

# DIY DECOMPRESSION

By Stuart Morrison

(c) 2000 Stuart Morrison/Liquid Technology

**The author accepts no responsibility for any errors contained within this article or injuries or illness arising from using the information contained here, or any applications to which it is put.**

To many divers, the decompression information generated by dive computers, tables and PC programs are mysterious and the products of either Einstein-like intelligence or a mysterious cabalistic form of alchemy. The reality is that they are simply the results of some straightforward equations.

The theory behind decompression modelling is very complex, but the calculations themselves are in practice within the realms of every diver. Decompression modelling is not representing what is going on within the body, it is simply a mathematical function which spits out figures which do not kill us. The protocol used to calculate decompression data is known as the algorithm, and there are many base types.

## **What is decompression?**

We cannot breathe underwater so we need to take a breathing gas with us. Oxygen is essential to us, but too much will cause serious physiological problems, too little will not support life – it is a very fine balance and all technical divers should understand that. Oxygen needs to be "diluted" with other gases which are inert ( i.e. not required by any physiological process, and not causing any toxic effects. ) Nitrogen is the easiest choice (air/nitrox) but to avoid narcosis then helium is added too (trimix). Underwater we breathe our gas mixes at pressures higher than atmospheric pressure which causes a certain amount of the inert gas(es) to dissolve into the body's tissues – the longer and/or deeper the dive then the more inert gas which will dissolve. Eventually saturation is reached – the point where no more gas will dissolve regardless of bottom time.

When we return to the surface, the pressure is reduced and the tissues can no longer hold the same amount of dissolved gas. The gas is released and the rate at which this happened controls whether or not the diver suffers from decompression illness (the bends). All a decompression model does is tell the diver how to ascend to the surface.

This article explains DIY decompression theory using the Buhlmann ZH-L16 model, and it is assumed that the diver will be familiar with partial pressures; the differences between saturation, desaturation and supersaturation; haldanean decompression models and half-times; tissue compartments; exponential decay and basic diving physiology. It is recommended that a good text book such as "Deeper Into Diving" by John Lippman is referred to.

## The Basics

The Buhlmann ZH-L16 model is the most commonly used because it is the only algorithm which has been freely published with all the needed information in one volume. It is also very well tested, and appears to be reasonably reliable.

This article assumes that the calculations will be done by a computer program and I have included some simple routines which will form the basis of a program. I have written the routines in simple English instructions which should be easy to follow for the non-programmer, and simple to convert to a programming language. These are all based on notes which I have used to write decompression software in both C and Visual Basic.

There should be one unit used throughout. It is important to think in terms of pressure rather than distance. For example, 10msw depth does not mean ten metres distance from the surface but the weight of a ten metre water column. It is important that all pressures are expressed in the same unit, either bar or msw. One atmosphere of pressure can be either 10msw or 1bar. Similarly the partial pressure of nitrogen in air on the surface is either 7.9msw or 0.79bar. Use one unit or the other, do not switch between them.

Water vapour in the lungs plays an important role. Air is 79% nitrogen and 21% oxygen, and at sea level the partial pressure of nitrogen is 0.79bar. You would assume that in the lungs this would be the same but because of the high humidity there the partial pressure of nitrogen is less. We need to correct for water vapour and its partial pressure ( $P_w$ ) must be subtracted from ambient pressure ( $P_a$ ):

$$\begin{aligned}ppN_2 &= FN_2 \times ( P_a - P_w ) \\ppN_2 &= 0.79 \times ( 1 - 0.0567 ) \\ppN_2 &= 0.745 \text{ bar}\end{aligned}$$

If we are diving at sea level then  $P_a = 1$ . If we are at altitude then  $P_a$  will be much less and depends on the height above sea level.

The basic concept is that the body is represented by sixteen tissue compartments, each with the half-times as shown in Table 1. These tissues absorb and release gas throughout the dive. None of the tissues have any connection to any of the others and cannot be affected by any other.

## Step 1 – Initialise the Model

The tissue model needs to be initialised. At the surface, breathing air, and not having dived for a while, every tissue in the model will be stable and have the same gas loading. In all the following examples, the tissue gas loading will be represented by Pn (nitrogen loading), Ph (helium loading) and Pt (total gas loading):

$$P_n = 0.79 \times (1 - 0.0567) = 0.745 \text{ bar}$$

As there is no helium yet:

$$P_h = 0$$

So the total is:

$$\begin{aligned} P_t &= P_n + P_h \\ P_t &= 0.745 + 0 = 0.745 \text{ bar} \end{aligned}$$

This means that all 16 tissues start off with a loading of 0.745bar in them. A common mistake is to forget to initialise the tissues. The program routine would be along the lines of:

```
Initialise_Tissues:
```

```
  For All 16 Tissues
```

```
    Pn = 0.79 * (ATM - WATERVAPOUR)
    Ph = 0
    Pt = Pn + Ph
```

where ATM = atmospheric pressure (bar) depending on height above sea level, WATERVAPOUR is the water vapour pressure. For programming I maintain two arrays of sixteen variables, one array for nitrogen and the other for helium.

## Step 2 – The Descent

The next thing to do is to get the depth of the dive. You can assume an instant descent to keep things simple but on deep dives there can be quite a lot of ongassing during depth changes. A normal descent rate is around 30m/min. There is an equation known as the Schreiner equation which calculates the gas loadings for a descent:

$$P = P_{io} + R(t - 1/k) - [P_{io} - P_o - (R/k)]e^{-kt}$$

where

$P_{io}$  = initial ambient pressure minus water vapor pressure  
 $P_o$  = initial compartment inert gas pressure  
 $R$  = rate of ascent/descent times the fraction of inert gas  
 $t$  = time interval  
 $k$  = half-time constant =  $\ln 2$ /half-time  
 $e$  = base of natural logarithms

You would need to do this for all sixteen tissues:

Descent\_Gas\_Loadings:

$t = \text{Depth} / \text{DESCENT\_RATE}$

For Each Tissue Loading:

```
;; Nitrogen first
Po = Pn [this is the current loading at the start]
Pio = (Depth - WATERVAPOUR) * fN2 [ppN2 in lungs]
R = DESCENT_RATE * fN2
k = log(2) / [nitrogen half-time of this tissue]
Pn = Pio + R*(t - (1/k)) - (Pio - Po - (R / k))* exp(-k*t)
;; Helium next
Po = Ph [this is the current loading at the start]
Pio = (Depth - WATERVAPOUR) * fHe [ppHe in lungs]
R = DESCENT_RATE * fHe
k = log(2) / [helium half-time of this tissue]
Pn = Pio + R*(t - (1/k)) - (Pio - Po - (R / k))* exp(-k*t)
```

where  $f_{N2}$  and  $f_{He}$  are the fractions of nitrogen and helium ( 79% = 0.79, 25% = 0.25, etc. ) Note that all descent rates are positive whereas all ascent rates are negative.

### Step 3 – The Dive

Next thing to do is to get the information for the actual portion of the dive spent on the bottom. You need the depth, the bottom time and the gas mix. The above Schreiner equation can be used and the term 'R' can be set to zero and 't' becomes the bottom time. Or you can use a simplified version known as the "Instantaneous Equation":

$$P = P_o + (P_i - P_o)(1 - 2^{-(t/\text{half-time})})$$

where:

P = compartment inert gas pressure (final)  
P<sub>o</sub> = initial compartment inert gas pressure  
P<sub>i</sub> = partial pressure of inspired inert gas  
t = time (of exposure or interval)

Again, you need to go through every tissue compartment and load it:

Dive\_Loadings:

t = [bottom time of the dive]

For Each Tissue Loading:

```
;; Nitrogen loadings first
Pi = (Depth - WATERVAPOUR) * fN2
Po = Pn [current gas loading in this compartment]
P = Po + (Pi - Po)*(1 - 2^(-t / [N2 half-time for this
tissue]))
Pn = P
;; Helium loadings first
Pi = (Depth - WATERVAPOUR) * fHe
Po = Ph [current gas loading in this compartment]
P = Po + (Pi - Po)*(1 - 2^(-t / [He half-time for this
tissue]))
Ph = P
```

For multi-level dives you simply repeat the entire process for every stage of the dive, ensuring that the ascent ceiling is not violated at any stage.

### Step 3b – Checking the Ascent Ceiling

The ascent ceiling is the shallowest depth to which a diver can ascend without serious risk of developing DCI. The ascent ceiling also represents the depth of the first decompressions stop ( normally rounded down to 3m intervals ) and shows if the dive is a no-decompression dive ( when the ascent ceiling is zero or negative. )

The ascent ceiling is calculated by using the a/b coefficients. Other Haldanean models use m-values which represents the maximum tissue loading for a specific depth which will not produce DCI in that tissue. I find a/b coefficients easier to work with. Each tissue has its own pair of a/b-values, nitrogen and helium. Table 2 lists them. Set A is the original set not intended for practical use, set B is recommended for software and table generation and set C is intended for incorporation into dive computers. The following equation is used to calculate the ascent ceiling for each tissue:

$$\text{Ceiling} = [(P_n + P_h) - a] * b$$

[ceiling is given as an absolute pressure e.g. 2.3bar = 13msw]

where 'P<sub>n</sub>' is the current nitrogen loading and 'P<sub>h</sub>' is the current helium loading. Each tissue will have a different ascent ceiling and the deepest ceiling is the one which should not be violated.

There are a/b coefficients for both helium and nitrogen so which one do you use? For a nitrox dive you use nitrogen, for a heliox dive you use helium, but what about trimix? You can either use the nitrogen set which will give a more conservative schedule or you can interpolate the value based on the proportions of each gas load in each tissue.

For example, if a tissue has 2.65bar of nitrogen in it and 1.34bar of helium then the total gas loading is 3.99bar. Nitrogen makes up 66% of the total loading and helium makes up 34% of it. So you would take 66% of the nitrogen a or b value and 34% of the helium a/b value:

$$a = [a(n_2) \times P_n + a(\text{he}) \times P_h] / (P_n + P_h)$$

$$b = [b(n_2) \times P_n + b(\text{he}) \times P_h] / (P_n + P_h)$$

where 'a(n<sub>2</sub>)' & 'b(n<sub>2</sub>)' are the a/b values for nitrogen and 'a(he)' & 'b(he)' are the a/b values for helium.

To calculate the ceiling:

Ascent\_Ceiling:

For Each Tissue:

```
;; Calculate the a/b values
A = ((an2 * Pn) + (ahe * Ph)) / (Pn + Ph)
B = ((bn2 * Pn) + (bhe * Ph)) / (Pn + Ph)
Ascent_Ceiling[This Tissue] = ((Pn+Ph) - A) * B
```

Find the deepest ascent ceiling of the 16 values

Critical Ascent Ceiling = Highest Value

## Step 4 – Get the First Decompression Stop

The first decompression stop is simply the ascent ceiling rounded to the next multiple of 3m. For example if the ascent ceiling is 11m then the first stop is at 12m, if the ceiling is 19m then the first stop is 21m, etc.

To find the no decompression limits use the following process:

1. Initialise the model (Step 1)
2. Descent loadings (Step 2)
3. Get the planned depth and gas mix and load the tissues, using a bottom time of 1 minute (Step 3)
4. Check the ascent ceiling (Step 4)

Keep repeating Step 3 until the ascent ceiling becomes greater than zero (i.e a depth below the surface). The number of 1 minute intervals (iterations) which you have to do to make the ascent ceiling not zero is the no-deco limit for that depth and gas mix.

## Step 5 – Calculating the Length of Decompression Stops

In step 4 you get the first deco stop depth. Now you have to use that to calculate the stop times. Normally the stop depths range in 3m intervals (3m, 6m, 9m, 12m, etc.).

You make your current depth the stop depth. You need to repeat step 3 (using the current stop depth as the depth, 1 minute as the bottom time, and the deco gas for the gas mix info) until the ascent ceiling changes to the next stop depth. For example:

...your first stop is 12m, your deco gas is EAN50. You repeat step 3 using these figures (and 1 minute bottom times) until the ascent ceiling changes to 9m (the next stop). The number of iterations (i.e. the repetitions of step 3) indicates the stop length i.e. four iterations means a four minute stop. You then move up to 9m and repeat the process until the ascent ceiling changes to 6m. At 6m you repeat until the ceiling is 3m (or the surface if you want 6m as your last depth) and so on.

The program code would be something like this:

StopCalc:

```
Get FirstStop [As step 4]
LastStop = [3msw] [or whatever you like]
For Each Stop From FirstStop to LastStop in 3m steps
Get DecoGas [do this at every stop]
StopLength = 0
Until AscentCeiling = ThisStopDepth - [3msw]
Do Stage3(Depth=ThisStopDepth, BT = 1min, GasMix = DecoGas)
StopLength = StopLength + 1
```

When this loop has ended then the total stop time = StopLength  
Move up to the next stop depth and repeat until the surface is reached

The code simply loops through the 1min iterations until the next shallowest stop depth is reached. For a closed circuit rebreather you would need to calculate what gas would actually be in the loop based on the unit's setpoint.

That is basically it. The steps are as follows:

1. Initialise the model based on the altitude
2. Get the depth of the dive and the breathing mix, then calculate what the tissue loadings will be during the descent
3. Get the bottom time of the dive and calculate the loadings based on that.
4. If it is a repetitive dive then repeat (3) again for each stage
5. Calculate the first decompression stop depth
6. Get the deco gas and do 1min loops at the stop depth until the ascent ceiling is equal to or less than this stop depth - 3 or the surface ambient pressure. The number of loops equals the length of the stop
7. If you haven't reached the surface yet then repeat (6) but get a new deco gas and make the stop depth 3m shallower.
8. Print the schedule at the end
9. If you want a repetitive dive then you just treat it as though it is a dive breathing air for the length of the surface interval and at a depth equal to the ambient pressure at the surface. For the next dive you would miss out Step 1, the initialisation stage and just use the tissue loadings which you are left with at the end of the surface interval. Simple. The first deco software I wrote had about 150 lines of instructions and took two hours to write. Programming with fancy graphics for Windows takes considerably longer and the original version of XS took around three months to complete.

## **Safety Factors and Conservatism**

It is well accepted that the unmodified Buhlmann algorithm is not conservative enough for actual dive planning. There are a few ways of adding in safety factors.

Proplanner increases the inert gas content depending on the safety factor which you enter.

Older versions of the Abyss assumes that offgassing is slower than ongassing. It also modifies the compartment half-times based on data which the user supplies: fitness, experience, work rate, comfort levels, etc.

Zplan lengthens bottom times before it plugs them into the equations.

How well these work is debatable, and the common thread is that they alter the tissue loadings. Many people feel that this is not the best approach as it simply lengthens the shallowest stops. Any damage will have already occurred by the time you reach 3m or 6m. Starting deco deeper is felt to be the best way to adapt the Buhlmann algorithm.

Multilevel alters the a-coefficient. It's author, Juergen Bohnert feels this is a safer approach. He also incorporates a seventeen tissue model rather than sixteen which generates much deeper stops. The later versions of XS also incorporate his ZH-L17TS model.

The big talking point is the Gradient Factor method for modifying the a/b coefficients (or m-values) and is incorporated in GUE's Decoplanner, GAP, DDPlan and the latest version of XS. This is based on Erik Baker's article on deep stops which was published in Immersed magazine and is widely available on many diving web sites. Rather than using a fixed conservatism value, it uses two which adapt as the deco progresses. This should be incorporated without any other fudge factors or the schedule becomes infeasibly long.

Other than that, good luck with your DIY deco program!