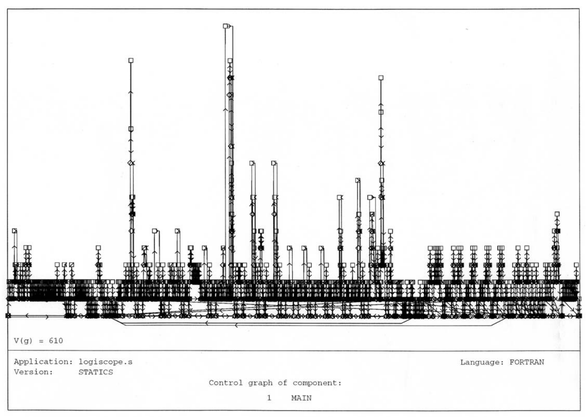
An Object Oriented Approach to Processing Accelerator Alignment Measurements

Introduction

The CERN least squares compensation program, LGC, has been gradually re-written in C++. Part of the program has been integrated into a software library, with the functionality specific to LGC built on top. The geodetic transformations used within the program have been updated and a sparse matrix representation added for the least squares matrices, this latter change contributes, in part, to a significant increase in the calculation speed. The program is designed to handle all the different measurement types that are used by the Survey group at CERN, and to process the data in a 3D coordinate system either locally defined, or, more rigorously, taking into account the horizontal and vertical geodetic reference surfaces that have been established for the CERN site.

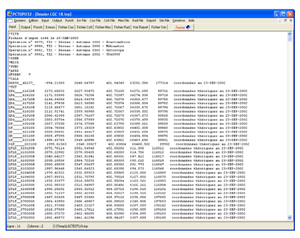
FORTRAN Code

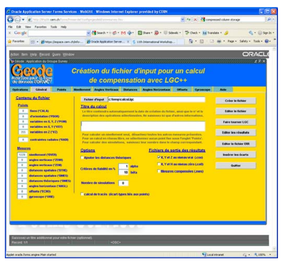
The LGC program was originally written in FORTRAN77 in the 1980's, at the time of the construction of LEP. The call graph of the main routine, which represented nearly a third of the code, shows a typical path through the application. The horizontal lines show the processing loops implemented with infamous GOTO statements.



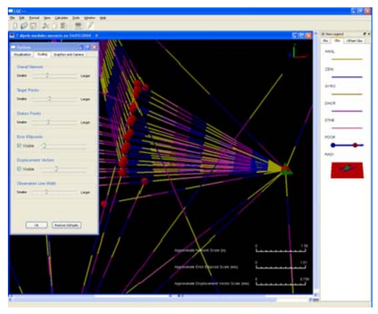
User Interface

The current user interface for LGC is provided either by PCTOPO (a utility program, providing a GUI to a number of our applications, including LGC), or by Geode (a web based front end for the SURVEY database).





During the development of the new C++ version a student project was also launched to create a GUI front end for LGC, with an emphasis on the graphical presentation of the measurement networks. The project will provide a basis for future work in this direction, and has shown the areas where more work is still required.



Sparse Matrices

$$A = \begin{bmatrix} 10 & 0 & 0 & 0 & -2 & 0 \\ 3 & 9 & 0 & 0 & 0 & 3 \\ 0 & 7 & 8 & 7 & 0 & 0 \\ 3 & 0 & 8 & 7 & 5 & 0 \\ 0 & 8 & 0 & 9 & 9 & 13 \\ 0 & 4 & 0 & 0 & 2 & -1 \end{bmatrix}$$

Value	10	3	3	9	7	8	4	8	8...9	2	3	13	-1
Row Index	1	2	4	2	3	5	6	3	4...5	6	2	5	6

Column Pointer	1	4	8	10	13	17	20

An implementation of a sparse matrix has now been included in LGC . The implementation is based around a compressed column storage algorithm which reduces the memory storage requirements and provides faster operations.

To explain this type of storage consider matrix A. The i^{th} entry for the row index, RowIndex(i), gives the row number on which Value(i) appears in A. For the Column Pointer, the value (ColPtr(i+1) - ColPtr(i)) indicates the number of non-zero elements in column i.

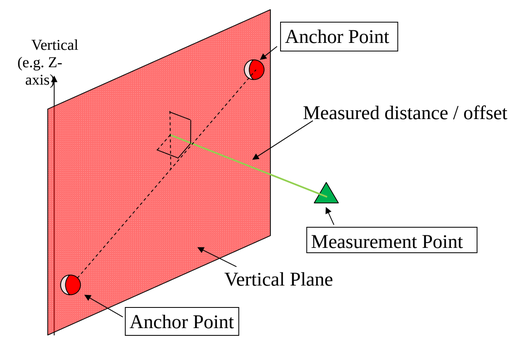
Reference Frame Transformations

The transformation algorithms used within LGC have been updated to a more rigorous implementation. The CERN Geodetic reference frame was included, and Local Astronomical reference frames added. These were integrated into the processing algorithm to assure the highest possible precision.

For the purposes of the transformation algorithm the reference frames are represented as nodes of a graph and the transformations as links between the nodes. The sequence of transformations to pass between any pair of reference frames is determined at run time by simply finding the shortest path between them on the graph.

Simple 3D Objects

Horizontal offsets are used extensively in most accelerator planimetric survey and alignment operations at CERN. These measurements have always been characterised by two spatial points (called anchor points), a measured point, and the measured offset distance (the horizontal distance between the measured point and the straight line joining the two anchor points). A graphical presentation is given on the right together with the observation equation (Eq. 1). In reality the instrument is actually placed on a point and the distance to a vertical plane, characterised by a stretched wire, is measured.



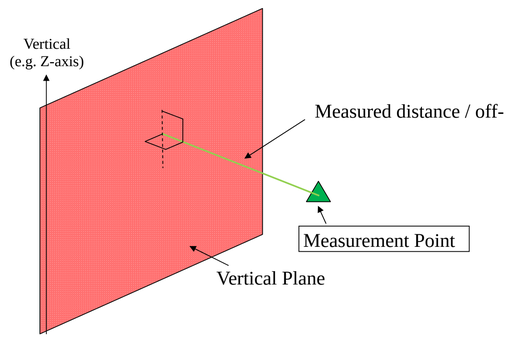
$$S^2 = (X_M - X_{A1}) \bullet (X_M - X_{A1}) - \left(\frac{(X_M - X_{A1}) \bullet (X_{A2} - X_{A1})}{|X_{A2} - X_{A1}|} \right)^2 \qquad (1)$$

where,

S = offset distance

X_M = 2D coordinates of the measurement point

X_{A1}, X_{A2} = 2D coordinates of the anchor points



$$S = X_M \bullet \hat{n} + d \qquad (2)$$

where,

S = offset distance

X_M = coordinates of the measurement point

$\hat{n}$ = horizontal unit vector normal to the plane

d = the plane distance parameter

$$S = (X_M - X_0) \bullet \hat{n} + d_0 \qquad (3)$$

where in addition to Eq. 2,

X_0 = coordinates of an arbitrary fixed point

d_0 = the plane distance parameter, from X_0

An alternative approach is to process these measurements directly as offsets to vertical plane (see graphical presentation of the observation on the left), and introduce the parameters of the plane into the least squares adjustment as unknowns. This observation equation (Eq. 2) is much simpler, has the advantage of using the actual field measurements, and does not require any specific pre-processing of the measurements. A small disadvantage of is that constraints must also be added to the least squares adjustment, e.g. a constraint to fix the length of the plane vector.

If we introduce a fixed point much closer to the area being measured (see Eq. 3), a small rotation of the plane should not induce significant changes in the plane distance parameter, and the least squares process will converge quicker.

Conclusions

Following many years of intermittent development, a new C++ version of LGC has now be produced to replace the original F77 version. This project has also resulted in the creation of a software library which we have called SurveyLib -also the basis for a couple of other new in-house applications. Further developments are now under way to allow the integration of LTD measurements, and simple 3D objects into an adjustment.

At the beginning of the project it was not easy to find the people with the right skills, but surveyors with C++ experience can now be found, and teaching surveying and least squares to mathematically minded computer programmers has also proven to be a good way forward.

11th International Workshop on Accelerator Alignment • DESY • Hamburg • Germany • 13th - 17th September 2010

Mark JONES • CERN • CH-1211 Geneva 23 • Switzerland • Mark.Jones@cern.ch