A Vivid Relic Under Rapid Transformation

Thomas Knudsen, Simon L. Kokkendorff, Karsten E. Engsager

Abstract

TRLIB, the transformation system of the Danish National Survey and Cadastre (KMS), has served as the official tool for transformation between Danish map projections and horizontal and vertical datums for half a century. TRLIB system originated as a fairly simple system using one of the first compilers for the Algol-60 language, but it has evolved dramatically since then. Recently the extended transverse mercator implementation of TRLIB was included in PROJ, the leading open source projection package. This is well in line with other efforts towards making TRLIB more interoperable with other open source projection and transformation packages.

The main objective of this paper is to briefly outline the historical origins of TRLIB, and to describe the architectural design considerations behind an ongoing effort to make TRLIB ready for today's requirements, most notably by making it 64-bit clean and thread safe. But since the TRLIB code is used in a large number of commercial GIS products, the design considerations are also influenced by the importance of maintaining a stable, backwards compatible version, while also utilizing this golden opportunity to simplify and improve the application program interface.

Historical introduction

In the late 1950s, the Danish Geodetic Institute (Geodætisk Institut, GI) realized the need for a high performance numeric computer in order to complete a diverse range of tasks, including the adjustment of geodetic trianglation networks, adjustment of data from high precision levelling campaigns, and general support for operational mapping.

The computer, christened GIER—Geodætisk Instituts ElektronRegnemaskine (Geodetic Institute's Electronic Computer), was delivered in September 1961 (Gram et al., 1963). It was built by the Danish organization RC (RegneCentralen: Literally "the computing center", formally "The Danish Institute of Computing Machinery at the Danish Academy of Technical Sciences"). While GIER was designed in close cooperation with GI and optimized for geodetic computations, it was also remarkably successful as a generic number cruncher: a series of GIERs were built by RC and deployed in organizations around Europe, where some of them remained active for almost 20 years.

GIER was the platform for development of some of the first ALGOL60-compilers, but prior to that, the programs needed for the geodetic operations were written directly in GIERs low level assembly language.

Through a journey spanning half a century, involving a sequence of at least 4 programming languages, numerous operating systems, and an innumerable number of organizational reorganizations, these original GIER programs have evolved into what is known today as the Geodetic Software System of the National Survey and Cadastre (National Survey and Cadastre is the Danish national mapping and geodata agency, Kort & Matrikelstyrelsen – KMS).

Basically, the Geodetic Software System consists of two parts:

1. ADJ, an adjustment package, solving the complex nonlinear least squares problems of geodetic network adjustment,
2. TRLIB, a 3D transformation package taking an integrated approach to coordinate transformations, providing unified access to map projections and generic reference system transformations, including horizontal and vertical datum shifts.

ADJ is primarily used for internal KMS operations, and will not be mentioned further in this paper. TRLIB, on the other hand, is widely distributed and built into a large number of commercial GIS systems, mostly in order to provide support for specific Danish map projections and datums (although TRLIB is actually a fairly generic transformation system).

The aim of this paper is to describe aspects of TRLIB, especially with respect to how its long history and ingrained 3D focus makes it differ from packages more narrowly focused on the map projection part of geodetic coordinate handling.

Approaches to numeric stability, algorithmic correctness, and evaluation of accuracy

Throughout TRLIB, care has been taken in selecting fast and numerically stable methods, e.g. Horner's scheme for evaluation of simple polynomia, and Clenshaw summation (Clenshaw, 1955, Tscherning and Poder, 1982) for evaluation of trigonometric series and other elements that can be defined as recurrence relations (Legendre polynomia et cetera).

Additional care has been taken to stand on the shoulders of giants, wherever possible. Most notably by building on the material compiled by König & Weise in their seminal reference work on the mathematical foundations of classical geodesy and cartography (König and Weise, 1951).

But selecting numerically stable and scientifically trustworthy algorithms is not enough to ensure neither numeric accuracy of results, nor correct implementation of the algorithms. But at least implementation errors lend themselves readily to analytical scrutiny and are evident once they are detected (although some pathological corner cases may be very hard to chase down).

Analytical scrutiny is much less fruitful for the other main class of "methods for obtaining wrong results": In the pioneer days of computers, subtle hardware failures were much more common than today. While not necessarily leading to catastrophic hardware breakdowns, such failures can flip the bits of the results in hard-to-detect, but exactly for that reason disproportionally catastrophic ways: once a slightly disturbed result has been entered into the geodetic databases, it may negatively influence the accuracy of network adjustments, and derived results, for years to come.

To avoid (or at least to detect) all classes of wrong results, whether induced by implementation or hardware errors, TRLIB has been designed around a policy of dual self-checking transformations: whenever possible a forward transformation is followed by a reverse, and concluded by a check of the resulting roundtrip deviation, before the transformation result is returned. Architecturally, this can often be implemented in ways where the forward and reverse transformations to a large degree share the same code, leading to easier maintenance.

While evidently having its roots in best practice from a time where hardware stability was shaky, the method of dual self-checking algorithms remains a useful feature: For an organization as KMS that among many other roles, is also a custodian of historical geodata, it is always important to ensure that any transformations that may be applied can also be reversed. While elaborate, the dual self-checks ensures that uncaught implementation blunders and asymmetric singularities are diagnosed and reported, which is essential for long term (multiple century scale) geodata interoperability.

Datum shifts

TRLIB includes support for datum shifts. The traditional way to handle datum shifts is by the application of either 3-parameter or 7-parameter Helmert transformations. While useful (and implemented in TRLIB), the Helmert transformation does not handle tensions in the original survey networks behind the two datums transformed to and from. Traditionally such tensions are handled by applying an empirically determined transformation—either represented as a high order polynomial, or as a (NADCON style) correction grid. TRLIB supports both these approaches.

For transformation of vertical datums, things may be further convolved by the question of which geoid model was in use for assigning level coordinates to non-levelled (e.g. GPS derived) heights. TRLIB handles these complications transparently through an extensive collection of background information for each datum supported.

TRLIB also includes some support for time varying transformations based on plate tectonic motion models, for implementing the proper transformations between a global datum (e.g. ITRF89) and regional, plate fixed reference systems (such as ETRS89) at different epochs.

Coordinate systems in TRLIB which include a vertical datum are viewed as truly 3-dimensional. When performing a (horizontal) transformation between systems based on the same datum, the third/height coordinate of a point is largely irrelevant and may be viewed as an extra attribute ("2.5D"). However when doing transformations between 3D-systems based on different horizontal datums, the actual 3-dimensional position of the point is taken into consideration and the resulting horizontal coordinates depend on the third coordinate. In this sense, TRLIB is a true 3D (and not just 2.5D) transformation system.

The major challenges

While TRLIB still plays a major role as the computational work horse of the KMS, it also shows its age through artefacts from historical design decisions, that are less fortunate when running the system on modern hardware and under modern operating systems.

Currently, we are planning a restructuring of the TRLIB code. The major problem lies in the use of global variables for the internal state, making the system inherently unsafe for execution in a parallel or multi-threaded environment. We intend to collect all internal state in an opaque object that is passed to the library functions on the stack, in a similar way to the application program interface (API) exposed by the PROJ map projection library (Evenden, 1990).

But since the TRLIB code is used in a large number of commercial GIS products, the design considerations are influenced by the importance of maintaining a stable, backwards compatible version, while also utilizing this golden opportunity to simplify and improve the API.

Comparison to PROJ

As already mentioned, we plan for a more PROJ like API for TRLIB. On the PROJ side, recently the transverse mercator implementation from TRLIB was included in the PROJ code base. In coming releases of PROJ the TRLIB code will be found as PROJ's new *etmerc* Extended Transverse Mercator projection (Poder and Engsager, 1998, Engsager and Poder, 2007, Karney, 2011). So called because it preserves roundtrip precision far longer from the central meridian (0.03 mm at 7500 km) than the existing PROJ *tmerc*.

So while PROJ and TRLIB certainly may converge in many ways, there are still some unavoidable differences, due to PROJ's background as a cartographic map projection library, and TRLIB's background as a geodetic transformation library. From a user's perspective, this becomes especially clear at the metadata level.

At the metadata level, TRLIB handles coordinate system descriptions using so called minilabels. A minilabel is a single string combining information on the projection, the horizontal datum and (if relevant) the vertical datum.

For example, the minilabel "utm32_wgs84" defines a horizontal coordinate system based on the UTM zone 32 projection of WGS84 coordinates. In PROJ, this corresponds to the definition string "proj=utm, zone=32, ellps=WGS84".

The somewhat longer minilabel "utm32Hed50_h_dvr90" specifies a 3D coordinate system with horizontal datum being ED50 (The European Datum 1950, based on the Hayford/International ellipsoid), and vertical datum being the Danish height system DVR90.

TRLIB includes some support for translating a minilabel to ESRI-WKT format. In combination with the spatial reference system abstractions provided by the widely used *gdal/ogr* package, this provides a mechanism for automatic translation of many minilabels representing 2D-systems to PROJ parameters. For 3D-systems this is (at present) not possible since PROJ still lacks a metadata vocabulary for the description of vertical datums.

In some rare cases, TRLIB fails to produce WKT in a form that is recognizable by gdal/ogr. We currently consider handling this in a simplistic but robust way, by supplying a manually compiled list of minilabels and their corresponding EPSG codes.

Conclusion

We have presented the half-centennial historical background of TRLIB transformation system. Despite its age it is still vivid and ready for the future. TRLIB was only recently released under an open source license (and made available through https://bitbucket.org/KMS/trlib), but in the near future we hope to implement means for better interoperability with the more well established libraries in the open source geomatics field.

Acknowledgements: We thank Willy Lehmann Weng and Knud Poder for commenting on the draft of this paper.

Bibliography

C. W. Clenshaw. A note on the summation of chebyshev series. *Math. Tables Aids Comput.*, 9(51):118–120, 1955. URL http://www.jstor.org/stable/2002068.

Karsten E. Engsager and Knud Poder. A highly accurate world wide algorithm for the transverse mercator mapping (almost). In *Proc. XXIII Intl. Cartographic Conf. (ICC2007), Moscow*, page 2.1.2, 2007.

Gerald I. Evenden. Cartographic projection procedures for the unix environment—a user's manual, 1990. US Geological Survey Open-File Report 90–284.

C. Gram, O. Hestvik, H. Isakson, P.T. Jacobsen, J. Jensen, P. Naur, B.S. Petersen, and B. Svejgaard. Gier - a danish computer of medium size. *IEEE Transactions on Electronic Computers*, EC-12(5):629–650, December 1963. URL http://www.datamuseum.dk/site_dk/rc/gierdoc/ieeeartikel.pdf.

Charles Karney. Transverse mercator with an accuracy of a few nanometers. *Journal of Geodesy*, 2011(1):1–11, 2011. doi: http://dx.doi.org/10.1007/s00190-011-0445-3.

R. König and K. H. Weise. *Mathematische Grundlagen der Höheren Geodäsie und Kartographie, Erster Band*. Springer, Berlin/Göttingen/Heidelberg, 1951.

Knud Poder and Karsten Engsager. Some conformal mappings and transformations for geodesy and topographic cartography, 1998. National Survey and Cadastre, Denmark, Publications, 4. series vol. 6.

Carl Christian Tscherning and Knud Poder. Some geodetic applications of clenshaw summation. *Bolletino di Geodesia e Scienze Affini*, XLI(4):349–375, 1982.

Thomas Knudsen
National Survey and Cadastre
kms.dk
thokn AT kms DOT dk