



The Solver Story

An Autobiography

INTRODUCTION

Many years ago Jack Champaigne asked me if I could write a computer program that would determine peening intensity quantitatively. This was to be an alternative to the subjective manual methods then being normally employed. No contract or payment was to be involved. The exercise was to be a purely academic one. Having recently retired from full-time employment at Coventry University, I had some time on my hands and so decided to take up the challenge. After all my experience with computer programs for X-ray residual stress analysis I thought that it shouldn't involve much effort—big mistake!

There are two essential elements that need to be considered:

- (1) Data points and
- (2) An appropriate curve-fitting equation.

Every measured data point throughout the Universe involves some variability, however small. Data values are affected by relevant variables. The effect of each variable can normally be expressed by an appropriate equation. This article is concerned with the way in which peening intensity can be derived by finding and using the parameters of an equation fitted to a set of data points.

DATA POINTS

Peening intensity data points involve two variables:

- (1) Measured deflection from flatness of an Almen strip and
- (2) The time of exposure to a shot stream.

Because only two variables are involved, we can easily plot peening intensity data points graphically as deflection against time—giving us the so-called 'Saturation Curve'. Normally between 4 and 6 data points are produced. One problem to consider, however, is whether or not both deflection and peening time suffer variability. I knew that deflection for any given exposure to a shot stream suffers variability but what about the corresponding peening time? This question gave me some food for thought. What might cause a variation in the time of exposure to a given shot stream? One reason is variability in the diameter of the shot stream itself as it strikes the Almen strip. The following is an explanation of this effect.

Relationship between shot stream diameter and time of exposure

Consider first an analogy of a train of length L traversing at a constant rate between two markers, A and B, that are L apart as illustrated schematically in fig.1. If the markers were 40 meters apart and the train was travelling at, say, 40 meters per second then it would take one second to pass both markers. If, however, the train was twice as long, $2L$, then it would take two seconds to pass both markers. It follows that the time to pass both markers is directly proportional to the length of the train.

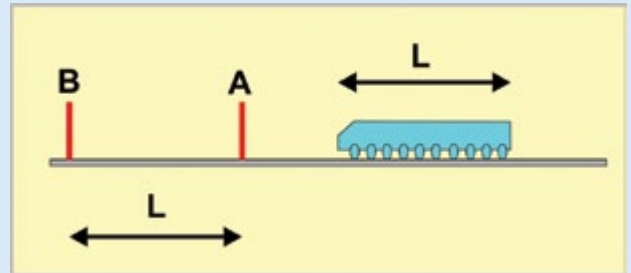


Fig.1. Representation of a train having to travel between markers.

Consider next a circular-section shot stream traversing along an Almen strip. The time of exposure of the strip must be directly proportional to the diameter of the stream's cross-section. Stream diameter variability is analogous to variability of train length.

I am not aware of any published quantitative evidence of stream diameter variability. Some variability must, however, occur although it is probably less than that of deflection variability.

CURVE-FITTING BY 'LEAST-SQUARES'

My approach was to apply the 'least-squares' technique that I was quite familiar with and which is built into the universally available Microsoft Excel program. This technique minimizes the squares of the differences between the data point values and the selected model equation. The following is an explanation of this effect.

Least-squares Curve-fitting

'Least-squares' means minimizing the sum of the squares of differences between data point values and the value lying on an assumed curve equation. This concept is illustrated in fig.2.

For simplicity of mental arithmetic, arbitrary units are used in fig.2. For point 1, the difference between the data point and the curve point is 1. 1 squared is also 1. For point 2 the square of the difference is also 1. For points 3 and 4 the difference between the data point and the curve point is 0.5 whose square is 0.25. The sum of the squares for the four points is therefore 2.5 (1 + 1 + 0.25 + 0.25). Consider next the dashed curve. For point 1 the difference between the data point and the curve point is 3 which squared is 9 —much larger on its own than the 2.5 for the much better-fitting continuous curve. The dashed curve is obviously not a good fit! Both of the curves shown in fig.2 have simple two-parameter exponential equations as in the EXP2P Solver programs.

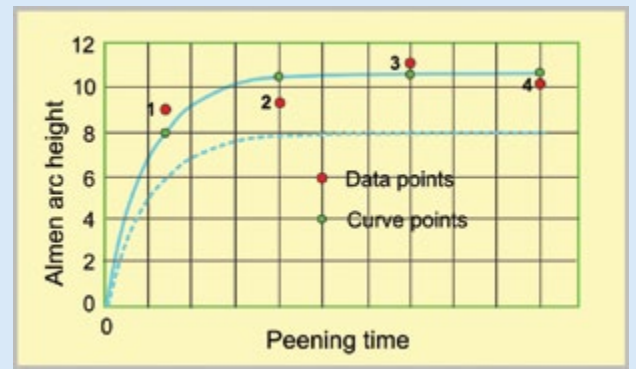


Fig.2. 'Least-squares' curve-fitting applied to shot peening data points.

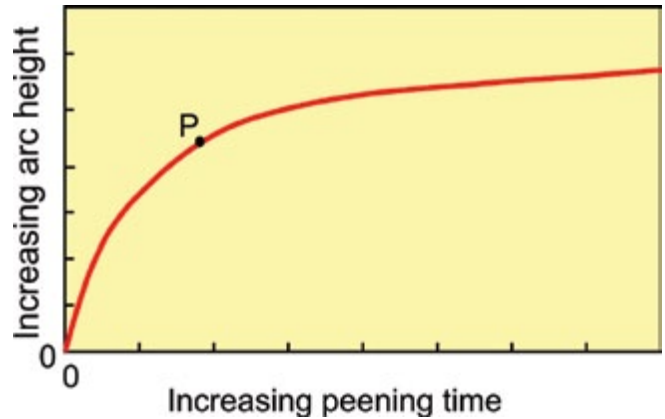


Fig.3. Continuous curve of arc height versus peening time.

that I had produced showed a continued increase of arc height with increase in amount of peening. This did not surprise me because shot peening is essentially a metalworking process for which any increase in the work done increases the amount of deformation and hence curvature. Another feature that I noticed was that there is always a 'point of maximum curvature' suggested here as being at P in fig.3.

A standard method is to select a set of data points and then use a curve-fitting program to find the equation of a curve that best fits those points. Having applied this method many times previously I used it to select exponential equations that could be appropriate.

Two-parameter Exponential Equation

The simplest relevant exponential equation has just two parameters, **a** and **b**, and is given as:

$$h = a*(1 - EXP(-b*t)) \tag{1}$$

where **h** is the Almen arc height and **t** is the peening time.

This type of equation is asymptotic to a limiting **h** value.

Three-parameter Exponential Equation

Three parameters, **a**, **b** and **c**, are involved and the equation is given by:

$$h = a*(1 - EXP(-b*t^c)) \tag{2}$$

This type of equation is also asymptotic to a limiting **h** value.

Four-parameter Exponential Equation

This type of equation adds a linear component, **d*t**, to the three-parameter equation to become:

$$h = a*(1 - EXP(-b*t^c)) + d*t \tag{3}$$

MECHANICS OF FITTING EQUATIONS

Microsoft's Excel program is, as mentioned previously, universally available and was therefore chosen for this project. Fig.4 illustrates the mechanics of how a selected equation, EXP2P, is fitted to a data set without using macros. The data

set comprises nine measured values which is far more than shot peeners would normally acquire. This set does, however, reveal some subtle features of the program.

Data Input Stage

This stage is straightforward as the values for the data set have to be entered within the blue frame. The next column uses a basic facility of Excel insofar as it allows values in one column to be automatically subtracted from corresponding values in another column to give 'Corrected Height' values.

Curve Solving Stage

Excel has a Solver function that I used to determine the least-squares best fit and from that the required peening intensity point. The word 'Solver' therefore appeared to be appropriate for titling purposes.

For these programs, Solver has to be given reasonable starting values for **a** and **b**. It then uses those initial values for **a** and **b** to calculate the **Y** values that lie exactly on the equation's curve. It then subtracts each calculated **Y** value from the corresponding data point's arc height to give what is called a 'Residual' difference. If, for example, the initial

values for **a** and **b** gave a **Y** value of 3.6 for the first data point then the corresponding residual value would be -0.4 (3.6 - 4). Each of the residual values is then squared and added up to give a **SUM** (as in fig.4). This sum has to be minimised to give a **least squares** value. Solver uses iteration to find the minimum value of the **SUM**. 'An **iterative process** is a **process** for calculating a desired result by means of a repeated cycle of operations. An **iterative process** should be convergent, i.e., it should come closer to the desired result as the number of **iterations** increases.' During iteration the values for **a** and **b** are repeatedly adjusted by tiny amounts causing the **SUM** to get smaller and smaller. When no further reduction is achieved the cycle stops to give values such as the 8.65 and 0.53 given in fig.4. The equation for the curve is now solved—in less than the blink of an eye!

Peening Intensity Point

Having now found the parameters **a** and **b** of the equation, the next step was to find the point on the curve that satisfied the requirement that 'When the peening time is doubled the arc height increases by 10%'. Excel's Solver could be employed to

SATURATION CURVE SOLVER, PROGRAM SCS1									
EXP2P STANDARD vers.09									
Strip No.	Time	Arc Height	Pre-bow	Corrected	Calc. Y	Residuals	Res. Sqrd.	Parameters	Values
				Height					
	0	0	0	0	0	0	0	a	8.65
								b	0.53
1	1	4	0.2	3.8	3.53419	-0.265813	0.070657	Start value for a=	9
2	2	6.3	0.5	5.8	5.62433	-0.175672	0.030861		
3	4	7.5	0.2	7.3	7.5915	0.2915015	0.084973	Sat. Intensity =	7.35
4	6	8	0.1	7.9	8.27954	0.3795433	0.144053	T =	3.61
5	8	8.3	0.1	8.2	8.52019	0.320194	0.102524		
6	13	8.8	0.3	8.5	8.64027	0.1402742	0.019677		
7	18	8.9	0.1	8.8	8.64896	-0.151038	0.022813		
8	22	9.2	0.2	9	8.64956	-0.350444	0.122811	If WARNING or ERROR appear	
9	26	9.4	0.4	9	8.64963	-0.350371	0.12276	above, read note below graph.	
						SUM =	0.72113	RMS-R=	0.2831
				Lower Spec.=					
				Upper Spec.=					
STEP 1	Enter data values within blue frame after deleting previous values. Arc heights to be in thousandths of an inch or micrometers.								
STEP 2	Enter largest CORRECTED arc height as a start value for "a" in J4 (required value appears in J6).								
STEP 3	Select box containing Sum of Residuals squared, H16, click Data (on Excel toolbar), select SOLVER, click on SOLVE, press OK.								
STEP 4	Enter required Lower and Upper specification values in red frame above - if you wish to, otherwise delete previous values.								

Fig.4. Excel Worksheet using EXP2P program.

find this point using ‘Goal Seek’. However I found a simpler method. Expressing the 10% requirement mathematically gives the following:

$$Y = a(1 - \text{EXP}(-b \cdot T)) \quad (4)$$

$$1.1Y = a(1 - \text{EXP}(-b \cdot 2T)) \quad (5)$$

Dividing equation (5) by equation (4) eliminates both **Y** and **a**:

$$1.1 = (1 - \text{EXP}(-b \cdot 2T)) / (1 - \text{EXP}(-b \cdot T)) \quad (6)$$

Equation (6) simplifies to give that the required value for **T** is given by $T = 1.897/b$. This required value calculation was incorporated into the spreadsheet, e.g., 3.61 in fig.4. The calculated required value for **T** is then substituted into the now-established equation to give the required peening intensity value, e.g., 7.35 in fig.4.

Goodness of Fit

Having found the parameters of the best fit of the selected equation, it is useful to be able to quantify how good the fit is. The **SUM** shown in fig.4 has the smallest value obtained using the least-squares method. This sum is, however, proportional to the number of data points. The greater the number of data points in a given set the larger will be the **SUM**. A term ‘Root Mean Square’ is generally used to quantify goodness of fit and eliminates the effect of data point number. This is achieved by taking the mean of the squared residuals (by dividing the **SUM** by the number of data points) and taking the root of the mean. This result is calculated automatically in the spreadsheet and displayed as **RMS-R** having a value of 0.2831 for the example shown as fig.4. The smaller the value of **RMS-R** the better is the fit.

Root mean square values are useful for quantifying the magnitude of random errors. They do not, however, reveal systematic errors. As a parallel, consider tossing a coin 40 times and finding that ‘heads’ appeared 20 times and ‘tails’ also 20 times. We might reasonably conclude that the coin was unbiased. But what if ‘heads’ appeared for each of the first 20 tosses and ‘tails’ appeared for each of the second 20 tosses? That would indicate a high probability of a systematic error. A systematic error is also possible for the ‘residuals’ of the example shown in fig.4. I built in an indicator for systematic error by highlighting negative residuals pink. Using the coin-tossing parallel we have two ‘tails’ followed by 4 ‘heads’ and then by three ‘tails’. This sequence is exceptional, hence indicating a possible systematic error.

DISPLAY OF FINDINGS

Throughout this exercise I decided to display all of the findings. This was so that users could discover how the findings had been determined. Many users are only interested in the results. A graphical type of display was therefore introduced that highlights the most important findings. Fig.5 is the graphical display that corresponds to the findings of fig.4. This display utilizes more of Excel’s excellent features.

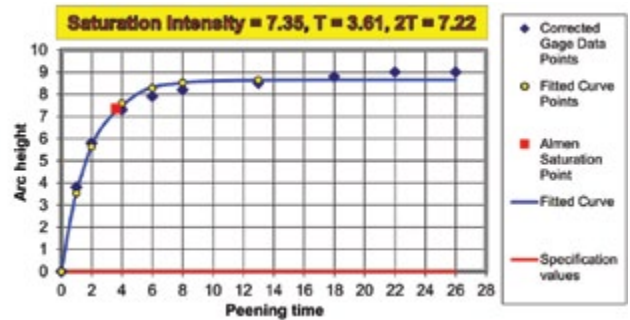


Fig.5. Graphical display using findings presented in fig.4.

The arc height for Saturation Intensity, aka Peening Intensity, is headlined together with its peening time, **T**, and **2T**. Corrected Gage Data Points can be compared with the Fitted Curve and Fitted Curve Points. In this case, the comparison reveals the possible systematic error considered previously for this particular data set. The spreadsheet shown in fig.4 also allowed for the specification limits to be entered. As it happens they were not specified for this data set. If they had been specified then two horizontal red lines would have appeared in fig.5. For example, specifying limits of 7 and 10 would highlight that the derived intensity of 7.35 was very close to the lower allowed limit.

MECHANICS FOR OTHER FITTED EQUATIONS

Having succeeded in producing a program using a two-exponent exponential equation, the next step was to modify the program to accommodate other equations.

Three-parameter Exponential - EXP 3P

This equation uses $h = a \cdot (1 - \text{EXP}(-b \cdot t^c))$ as a fitting equation. The only significant modification required was to employ Excel’s **GOAL SEEK** function to calculate the ‘10% time’ and thence the arc height for that time. This was achieved by asking Goal Seek to minimize the following function:

$$f(x) = 1.1 (1 - \text{EXP}(-b \cdot T)) - (1 - \text{EXP}(-b \cdot 2T)) \quad (7)$$

Goal Seek starts by substituting a value for **T** into equation (7) in order to find a value for $f(x)$. It then makes tiny changes to the value of **T** to see if they reduce the magnitude of $f(x)$. Eventually it finds a value for **T** that minimizes $f(x)$. That minimum value is normally the required peening intensity point’s time.

The only problem with Goal Seek is that there are two possible answers! If it started its search with a time close to zero it would hone in on $T = 0$ as being the solution. In practice this rarely happens but when it does Goal Seek has to be given a value at least as great as the time for the first data point in the set.

Four-parameter Exponential - EXP 4P

This equation uses $h = a \cdot (1 - \text{EXP}(-b \cdot t^c)) + d \cdot t$ as a fitting

equation. The $d \cdot t$ adds a linear component to EXP3P. I felt intuitively that this would achieve an excellent fit and so it turned out. One restriction, however, was that it could only be applied using the least-squares method to data sets that comprised at least six points. Also a four-parameter exponential equation is mathematically relatively unstable. I later found out that the improved fit was not necessary with respect to locating the peening intensity point. For that reason (discussed later), I did not include it in the Solver Suite of programs.

French Specification - 2PF

The French specification for computer-based analysis requires users to use a particular fitting equation. This is termed a ‘rational function’ and has the following equation:

$$h = a \cdot t / (t + b) \tag{8}$$

Equation (8) is exponential to the value of a . The value of the peening point time does not need Goal Seek as I found that it was equal to $9b/2$. This is because we have the following relationship:

$$1.1 \cdot a \cdot T / (b + T) = a \cdot 2 \cdot T / (b + 2T)$$

$a \cdot T$ in the numerators cancel each other out leaving:

$$2 \cdot b + 2 \cdot T = 1.1 \cdot b + 2.2T \text{ which reduces to give:}$$

$$0.9 \cdot b = 0.2 \cdot T \text{ or } T = 0.9b/0.2$$

PROGRAM VARIANTS

Having succeeded in producing three workable programs, EXP2P and EXP3P, I modified them to accommodate the then published corrections needed for flapper peening and also to allow the findings from one data set to be compared with one from a previous data set. This led to a total of 12 separate programs forming the Solver Suite. I pondered about whether or not such a number might be found confusing. Eventually I decided that in practice users would normally only select one or two from the suite to use regularly.

PROGRAM COMPARISONS

The primary objective of the exercise was to estimate the peening intensity point rather than finding the best fit. Estimates of the derived peening intensity point were therefore carried out using all four available programs (EXP2P, EXP3p, EXP4P and 2PF) and the same nine-point data set as shown in fig.4. This gave the results shown as Table 1.

Table 1

Effect of Program on derived Peening Intensity Point

Program	Intensity	T
EXP2P	7.35	3.61
EXP3P	7.69	4.30
EXP4P	7.16	3.59
2PF	7.79	6.06
Average	7.50	4.49

The results show clearly that these widely different programs all produce very similar derived peening intensity values albeit with substantial differences in the corresponding times, T.

Years after producing the Solver Suite of programs, the SAE introduced J2597. This specification gives guidelines on computer-derived peening intensity values. Any program was allowed provided that it met a stated criterion when applied to its set of 10 data sets—the criterion being that all of the derived peening intensities should deviate less than ± 0.001 " from the stated values. Fig.6 illustrates the deviations from the stated values for Solver Suite programs based on EXP2P and 2PF. Both meet the requirement.

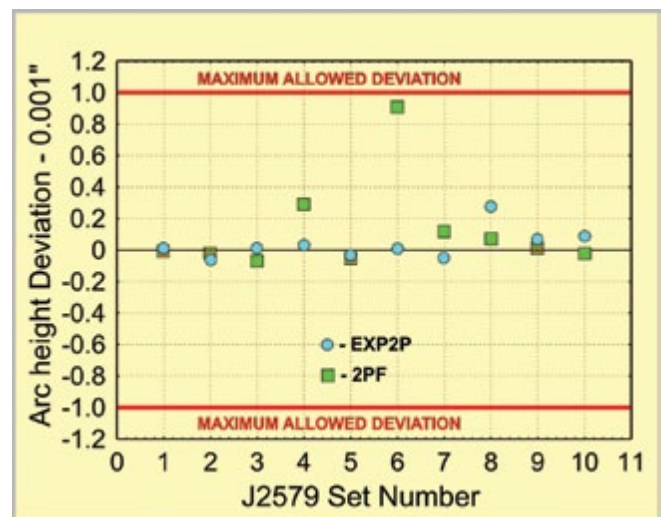


Fig.6. Deviations from the J2579 data set for EXP2P and 2PF programs.

DISCUSSION

A set of programs was successfully produced that allowed peening intensity to be derived using features embodied in Microsoft’s Excel. It is believed that this free set of programs has stimulated the almost universal adoption of computer-based intensity derivation in place of subjective manual techniques. All of the programs are ‘open source’ as I insisted that a primary feature was for them to be ‘educational’—allowing users to know how values are being derived. In particular the fitting equation is always declared unlike commercial programs.

As a word of warning, specifications are regularly updated which may impinge on some Solver Suite programs. For example, AMS2590B for flapper peening was updated in January 2019. This specification requires data corrections prior to creating a saturation curve. Only some of the proposed data corrections are applied automatically by the corresponding Suite programs. ●