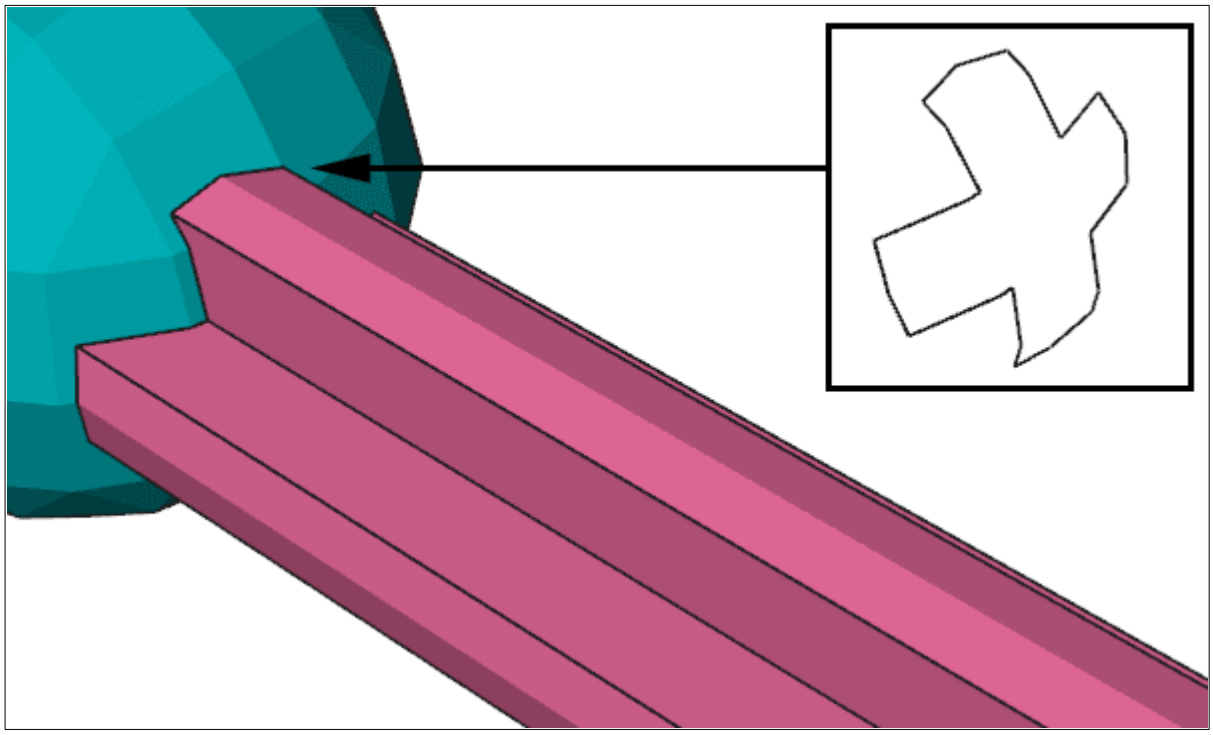
Isecalc, Intersection Line Calculator



Isecalc utility computes the intersection line of two sets of triangles or quads. Each set is provided to the utility as separate LDraw files. A third file containing the intersection lines is created.

It is a simple console application, source code is provided below to anyone willing to integrate it in a more palatable interface.

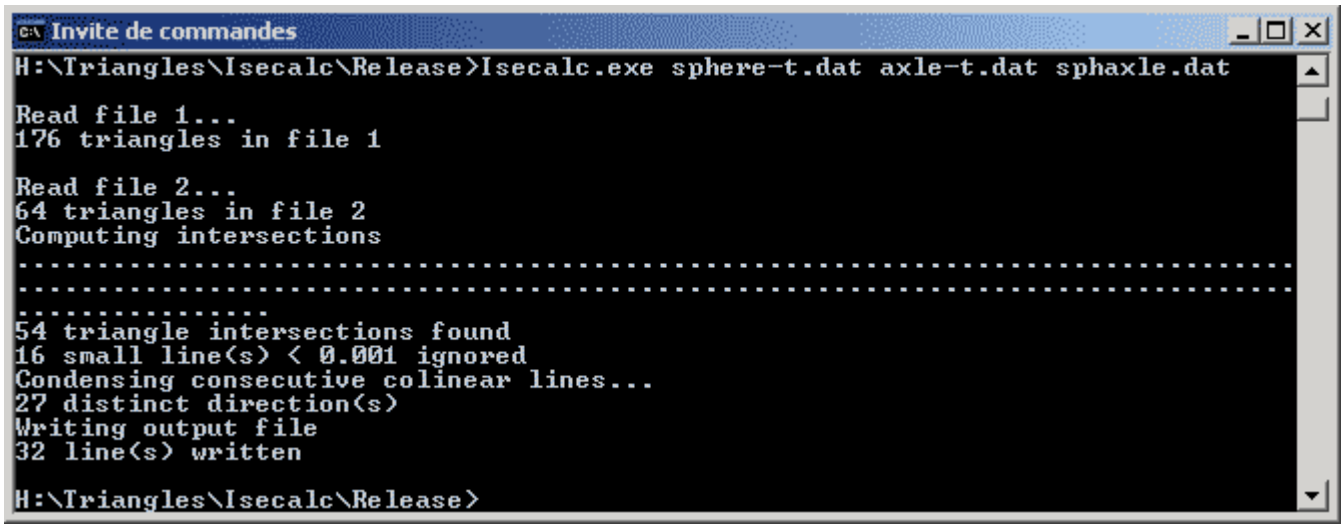
Download

- **Isecalc** program for Windows
- **Isecalc** program for Macintosh (universal binary). Compilation courtesy of Jim DeVona
- **Isecalc** source files (Visual C++ 6.0)
- Sample files
  - sphaxle.ldr contains a Technic axle stuck in a sphere primitive, slightly off-center..
  - sphere-t.dat is the inlined sphere file.
  - axle-t.dat is the inlined axle file.
  - sphaxle.dat is the intersection line as created by **Isecalc**.
  - sphaxle2.dat contains the intersection line with the original axle and sphere.
- Or get the full package.

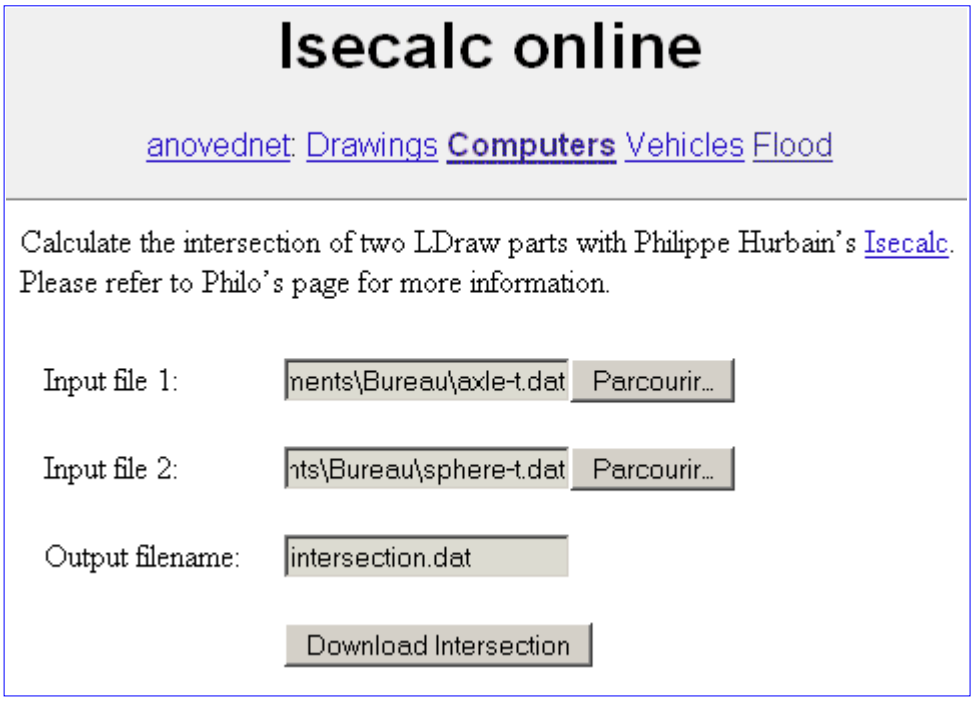
Usage

- Prepare the input LDraw files. **Isecalc** calculates only the intersections of triangles and quads. Other LDraw line types are ignored. If you want to compute an intersection with parts or primitives, they must be inlined down to triangles and quads. LDDesignPad does this very conveniently.
- Launch a command prompt
- Type the command line: Isecalc LdrawFile1 LdrawFile2 LdrawIntersectionFileOut. Isecalc will create LdrawIntersectionFileOut, containing the intersection line. Note that if file LdrawIntersectionFileOut exists it will be overwritten without warning.
- **Isecalc** output file with 6 digits after decimal point, this precision is excessive for most usages and values should be rounded. Here again, LDDesignPad does that very well.

Here is a screen shot of a sample run:



If you don't like command line, you will prefer Jim DeVona's **Isecalc** online version:



How Isecalc works

- Both input files are read and parsed. Quads are split into 2 triangles. Quads with bad winding ("bow-tie") will not be properly processed. All triangle vertexes are stored in arrays (limited to 1000 triangles, should be more than enough!).
- All triangles of one set are tested for intersection with all triangle of the second set. If an intersection line is found, it is stored in an array (limited to 2000 lines). The triangle intersection program originate in Tomas Möller thesis. Very short lines (length < 0.001 ldu) are ignored.
- The lines are then sorted according to their direction, all lines with the same direction receive the same tag.
- Lines with matching direction tags are scanned for identical endpoints. In that case they are condensed in a single line.
- Output file is created.