Subject: Re: Function Maximum
Posted by R.Bauer on Thu, 19 Jun 2003 14:44:07 GMT
View Forum Message <> Reply to Message

Kate wrote:

> If I have a function definition, say it is
>
> FUNCTION MYGAMMA, X, P
> RETURN, P[0]*(X^P[1])*EXP(-1*X/P[2])
> END
>
> Where P holds my fit parameters that are already determined.
>
> If I want to find a local maximum in a certain range is it possible to
> do this with an existing written IDL function?

What's with MAX()

regards
Reimar

>
> Thanks
>
> Kate Morgan
> Research Technician
> Department of Medical Imaging
> University of Toronto

--
Forschungszentrum Juelich
email: R.Bauer@fz-juelich.de
http://www.fz-juelich.de/icg/icg-i/
========== ======
a IDL library at ForschungsZentrum Juelich
http://www.fz-juelich.de/icg/icg-i/idl_icglib/idl_lib_intro. html

Subject: Re: Function Maximum
Posted by Craig Markwardt on Thu, 19 Jun 2003 14:45:36 GMT
View Forum Message <> Reply to Message

faeriepunk@aol.com (Kate) writes:

> If I have a function definition, say it is
>
> FUNCTION MYGAMMA, X, P

> RETURN, P[0]*(X^P[1])*EXP(-1*X/P[2])
> END
>
> Where P holds my fit parameters that are already determined.
>
> If I want to find a local maximum in a certain range is it possible to
> do this with an existing written IDL function?

You question has an analytical solution. The function you describe has a global maximum at X_MAX = P[1]*P[2], with no other saddle-points.

For more complicated functions, you can use an optimizer. For example CONSTRAINED_MIN, or AMOEBA. Both of those are minimizers so you would have to minimize the *negative* of your function in order to maximize it. I also have a routine on my web page called TNMIN. It's a tad rough at the edges, but it doesn't require function derivatives, and has a MAXIMIZE keyword. [I use it all the time to maximize huge functions.]

Craig

http://cow.physics.wisc.edu/~craigm/idl/idl.html (under fitting)

--
---------- ----------
Craig B. Markwardt, Ph.D. EMAIL: craigmnet@cow.physics.wisc.edu
Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response
---------- ----------

Subject: Re: Function Maximum
Posted by Benjamin Panter on Thu, 19 Jun 2003 15:02:11 GMT
View Forum Message <> Reply to Message

Reimar Bauer wrote:
> Kate wrote:
>
>
>> If I have a function definition, say it is
>>
>> FUNCTION MYGAMMA, X, P
>> RETURN, P[0]*(X^P[1])*EXP(-1*X/P[2])
>> END
>>
>> Where P holds my fit parameters that are already determined.
>>

>> If I want to find a local maximum in a certain range is it possible to
>> do this with an existing written IDL function?
>
>
> What's with MAX()

<hopping out from lurking for a moment>

I think that Kate is asking to find the maximum of an expression rather
than the maximum of an array.

An inelegant solution that might work is to evaluate the funtion in the
range of interest to as high a precision as is computationally possible
- and then run MAX() on it - but there must be a nicer way? I think this
method will fall down if there is a very sharp global max but a wider
local max

Apologies for not being much of a mathematician and failing to provide a
better way!

<back to lurking & learning...>

Ben

--
Ben Panter, Edinburgh
My name (no spaces)@bigfoot which is a com.

Subject: Re: Function Maximum
Posted by James Kuyper on Thu, 19 Jun 2003 16:27:49 GMT
View Forum Message <> Reply to Message

Benjamin Panter wrote:
...
> An inelegant solution that might work is to evaluate the funtion in the
> range of interest to as high a precision as is computationally possible
> - and then run MAX() on it - but there must be a nicer way? I think this
> method will fall down if there is a very sharp global max but a wider
> local max
>
> Apologies for not being much of a mathematician and failing to provide a
> better way!

Actually, for an arbitrary function, the method you describe is the only
method that is absolutely guaranteed to find the true maximum value. Any
method that is faster than that one is based upon assumptions about the
function, such as the assumption that it is reasonably smooth.

Subject: Re: Function Maximum
Posted by Craig Markwardt on Thu, 19 Jun 2003 16:50:02 GMT
View Forum Message <> Reply to Message

James Kuyper <kuyper@saicmodis.com> writes:

> Benjamin Panter wrote:
> ...
>> An inelegant solution that might work is to evaluate the funtion in the
>> range of interest to as high a precision as is computationally possible
>> - and then run MAX() on it - but there must be a nicer way? I think this
>> method will fall down if there is a very sharp global max but a wider
>> local max
>>
>> Apologies for not being much of a mathematician and failing to provide a
>> better way!
>
> Actually, for an arbitrary function, the method you describe is the only
> method that is absolutely guaranteed to find the true maximum value. Any
> method that is faster than that one is based upon assumptions about the
> function, such as the assumption that it is reasonably smooth.

Even the brute force method described above assumes that the function
is smooth enough that it doesn't vary in between grid samples. For
example, a finite sum of delta functions at random positions would
probably be missed by any approach.

Craig

--
-- --------------
Craig B. Markwardt, Ph.D. EMAIL: craigmnet@cow.physics.wisc.edu
Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response
-- --------------

Subject: Re: Function Maximum
Posted by James Kuyper on Thu, 19 Jun 2003 17:10:01 GMT
View Forum Message <> Reply to Message

Craig Markwardt wrote:
>
> James Kuyper <kuyper@saicmodis.com> writes:
>
>> Benjamin Panter wrote:
>> ...
>>> An inelegant solution that might work is to evaluate the funtion in the
>>> range of interest to as high a precision as is computationally possible

>>> - and then run MAX() on it - but there must be a nicer way? I think this
>>> method will fall down if there is a very sharp global max but a wider
>>> local max
>>>
>>> Apologies for not being much of a mathematician and failing to provide a
>>> better way!
>>
>> Actually, for an arbitrary function, the method you describe is the only
>> method that is absolutely guaranteed to find the true maximum value. Any
>> method that is faster than that one is based upon assumptions about the
>> function, such as the assumption that it is reasonably smooth.
>
> Even the brute force method described above assumes that the function
> is smooth enough that it doesn't vary in between grid samples. For
> example, a finite sum of delta functions at random positions would
> probably be missed by any approach.

I was assuming that the grid samples would consist of every distinguishable floating point number within the domain over which you want to find the maximum. In a certain practical sense, a computer function is meaningfully defined only at those sample points. Still, the computer function is usually meant to be a discrete approximation of an abstract mathematical function; the abstract function might have a maximum that isn't represented correctly in the discrete approximation.

Subject: Re: Function Maximum
Posted by Haje Korth on Thu, 19 Jun 2003 18:48:20 GMT
View Forum Message <> Reply to Message

This sounds more like a task for Mathematica and not IDL!?

Haje

--

"Kate" <faeriepunk@aol.com> wrote in message
news:c3c94d97.0306190612.542583f9@posting.google.com...
> If I have a function definition, say it is
>
> FUNCTION MYGAMMA, X, P
> RETURN, P[0]*(X^P[1])*EXP(-1*X/P[2])
> END
>
> Where P holds my fit parameters that are already determined.
>
> If I want to find a local maximum in a certain range is it possible to

> do this with an existing written IDL function?
>
> Thanks
>
> Kate Morgan
> Research Technician
> Department of Medical Imaging
> University of Toronto

Subject: Re: Function Maximum
Posted by R.Bauer on Thu, 19 Jun 2003 19:55:01 GMT
View Forum Message <> Reply to Message

Haje Korth wrote:

> This sounds more like a task for Mathematica and not IDL!?
>
> Haje
>

Why?

Normally the member of this group could build everything what's someone is missing or already have done it.

Mathematica could have different mathematics functions and tools as idl but if you already know idl and it mostly did all the things you want why to switch for some percent of the work to another tool.

For a lot of years before I started with IDL I have had a lot of different tools for different parts of a project. After an upgrade of some of this tools I have had to learn everything new. This tooks a lot of time. The tools I used were not compatible to the tools my neighbor uses. That's make much fun too.

After a few month of work with idl I examined that it is am atomic tool for our work. We developed over several years a lot of common tools. I myself use it for all what I could not do by latex. ;-) If something isn't available what's we need then someone has to build it.

The members of this newsgroup in combination with google search group is a very good place of knowledge and inspiration.

Reimar

--
Forschungszentrum Juelich
email: R.Bauer@fz-juelich.de
http://www.fz-juelich.de/icg/icg-i/
========== ======
a IDL library at ForschungsZentrum Juelich
http://www.fz-juelich.de/icg/icg-i/idl_icglib/idl_lib_intro. html

---

Subject: Re: Function Maximum
Posted by Paul Van Delst[1] on Thu, 19 Jun 2003 21:06:46 GMT
View Forum Message <> Reply to Message

Kate wrote:
>
> If I have a function definition, say it is
>
> FUNCTION MYGAMMA, X, P
> RETURN, P[0]*(X^P[1])*EXP(-1*X/P[2])
> END
>
> Where P holds my fit parameters that are already determined.
>
> If I want to find a local maximum in a certain range is it possible to
> do this with an existing written IDL function?

How about something like:

```
maxY = MAX( mygamma(X,P) )
```

?

You already have your P's and your X's define the range in which the maximum is sought. If you need the X-value of the maximum, then

```
maxY = MAX( mygamma(X,P), maxXLoc )
maxX = X[maxXLoc]
```

If you don't want to carry around a bunch of X-vectors, then why not do something like:

```
FUNCTION FINDMAX, Xr, P, UserFunction
  X1 = DOUBLE(Xr[0])  ; The beginning of the range
  X2 = DOUBLE(Xr[1])  ; The end of the range
  dX = DOUBLE(Xr[2])  ; The resolution within the range

  ; Compute the ordinate vector
  nX = LONG( (X2-X1)/dX ) + 1.5d0 )
  X = ( DINDGEN( nX ) / DOUBLE( nX-1 ) ) * ( X2-X1 ) + X1
```

```
    ; Find the maximum value of your user function
    Y = CALL_FUNCTION( UserFunction, X, DOUBLE(P) )
    maxY = MAX( Y, maxXLoc )
    maxX = X[ maxXLoc ]

    ; Return the coordinate of the local max
    RETURN, [ maxX, maxY ]

  END
```

where the Xr input defines the function range [x1,x2] and the resolution at which the function is calculated (Sort of like a loop triplet.)

Otherwise maybe you could use something like the FX_ROOT function to find the roots of the first derivative of your function (via DERIV or your own calculated derivative) and use the second derivative to determine if it's a max or min.

paulv

--
Paul van Delst
CIMSS @ NOAA/NCEP/EMC
Ph: (301)763-8000 x7748
Fax:(301)763-8545

Subject: Re: Function Maximum
Posted by faeriepunk on Fri, 20 Jun 2003 13:25:47 GMT
View Forum Message <> Reply to Message

Thanks everyone for your suggestions, I ended up writing a function to do a windowed brute force search because I wanted the continuous not discrete maximum of the function. Sometimes it's unfortunate not being a mathemetician, but thanks Craig for your analytical solution, it will work for this function!

Kate Morgan
Research Technician
Dept. of Medical Imaging
University of Toronto

Paul van Delst <paul.vandelst@noaa.gov> wrote in message news:<3EF225E6.33D91065@noaa.gov>...
> Kate wrote:

>>
>> If I have a function definition, say it is
>>
>> FUNCTION MYGAMMA, X, P
>> RETURN, P[0]*(X^P[1])*EXP(-1*X/P[2])
>> END
>>
>> Where P holds my fit parameters that are already determined.
>>
>> If I want to find a local maximum in a certain range is it possible to
>> do this with an existing written IDL function?
>
> How about something like:
>
> maxY = MAX(mygamma(X,P))
>
> ?
>
> You already have your P's and your X's define the range in which the maximum is sought. If
> you need the X-value of the maximum, then
>
> maxY = MAX(mygamma(X,P), maxXLoc)
> maxX = X[maxXLoc]
>
> If you don't want to carry around a bunch of X-vectors, then why not do something like:
>
> FUNCTION FINDMAX, Xr, P, UserFunction
> X1 = DOUBLE(Xr[0]) ; The beginning of the range
> X2 = DOUBLE(Xr[1]) ; The end of the range
> dX = DOUBLE(Xr[2]) ; The resolution within the range
>
> ; Compute the ordinate vector
> nX = LONG((X2-X1)/dX) + 1.5d0)
> X = (DINDGEN(nX) / DOUBLE(nX-1)) * (X2-X1) + X1
>
> ; Find the maximum value of your user function
> Y = CALL_FUNCTION(UserFunction, X, DOUBLE(P))
> maxY = MAX(Y, maxXLoc)
> maxX = X[maxXLoc]
>
> ; Return the coordinate of the local max
> RETURN, [maxX, maxY]
>
> END
>
> where the Xr input defines the function range [x1,x2] and the resolution at which the
> function is calculated (Sort of like a loop triplet.)
>

> Otherwise maybe you could use something like the FX_ROOT function to find the roots of the
> first derivative of your function (via DERIV or your own calculated derivative) and use
> the second derivative to determine if it's a max or min.
>
> paulv
