4:: i,iRow,j,jColumn,k,iValMax

B=0.d0
B(1:4,1:4)=A(1:4,1:4)
do i=1,4
  B(i,i+N)=1.d0
Enddo !i

do iRow =1,N       …Proceeding down the diagonal of the matrix:
  jColumn=iRow     …Find the largest absolute B(,) in column jColumn
  iMax=0
  AbsValMax=0.d0
  do i= iRow,N
    if(abs(B(i,jColumn)).gt.abs(AbsValMax)) then
      AbsValMax= B(I,jColumn)
      iValMax=iRow
    endif
  enddo
  if(iValMax.eq.0) stop “Matrix A is ill-conditioned.”
               …because all the remaining jColumn values were zero’s.
                            [ Being “ill-conditioned” and
                             “having redundant variables”
                                 are related concepts.   ]
  do j=1,2*N        …swap row iValMax with row iRow
    Temp=B(iValMax,j)
         B(iValMax,j)=B(iRow,j)
         B(iRow   ,j)=Temp)/AbsValMax    …while rescaling iRow
  enddo !j

  do i=1,N          …reduce the other jColumn values to zero.
    if(i.eq.iRow) cycle   (=skip this trip through the loop)
    do j=2*N,jColumn,-1
      B(i,j)= B(i,j)-B(i,jColumn)*B(iRow,j)
    enddo !j
  enddo !i
enddo !iRow

do i=1,N
  do j=1,N
    Ainverse(i,j)=B(i,j+N)
  enddo !i
enddo !j
```

To test that your inverter computed the correct answer:

Ainverse*A = an identity matrix   (1.’s on the diagonal, 0.’s elsewhere)

which makes sense because transforming “out”, & back “in”, should leave you where you started.

H describing a rigid body's Six Degree-Of-Freedom (6DOF) position & attitude:

```
Within a "standard flight simulation coordinate system":
  Let the position = (x,y,z), as in drawing #1 and
  let the attitude = (roll,pitch,yaw)  where: +yaw   takes +x toward +y
                                               +roll  takes +y toward +z
   sr=  sine(roll); sp=  sine(pitch); sy=  sine(yaw)  +pitch takes +z toward +x
   cr=cosine(roll); cp=cosine(pitch); cy=cosine(yaw)
                        with gimbal (concatenation) sequence: yaw <-Pitch <-Roll
   H(1,1)= cy*cp; H(1,2)=-sy*cr+cy*sp*sr; H(1,3)= sy*sr+cy*sp*cr;  H(1,4)= x
   H(2,1)= sy*cp; H(2,2)= cy*cr+sy*sp*sr; H(2,3)=-cy*sr+sy*sp*cr;  H(2,4)= y
   H(3,1)=   -sp; H(3,2)= cp*sr;          H(3,3)= cp*cr;           H(3,4)= z
   H(4,1)=    0.; H(4,2)=    0.;          H(4,3)=    0.;           H(4,1)= 1.
```

Numerical example:

Position=(x ,   y , z ,w) =(1. ,2., 3.,1.) e.g. in meters

Attitude=(roll,pitch,yaw,w) =(10.,20.,30.,1.) degrees - become direction cosines

H_{6DOF} = …which are unitless and independent of "angle" definition!

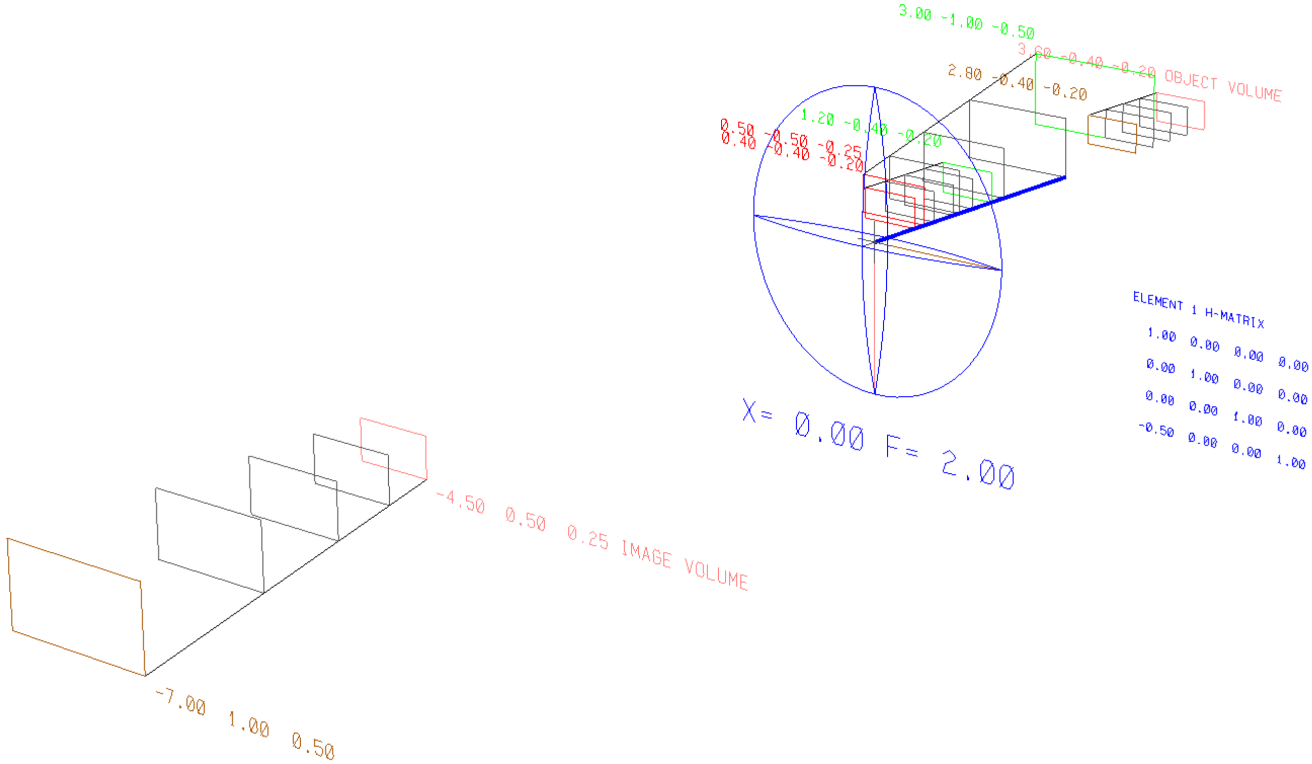
0.813797681	-0.440969611	0.378522306	1.000000
0.469846310	0.882564119	0.018028311	2.000000
-0.342020143	0.163175911	0.925416578	3.000000
0.000000000	0.000000000	0.000000000	1.000000

--- End of Appendix "A" ---

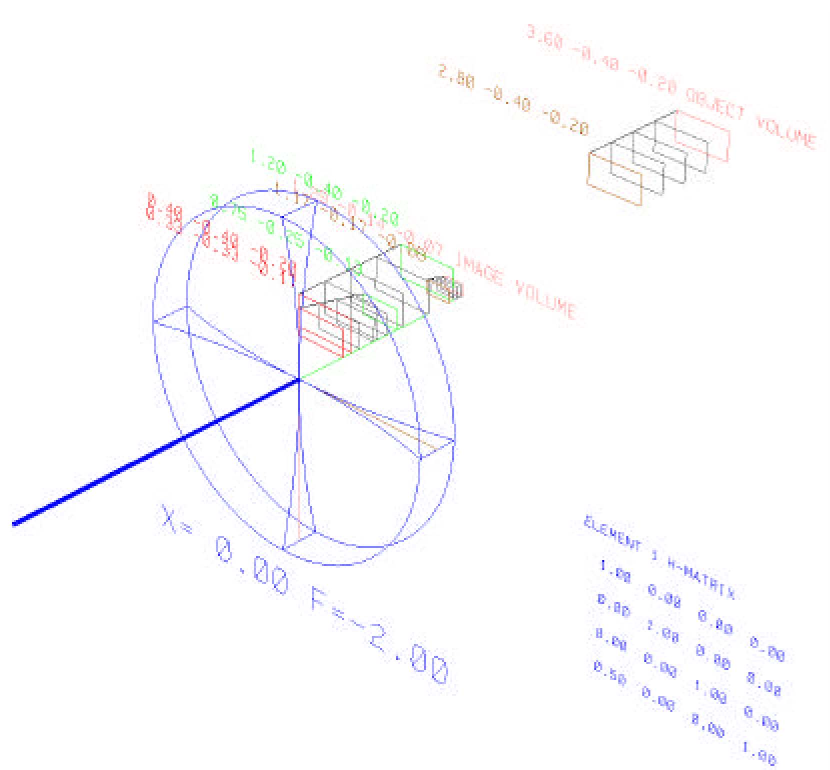
Appendix “B” Further visualizations of H-Optics volumetrics.

```
                      | 1.  , 0., 0., 0.|
   HRefraction      = | 0.  , 1., 0., 0.|     for refractive lenses:
                      | 0.  , 0., 1., 0.|
                      |-1./F, 0., 0., 1.|
```

F=+2. Convergent:

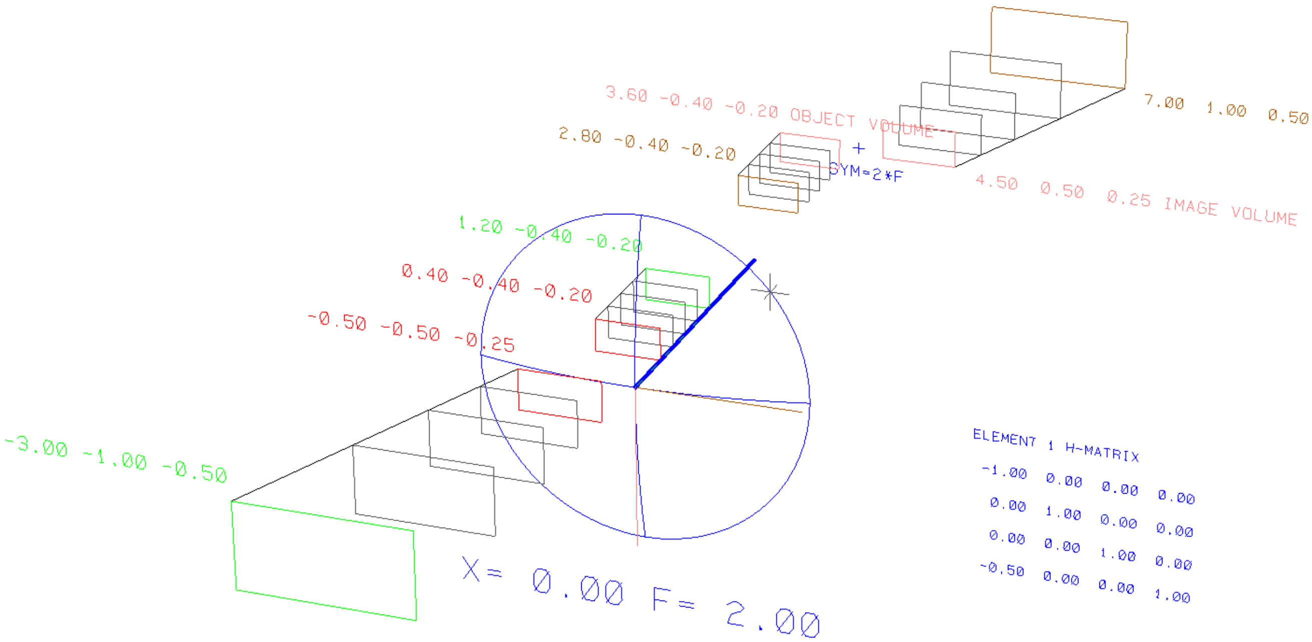


F=-2. Divergent:

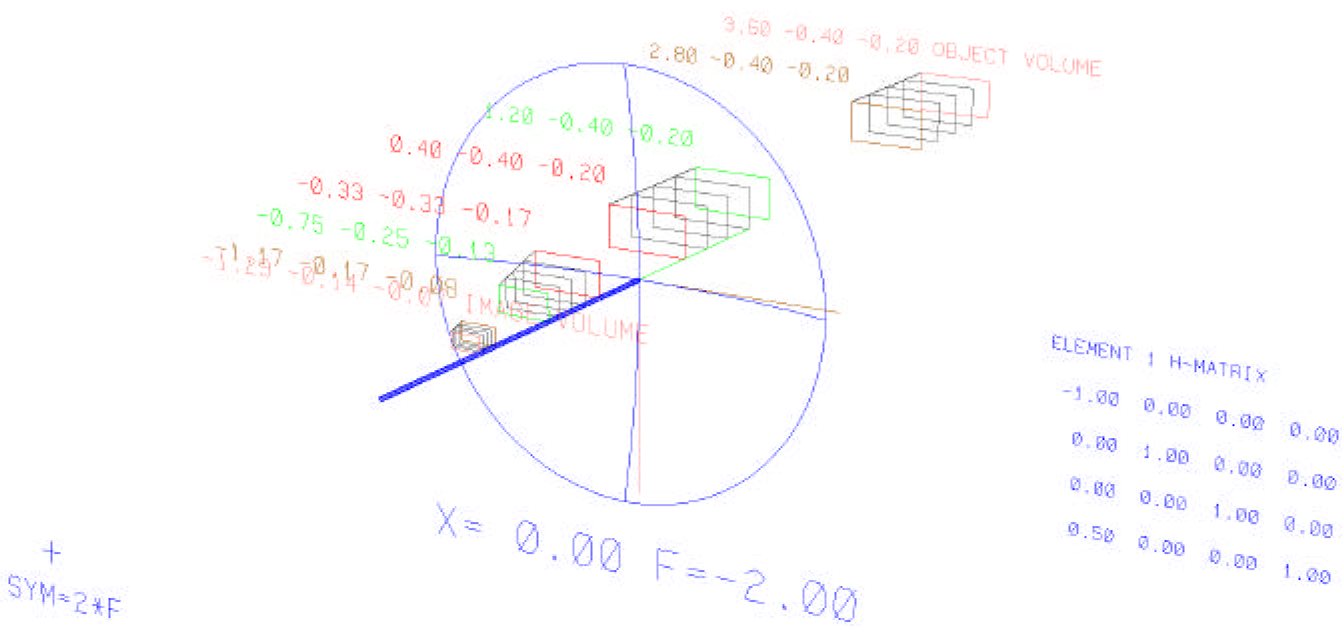


```
                  | 1.  , 0., 0., 0.|
                  |-1.  , 0., 0., 0.|
HReflection  =    | 0.  , 1., 0., 0.|      for reflective lenses:
                  | 0.  , 0., 1., 0.|
                  |-1./F, 0., 0., 1.|       (Note the symmetry points at 2.*F)
```

F=+2. Convergent:



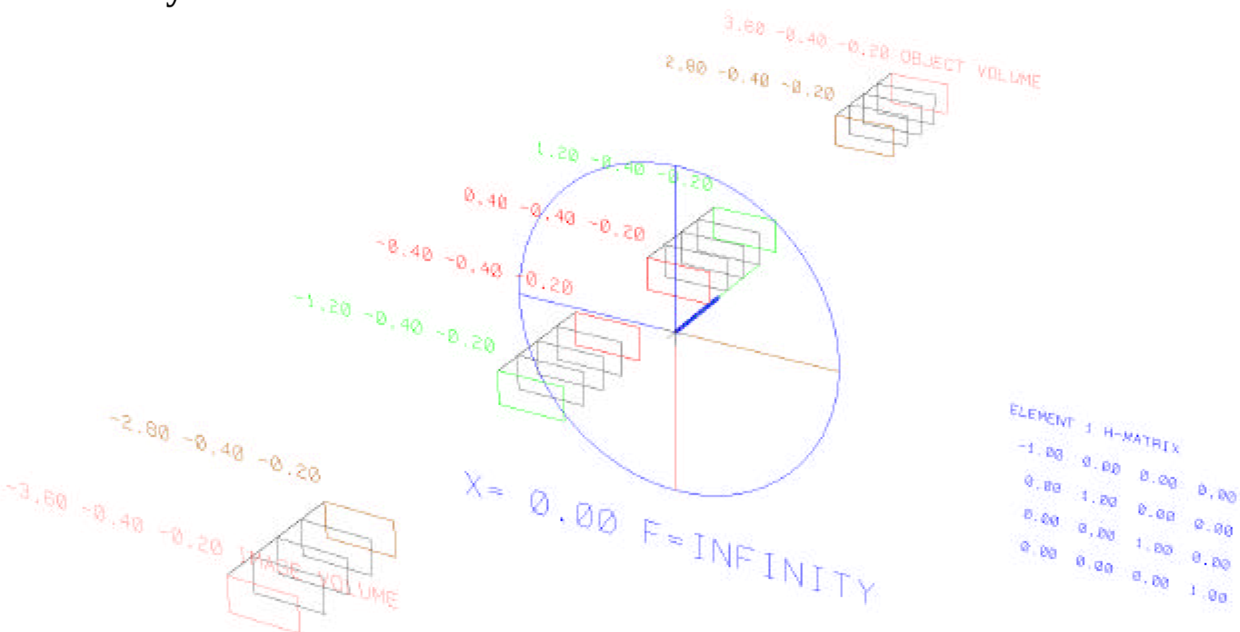
F=-2. Divergent:



$$H_{Mirror} = \begin{vmatrix} -1. , & 0., & 0., & 0. \\ 0. , & 1., & 0., & 0. \\ 0. , & 0., & 1., & 0. \\ 0. , & 0., & 0., & 1. \end{vmatrix}$$

for mirrors:

F = Infinity



--- End of Appendix "B" ---

Disclaimers:

1. This is my first attempt at using homogeneous transforms in optics.

2. 6DOF spatial analytics are not trivially easy to "get right"
   (properly define & sequence), particularly the first time,
   whether or not homogeneous transforms are the tools employed. &

3. Although I mean well, I sometimes make mistakes. &
   In fact, the price of pursuing cutting-edge R&D is being wrong almost all the time; physical reality (~"Mother Nature") acts as an impartial but relentless judge.

4. **This write-up is positively and certainly not fool-proof:**

Disclaimer #5:

THE ANALYTICS, DATA, IMAGES, AND DOCUMENTATION HERE ARE PROVIDED "AS IS" WITHOUT GUARANTEES OR WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, FITNESS FOR ANY PARTICULAR PURPOSE.