Mathematical platform for studies on VPM and Buehlmann decompression algorithms

B. Kuch[1)], R. Bedini[2)], G. Buttazzo[1)], A. Sieber[2)3)]

1) *Scuola Superiore Sant Anna, Pisa, Italy*

2) CNR Institute of Clinical Physiology, Pisa, Italy

3) *SZS Research and Solutions, Italy*

Abstract:

The present paper describes a mathematical platform for studies on decompression algorithms. It is scalable and provieds a tool set for deep analysis of the Buehlmann ZHL-16 decompression algorithm and the Varying Permeability Model. It is implemented in MatLab.

Introduction:

Nowadays there exist a wide range of decompression algorithms, which basically can be divided into Haldane models like Workman [1] or Buehlmann [2] and bubble models like the Varying Permeability Model [3] (VPM) or the Reduced Gradient Bubble Model [4](RGBM).

Haldane decompression models use hypothetical tissue compartments with a corresponding half-time to simulate the human bodies inert gas absorbion and release throughout the dive. The Schreiner equation calculates the gas loadings for a descent/ascent. The Haldane equation is used to calcualted the ongasing during the bottom time. M-values for every tissue compartment describe the supersaturation tension at a given depth via a linear equation. By comparing the tissue saturation with the corresponding M-value, a decompression schedule can be generated. M-values were found empirically [5][6].

The VPM uses the same dissolved gas tissue model like Haldane models. Instead of using static empirical M-values, VPM dynamically computes its 'M-values' dependent on the specific characteristics of the dive profile, the VPM parameter, and their combined effect on the Yount bubble [7][8][9].

The present paper describes a mathematical platform developed under Matlab. It is envisaged for studies on the Buehlmann ZHL-16 decompression algorithm and the VPM. This platform is open source, scalable, provides a tool set for deep analysis of present decompression algorithms and functionalities for the validation of improved, optimized or new decompression algorithms.

Methods:

The platform supports the Buehlmann ZHL-16 decompression algorithm and the Varying Permeability Model. Both algorithms are in general straight translations of Erik Baker's original FORTRAN code into MatLab R2008a [MathWorks].

MatLab is a numerical computing environment and programming language. MatLab as programming language has the benefit, that single functionalities can be executed in an isolated way or in arbitrary order. Intermediate results, for example the tissue saturation at a certain point in time, can be visualized easily in 2D or 3D diagrams without the effort of writing additional code. Timings, bottlenecks and dependencies can be measured or figured

out by available libraries.

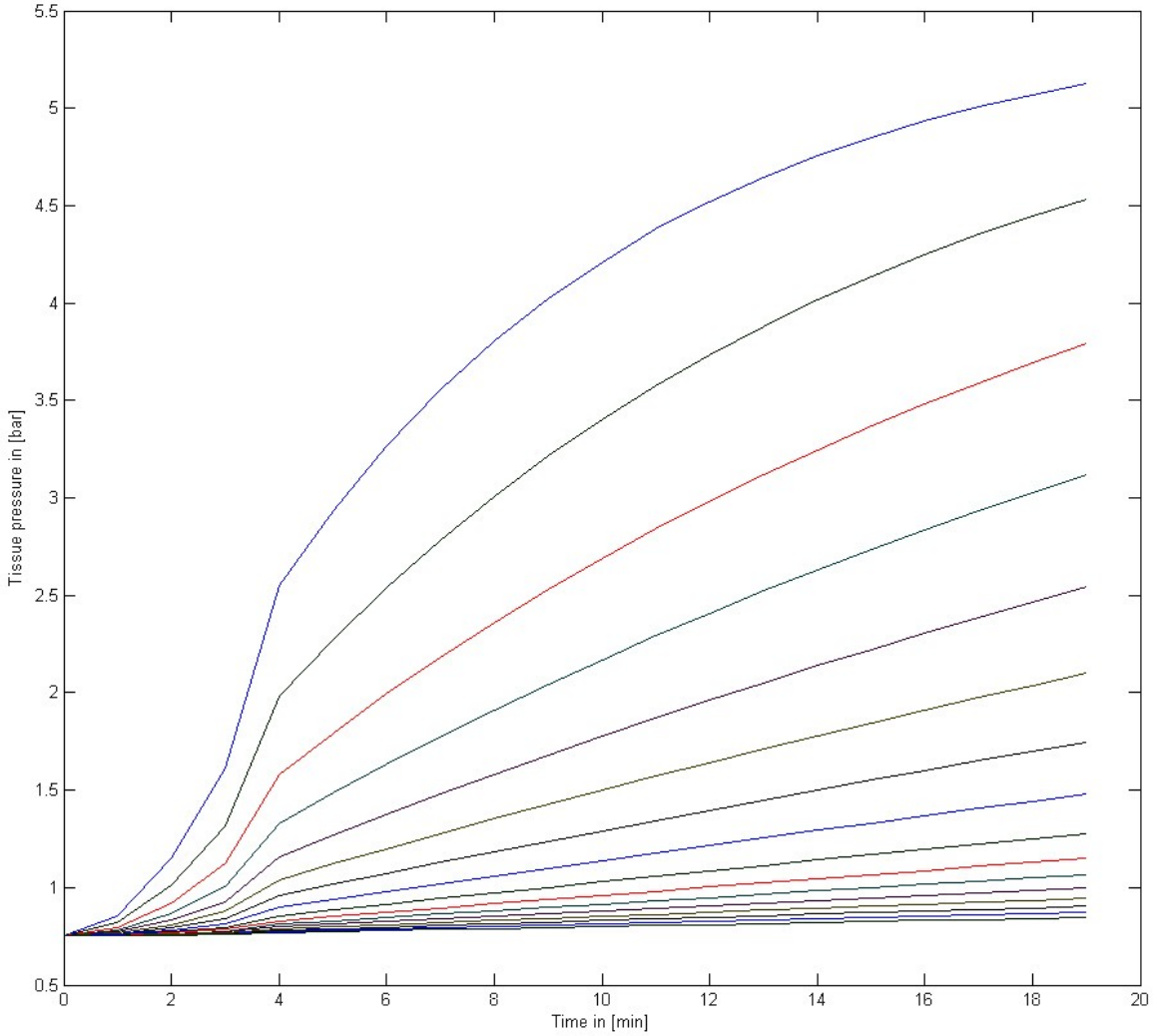

Figure 1: Tissue saturation of nitrogen after 19min bottom time at 60m – Gas: AIR / descent rate 20m/min

Both Buehlmann ZHL-16 and VPM support various levels of conservatism to provide a basis for wide studies. To increase conservatism to the ZHL-16, Erik Baker's Gradient Factors are included [10]. The conservatism level of the VPM is supplied by manipulating the initial critical bubble radius. Furthermore the VPM algorithm uses Boyle's Law to increase the bubble radius during ascent (generally known as VPM-B) to provide additional conservatism [11].

Both algorithms support arbitrary kinds of nitrogen and helium mixes. All constants of both algorithms (like Buehlmann's halftimes, A and B values or VPM's bubble skin compression value, bubble regeneration time, etc.) can be modified.

Results:

The platform was developed under MatLab. It supports Buehlmann ZHL-16 as Haldane decompression algorithm and VPM-B as bubble model.

The decompression algorithms were validated against several state of the art decompression planners like V-Planner or Freeplanner. V-Planner is a proprietary decompression planning

software and state of the art for technical divers, who prefer to decompress according to VPM or Buehlmann. Freeplanner is the Freeware solution of V-Planner. It is in principal a GUI application (graphical user interface) which controls Baker's Fortran shell applications [12]. Both algorithms deliver equal results like the reference software.

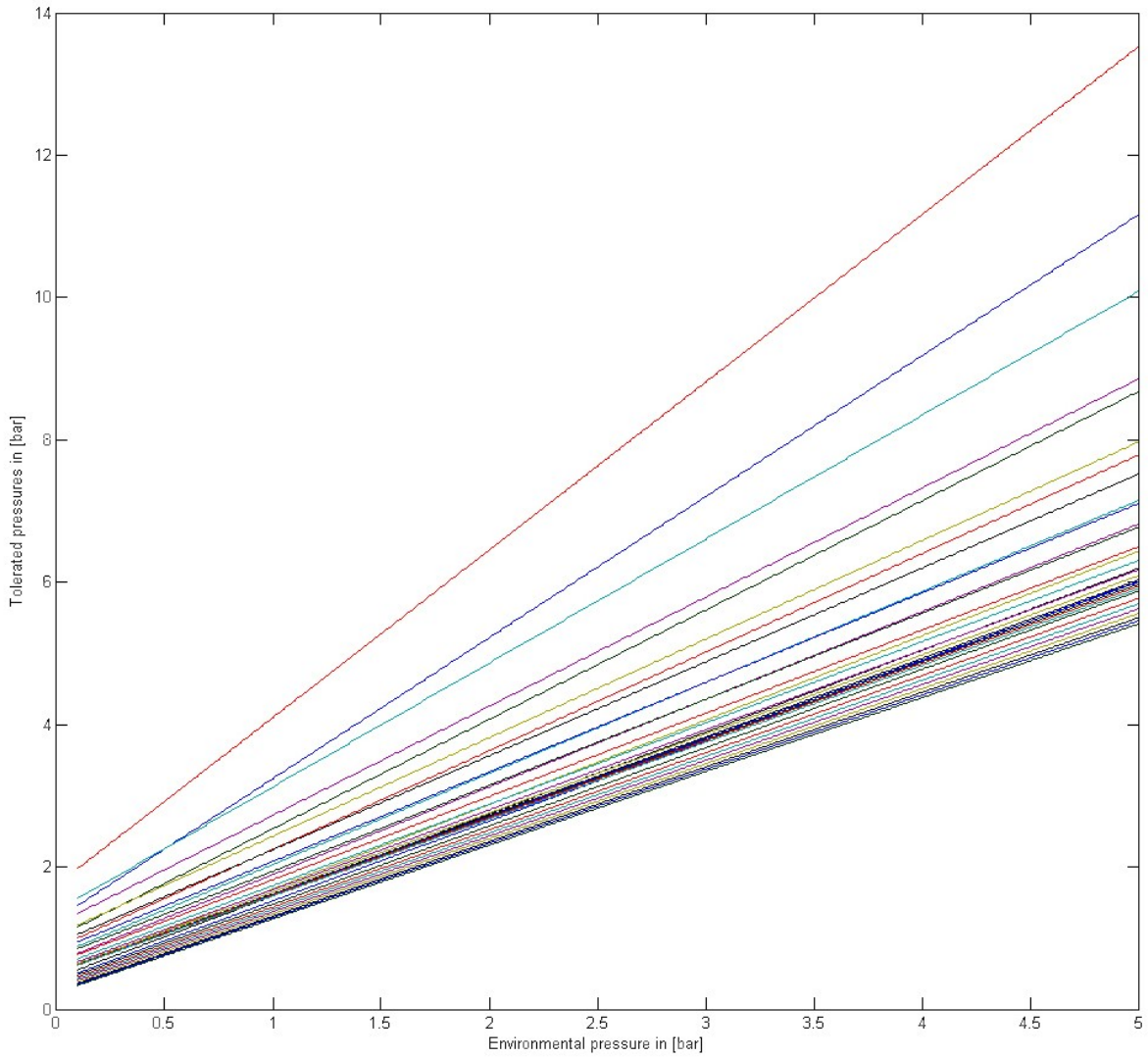

Figure 2: Tolerarated tissue pressure (nitrogen and helium) vs. environmental pressure of Buehlmann ZHL-16 without Gradient Factors

Run time comparisons and 2D/3D plots of arbitrary data can be performed easily within this platform.

Figure 1 shows the tissue saturation of nitrogen in dependance of time for a 60m diver with air and a descent rate of 20m/min. Figure 2 illustrates the maximum tolerated tissure pressure for nitrogen and helium in dependance of depth (environmental pressure) according to ZHL-16.

Discussion:

Buehlmann's ZHL-16 algorithm and the Varying Permeability Model are well documented and information about these decompression algorithms can be found easily. There exists a wide range of proprietary and free (also open source) decompression applications to compute

arbitrary decompression schedules. But for research / deep analysis purposes, these implementations are not suitable. Of course it is possible to expand these applications. But this is very time-consuming in state of the art programming languages like Java or C++. MatLab indeed provides innately a tool set for analysis and simulation.

Disadvantage of the presented MatLab platform is, that it requires at least basic knowledge in MatLab to be able to use it. It has no graphical user interface (yet) and everything has to be changed inside the code. Thus the present platform is not envisaged to be used by recreational or technical divers as substitution for a decompression planner.

Additionally the presented platform should not be seen a final product. Rather it should provide a first stage of a research platform in decompression theory, which is available for everyone and everyone can further develop and expand it.

Conclusion:

The introduced platform provides a MatLab platform to simulate arbitrary dive profiles. This platform is open source, scalable and provides a tool set for analysis and simulation. Intermediate and final results of the computed decompression schedules can be visualized in 2D and 3D diagrams to provide a basis for further studies.

Further work:

The source code of the platform will be published under the GNU General Public License (GPL) at SourceForge and possibly on the EUBS database so that the whole diving community can use and further develop it.

References:

[1] Workman, R.D. (1965). Calculation of Decompression Schedules for Nitrogen-Oxygen and Helium-Oxygen Dives. Interim Report Research Report 6-65, U.S. Navy Experimental Diving Unit, Washington, D.C., USA.

[2] Buehlmann, A. A. (1995) Tauchmedizin. Springer-Verlag. ISBN 3540555811.

[3] Baker, E.C., Maiken, E.B., Yount, D.E. (1999). Implications of the Varying Permeability Model for Reverse Dive Profiles. Reverse Dive Profiles Workshop. Smithsonian Institution, Washington, D.C.

[4] Bruce Wienke. Reduced Gradient Bubble Model. International Journal of Biomedical Computing, 26:237–256, 1990.

[5] Baker, E.C. (2001). Some Introductory “Lessons” About Dissolved Gas Decompression Modeling. ftp.decompression.org

[6] Baker, Erik C.: *Understanding M-values,* Immersed – International Technical Diving Magazine, Vol. 3, No. 3, Fall 1998

[7] Yount, D.E., Hoffman, D.C. (1986). On the use of a bubble formation model to calculate diving tables. Aviat. Space Environ. Med. 57:149-156.

[8] Yount, D.E., Hoffman, D.C. (1983). Decompression theory: a dynamic critical-volume hypothesis. In: Bachrach A.J., Matzen, M.M. Underwater physiology VIII: Proceedings of the eighth symposium on underwater physiology. Undersea Medical Society, Bethesda, 131-146.

[9] Baker, E.C. VPM: Solving for radius in the impermeable regime.

ftp.decompression.org

[10] Baker, Erik C.: *Clearing up the Confusion about Deep Stops*, in Immersed - International Technical Diving Magazine, Vol. 3, No. 4, Winter 1998

[11] Baker, E.C. (2002). VPM-B Program Update Explanation. ftp.decompression.org

[12] Baker, E.C. (2001) VPM Fortran Source Code. ftp.decompression.org