
Subject: Question about CURVEFIT function
Posted by [fd_luni](#) on Thu, 29 Aug 2013 13:14:43 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi all

I have some questions about the CURVEFIT function. I didn't use any programming language before and there are some things in this function that I don't understand.

The keyword TOL which corresponds to tolerance what actually do in the algorithm?

When you specifies the number of iterations what does this means? What is the role of this keyword?

Finally, the CURVEFIT function how does it works? For instance, It uses the derivatives of the points?

Many Thanks
Mar

Subject: Re: Question about CURVEFIT function
Posted by [Paul Van Delst\[1\]](#) on Thu, 29 Aug 2013 14:45:56 GMT
[View Forum Message](#) <> [Reply to Message](#)

Here's a place to start:
http://en.wikipedia.org/wiki/Non-linear_least_squares

IDL docs tell me CURVEFIT uses a gradient expansion method. Directly from the docs:

PROCEDURE:

Copied from "CURFIT", least squares fit to a non-linear function, pages 237-239, Bevington, Data Reduction and Error Analysis for the Physical Sciences. This is adapted from: Marquardt, "An Algorithm for Least-Squares Estimation of Nonlinear Parameters", J. Soc. Ind. Appl. Math., Vol 11, no. 2, pp. 431-441, June, 1963.

"This method is the Gradient-expansion algorithm which combines the best features of the gradient search with the method of linearizing the fitting function."

Iterations are performed until the chi square changes by only TOL or until ITMAX iterations have been performed.

The initial guess of the parameter values should be as close to the actual values as possible or the solution

may not converge.

On 08/29/13 09:14, fd_luni@mail.com wrote:

- > Hi all
- >
- > I have some questions about the CURVEFIT function. I didn't use any
- > programming language before and there are some things in this
- > function that I don't understand.
- >
- > The keyword TOL which corresponds to tolerance what actually do in
- > the algorithm?
- >
- > When you specifies the number of iterations what does this means?
- > What is the role of this keyword?
- >
- > Finally, the CURVEFIT function how does it works? For instance, It
- > uses the derivatives of the points?
- >
- > Many Thanks Mar
- >

Subject: Re: Question about CURVEFIT function

Posted by [Paul Van Delst\[1\]](#) on Thu, 29 Aug 2013 14:48:43 GMT

[View Forum Message](#) <> [Reply to Message](#)

BTW, I would use MPFIT rather than CURVEFIT:

<http://www.physics.wisc.edu/~craigm/idl/fitting.html>

On 08/29/13 10:45, Paul van Delst wrote:

- > Here's a place to start:
- > http://en.wikipedia.org/wiki/Non-linear_least_squares
- >
- > IDL docs tell me CURVEFIT uses a gradient expansion method. Directly
- > from the docs:
- >
- > PROCEDURE:
- > Copied from "CURFIT", least squares fit to a non-linear
- > function, pages 237-239, Bevington, Data Reduction and Error
- > Analysis for the Physical Sciences. This is adapted from:
- > Marquardt, "An Algorithm for Least-Squares Estimation of
- > Nonlinear Parameters", J. Soc. Ind. Appl. Math., Vol 11,
- > no. 2, pp. 431-441, June, 1963.
- >
- > "This method is the Gradient-expansion algorithm which
- > combines the best features of the gradient search with

> the method of linearizing the fitting function."
>
> Iterations are performed until the chi square changes by
> only TOL or until ITMAX iterations have been performed.
>
> The initial guess of the parameter values should be
> as close to the actual values as possible or the solution
> may not converge.
>
> On 08/29/13 09:14, fd_luni@mail.com wrote:
>> Hi all
>>
>> I have some questions about the CURVEFIT function. I didn't use any
>> programming language before and there are some things in this
>> function that I don't understand.
>>
>> The keyword TOL which corresponds to tolerance what actually do in
>> the algorithm?
>>
>> When you specifies the number of iterations what does this means?
>> What is the role of this keyword?
>>
>> Finally, the CURVEFIT function how does it works? For instance, It
>> uses the derivatives of the points?
>>
>> Many Thanks Mar
>>

Subject: Re: Question about CURVEFIT function
Posted by [fd_luni](#) on Fri, 30 Aug 2013 08:17:59 GMT
[View Forum Message](#) <> [Reply to Message](#)

> "This method is the Gradient-expansion algorithm which combines the best features of the
gradient search with the method of linearizing the fitting function."

I still don't understand clearly how the CURVEFIT function works. The method of linearizing the
fitting algorithm is the Levenberg-Marquart?

Subject: Re: Question about CURVEFIT function
Posted by [Craig Markwardt](#) on Sat, 14 Sep 2013 05:57:07 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Friday, August 30, 2013 4:17:59 AM UTC-4, fd_...@mail.com wrote:
>> "This method is the Gradient-expansion algorithm which combines the best features of the
gradient search with the method of linearizing the fitting function."

>
>
>
> I still don't understand clearly how the CURVEFIT function works. The method of linearizing the fitting algorithm is the Levenberg-Marquart?

Maria wrote to me and asked this question privately. Here is what I wrote...

The CURVEFIT algorithm is copied from Bevington's "Data Reduction and Error Analysis for the Physical Sciences." Bevington is a very good book. Also, Numerical Recipes has a very good discussion of how this works.

The goal is to find the minimum of a function, in this case, chi-square. You can think of the function to be optimized as a valley in parameter space and the goal is to find the bottom of the valley (coordinates of lowest value of function). At a basic level, one might consider two ways to find a minimum of a function, steepest descent or Newton's method.

One way is to find the direction of steepest descent and follow that. You just calculate the gradient direction of the function and take your next step in that direction. This is good when the gradient is steep, but bad when you are near the bottom of the valley where the function is more rounded and less steep.

http://en.wikipedia.org/wiki/Gradient_descent

Another way is to use Newton's method. This method basically treats the shape of the valley as a paraboloid. You can know the shape of a paraboloid by measuring the derivatives at one position, and use those to solve for the position of the most extreme value (minimum function value). This type of method works well near the bottom of the valley where the shape of the function is more rounded but does not work well when the gradient is steep.

http://en.wikipedia.org/wiki/Newton%27s_method_in_optimization

Levenberg and Marquardt came up with a scheme to mix both kinds of solutions. When you are far from the bottom of the valley, you use the steepest gradient method. When you are close to the bottom, you use the Newton's method. This algorithm has the benefits of both techniques.

http://en.wikipedia.org/wiki/Levenberg%E2%80%93Marquardt_algorithm

CURVEFIT uses Levenberg-Marquardt algorithm. It is concerned with calculating the gradients (derivatives) of the function, and then combining those to estimate the direction of steepest descent and Newton minimum. It forms a new trial solution, and starts a new iteration to try again. When it finds a minimum value to within the desired tolerance (CURVEFIT's TOL keyword), it reports that solution.

The MPFIT family of functions are also advanced Levenberg-Marquardt-style algorithms. The internals are also more robust to round-off errors and degenerate parameters.

Craig Markwardt
