Calculating deco schedule with VPM

> *This note is mainly addressed to divers and is intended to be simply an interface between the several introductory presentations of VPM and the explainations addressed to those with matematical background. As anyone who knows VPM can easily see I oversimplified nearly all the features and concepts to an extent sometimes I would call "extreme". But as far as I know there isn't any presentation trying to fill the gap between the bubble stabilization mechanisms and the way a VPM-based deco schedule is calculated. For this reason I hope that what follows will be useful for those interested.*

To understand the *critical volume algorithm* is to understand the way VPM is adapted to calculate deco schedule. The former is a single step among the several the latter is made of. While the classical approach (Haldane and its derivations like those of Bühlmann and Workman) is quite straightforward as for the logical sequence of steps, VPM is a bit less. So, to make clear VPM deco calculation algorhitm, one must understand each plug of the puzzle; once done, it is easy to tie everything toghether. The plugs are in essence the following two.

- the way a single bubble behaves;
- the way many bubbles behave. In other words, the concept of "distribution".

I shall explain the two points.

The first plug: understanding a single bubble

As it is known, according to VPM a bubble is stabilized by a surfactant monolayer (as for the adjective "monolayer" dr Kunkle would not agree, perhaps!). There is a lot of literature, presentations and articles on the net about this topic and I shall not deal with it too much. The point for us is this. We can imagine the surfactant monolayer as being "intelligent". It is capable to gather surfactant molecules from the immediate neightborhood of the outer side if the bubble grows and to expel molecules from the layer if the bubble shrinks, all this to accommodate changes in radius. As it is known, the monolayer opposes to the surface tension according to Laplace law I assume known. In the following, to make reading easy, I'll sometimes refer to "surface tension" to mean "the pressure due to surface tension". Turning to us, since the opposite forces are equal the bubble is stable. The forces involved are inversely proportional to the radius; in other words, if the bubble is tiny the forces are (relatively!) great and if the bubble is great the forces are small. This is a key point in understanding VPM deco calculation.
Once the bubble is stabilized at some pressure, it is notewhorty if not crucial to ask what happens if we relieve the pressure; well, even if we relieve the pressure by a small amount, the surfactant layer looses its packed structure. From a practical point of view, everything goes as if there would be no surfactant at all and we are left only with Laplace (better, with the pressure directed inside the bubble, due to surface tension).
Now, let us imagine a stable bubble in some medium with our system at equilibrium. The situation is that of fig. 1 in which the initial pressure/tension is the same everywhere. We have 60 *bar* tension (the pressure in the liquid surrounding

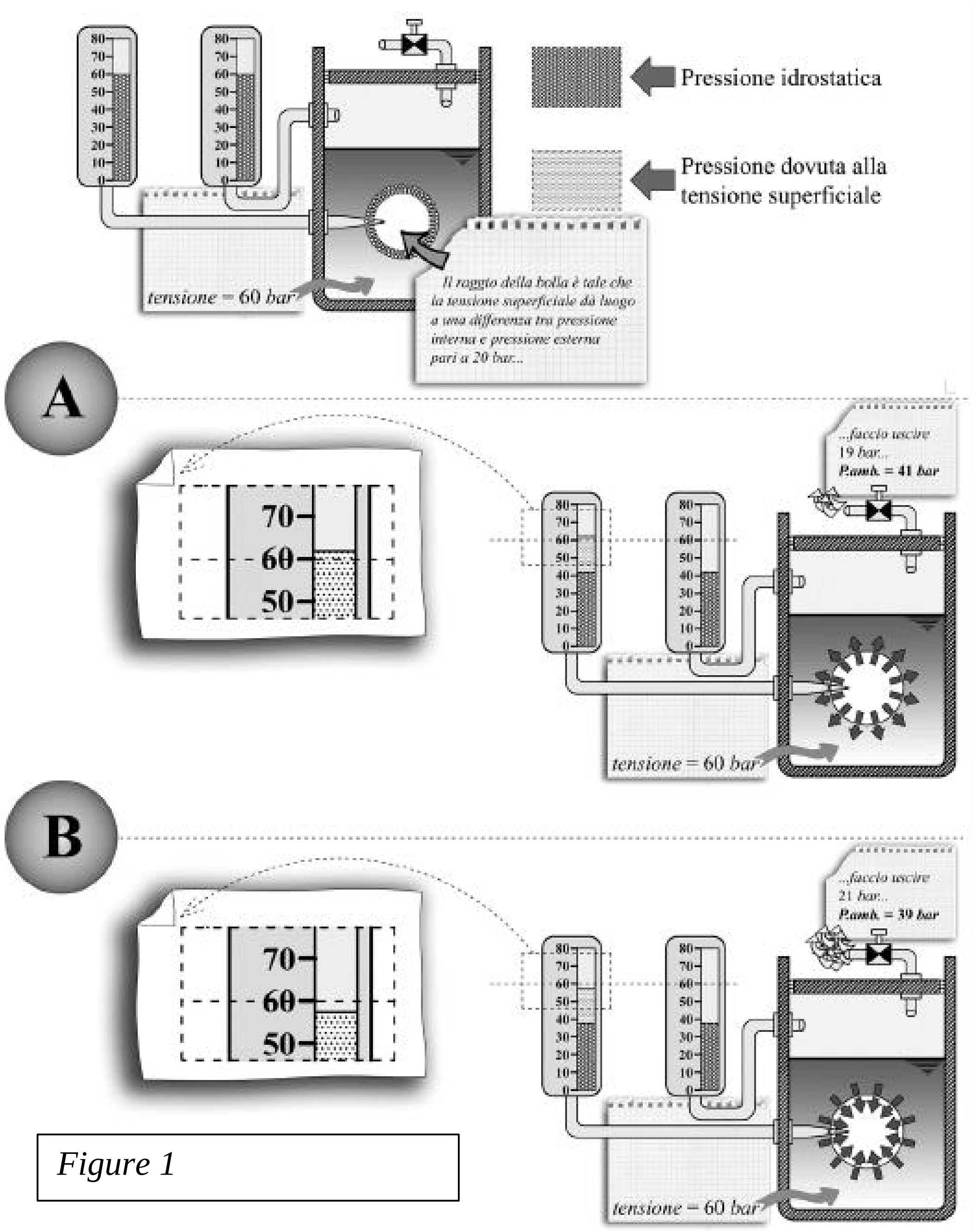
Pressione idrostatica
Pressione dovuta alla tensione superficiale
tensione = 60 bar
Il raggio della bolla è tale che la tensione superficiale dà luogo a una differenza tra pressione interna e pressione esterna pari a 20 bar...
A
...faccio uscire 19 bar...
P.amb. = 41 bar
70
60
50
tensione = 60 bar
B
...faccio uscire 21 bar...
P.amb. = 39 bar
70
60
50
tensione = 60 bar

Figure 1

the bubble), 60 *bar* gas pressure above the liquid and 60 *bar* inside the bubble. At equilibrium surface tension does not act since surfactant opposes, so we find 60 *bar* in the bubble also. Let now suppose that the bubble radius is such that if it were not the surfactant monolayer, surface tension would be 20 *bar* (60 and 20 *bar* don't have any specific meaning, are only for example). First, ***remember that surface tension sums up with the external pressure*** then let us go see what happens in a couple of ideal experiments. First, we relieve the gas pressure by 19 *bar*. This cancels the effetct of the surfactant, so surface tension acts alone. We find (60-19)=41 *bar* outside the bubble. Surface tension adds 20 *bar*, so we have (41+20)=61 *bar* inside the bubble. The tension is 60 *bar*, so the gradient is directed toward the extern of the bubble; in other words the bubble shrinks. Now, we make the same but we relieve the pressure by 21 *bar*. Exactly as before, pressure is (60-21)=39 *bar*, plus 20 of surface tension, total is 59 *bar* inside the bubble. This time, gradient is opposite and the bubble grows. With the values we used, it is intuitive that the critical pressure is 20 *bar*, exactly the value of the pressure due to surface tension. Now, we are ready for some conclusions.

1. By relieving the pressure by less than 20 *bar* (= by less then the surface tension) the bubble shrinks; by relieving the pressure by more than 20 *bar* (=by more then the surface tension) the bubble grows. If we eliminate the redundancies from the previous statement we get:
2. by relieving the pressure by less than the surface tension the bubble shrinks; by relieving the pressure by more than the surface tension the bubble grows. ***Now:***
3. ***since*** the bubble is stable and surface tension is low when a bubble is big and is high when a bubble is small, ***it follows that:***
4. ***to initiate the growth of a big stable bubble I have to relieve the hydrostatic pressure by a small amount; to initiate the growth of a little stable bubble I have to relieve the hydrostatic pressure by a large amount***.

Now, let us name “stable bubble” as “nucleus” and we are almost done for now. Our last step is depicted in fig. 2 in which we find 3 nuclei (black, left) of different size. It should be easy to understand what happens; as the difference between the bottom and the first stop grows, more and more little nuclei are said to be “activated” and grow into bubbles. *This is another key point in understanding VPM*, so it is important to think over the figure until it becomes clear.

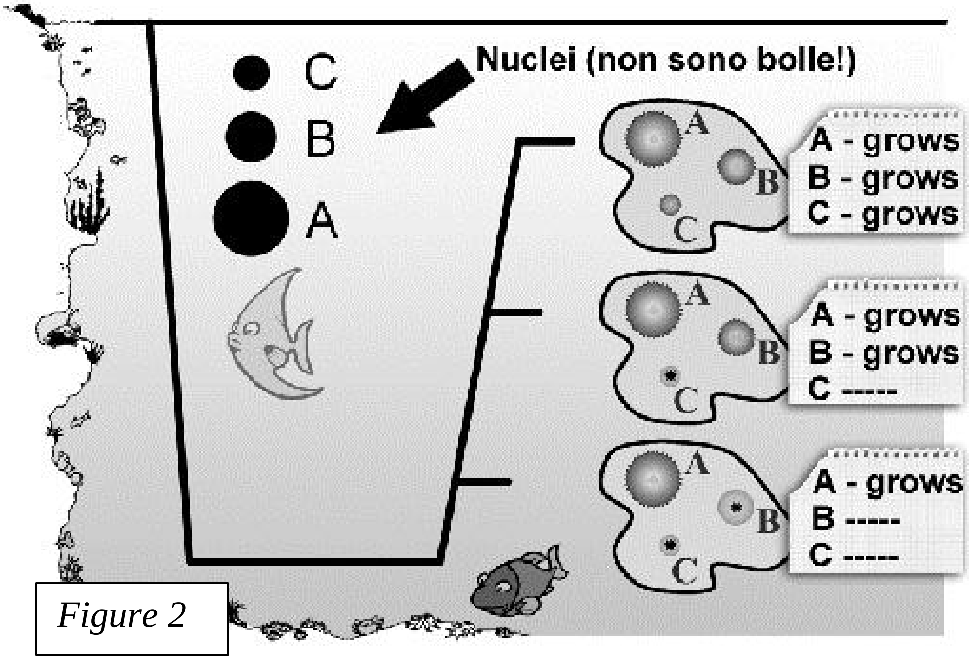

Figure 2

The second plug: understanding a population of bubbles

Few words about this point. At least in gel (scientists say “in vitro” from latin “in the glass” or better, in this context, “in a test tube”) a population of bubbles would appear something like in fig 3.

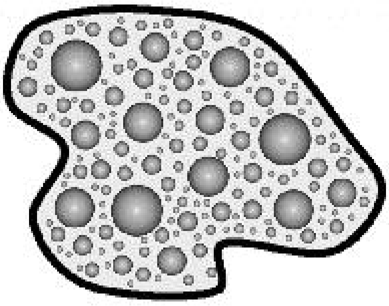

Figure 3

By eye inspection we can see from now that there are few big bubbles and more and more bubbles as the radius decreases. When dealing with these situations, it is customary to introduce the concept of *distribution* I am going to explain with the help of fig. 4. Let us start from the top, and imagine that we are dealing with pearls. One of the first things to do after fishing is to split the pearls according to their dimensions. There are several boxes each of them contains pearls of some radius plus or minus some tolerance. Referring to fig. 4 top, we can think the numbers on X-axis as inches (do not be tempted to steal the pearls!). So, for example, the box labeled 6’’ contains pearls whose radius is between 5.5 and 6.5’’, since the radii are spaced by 1’’ (see X-axis marks). Then, we count the pearls in each box and graph the results. Everything is quite straightforward for now. The next step is less intuitive; so we have to pay some attention, or, better, think it over and over if needed. For reasons that will be clearer soon, we want to know how many pearls are *greater or equal* than

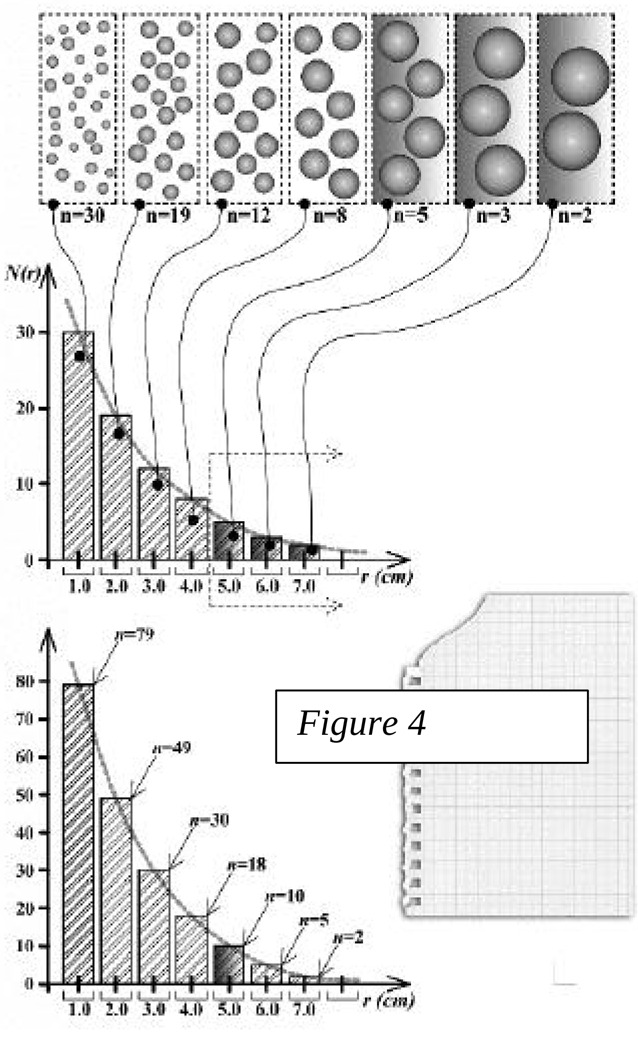

Figure 4

some value of radius, say 5’’. To find the answer, we sum up all the values starting from 5’’ forward. According to the figure, we have (5+3+2)=10 (see shaded and hatched zones). By doing the same for all the values, we get the graph of fig 4 bottom, to be read as follows: ***given some value X of the radius, Y is the number of items (of course, not necessarily pearls) whose radius is greater or equal to X***. Be sure to check the figure in some cases to understand how things go. Now, let us substitute bubbles for pearls and we are done again. Note that the figure at the bottom contains exactly the same information of that above; for VPM purposes it is more useful the last form (the one at the bottom) that is named *integral distribution*.

The “real” distribution of the bubbles is similar to that in fig. 4; the shape is exponential but we must beware of being mislead by gas exchange law, radioactive decay and so on. Exponential function is a direct consequence of a very basic kind of phisical “behaviour” that it is not necessary to understand in depth here. Exponential distribution has nothing to do with gas exchange, unless we fathom an extremely basic level of ideas into the hidden secrets of the phisical world.

Tying the plugs

Now let us consider fig. 5 that summarises all we have learned until now (nuclei are depicted as little stars). Consider a tissue with its exponentially distributed population of bubbles, and consider that – of course – what is valid for a bubble remains valid for all the bubbles in the population. The basic statement is this: as the supersaturation (=difference between bottom and first stop) grows, more and

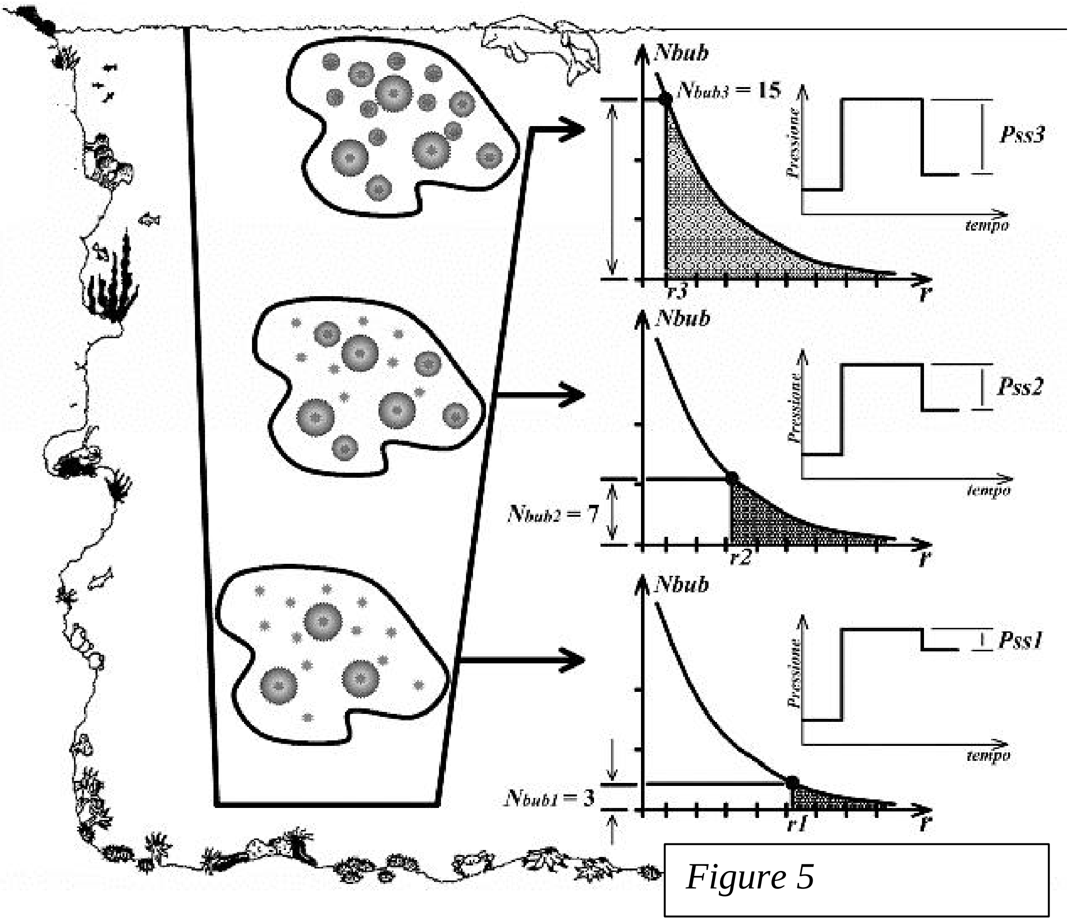

Figure 5

more nuclei grow into bubbles. Still referring to fig 5, we see that the number of nuclei initated into growth is directly found on the Y-axis ***if*** we use the integral distribution as we have done here. Count the number of bubbles for each of the 3 cases and check it in the graphs on the right; for purposes of understanding how the things go, it is very important to make such kinds of verifications.

Calculating the deco schedule

We are almost ready to understand how VPM can be used to calculate a deco schedule. There are several if not many assumptions and details behind this, but those important for us are the following:

1) the body is modeled with the “usual” compartments a-la-Haldane (exponential gas exchange);
2) the body tolerates a maximum amount of free gas (that in the bubbles, of course) we name from now on V_{gas};
3) the volume of free gas is the product of:
   a) **number of bubbles** (if with 100 bubbles I have 1 *cf* of gas, with 200 bubbles I have 2 *cf* of gas);
   b) **the time the supersaturation acts** (if a gradient of 1 *bar* acts for 1 *h* and produces 1 *cf* of gas, 1 *bar* for 2 *h* produces 2 *cf* of gas);
   c) **the gradient acting** (if a gradient of 1 *bar* acts for 1 *h* and produces 1 *cf* of gas, a gradient of 2 *bar* for 1 *hr* will produce 2 *cf* of gas);

I believe that due at least to the remarks in parentheses the assumptions be quite intuitive, or, at least I hope so. Firmly believing that I am right ☺ I shall not comment on them. Rather, let us follow the way the calculation proceeds.

The idea behind is quite simple. We start with some value for the gradient *G*, say 1 *bar*. But this starting point requires some explainations. The VPM deco schedule is calculated exactly the same way the classical method would do. In other words everything goes exactly as if we used for example Abyss software by setting all the *b* values equal to 1.0. We are left with the *a*’s being no more no less than the different gradients each tissue tolerates; since all *b*’s are equal to one, it means that the supersaturation tolerated by each compartment is depth independent. As for the classical method there is a lot of easily understandable literature on the net so, again, I shall not deal with this in detail. The important point is that at this level VPM works exactly as the classical method does. The (how great!) differences start from what follows.

In fig 6 we can see how the deco schedule calculation is performed. Basically, we can divide the process of re-equilibration with the environment in two parts, the first one in the water, the second one after surfacing. The dashed line is that of the true time course of the gradient *G*. Of course, *G* attains its maximum value at the beginning of each step, than decreases until the tissue tension allows for ascending to the next step where – by definition – we find *G* again and so on up the surface. To simplify calculations, VPM assumes that *G* is constant along the ascent and decreases exponentially after surfacing (continuous line). Even if we lack

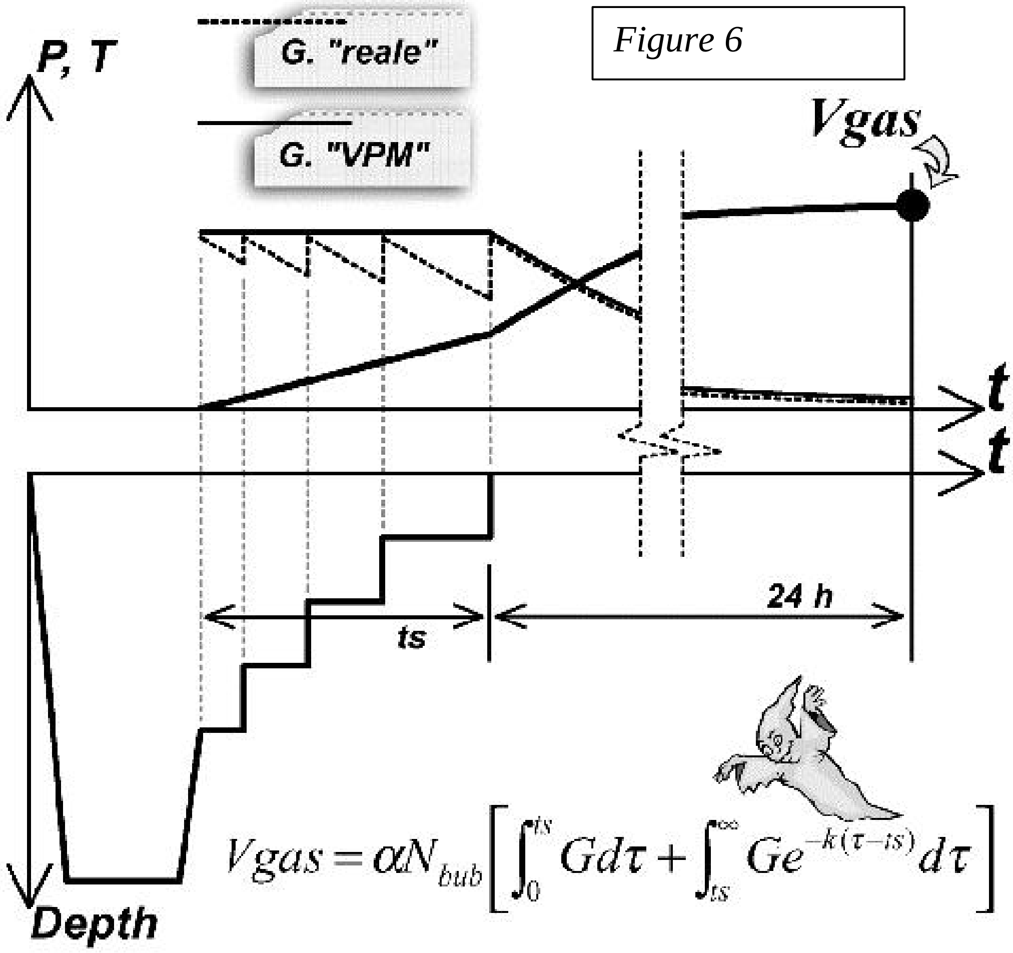
P, T
G. "reale"
G. "VPM"
Vgas
t
t
ts
24 h
Depth
$Vgas = \alpha N_{bub}\left[\int_0^{ts} G d\tau + \int_{ts}^{\infty} G e^{-k(\tau - ts)} d\tau\right]$

Figure 6

mathematical background we can easily calculate the free gas volume during the ascent in water. It is simply:

- the number of bubbles (more on this below!) by
- the ascent time by
- the gradient *G*,

because all these quantities are constant (and *G* is constant only because so we assumed). But as for the second part, the exponential decay of *G* introduces complications that can be overcome by some math; this happens because *G* is no longer constant along the time. And we can't assume it constant because – unlikely the first part underwater - we would introduce an unacceptable error (consider ascent time vs. 24/48 *h*). Math details are not too important here; it suffices to know that the calculation is possible (and quite easy for those who know calculus). We can see how the gas volume grows (thick continuous line) until the equilibrium is reached. BTW this is the reason V-planner wants to know something about the time we assume for the completion of the process; it does not allow for calculation of deco schedule until we choose 24 or 48 *h*. VPM must to "know" in advance the time for complete equilibration since unlikely classical methods, ***decompression still continues once surfaced and the gas phase volume depends upon the entire***

history, including that part outside the water. Gas phase continues growing after surface, as – for example - those bent well know.

Last, the *critical volume algorithm*

What happens with different values for *G*? Some interesting facts. Remember that *G* dictates:

- how many nuclei will become bubbles and…
- …that part of the total deco time we spend in water. This because as *G* decreases stops duration times increase, and the opposite holds.

It happens that since the total volume of free gas depends upon all the three factors, a straigthforward calculation is no longer possible. What to do? Nothing from our side because from this point on, everything becomes mainly a matter of math machinery. Neverthless, we can still understand the basic nature of the algorithm and how it works. Let us see.

1. We start with some small (but not casual, since it is calculated from a precise rationale) value for *G*;
2. Since *G* is known, we can calculate the deco schedule because all we need for doing this is *G* itself (along with mixes, profile, and so on, of course, all fixed factors with respect for this context); from the calculated schedule, we get also the total deco time underwater, say T_{deco}, by summing up the stop times;
3. Since we have *G* we have also the number of nuclei activated into growth, in other words, the number of bubbles N_{bub} (fig. 5);
4. Since we have *G*, T_{deco} and N_{bub}, we can calculate the total free gas volume V_{iter} (more about this "new name" below) that developes along (T_{deco}+24/48 *h*).

Is the above volume the "right one"? Not necessarily. Generally speaking the answer is obviously no. Rather, it should be remembered that assumption 2) holds; according to it there is a definite limit of V_{gas} we can tolerate. The value of V_{gas} would require some comments; for our purposes we can say that this is in some sense one of the "weak points" of VPM, but it suffices to say that V_{gas} is a fixed and known value (for software users, it is closely related to the parameter lambda).

So, we are – apparently – faced with a hard problem of reverse engineering. Starting with V_{gas}, the profile, and so on, we must to find that value of *G* such that the schedule calculated using *G* is "the right one", in other words the one that woud result in the value of V_{gas}. Of course we could accept a value less than V_{gas} but this means a too conservative deco. In VPM, V_{gas} plays the role of the various *a* and *b* of Bühlmann, the Workman *M* values and so on. But there is no need for reverse engineering. The nature of the algorithm helps us because the result V_{iter} (see above) embeds all the informations needed to find another and better value for *G* and start again the procedure from point 2. It is easy now to understand the reason VPM is said to be an *iterative* algorithm. What happens is depicted in fig. 7; for

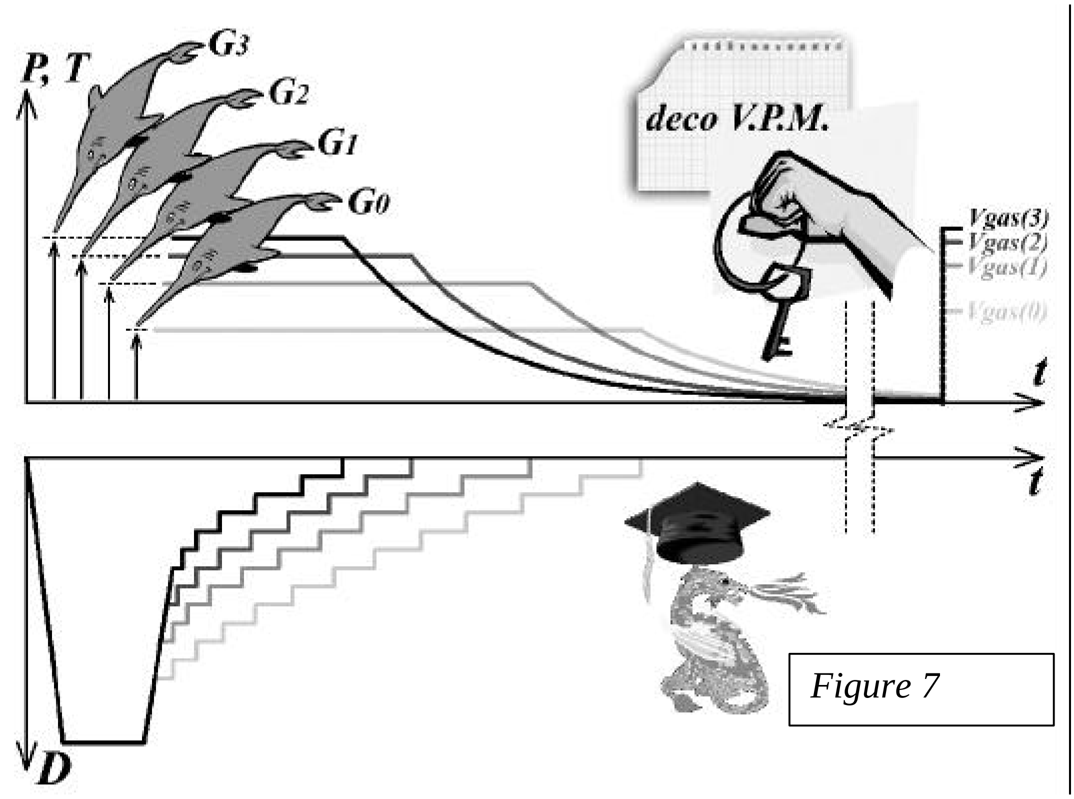

Figure 7

each iteration we can calculate a value of V_{iter} that approaches V_{gas} closer and closer (it is said *asymptotically*) and we say that the algorithm *converges*. Each iteration a new value of *G* is found and the entire calculation is made again. Note that as iterations proceed:

- the gradient *G* increases;
- More and more nuceli of decreasing radius are activated into growth (this last point is important also to understand how VPM deals with conservativism, see below).

In principle, this process is endless. In practice, after few iterations (typically 2-3) the difference from some value of V_{iter} and that of the previous step is little enough to allow stop the procedure; this is due also to the fact that stop times are rounded to the nearest integer upper value (and to some other reasons). BTW, in practice, as typical dives are concerned the algorithm starts so close to the correct value for *G* that often no iterations are needed. This is the reason V-planner (in this case, the VPM program by Erik working behind the scene) allows the user for choosing to use the critical volume algorythm or not. Unless we are planning very aggressive dives, usually the error between the "true" profile and the starting one is practically small enough to be neglected. In any case, since *G* grows along the iterations, if we choose not to use the critical volume algorithm, that means adding some safety factor to our profile. On the contrary, if - only for reasoning purposes - *G* decreased along the procedure, it would mean that the deco schedule sequence would be more and more conservative, and the critical volume algorithm wouldn't be optional but a more intrinsic component of the VPM algorithm. It should be an essential part of the calculation procedure to ensure profile safety.

Some other points

We are very close to that point beyond which we would be entraped in many technical details, so it is the time to stop. As a last remark and since the interest in VPM deco is growing, let us spend few words about the way VPM algorithm deals with conservativisms. Basically, everything is pivoted around a feature of VPM we have not yet dealt with. To allow for a better fitting of experimental data, Yount and Hoffman made another assumption, that there is a treshold V_{undef} such that the gas volume not exceeding V_{undef} is tolerated undefinitely (this accounts for the weeks needed for ascent from saturation, for example). As we can now understand V_{undef} is closely related to some corresponding value for G and the consequent number of bubbles. But in turn these are related to some value r_0 of the bubble radius. The gas in all the bubbles whose radius is greater or equal to r_0 is tolerated undefinitely and "does not play any role in the calculations" (note the quotation marks and forgive me if you are a VPM guru!). The situation is that in figs. 8 and 9; as we can see it is better to reason in terms of the bubble radius. The only gas "to be decompressed" is that in the gray zone between the radius corresponding to V_{gas} (left) and V_{undef} (right). In other words, we can think G as the sum of two G's, the first is fixed and accounts for V_{undef}, the second grows along the iterations and accounts for the gas we couldn't tolerate unless we wait the necessary time (in other words, the deco stops). It is easy to see that once fixed all the other parameters, the schedule is more and more aggressive as r_0 decreases, as one can check by playing a bit with V-planner for example. Now, once understood the concept, r_0 is nothing else than the "critical radius" found in the program (see for example, the advanced settings of V-planner).

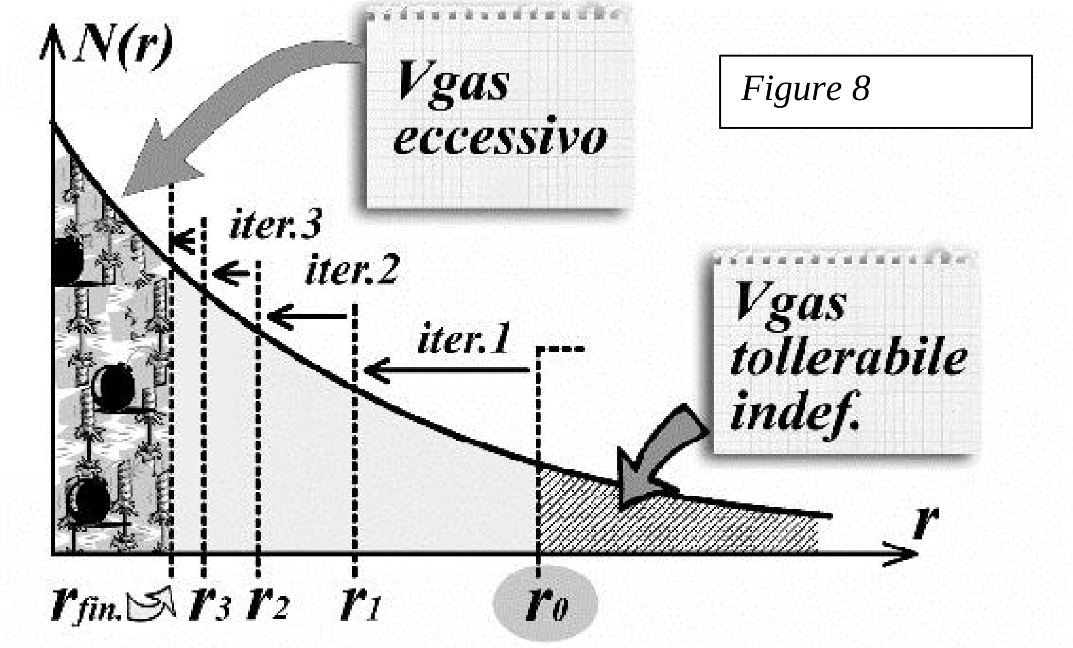

Figure 8

Even a brief explaination of the other parameters involved in the algorytm would be outside the scope of this note, so I refer the reader to the literature available. A few words for lambda (as I said, closely related to the maximum volume V_{gas}) and

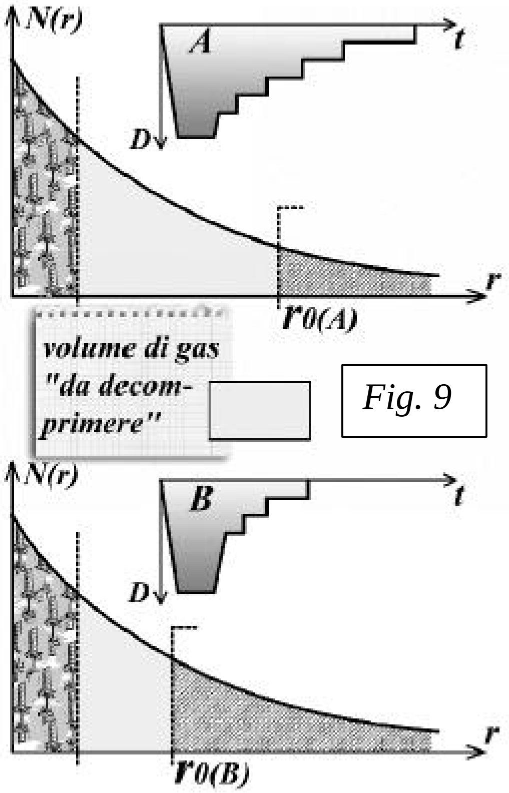

Fig. 9

for the so-called *bubble regeneration constant*. The latter accounts for the following phenomenon. Imagine we submit some bubble of some radius r_1 to compression to some pressure. As we know, bubble shrinks and through the expulsion of surfactant molecules the bubble finds its new equilibrium at some radius r_2, smaller than r_1. In general, all the bubble of the population in our sample will shrink (obeying to the so called *ordering hypotesis* the reader can find dealt with in the literature). It happens that due to complex phenomena pertaining to statistical physics, the original distribution is “recovered” after some time (of the order of weeks). The recovering time function is, once again, exponential. So, the regeneration constant resambles the compartiment halftime, the difference being that the halftime is now that of the exponential time course of the recovering. As we can expect from its range, this parameter becomes important as our dives span along days and weeks. But, as the reader can see by experimenting with the program and changing the parameters found in the “advanced setting” window, by far the most important parameter is r_0 since deco schedule is most sensitive to r_0 and much less to the other parameters. As a first approach to VPM understanding r_0 is the first thing to do.

Last but not least, it is very interesting to compare the gradient *G* according to the classical method and to VPM (fig. 10). The curves refer to a fast tissue and the values are those of the VPM program by Erik Baker; we can see the gradient resulting from the Bühlmann *a* and *b* (straight line) and the gradients of VPM. As we can expect, while the gradient is time independent for Bühlmann, as VPM is concerned *G* varies according to depth ***and*** bottom time of the dive; this is the reason because we find a family of curves rather than only one (in this sense, the Bühlmann straight line is a particular kind of curve).

It can be seen how much VPM is rich and interesting as for phenomenology and behaviour. Moreover, it is intrinsically free from the many haunting J-factors and other trimming points often, if not always, completely unrelated with the real word. This is not to say that VPM is a “perfect” model. But at least for its connection with the basic level of the involved phenomena, VPM is certainly a great step forward.

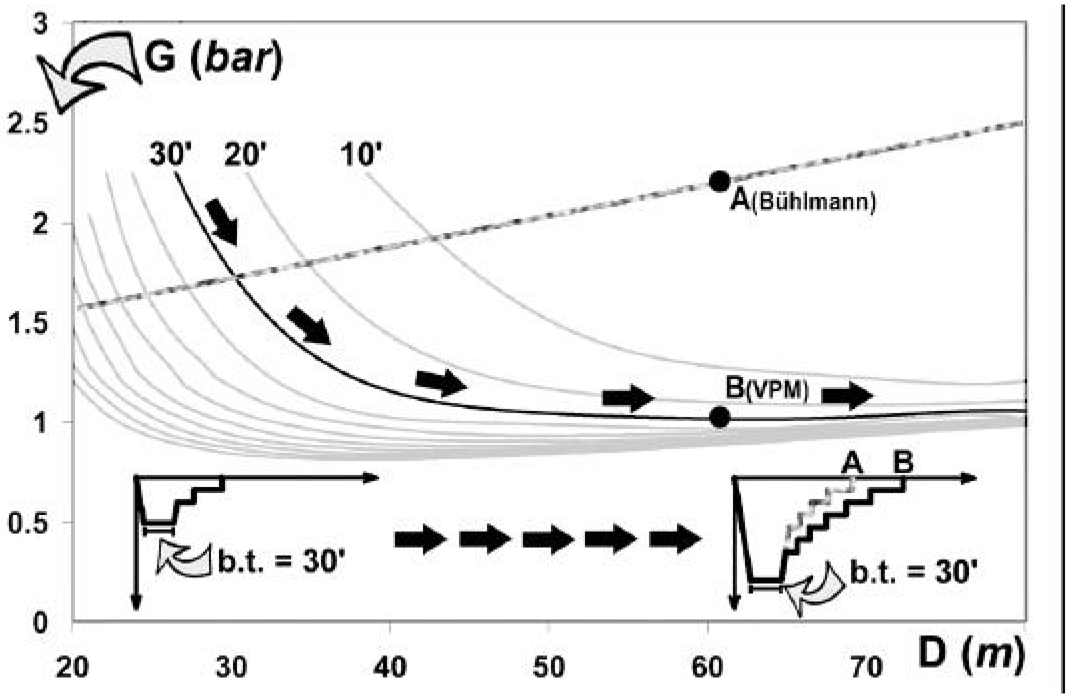
G (*bar*)
3
2.5
2
1.5
1
0.5
0
30'
20'
10'
A(Bühlmann)
B(VPM)
A
B
b.t. = 30'
b.t. = 30'
20
30
40
50
60
70
D (*m*)

Figure 10