# Ray Tracing

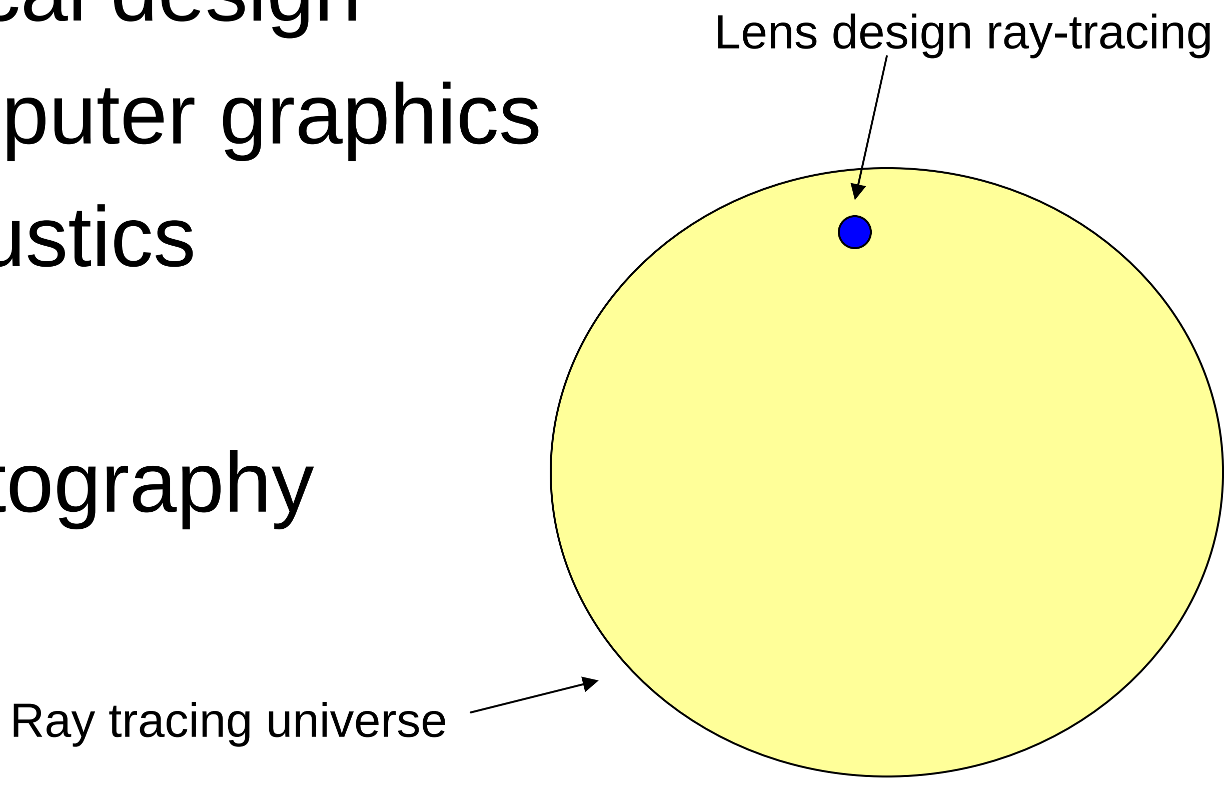
Lens Design OPTI 517

Prof. Jose Sasian

College of Optical Sciences
THE UNIVERSITY OF ARIZONA®

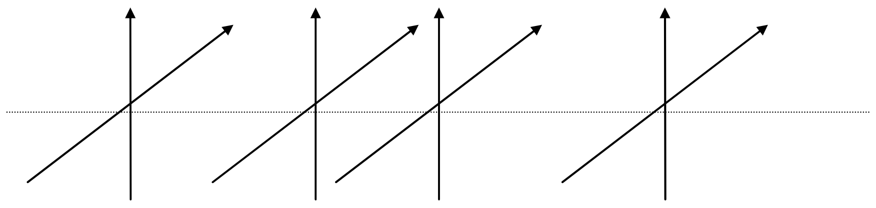
# Use of rays

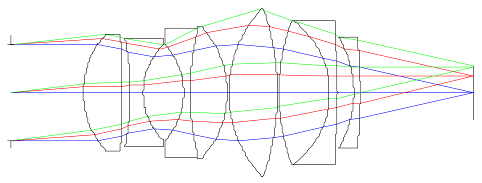
- In optical design
- In computer graphics
- In acoustics
- In art
- In photography

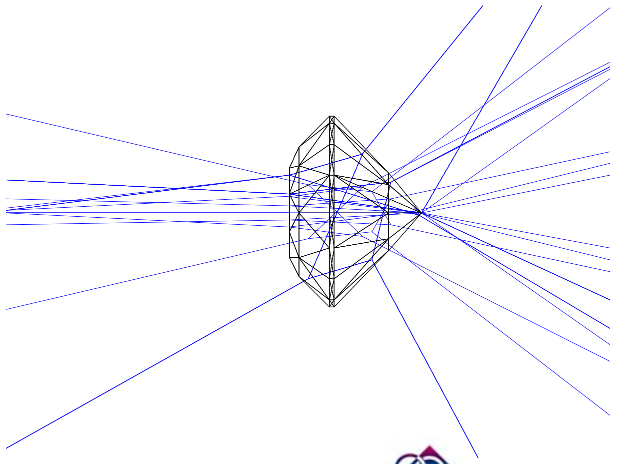


# Ray tracing

- It is important to have an understanding about how an optical design program works.
- Most calculations are done by real ray tracing to within the computer precision.
- XYZ coordinates, ray fans, wave fans, spot diagrams, apertures, wavefront deformation, optimization, field curves, etc.
- Sequential ray tracing vs. non-sequential ray tracing







Prof. Jose Sasian

College of Optical Sciences
THE UNIVERSITY OF ARIZONA®

## Ray tracing Algorithm

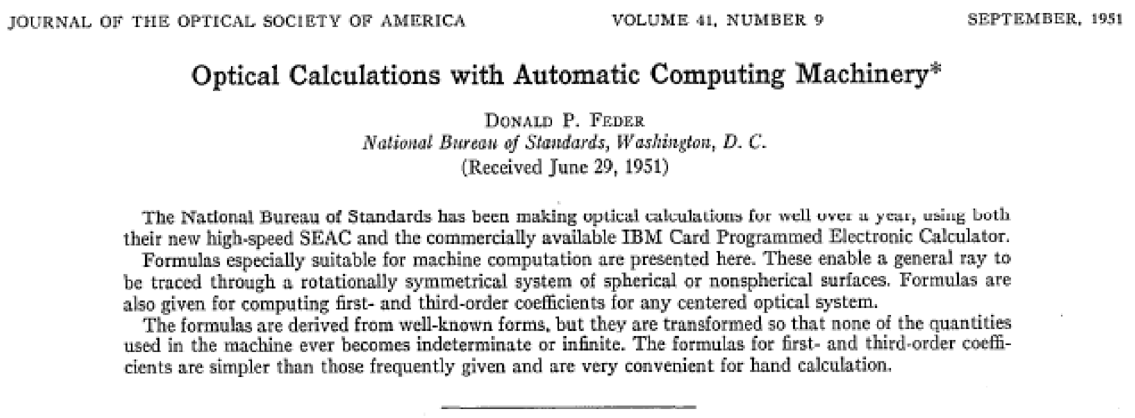
- Each coordinate system is local to a surface and it is referenced with respect to the previous surface.
- We have local XYZ coordinates for a ray intersection.
- A surface is locally defined by a function $F(x,y,z) = f(x,y) - z= 0$
- Local vs. Global coordinate systems
- UDS and other surface types
- Standard surface
- Close form equation for ray intersection for conics
- Iterative approximation for aspherics and other surfaces

## A bit of historical perspective

- Graphical ray tracing
- Paraxial ray tracing
- Logarithmical or Trigonometrical ray tracing
- Meridional rays, (L,U) Method, (Q,U) Method
- Skew rays
- Hundreds of papers on the subject of ray-tracing (efficiency and precision)

References
R. Kingslake’s Chapter 2
R. Shannon’s Chapter 2.1.2
W. Welford, Aberrations of optical systems, “finite ray tracing.”

# Some early papers

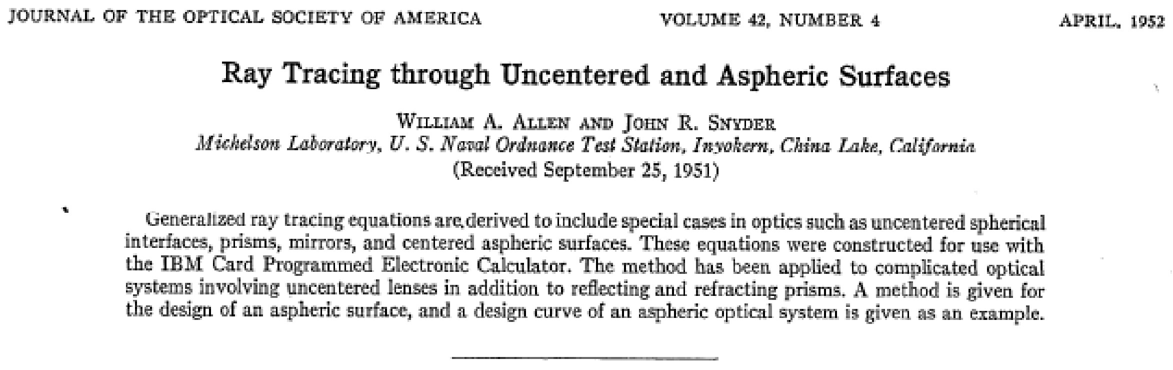


JOURNAL OF THE OPTICAL SOCIETY OF AMERICA VOLUME 41, NUMBER 9 SEPTEMBER, 1951

**Optical Calculations with Automatic Computing Machinery***

DONALD P. FEDER
*National Bureau of Standards, Washington, D. C.*
(Received June 29, 1951)

The National Bureau of Standards has been making optical calculations for well over a year, using both their new high-speed SEAC and the commercially available IBM Card Programmed Electronic Calculator.
Formulas especially suitable for machine computation are presented here. These enable a general ray to be traced through a rotationally symmetrical system of spherical or nonspherical surfaces. Formulas are also given for computing first- and third-order coefficients for any centered optical system.
The formulas are derived from well-known forms, but they are transformed so that none of the quantities used in the machine ever becomes indeterminate or infinite. The formulas for first- and third-order coefficients are simpler than those frequently given and are very convenient for hand calculation.



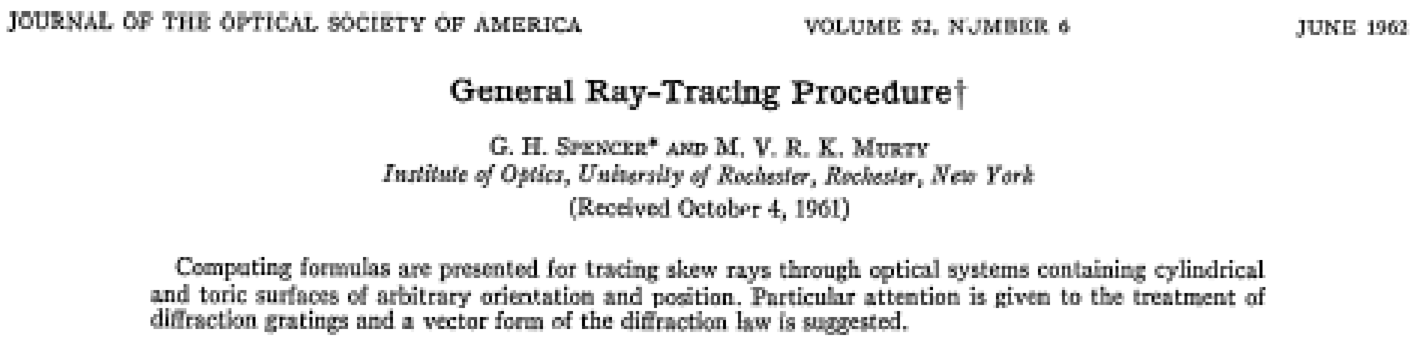
JOURNAL OF THE OPTICAL SOCIETY OF AMERICA VOLUME 42, NUMBER 4 APRIL, 1952

**Ray Tracing through Uncentered and Aspheric Surfaces**

WILLIAM A. ALLEN AND JOHN R. SNYDER
*Michelson Laboratory, U. S. Naval Ordnance Test Station, Inyokern, China Lake, California*
(Received September 25, 1951)

Generalized ray tracing equations are derived to include special cases in optics such as uncentered spherical interfaces, prisms, mirrors, and centered aspheric surfaces. These equations were constructed for use with the IBM Card Programmed Electronic Calculator. The method has been applied to complicated optical systems involving uncentered lenses in addition to reflecting and refracting prisms. A method is given for the design of an aspheric surface, and a design curve of an aspheric optical system is given as an example.

# Some early papers



JOURNAL OF THE OPTICAL SOCIETY OF AMERICA VOLUME 52, NUMBER 6 JUNE 1962

**General Ray-Tracing Procedure**†

G. H. SPENCER* AND M. V. R. K. MURTY
*Institute of Optics, University of Rochester, Rochester, New York*

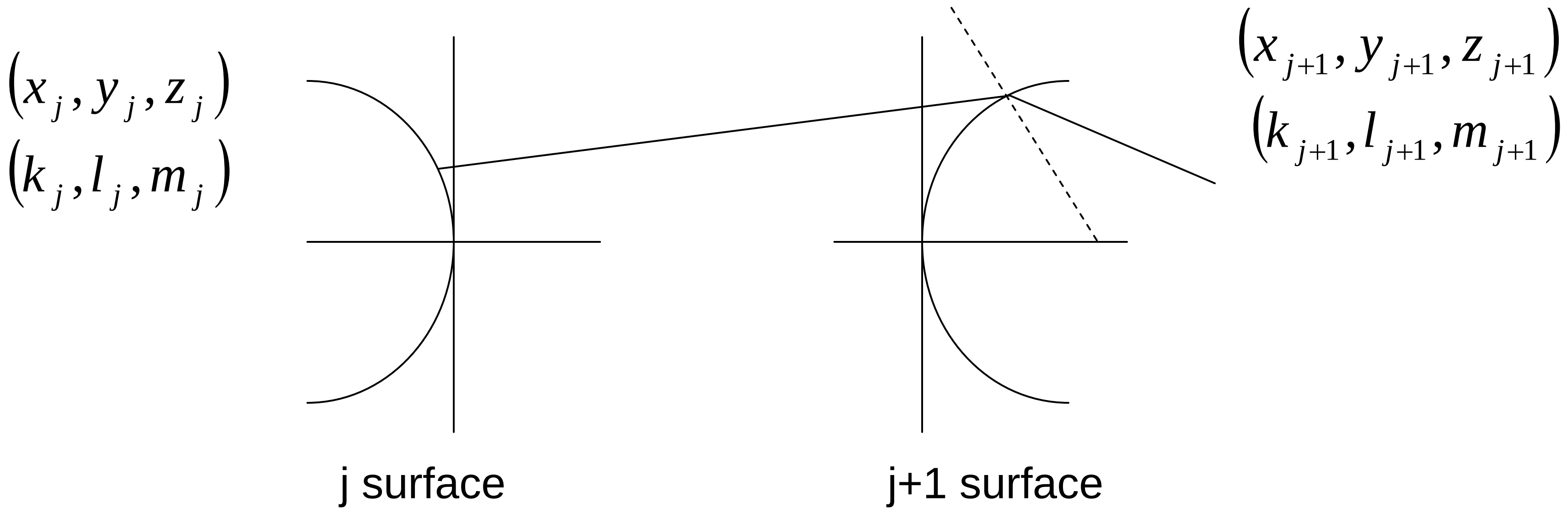
(Received October 4, 1961)

Computing formulas are presented for tracing skew rays through optical systems containing cylindrical and toric surfaces of arbitrary orientation and position. Particular attention is given to the treatment of diffraction gratings and a vector form of the diffraction law is suggested.

Prof. Jose Sasian

College of Optical Sciences
THE UNIVERSITY OF ARIZONA®

# Ray tracing algorithm



Given the $X_j$, $Y_j$, $Z_j$ coordinates of a ray at the (j) surface of an optical system with the optical direction cosines $K_j$, $L_j$, $M_j$ of the ray in the space following that surface, we wish to find the coordinates of the ray at the j+1 th surface and the optical direction cosines of the ray after refraction, reflection, or diffraction at that surface.

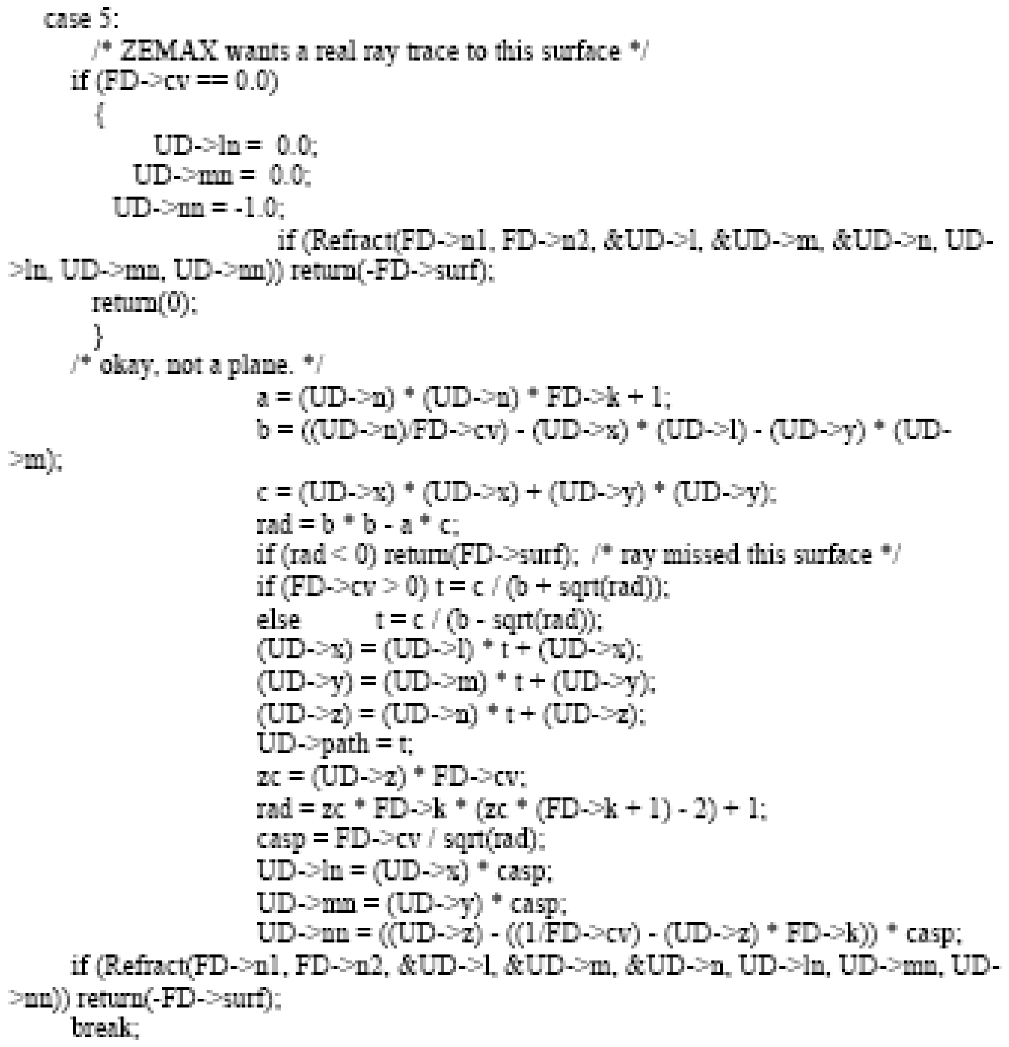
# Single ray trace

1) Transfer to next coordinate system; account for displacements and rotations.
2) Conic or aspheric intersection
3) Refraction, reflection, diffraction
4) Return new coordinates, direction cosines, and optical path.
5) Next surface

## Multiple ray trace (ray fan)

1) Define field point
2) Generate loop that defines fan of rays through range of direction cosines.
3)Trace chief ray first, surface after surface through image plane
4) Save coordinates
5) Trace other rays, subtract chief ray coordinates, plot.

# Zemax DLL: code efficiency

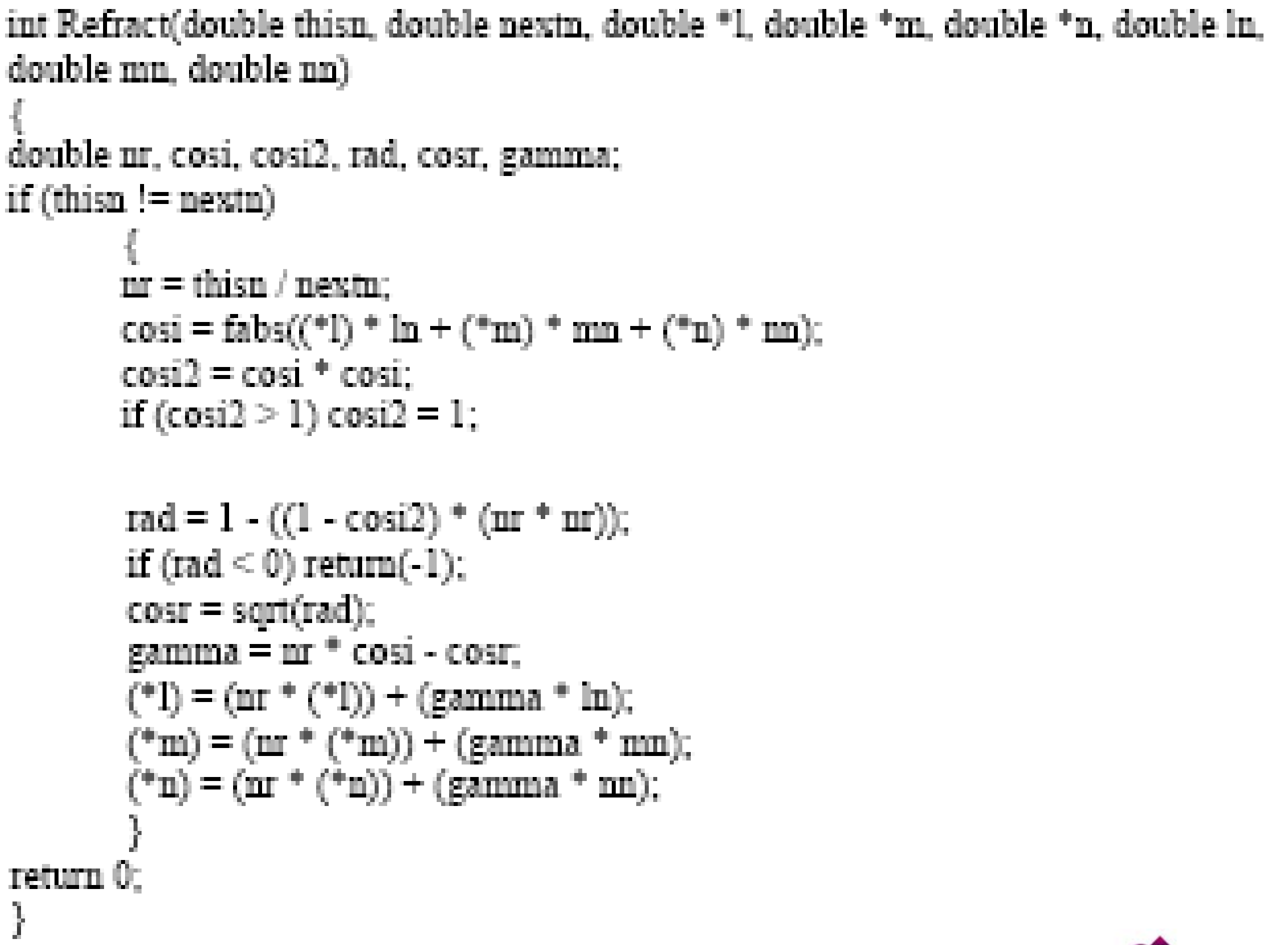


```
      case 5:
            /* ZEMAX wants a real ray trace to this surface */
         if (FD->cv == 0.0)
            {
                 UD->ln =  0.0;
               UD->mn =  0.0;
             UD->nn = -1.0;
                              if (Refract(FD->n1, FD->n2, &UD->l, &UD->m, &UD->n, UD-
>ln, UD->mn, UD->nn)) return(-FD->surf);
            return(0);
            }
         /* okay, not a plane. */
                            a = (UD->n) * (UD->n) * FD->k + 1;
                            b = ((UD->n)/FD->cv) - (UD->x) * (UD->l) - (UD->y) * (UD-
>m);
                            c = (UD->x) * (UD->x) + (UD->y) * (UD->y);
                            rad = b * b - a * c;
                            if (rad < 0) return(FD->surf);  /* ray missed this surface */
                            if (FD->cv > 0) t = c / (b + sqrt(rad));
                            else          t = c / (b - sqrt(rad));
                            (UD->x) = (UD->l) * t + (UD->x);
                            (UD->y) = (UD->m) * t + (UD->y);
                            (UD->z) = (UD->n) * t + (UD->z);
                            UD->path = t;
                            zc = (UD->z) * FD->cv;
                            rad = zc * FD->k * (zc * (FD->k + 1) - 2) + 1;
                            casp = FD->cv / sqrt(rad);
                            UD->ln = (UD->x) * casp;
                            UD->mn = (UD->y) * casp;
                            UD->nn = ((UD->z) - ((1/FD->cv) - (UD->z) * FD->k)) * casp;
         if (Refract(FD->n1, FD->n2, &UD->l, &UD->m, &UD->n, UD->ln, UD->mn, UD-
>nn)) return(-FD->surf);
         break;
```

Prof. Jose Sasian

College of Optical Sciences
THE UNIVERSITY OF ARIZONA®

# Zemax DLL : code efficiency



```
int Refract(double thisn, double nextn, double *l, double *m, double *n, double ln,
double mn, double nn)
{
double nr, cosi, cosi2, rad, cosr, gamma;
if (thisn != nextn)
          {
          nr = thisn / nextn;
          cosi = fabs((*l) * ln + (*m) * mn + (*n) * nn);
          cosi2 = cosi * cosi;
          if (cosi2 > 1) cosi2 = 1;

          rad = 1 - ((1 - cosi2) * (nr * nr));
          if (rad < 0) return(-1);
          cosr = sqrt(rad);
          gamma = nr * cosi - cosr;
          (*l) = (nr * (*l)) + (gamma * ln);
          (*m) = (nr * (*m)) + (gamma * mn);
          (*n) = (nr * (*n)) + (gamma * nn);
          }
return 0;
}
```

# Program output

```
Title: P RUDOLPH TESSAR USP#721240
Date : TUE SEP 23 2003
Units      : Millimeters
Wavelength  : 0.587562 microns
Coordinates : Local
Direction cosines are after refraction or reflection from the surface or object.
Normalized X Field Coord (Hx) :     0.000000
Normalized Y Field Coord (Hy) :     0.000000
Normalized X Pupil Coord (Px) :     0.000000
Normalized Y Pupil Coord (Py) :     1.000000
Real Ray Trace Data:
Surf       X-coord       Y-coord       Z-coord   X-cosine   Y-cosine   Z-cosine   X-normal   Y-normal   Z-normal   Angle in    Path length
OBJ        Infinity      Infinity      Infinity  0.0000000  0.0000000  1.0000000          -          -          -          -            -
  1   0.000000E+000  3.934634E+001  0.000000E+000  0.0000000  0.0000000  1.0000000  0.0000000  0.0000000 -1.0000000    0.00000
0.000000E+000
  2   0.000000E+000  3.934634E+001  1.655511E+001  0.0000000 -0.2905639  0.9568556  0.0000000  0.7149388 -0.6991870   45.63818
6.155511E+001
  3   0.000000E+000  3.773702E+001 -2.145273E+000  0.0000000 -0.4950848  0.8688447  0.0000000 -0.1133297 -0.9935574   23.39901
5.538579E+000
  4   0.000000E+000  3.710166E+001 -1.103024E+001  0.0000000 -0.0559880  0.9984314  0.0000000 -0.5463095 -0.8375834   62.78954
1.283350E+000
  5   0.000000E+000  3.637227E+001 -2.023041E+000  0.0000000 -0.1408544  0.9900303  0.0000000 -0.1108978 -0.9938318    9.57662
1.302764E+001
  6   0.000000E+000  2.612534E+001  0.000000E+000  0.0000000 -0.1408544  0.9900303  0.0000000  0.0000000 -1.0000000    8.09729
7.274832E+001
Paraxial Ray Trace Data:
Surf       X-coord       Y-coord       Z-coord   X-cosine   Y-cosine   Z-cosine
OBJ        Infinity      Infinity      Infinity  0.0000000  0.0000000  1.0000000
  1   0.000000E+000  3.934634E+001  0.000000E+000  0.0000000  0.0000000  1.0000000
  2   0.000000E+000  3.934634E+001  0.000000E+000  0.0000000 -0.2278490  0.9736965
  3   0.000000E+000  3.373024E+001  0.000000E+000  0.0000000 -0.3691049  0.9293877
  4   0.000000E+000  2.975875E+001  0.000000E+000  0.0000000 -0.1173775  0.9930874
  5   0.000000E+000  2.928597E+001  0.000000E+000  0.0000000 -0.2169305  0.9761871
  6   0.000000E+000  1.373042E+001  0.000000E+000  0.0000000 -0.2169305  0.9761871
```

## Ray tracing pitfalls

- Total internal reflection
- The ray misses the surface
- Two possible intersections
- Ray intersection point at some aspheric surfaces may not be found with the iterative algorithm

## Surface types

- Sphere
- Conics
- Cartesian ovals (no implemented)
- Even polynomials
- Odd polynomials
- Zernike surfaces
- Many, many more
- User defined surfaces

## Summary

- Understanding ray tracing is critical in optical design
- Ray tracing pitfalls
- Speed and precision
- Skill of user defining surfaces (UDS)