Plans for the Vienna VLBI Software VieVS

J. Boehm, H. Spicakova, L. Plank, K. Teke, A. Pany, J. Wresnik, S. Englich, T. Nilsson, H. Schuh
Institute of Geodesy and Geophysics, Vienna University of Technology, 1040 Vienna, Austria

T. Hobiger, R. Ichikawa, Y. Koyama, T. Gotoh, T. Kubooka
National Institute of Information and Communications Technology, Tokyo, Japan

T. Otsubo
Hitotsubashi University, Tokyo, Japan

Abstract. A new VLBI (Very Long Baseline Interferometry) data analysis software (called Vienna VLBI Software VieVS) is developed at the Institute of Geodesy and Geophysics in Vienna taking into consideration all present and future VLBI2010 requirements, e.g. phase delay solutions and a significantly denser observation schedule. The programming language Matlab is used, which considerably eases the programming efforts because of many built-in functions and tools. Matlab is the high-end programming language of the students at Vienna University of Technology and at many other institutes worldwide. Thus, the new software will attract students and scientists to get interested in VLBI and to contribute to VLBI analysis. Together with the National Institute of Information and Communications Technology (NICT, Japan) phase delay solutions will be implemented in VieVS and initial steps will be taken to equip the software with tools for spacecraft tracking and space VLBI. Furthermore, the gap existing between the correlator output and the presently used Occam software will be closed in the Vienna VLBI software. The common efforts will contribute to the new specification of the VLBI data format, which is defined within Working Group 4 on VLBI Data Structure of the International VLBI Service for Geodesy and Astrometry (IVS).

Keywords. VLBI software, Matlab

1 Introduction

So far, at the Institute of Geodesy and Geophysics (IGG) of the Vienna University of Technology, we have been using the Occam software package (Titov et al., 2004) for the analysis of Very Long Baseline Interferometry (VLBI) observations. Occam (previously called Bonn VLBI Software System, BVSS) has been in use since the beginning of the eighties, and many subroutines and source code lines have piled up which are obsolete and no longer necessary. Some technical Fortran-related details date back to the eighties, e.g. the use of COMMON blocks to exchange variables between the subroutines. Thus, the source code of Occam is rather difficult to read for persons who have not been involved in the development of the software.

As an Associated IVS Analysis Center, IGG Vienna is mostly dealing with research tasks and not with operational (routine) VLBI processing, i.e., we often make modifications to the source code by introducing new models and algorithms. Our students are experts in Matlab but they are not experienced in writing Fortran source code, which makes the work very difficult and time consuming for them (and consequently also for the other staff at IGG). This is our main motivation to develop new VLBI software (VieVS) in Matlab.

Another reason for developing new VLBI software is that right now we need to carry out several updates of our present VLBI software package Occam 6.1, and this task would be much easier in Matlab than in Fortran. In particular, we want to make the software fully consistent with the most recent IERS Conventions (McCarthy and Petit, 2003) (e.g., non-rotating origin and the corresponding partials for the nutation parameters), and for clarity we will remove all former models which are no longer necessary.

At the IGG Vienna, we have been using the classical Gauss-Markov least-squares adjustment of Occam 6.1 LSM which is based on piecewise linear functions using rates for the representation of zenith wet delays, clocks, or Earth orientation parameters. These rates do not correspond to intervals between integer hours as their first epoch is arbitrary. With the new software VieVS, we estimate the paramters as piecewise linear offsets at integer hours (see below), which makes our results easier compatible with those from other space geodetic techniques like the Global Navigation Satellite Systems (GNSS) or Satellite Laser Ranging (SLR).

2 Why Matlab?

In many courses of their curriculum the students of the Vienna University of Technology and at many other universities use Matlab. Thus, new VLBI software in Matlab will attract more students to get interested in VLBI and to write their diploma or phd theses about VLBI related topics. The use of Matlab will ease the implemention of new ideas and models in the software (for them and their supervisors). Furthermore, Matlab has many built-in tools and functions which make the writing of the code much faster, and the source code is significantly shorter and more concise. Examples for built-in tools are:

- Matrix operations like matrix inversion and multiplication or matrix decomposition, but also easy addressing of columns/rows;
- Plots can be made easily and routines for graphical interfaces are available;
- Netcdf readers and writers;
- Structure arrays which are very useful to store scan-based information.

Of course, there are also arguments against using Matlab - the main of which is certainly that Matlab is a commercial software. However, many institutes worldwide have access to Matlab and use it. Alternatively, we can provide executables of VieVS which run on any machine (whithout having Matlab installed). Another possibility is the use of non-commercial counterparts of Matlab like Octave which will be tested in the next months. If possible, we will modify VieVS in a way that it can be run with Octave or other counterparts.

The other argument against Matlab is that it is slower than Fortran or C/C++. This is certainly true, but tests showed that this is not critical in our case. Even if VieVS takes twice as long as e.g. the Fortran-based Occam, we think that this is fast enough for most of the research purposes, but also for the determination of global solutions.

3 Concept

3.1 Occam to VieVS

We did not start from scratch when developing VieVS, but we heavily rely on the Occam 6.1 LSM software package. Occam LSM consists of five parts: *dtau0*, *pn*, *station*, *geomet*, *lsm*. At first, we made a 'line by line' transition from Fortran to Matlab to guarantee that we get identical results. In a second step we deleted all obsolete parts and comments, and we applied Matlab tools to shorten the source code considerably. Finally, we re-structured the code and the results were still identical to those of Occam. E.g., we connected *pn*, *station*, and *geomet* into one part called *vie_mod*. *dtau0* is now called *vie_init* and *lsm* becomes *vie_lsm*.

One important change with *vie_lsm* is that we are using piecewise linear offsets for all parameters which can be estimated. These piecewise linear offsets are estimated at integer hours (e.g., at 18 UTC, 19 UTC, ...), at integer fractions of integer hours (e.g., 18:20 UTC, 18:40 UTC, ...) or at integer multiples of integer hours (e.g. 18:00 UTC, 0:00 UTC, 6:00 UTC, ...). This representation is not only possible for troposphere zenith delays and gradients, station clocks, or Earth orientation parameters, but also for stations coordinates.

As an example Equation (1) shows the troposphere delay L at one station represented by piecewise linear offsets x_1 and x_2 of the zenith delays at the integer hours t_1 and t_2. $m(t)$ denotes the mapping function at the epoch t of the observation which is in between the integer hours. The partial derivatives which have to be entered in the design matrix are shown in Equation (2) and (3). This concept is similar for all parameters, and future combinations (at the normal equation level) with other space geodetic techniques will be easier with this kind of parameterization.

$$L(t) = m(t) \cdot x_1 + m(t) \cdot \frac{t - t_1}{t_2 - t_1} (x_2 - x_1) \quad (1)$$

$$\frac{dL}{dx_1} = m(t) - m(t) \cdot \frac{t - t_1}{t_2 - t_1} \quad (2)$$

$$\frac{dL}{dx_2} = m(t) \cdot \frac{t - t_1}{t_2 - t_1} \tag{3}$$

As already mentioned, we will make VieVS fully compatible with the IERS Conventions (McCarthy and Petit, 2003), which is not the case with the Occam 6.1.

3.2 Related tasks

The Vienna VLBI Software (VieVS) comes at an appropriate time, because it is strongly related to many activities within the VLBI community and at IGG Vienna:

- **IVS WG4**: IVS Working Group 4 (chaired by John Gipson) is developing new VLBI data structures. We plan to fully incorporate the new format in VieVS and we also will contribute to the definition of the format. With the new Matlab software VieVS we are very flexible to consider any new development. E.g., if the new data structures will be based on netcdf, built-in Matlab tools will be available to read and write the data.

- **VLBI2010**: IGG Vienna has been contributing to the simulation studies for VLBI2010 (Petrachenko et al., 2009). We plan to continue these simulations with VieVS, and we will take advantage of the experience that we gained with our Occam simulations. E.g., we can use the generators of the turbulent delays and the clock values. In addition to the classical Gauss-Markov least-squares model *vie_lsm*, we plan to set up a Kalman Filter solution in VieVS as well.

- **SCHED2010**: We work on new scheduling software for geodetic VLBI which will be closely related to VieVS. In particular, we plan to use the simulation capabilities of VieVS to determine and validate optimized schedules.

A schematic workflow of VieVS and the related tasks are shown in Figure (1).

4 Co-operation between IGG and NICT

There will be close co-operation between the IGG Vienna and the National Center of Information and Communications Technology (NICT). Although NICT works on its own space geodetic software (for GNSS, SLR, and VLBI) which is based on Python and bindings to C/C++, both organizations will share their experience gained with the development of each software package.

Additionally, IGG Vienna can profit from NICT's experience concerning group delay ambiguity resolution and phase solutions. Presently, VieVS is using the 'NGS cards' as input files, but it is planned to also cover earlier steps in the processing chain when the new data structures are defined by the IVS WG4.

There are plans and ideas to equip VieVS with satellite tracking and space VLBI capabilities. In this respect, the co-operation between NICT and IGG will also be very beneficial and important.

5 Conclusions and outlook

The IGG Vienna is developing new VLBI software (VieVS) in Matlab. Presently, it has about the same capabilities as Occam 6.1, but by using Matlab the source code is more concise. VieVS will be improved and extended in many aspects,

- it will be equipped with satellite tracking and space VLBI capabilities (in co-operation with NICT),
- steps 'backwards' in the processing chain will be covered, such as group delay ambiguity resolution,
- it will get its own features to set up global solutions.

The Vienna VLBI Software (VieVS) will be made freely available to get feedback from as many groups as possible.

Acknowledgements Hana Spicakova is grateful to Mondi Austria Privatstiftung for financial support during her phd study at TU Vienna. Andrea Pany is recipient of a DOC-fForte fellowship of the Austrian Academy of Sciences at the Institute of Geodesy and Geophysics, Vienna University of Technology. Joerg Wresnik is grateful to the Austrian Science Fund (FWF) for supporting his work within project P21049, and Tobias Nilsson to the Deutsche Forschungsgemeinschaft (DGF) (project SCHUH 1103/3-1).

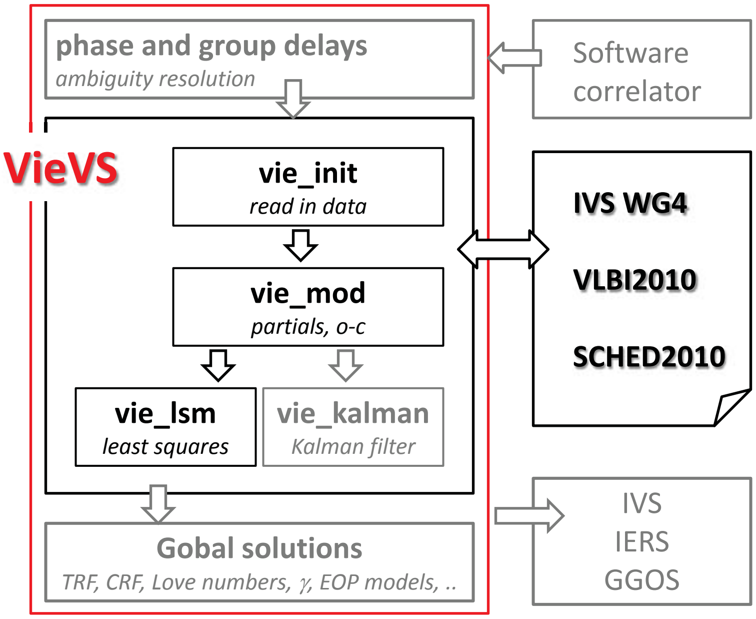

Figure 1. Schematic workflow for the Vienna VLBI Software VieVS. See the text for further details. Grey colours indicate that these tasks have not been started yet.

References

Titov O., Tesmer V. & Boehm J., 2004, OCCAM v.6.0 Software for VLBI Data Analysis, in: Vandenberg N.R. & Baver K. (eds.) International VLBI Service for Geodesy and Astrometry 2004 General Meeting Proceedings, NASA/CP-2004-212255, pp 267-271

Petrachenko B., Niell A., Behrend D., Corey B., Boehm J., Charlot P., Collioud A., Gipson J., Haas R., Hobiger T., Koyama Y., MacMillan D., Malkin Z., Nilsson T., Pany A., Tuccari G., Whitney A. & Wresnik J., 2009, Design Aspects of the VLBI2010 System, Progress Report of the IVS VLBI2010 Committee, in: Behrend D. & Baver K. (eds.) IVS Annual Report 2009

McCarthy D.D. & Petit G., 2003, IERS Conventions, IERS Technical Note 32