SOFA Software Progress Report

P. Wallace

Rutherford Appleton Laboratory, Chilton, Didcot, Oxon, OX14 4HH, United Kingdom

Abstract. The IAU set up the SOFA initiative at the 1994 General Assembly, to promulgate an authoritative set of fundamental-astronomy constants and algorithms. At the subsequent General Assembly, in 1997, the appointment of a Reviewing Board and the selection of a site for the SOFA Center (the outlet for SOFA products) were announced. The time since then has been spent developing the specifications for a basic collection of fundamental-astronomy algorithms. Progress has been slow, for a number of reasons: the task is difficult and requires great circumspection; a substantial toolkit of support routines must be adopted or developed before work on the astronomy routines themselves can begin; and it takes time for a geographically dispersed committee to reach a consensus on matters that involve value judgements, such as programming style. However, despite these formidable difficulties, significant progress has been made. A library of 52 vector/matrix routines has been developed, supporting a set of 25 astronomy algorithms. The Reviewing Board is about to start validating these algorithms, at which point they can be released through the SOFA Center.

1. Introduction

The SOFA initiative, proposed by the Working Group on Astronomical Standards, was ratified by the IAU at the 1994 General Assembly (Fukushima, 1995). The objective was to make...

> "...new arrangements to establish and maintain an accessible and authoritative set of constants, algorithms and procedures that implement standard models used in fundamental astronomy."

The four elements of the scheme were (i) a Maintenance Committee, (ii) a Relativity Subgroup, (iii) a Reviewing Board and (iv) the SOFA Center. The present paper deals with the last two, the SOFA Reviewing Board and the the SOFA Center.

In the intervening period, the Reviewing Board has been created and is active, the SOFA Center has been set up and algorithms for a first release have been drafted.

2. The SOFA Reviewing Board

The current Board membership is as follows:

Wim Brouw	Australia Telescope National Facility
Anne-Marie Gontier	Paris Observatory
Catherine Hohenkerk	Royal Greenwich Observatory
Wen-Jing Jin	Shanghai Observatory
George Kaplan	United States Naval Observatory
Zinovy Malkin	Inst. Appl. Astron., St Petersburg
Dennis McCarthy	United States Naval Observatory
Jeffrey Percival	University of Wisconsin
Patrick Wallace	Rutherford Appleton Laboratory

The limited funding available to Board members for overseas travel makes it difficult, if not impossible, to hold meetings of the whole Board, and so almost all business has to be conducted *via* e-mail. However, there have been meetings of subgroups and between individuals, and these have been an invaluable supplement to the e-mail correspondence.

3. The SOFA Center

The SOFA Center is where users can access information about SOFA and obtain the SOFA products themselves. Bids to host the Center were invited at the beginning of 1997 through an Announcement of Opportunity (IAU, 1997). Only two bids were received, from the US Naval Observatory and from the Rutherford Appleton Laboratory. In each case the proposed service was on a fairly small scale, and no specific levels of effort were committed. The bid from RAL was successful, and in the middle of 1997 the Center was established.

The Center is now awaiting the first release of software. Distribution of software and other products will be through a Web-based self-service "software store", and will draw on the experience and techniques of the Starlink Project.

4. Algorithms

The initial goal has been to create a set of callable subprograms. Whether "subroutines" or "functions", they are all referred to simply as "routines". They are used by software developers to write applications. No complete, runnable, free-standing applications are included in SOFA's preliminary plans.

The SOFA policy is to organize the calculations so that the machine accuracy is fully exploited. The gap between the precision of the underlying model or theory and the computational resolution has to be kept as large as possible, hopefully leaving several orders of magnitude of headroom. This point needs careful attention, especially in cases involving epochs and cyclic phenomena. The draft SOFA routine for converting Universal Time into Greenwich Sidereal Time is a good example. Firstly, the epoch, which is a Julian Date, is supplied as two double precision numbers, apportioned in any way the caller wishes. If maximum accuracy is desired, the caller provides a Part 1 that is an integer,

which in practice all floating-point architectures encode without error, and a Part 2 that is small and therefore relatively immune from rounding errors. Secondly, the canonical algorithm is organized so that integer and fractional parts are kept separate as far as possible. Integer multiples of one day are omitted from the calculation and never deplete the available precision.

One of the difficulties in designing the SOFA routines is that there is an inevitable compromise between rigor and ease of use (and also speed, though nowadays this is seldom a major concern). There will inevitably be some calculations where multiple routines have to be provided, each offering a different trade-off.

5. The SOFA routines

So far, a library of 52 Fortran vector/matrix routines has been drafted, supporting a set of 25 astronomy routines, also Fortran. The Reviewing Board is about to embark on a process of validating these algorithms prior to their release through the SOFA Center. At this stage, versions in other languages are planned.

Each SOFA routine consists of an ample set of preamble comments, followed by the code itself, followed by a lengthy set of standard comments setting out the copyright and disclaimer conditions. In most cases the code, which often is only a few lines, is dwarfed by the commentary. However, this is of little consequence as far as the user is concerned. The preamble includes comprehensive instructions on how to use the routine, the code is clearly set off and can be quickly found, and the formalities are left until last.

5.1. The vector/matrix library

The SOFA Vector-Matrix Library contains a collection of simple tools for manipulating the vectors, matrices and angles used by the positional-astronomy routines. Although this library could be used in its own right, it is limited in scope to what the other SOFA routines require, and does not attempt to compete with the many specialist libraries available elsewhere.

As well as ordinary 3-vectors, which are referred to as "p-vectors" (p standing for position), there is support for 6-element "pv-vectors" (v standing for velocity). The six elements are, in fact, expressed as two successive groups of three, to highlight the relationship between the respective Cartesian components. The order of the elements, (3,2) in Fortran and [2][3] in C, enables each triple to be used in its own right as a 3-vector, which makes for neater coding.

The only matrices which it has been so far necessary to use are ordinary 3×3 real orthogonal "rotation" matrices, referred to as "r-matrices". Routines also exist which support a second 3×3 layer containing the derivative of the corresponding r-matrix element. Such matrices are called "rs-matrices", where rs stands for rotation/spin. However, these have yet to be used, and so are being held in reserve until they are needed. For expressing spins, it has so far proved most convenient to specify the axis of rotation directly as a 3-vector, the Euler axis scaled by the Euler angle in radians.

Positional calculations are all vector based of course; spherical coordinates are used only for purposes involving the interface to users or catalogs.

The 52 routines offer a wide range of operations on and between these entities: initialize, copy, build rotations, rotate, add, subtract, products, spherical to/from Cartesian, range normalization, separation, position-angle, r-matrices to/from spin vectors, and so on.

5.2. The astronomy library

The full list of astronomy routines that have been drafted so far is as follows:

- Calendars
  - Gregorian Calendar to Julian Date
  - Julian Date to Gregorian year, month, day, fraction
  - Julian Date to Gregorian date for formatted output
  - Julian Date to Besselian Epoch
  - Julian Date to Julian Epoch
  - Besselian Epoch to Julian Date
  - Julian Epoch to Julian Date
- Timescales
  - ΔAT (=TAI−UTC) for a given UTC date
  - TDB−TT
  - Greenwich Mean Sidereal Time for a given UT1
  - equation of the equinoxes, IAU 1994
- Ephemerides (limited precision)
  - Earth position and velocity
  - major-planet position and velocity
- Precession/Nutation
  - nutation, IAU 1980
  - mean obliquity, IAU 1980
  - nutation matrix, IAU 1980
  - precession matrix, IAU 1976
  - precession, IAU 1976
  - precession/nutation matrix, IAU 1976/1980
- Star space motion
  - star catalog coordinates to position+velocity vector
  - star position+velocity vector to catalog coordinates
- Star catalog conversions
  - proper motion between two epochs

- transform FK5 star data into the Hipparcos frame
- transform Hipparcos star data into the FK5 frame
- FK5 to Hipparcos assuming zero Hipparcos proper motion
- Hipparcos to FK5 assuming zero Hipparcos proper motion

The routines fall under two main headings: utility functions and canonical models. The calendar conversions are an example of the former; the 1976 precession is an example of the latter (or, more correctly, is a SOFA implementation of an existing, published, canonical model).

5.3. Rate of progress

The SOFA Reviewing Board has been in operation for several years, and a first release of software has yet to occur. Why is it taking so long? There are four main reasons:

1. The task demands great care and circumspection.

   To be of any value as standards, SOFA routines must be correct at their first release and may not subsequently be revised. This is a heavy responsibility for the SOFA Reviewing Board and cannot be rushed. If a routine was found to deliver imperfect results, the only recourse would be to issue a new, additional, routine under a different name, a laborious exercise leaving an ugly and potentially confusing result.

   In any case, designing a system which is comprehensive, integrated and rigorous, yet easy to use, is a difficult task.

2. Providing the basic facilities is a major enterprise in itself.

   Most of the the 70-odd routines developed so far do familiar things in a conventional way, and their counterparts exist in many existing collections of software. Nevertheless, the routines still have to be designed, documented, written, and tested.

3. Working with a committee takes time.

   From the start, Board decisions have been made, whenever possible, on the basis of consensus. Debates on every aspect of the work have been permitted and indeed encouraged, so that consensus can emerge and be embraced. There have, for example, been lengthy exchanges about programming style as well as more fundamental issues such as choice of programming language. The Board membership was chosen on the basis of interest and expertise, and it would be worrying if such debates did not arise! Reaching a consensus is an annealing process, and cannot be rushed: discussions must be allowed run their course. Sometimes a majority view emerges but fails to convince all Board members. When this happens, there is no choice but to hold a vote, which again takes time.

4. There is only limited effort available.

   All of the Board members are busy people, under heavy pressure from their other work. Formal funding for their participation in SOFA exists

only in a minority of cases, and then covers only a small fraction of their time. (These difficulties are not confined to SOFA of course but must afflict many other IAU projects.) Moreover, the available time is patchy, and Board members may not be in a position to react promptly to a new development.

6. Fundamental issues

Although the SOFA initiative presents the Reviewing Board with many fascinating questions of detail, there are also some basic issues that have never been resolved to everyone's satisfaction. The three that I am about to address are who the software is for, what the software is for, and what language should it be in. They are to some extent interrelated.

6.1. Who are the prospective users?

Given a large enough volume of software, all possible classes of user could be satisfied. However, the limited effort available means that the SOFA software must be targeted and cannot afford simply to duplicate facilities available elsewhere. One way to subdivide the potential user community is as follows.

- Fundamental-astronomy specialists.

  In their special fields of expertise, researchers in the fundamental-astronomy community are, by definition, ahead of SOFA and indeed are, in effect, working on the next generation of SOFA algorithms. Under these circumstances, the SOFA routines are of little relevance. Even away from their specialist topics, these potential users are working at the highest levels, and they need to understand every detail of their calculations. This means that they are unlikely to want to use fundamental-astronomy software from an outside source. However, they may find SOFA routines useful as style templates for new algorithms, and for comparison purposes.

- Professional astronomers.

  This class of user wants practical tools that are easy to use. They want to be confident that the answers are right, but are not particularly interested in minutiae or issues of principle. They are already well-served by several existing packages. However, there is no reason why a well-written SOFA Software Collection could not offer an attractive alternative to established tools in the areas where their functions overlap.

- Software Developers.

  Those developing stand-alone astronomy applications may find the SOFA software a useful resource. The key advantages will be accessibility, long-term support and trustworthiness. These characteristics may even appeal to the developers of existing subroutine libraries, who may wish to replace the lower levels of their packages with calls to the SOFA routines. And for all such users, the SOFA software should be an invaluable resource for testing and validation.

- Amateurs.

  The hobbyist user is in a similar position to the professional astronomer in wanting practical tools that are straightforward to apply. However, there is an even greater demand for simple, clear documentation, and ease of use has much greater weight than milliarcsecond accuracy. Such users are likely to be best served by book-based solutions such as those provided by Meeus and Duffett-Smith. However, SOFA routines may still be of interest in some cases.

6.2. What should the SOFA software include?

There clearly are practical limits to what can be achieved, given the nature of the problem and the resources available. Although opinions among Board members vary, we have to accept that, in the short term at least, the SOFA software probably cannot be:

- Comprehensive.

  The SOFA acceptance procedures must be careful and thorough, and hence take time. Combined with the shortage of effort, it is not feasible to support a collection of more than a few dozen astronomy routines.

- Foolproof.

  Rigor and ease of use may not always be achievable in one and the same routine. It is dangerous to give the impression of great accuracy if the user has supplied too little information for this to be possible. Hidden defaulting rules or silent correction of errors are also ruled out.

- Elaborate.

  It is necessary to make the minimum of assumptions about the user's environment. This precludes techniques such as graphical user interfaces and even access to files.

But there are many desirable qualities that the software *can* aspire to. The SOFA Collection can be:

- Authoritative.

  It is important to the astronomical community in general that there be an IAU-approved way of performing all the key calculations, even if not state-of-the-art.

- Canonical.

  Computer code is the only practical way to pin down an algorithm unambiguously. A written description can only go so far, and it is all too easy to omit a small detail that then leaves the procedure open to multiple interpretations. Computer code (as long as it is properly written and platform-independent) defines exactly what happens, in what order.

- Practical.

  Packages such as NOVAS (Kaplan, 1996) and SLALIB (Wallace, 1997) have shown that it is possible to design tools that are precise and rigorous

and yet straightforward to use. Documentation is obviously an important factor, but a number of other aspects must be right as well:

- The functional space must be split into modules with the boundaries in just the right places and modules of just the right size.
- There should be no gaps that the non-specialist user will find hard to bridge. For example, if some of the routines require Julian Date, it is helpful also to provide ones that translate between calendar date and Julian Date.
- There has to be consistency, for example in naming conventions, status handling, units, choice of number representation and so on.
- It is crucial to pick the right arguments. For example, an atmospheric refraction routine that asks for the partial pressure of water vapour is likely to put off the typical user, who wants to start from relative humidity.

- Accessible and supported.

  Many fine implementations of all the key fundamental-astronomy algorithms exist. But only a few are widely used outside the group that developed them, because correct behaviour and high standards of coding are only part of the story. To be attractive to outsiders, the routines also have to be downloadable from an easy-to-find place, be in an integrated and consistent form, and come with documentation. And there has to be some guarantee that the user will be able to get help if at any point the routine stops working properly. All of this requires effort and planning.

The balance between these positive qualities and the inevitable limitations that were outlined earlier will determine whether the SOFA software is successful and who its users are.

6.3. Choice of programming language

Choice of programming language has been a source of much debate within the SOFA Reviewing Board from the start.

The key problem is a mismatch between the fundamental-astronomy and the potential user community. All suppliers of new algorithms (and most Board members) are still Fortran-based (and Fortran 77 rather than Fortran 9X). But most potential users are C-based, and many are using C++ and other Object Oriented (OO) languages such as JAVA. Two members of the Board never use Fortran.

There is little to choose between Fortran 77 and C for expressing astronomy algorithms, especially if some of the almost universally-available extensions to Fortran 77 are used. There are differences, certainly, affecting certain types of calculation, but in most cases either language will do. On balance, Fortran 77 offers rather less freedom than C, but this can even be an advantage to the SOFA Reviewing Board by reducing the scope for debate and focusing the discussion on the core functionality.

The choice faced by the Board is either to use Fortran and limit the appeal of the result to a wider audience, or to use C and break the link with the algorithm suppliers.

The obvious solution to this dilemma is not to limit the SOFA Software Collection to just one language, but to provide multiple implementations. However, during the initial period of trial and error, only one language is being used, and that is Fortran. Once the Collection has stabilized, a C version will be produced. The SOFA routines are very straightforward and self-contained—they do not even use input/output—and producing a C routine starting from Fortran is an easy process in which several Board members have considerable experience. There are even some benefits from the translation task. The process involves close scrutiny of the code, offering a further opportunity for spotting errors, and comparing the results of running the two versions can sometimes expose problems.

The dialect of Fortran being used is ANSI Fortran 77 (American National Standards Institute, 1978), the lowest-common-denominator choice. The standard is strictly complied with, except for the > 6-character routine names and the use of `IMPLICIT NONE`. The long names (for example `iau_PREC76`) are designed so that the prefixes can be eliminated by a global edit without affecting the result. The `IMPLICIT NONE` statements will be commented out in the released code.

Once the Fortran and C versions are stable, it is safe to begin looking at the prospects for OO-based products involving C++ and JAVA. Because the SOFA routines avoid the use of context that persists between calls, a C++ or JAVA class that simply provided all the SOFA calls as methods would offer little more than the original C code. However, the OO languages allow new and more sophisticated tools to be developed, based on the core algorithms, and this possibility will be studied. Finally, users of interactive languages such as Perl and Tcl can be provided with SOFA functionality simply by writing the appropriate set of wrappers.

7. Concluding remarks

The SOFA routines so far developed contain very little that is new, and no one is anxiously waiting to begin using them. On the face of it, this is not a very encouraging start, but in fact it is to be expected, because the first generation of SOFA routines is in many respects merely a dress rehearsal. Once the routines have been brought to the state where they can be released, they can act as a framework, ready to accept new algorithms. This should prove a much quicker process than it has been getting the first generation together.

The intention is to make the first release at the end of 2000, possibly including some extra routines to fill in gaps in timescale and catalog conversions and to provide apparent place capabilities. It will then be possible to develop the C version and begin looking at the possibility of C++ and JAVA products. The first new astronomy algorithms will implement the revised precession/nutation models and other algorithms by then adopted by the IAU. At that stage the SOFA Software Collection should have become an important resource.

References

American National Standards Institute, ANSI X3.9-1978.

Fukushima, T., 1995, "Report of the IAU WGAS Subgroup on Standard Procedures," in *Highlights of Astronomy,* **vol. 10**.

IAU Information Bulletin 79, January 1997, 4.2, 13.

Kaplan, G. H., 1996, Naval Observatory Vector Astrometry Subroutines.

Wallace, P. T., 1997, *SLALIB – Positional Astronomy Library,* Starlink User Note 67.36.