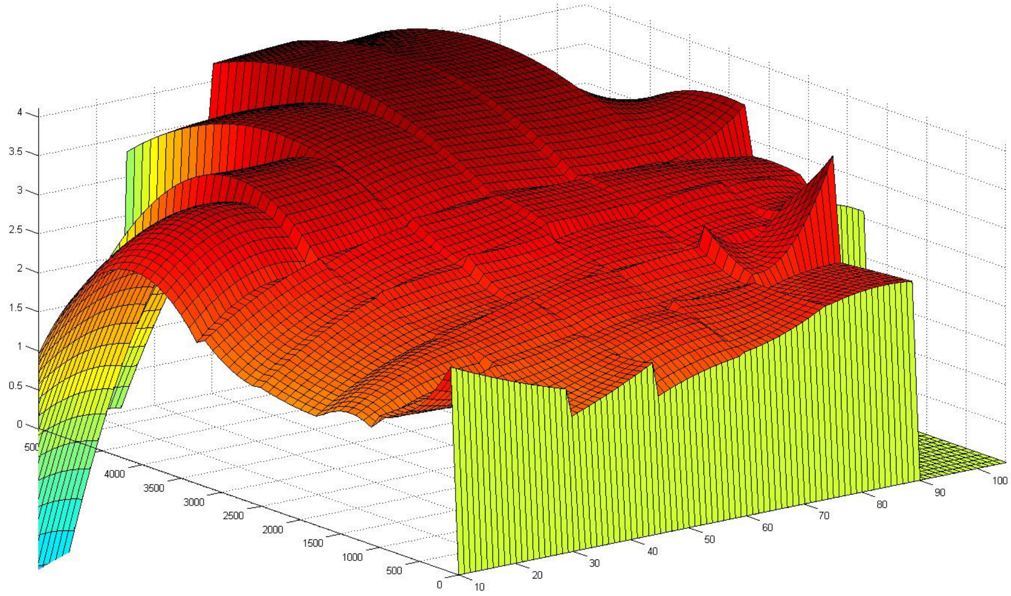


Parametric VE

Version 1.0

By Marcin Pohl, started June 2007, finished June 25, 2009

What is VE?

For years now, the definition of Volumetric Efficiency is the ratio of the mass of air pumped through the engine and the theoretical, potential mass of air you could fit into the engine. Traditionally, we looked at the VE table as an array of values between 0-100%, for varying values of MAP and RPM. With the arrival of E40 and newer PCM's, we got to see some new 'versions' of the VE table. Of course conceptually VE is the same, however the way how it's expressed has changed quite a bit. E40's VE table has some strangely high values, not particularly relating to anything we could easily identify or understand. However, the table seemed to behave the same way as the traditional VE table; increasing the value 10% seemed to estimate 10% more air mass, effectively yielding 10% more fuel delivered, thus giving us an impression of just some strangely scaled VE table. This is where most thinking about the new VE table stopped.

Even though the new VE (I call it GMVE as I have not seen it anywhere else but GM PCMs) behaves the same, looking at it had a profound impact on my understanding of its true nature, and the role of the VE table in the air mass estimation and fueling process. Looking at the unit of the new GMVE values made me wonder what it really is. Gram*Kelvin/kPascal is not what you would expect a traditionally dimensionless (unitless) measurement of Volumetric Efficiency. And then it hit me: GMVE is not a Volumetric Efficiency table—it is a normalized air mass table. What do I mean by 'normalized?' The unit explains it best; it is how much air mass you get at a particular temperature and pressure. If the value is 3, it means that at 1kPa of pressure and 1K the engine contains 3 grams of air. Or in a more realistic conditions of 100kPa and 300K the GMVE of 3 will be 3*100/300=1g of airmass. This is a really neat and elegant way of dealing with it. Instead of expressing VE as a ratio of real and theoretical air mass, it characterizes the flow characteristics of the engine, abstracting of actual conditions.

Current state of VE tuning

AFR is by definition a ratio of air to fuel. This means that to know airflow, we need to measure AFR and know the fuel mass. Or you can think about it as more of a cause and effect dependency, and think that the wrong amount of predicted air mass dictated a wrong amount of fuel, causing AFR from a wideband to a different than the commanded AFR. With such mindset, it only makes sense to adjust the table that affects the air mass calculations the most, which everyone agreed to be the VE table.

Currently, we measure AFR using a wideband, and assuming the injectors flow exactly as much fuel as their tables describe, it would yield the exact mass of air digested by the engine. This approach, while theoretically correct, has few practical complications. For example, how do we know that the injectors flow exactly as much we have them configured in the PCM, and continue doing so under all conditions from deep deceleration to high RPM WOT runs?

This approach also forces a tuner to think that all changes in airmass are caused by the changes in the volumetric efficiency (the real aspect of an engine, not a table in software). The Ideal Gas Law says that pressure, temperature and mass affect each other in a closed system. VE multiplied out with a cylinder volume will give us the actual volume of the gas in a cylinder. Pressure and temperature will determine the air density. Volume and density is mass. So the airmass is a function of VE, cylinder volume, pressure, and temperature, but we are only adjusting VE. So what happens if the air mass increase was really caused by temperature or pressure changes? How do we account for these? Why should VE change after the sunset or in the winter? This is what I call a 'problem of proper attribution' and it's absolutely crucial to understand why I consider the current VE tuning techniques to be useful only on Hawaii, a high-end dyno cell, or any other environment with minimal weather changes. This is exactly the reason why you can tune your car on one occasion and have it be perfect, and few days later, after some temperature or pressure changes that very same tune results in a scan with a lot of noise and discrepancies.

The new VE non-table

The new E38/E67 PCMs showed up, and to everyone's amazement, there were no VE tables! Instead, we gained in a whole bunch of tables of various coefficients. The numbers were meaningless, and no one quite knew what to make of them.

It turns out that GM is preparing their PCM's for the next generation of engines. The VE table works great, but it assumes fixed displacement and breathing characteristics. Most new cars come with butterflies in exhaust, variable cam lift and timing, variable intake runner lengths, etc. All this variability will result in changing the breathing characteristics, and the changes can occur within what we used to think of as one VE cell. While it would be possible to extend the VE table into more dimensions, it will quickly fall victim to 'the curse of dimensionality.' Even with just one more dimension, let's say Intake Cam duration, you will need a fully calibrated table for every different Intake Cam duration. So if you think tuning a single VE table was a tedious process, how about tuning 50 of them for 50 different intake cam durations? How about adding another variable (ie. Exhaust Cam duration), and having another 50 settings for that one? This would create 2500 VE tables. It only gets worse from then on.

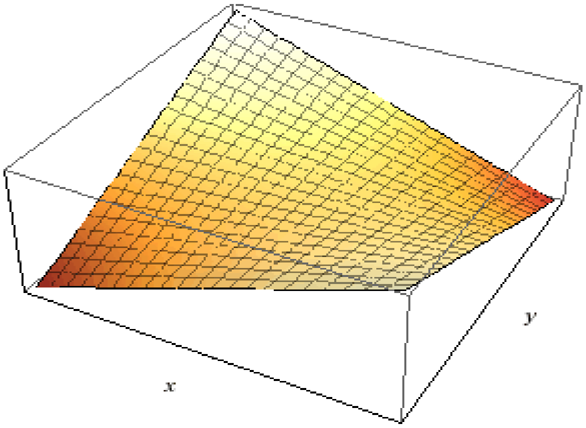
Response Surfaces

As you saw in the previous paragraph, the typical setup very quickly runs out of ability to express VE when facing more than two variables. Obviously a new way of expressing the breathing ability had to be highly resistant to dealing with higher numbers of dimensions. At the same time it must be able to be calibrated without going through explicitly calibrating a VE table for each and every possible combination.

Response Surfaces are a mathematical trick to describe dependencies that are not fully understood, or for which we do not have a clear-cut physics based model. If you look at a nicely smooth stock surface of the stock VE table, you can see there is a particular shape to it. At the same time the shape is nowhere near simple enough to be expressed with a formula. If we had an equation for this surface, instead of finding numbers for every single separate VE table cell, we would need to find the coefficients describing the function, and values could be calculated for any MAP-RPM tuple. No more dealing with various table resolutions, interpolations, searching for an average for a cell, which is really not just an average but a range of values from one corner of each VE cell to another.

Another neat aspect of Response Surfaces stems from their math background. They can be easily expanded into higher number of dimensions. Even if we cannot visualize it, the math checks out just fine.

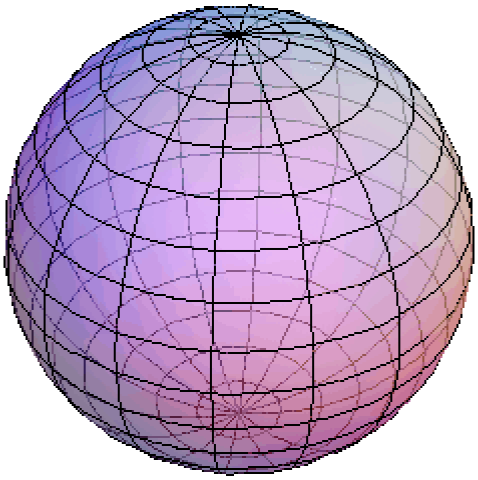
Let's take a look at the simplest surface—a plane.



A formula for a plane in 3D is:

Ax + By + Cz = D or

A more complicated surface would be:



(x-A)^2+(y-B)^2+(z-C)^2=D this one creates a sphere

You could go higher in dimensions, add **w** and **v** and you can have a 5D sphere. It is not easy to imagine, but the math for it works flawlessly; such a sphere can have volume and area just like the common 3D version we can easily imagine. This is where abstracting away from regular geometry based shapes and into math gains flexibility. The good part is that everything that works on 3D will work on 16D if need be, and it can be verified with the easy 3D model. Making it work for the higher dimensions just uses more numbers, but the mathematical procedures are the same for any number of dimensions.

What the equations mentioned above have in common is that they have a very similar form. It is a summation of terms (x,y,z…), some of them linear, some of them squared.

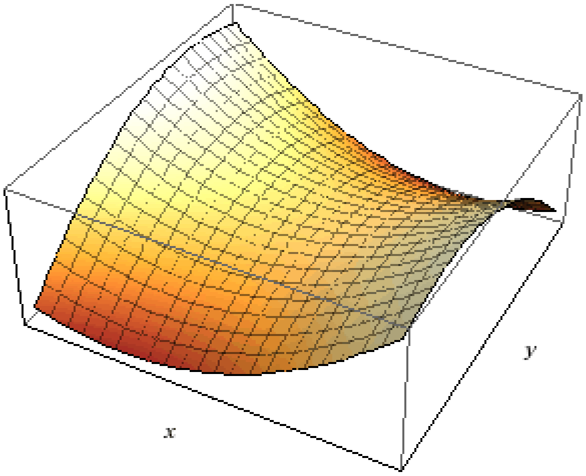
Response Surfaces describe relationships that cannot be described in any other way. If the shape of a VE table was a part of a sphere, you could describe it as:

(RPM-A)^2+(MAP-B)^2+(GMVE-C)^2=D

If this was precise enough, instead of trying to hit all the cells in the VE table, you'd have to find enough points to fit this surface to come up with the coefficients A, B, C, and D. Once you had these, you could calculate the VE value for *any* RPM and MAP pair, without any limits on the VE table, as it is not needed. Suddenly the 6800RPM limitation on E40's VE table would be irrelevant—if you built a motor to spin to 9000rpm, you just plug that 9000RPM into the equation, and out come the numbers.

Of course, the whole VE table does not lie on a sphere, and you should not try to approximate it as such, unless you are more into this car math magic than I am. However, these simple examples are needed to be understood and conceptually familiar to proceed with what the GM did to the VE table. '07 Z07 corvette has 30 zones, and each zone is described with one set of parameters. So instead of having one complicated function, you have 30 zones described by 30 simple, generic functions described by six terms with six coefficients each.

This is a picture of one full response surface to give you an idea of what kind of shapes it can take on:



These generic functions describing arbitrary surfaces are called Response Surfaces. A Response Surface can take on any shape that follows the formula:

A+B*MAP^2+C*MAP+D*MAP*RPM+E*RPM+F*RPM^2=0

The great part is that we can describe *any* number of 'cells' with it, and all it takes is the six coefficients (A, B, C, D, E, F).

There are four types of terms:

1. Constants (A) not dependent on any variables of the model
2. Linear (C, E) dependent on the linear (^1) terms of the model (RPM, MAP)
3. Quadratic (B, F) dependent on the quadratic (^2) terms of the model (MAP^2, RPM^2)
4. Interactive (D) dependent on the combinations between the terms of the model. (MAP*RPM)

Summing all the terms together results in the formula above.

Just like before with the 5D sphere example, we can expand the Response Surface into higher dimensions, let's add Intake and Exhaust Cam variables and see how many terms we will get for the each type of terms, just like we did above:

1. Constants: (A) still just one constant, if we had more they would just add up to one constant
2. Linear: (B,C,D,E) one for each variable, and we have 4 now (RPM, MAP, INT, EXH)
3. Quadratic: (F, G, H, I) one for each quadratic version of a variable (RPM^2, MAP^2, INT^2, EXH^2)
4. Interactive: (J, K, L, M, N, O) six total, for each combination of terms (RPM*MAP,RPM*INT,RPM*EXH, MAP*INT,MAP*EXH, INT*EXH)

Then you add them all up and end up with 15 terms.

A+B*RPM+C*MAP+D*INT+E*EXH+F*RPM^2+G*MAP^2+H*INT^2+I*EXH^2+…

…+J*RPM*MAP+K*RPM*INT+L*RPM*EXH+M*MAP*INT+N*MAP*EXH+O*INT*EXH

Ok, so that's bigger than the six coefficients in the two variable case, but it is still a vastly superior approach than separate calibrations of 2500 VE tables, especially if you consider that will take care of a describing a 5 dimensional airmass model. I am not providing a picture of a 5D surface as it's something only to be experienced during a 'Laser Floyd' show.

Cons and Pros of Parametric VE

The cons are simple: it is a lot of numbers, coefficients that do not mean anything until you put them together. With VE you look at it, you see 90, and you know it flows better than when it was 80. In this case, 2.3 is not comparable to 3452 because they are just one of many coefficients for different terms. Finding these terms is another story. You would think that finding 16 coefficients is difficult, but conceptually it is no different than finding 6 of them. The arrays of numbers and vectors of coefficients just get slightly longer, but the procedure is exactly the same. Another big problem with the Parametric VE is that it's simply a lot of rather esoteric math put together. None of it is particularly hard (squares and additions, basic vector multiplication) but all together it's rather scary to operate on, as it is not something you can easily double check with intuition or graphing.

There are many pros to using Response Surfaces. The fact that you can model *arbitrary number of variables* just by making longer vectors of coefficients is amazing. Record all the variables needed, estimate airflow, throw it into a solver, and you just tuned your VE coefficients without even seeing what the final shape built from the new coefficients looks like, and regardless of number of dimensions involved. If you need to tune with more variables, the same procedure applies, you just have more numbers to crunch.

Infinite resolution is another great aspect of using Response Surfaces. No more interpolating between 4 corners and hoping it is close enough. Each surface has an exact value for *every* point on it. What used to be 'between 4000 and 4400rpm and 70 and 75kPa I have 70% VE, but at lower RPMs you had only 62% VE, while at the same RPM but lower MAP you had 69% VE-so does it mean that exactly at 4400RPM-70kPa I should have 66% or 69%? Interpolation is a viable option, but extending it to five dimensions creates more problems than it solves. With Response Surfaces, you always get a precise answer, regardless of number of dimensions involved.

No more smoothing as surfaces are naturally smooth. No more 'rocky mountains' VE tables causing hesitation, bucking, knock and spikes in AFR. The only problem I see here is how does each of the surfaces 'connects' to another? With a set of bad sample of data, or data gathered only for a small area of a particular zone, the two surfaces might not 'touch' exactly along on the edges.

However, this could be easily remedied, by *changing the zone boundaries* on MAP and RPM. This could use used creatively—if you have a big cube supercharged motor that does not spin above 6000rpm but it does not need to either, you could adjust the zones so they cover the area you use your engine, instead of trying to tune some 7000RPM+ areas that the car will never experience. But it can be used the other way too: if you have a big stall in a high rpm drag only car, then have one zone for getting off the trailer and into the pits, and cram the all the other zones into where the action happens—5000RPM and up. All you need to do is to rerun the math on the latest data, and you'll get new surfaces shaped for the new zones. This is the very essence of tuning—adapting your car to do well what you want it to do, refusing to go along with the factory 'one size fits all' approach.

No more limits. E40 has a max value of 4.095 g*K/kPa. While this is OK for even big cube NA applications, FI setups can max it out easily. With VE values calculated on the fly, there is no limit. Well, that is not completely true; eventually it has to be stored in processor's register, but because the final number is stored only temporarily instead of saving entire matrixes of them, you can allocate a one large variable to ‘current air mass’ and have a very large range or precision, depending on what you need.

2 and 3bar *enhancements should be trivial to implement*. Since there is no need to store more numbers, it will not take any more memory. All you have to do is put in a 2 or 3 bar MAP sensor, and calibrate it. Then adjust your MAP boundaries for the different zones. It is now possible to adapt it to the boost levels you need; if you run up to 140kPa MAP, there’s no need to waste resolution on what would be the rest of the 2bar table (145-210kPa).