

# Shape Interrogation for CAD/CAM

**Note:** These codes were tested on the 32-bit Linux boxes with GNU's C/C++ (gcc/g++)

## (1) solpow.cc

Program to illustrate the projected polyhedron nonlinear system solver (w/ input as *power basis* polynomial equations)

### **Input file format:**

*number of equations (N), number of variables*  
*degree list for equation 1*  
*coefficients for equation 1*  
.....  
*degree list for equation N*  
*coefficients for equation N*

How to make:

**prompt> make**

How to run:

- (1) for floating point arithmetic:  
    **prompt> solpow-fpa input\_file\_name**
- (2) for rounded interval arithmetic:  
    **prompt> solpow-ria input\_file\_name**

**Example:** intersection between a circle  $x^2 + y^2 = 1$  and an ellipse  $x^2/4 + 4y^2 = 1$  in the first quadrant:

**prompt> solpow-fpa solpow.in**  
**prompt> solpow-ria solpow.in**

## (2) solbern.cc

Program to illustrate the projected polyhedron nonlinear system solver (w/ input as *Bernstein basis* polynomial equations)

### **Input file format:**

*number of equations (N), number of variables*  
*degree list for equation 1*  
*coefficients for equation 1*  
.....  
*degree list for equation N*  
*coefficients for equation N*

How to make:

**prompt> make**

How to run:

(1) for floating point arithmetic:

prompt> solbern-fpa input\_file\_name

(2) for rounded interval arithmetic:

prompt> solbern-ria input\_file\_name

### Examples:

(1) intersection between a circle  $x^2 + y^2 = 1$  and an ellipse  $x^2/4 + 4y^2 = 1$  in the first quadrant:

**prompt> solbern-fpa solbern.in**

**prompt> solbern-ria solbern.in**

(2) degree 20 Wilkinson polynomial in Section 4.9 of the hyperbook

**prompt> solbern-fpa wil.in**

**prompt> solbern-ria wil.in**

Note solbern-fpa misses some roots due to the robustness issues described in section 4.6 of the hyperbook.

(3) finding the extrema of the squared distance between two spheres, which results in 6 equations w/ 6 unknowns:

$$(x_1 - 0.2)^2 + (y_1 - 0.2)^2 + (z_1 - 0.2)^2 - 0.04 = 0$$

$$(x_2 - 0.2)^2 + (y_2 - 0.2)^2 + (z_2 - 0.8)^2 - 0.04 = 0$$

$$(x_1 - x_2)(z_1 - 0.2) - (z_1 - z_2)(x_1 - 0.2) = 0$$

$$(y_1 - y_2)(z_1 - 0.2) - (z_1 - z_2)(y_1 - 0.2) = 0$$

$$(x_1 - x_2)(z_2 - 0.8) - (z_1 - z_2)(x_2 - 0.2) = 0$$

$$(y_1 - y_2)(z_2 - 0.8) - (z_1 - z_2)(y_1 - 0.2) = 0$$

**prompt> solbern-fpa solbern6.in**

**prompt> solbern-ria solbern6.in**

### (3) sollex.cc

Program to illustrate the projected polyhedron nonlinear system solver (w/ input as *lexical representation* of power basis polynomial equations)

#### Input file format:

*number of equations (N)*

*degree list for equation 1*

*body of equation 1*

.....

*degree list for equation N*

*body of equation N*

*end*

### **Notes on the input:**

Every variable must be written in  $x_i$  i.e.  $x$  followed by an integer. Elevation of a variable to a certain power should be expressed by the symbol  $^$  i.e. square of  $x_1$  is  $x_1^2$ . There should not be any  $=$  sign (followed by 0) i.e. we assume the right hand side of all the equations is equal to zero. All the coefficients are real numbers and may be written as integers, floating point numbers, or in rational form (with a  $/$ ) e.g.  $0.5x_1$  can be also written as  $1/2x_1$ . There should not be any multiplication symbol such as  $*$  in the input. For instance,  $x_1x_2$  should be written as  $x_1x_2$ . The last line should be "end" indicating the end of input file – see e.g. [sollex.in](#).

How to make:

**prompt> make**

How to run:

(1) for floating point arithmetic:

prompt> sollex-fpa input\_file\_name

(2) for rounded interval arithmetic:

prompt> sollex-ria input\_file\_name

**Example:** 3 equations and 3 unknowns:

$$10x - 20y + 30z = 14$$

$$20x + 10y - 40z = -2$$

$$-30x + 40y - 10z = -2$$

**prompt> sollex-fpa sollex.in**

**prompt> sollex-ria sollex.in**

For further information, please read [./solver.pdf](#) document.