Subject: Re: Point of intersection Posted by Juggernaut on Wed, 30 Jul 2008 15:55:14 GMT View Forum Message <> Reply to Message

On Jul 30, 11:46 am, kishore1...@gmail.com wrote: > Hello. > > I hope this is simple question for experienced guys. > How to find out perfect point of intersection of x value and > corresponding y value. > For example: > x1=[0.1,0.2,0.6,0.7]> x2=[0.5,0.4,0.5,0.3]> y=[1,2,3,4]> plot,x1,y,xran=[0.,0.8]> oplot,x2,y > In this, two plots are intersection at one point, how to find out that > particular interesection x and y value. Thanking you, > Kishore

Well if the y values are always equal like you have there then they will intersect where the x values are equal. Unless I'm thinking the wrong way? Which is entirely possible....

Subject: Re: Point of intersection
Posted by Juggernaut on Wed, 30 Jul 2008 16:11:27 GMT
View Forum Message <> Reply to Message

```
On Jul 30, 11:55 am, Bennett <juggernau...@gmail.com> wrote:

> On Jul 30, 11:46 am, kishore1...@gmail.com wrote:

> Hello,

> Hello,

> How to find out perfect point of intersection of x value and

> corresponding y value.

> For example:

> x1=[0.1,0.2,0.6,0.7]

> x2=[0.5,0.4,0.5,0.3]

> y=[1,2,3,4]

> plot,x1,y,xran=[0.,0.8]
```

```
>> oplot,x2,y
>
>> In this, two plots are intersection at one point, how to find out that
>> particular interesection x and y value.
>
>> Thanking you,
>
>> Kishore
>
> Well if the y values are always equal like you have there then they
> will intersect where the x values are equal. Unless I'm thinking the
> wrong way? Which is entirely possible....
```

Think I may have jumped the gun there...what you really want to do since your dataset does not have any x that match (what I neglected to see because I'm not the brightest bulb) is to interpolate both your x's over a longer range and then find where they match within some error...like the following

```
x1_int = interpol(x1, 100)
x2_int = interpol(x2, 100)
location = where(abs(x1_int-x2_int) LT 0.001) ;- Where you can set
your error to whatever it is that you want..which I assume will depend
on the degree to which you interpolate
```

For your case this gives the intersection to be 0.4666

Hope this helps get you in the right direction....as well as myself

```
Subject: Re: Point of intersection
Posted by Bob[3] on Wed, 30 Jul 2008 16:14:05 GMT
View Forum Message <> Reply to Message
```

```
On Jul 30, 11:46 am, kishore1...@gmail.com wrote:

> Hello,

> I hope this is simple question for experienced guys.

> How to find out perfect point of intersection of x value and

> corresponding y value.

> For example:

> x1=[0.1,0.2,0.6,0.7]

> x2=[0.5,0.4,0.5,0.3]

> y=[1,2,3,4]

> plot,x1,y,xran=[0.,0.8]

> oplot,x2,y

>
```

> In this, two plots are intersection at one point, how to find out that

> particular interesection x and y value.

>

> Thanking you,

>

> Kishore

The lines cross where (x1(i)-x2(i)) changes sign. ie. between the indices given by MAX(WHERE((x2-x1)>0)) and MIN(WHERE((x2-x1)<0))

The exact intersection can then be found by determining the intersection of the 2 straight lines defined by those points.

Bob.

Subject: Re: Point of intersection

Posted by Wox on Thu, 31 Jul 2008 08:51:15 GMT

View Forum Message <> Reply to Message

On Wed, 30 Jul 2008 08:46:30 -0700 (PDT), kishore1818@gmail.com wrote:

- > ...how to find out that
- > particular interesection x and y value.

The could below works for any x and y values but might be simplified in your case (special y values).

function segmentintersect,L1x,L1y,L2x,L2y,xy=xy

; code:

; 0: no intersecting

; 1: intersect in 1 point

; 2: parallel

; 3: coincident

$$\begin{split} &\text{denom=float(L2y[1]-L2y[0])*(L1x[1]-L1x[0])-(L2x[1]-L2x[0])*(L1y[1]-L1y[0])} \\ &\text{numa=(L2x[1]-L2x[0])*(L1y[0]-L2y[0])-(L2y[1]-L2y[0])*(L1x[0]-L2x[0])} \\ &\text{numb=(L1x[1]-L1x[0])*(L1y[0]-L2y[0])-(L1y[1]-L1y[0])*(L1x[0]-L2x[0])} \end{split}$$

if denom eq 0 then code= (numa eq 0 and numb eq 0)+2 \$ else begin

ua = numa / denom

ub = numb / denom

```
code= ua ge 0 and ua le 1 and ub ge 0 and ub le 1
if code then $
xy=[L1x[0]+ua*(L1x[1]-L1x[0]),L1y[0]+ua*(L1y[1]-L1y[0])]
endelse
return,code
end;function segmentintersect
pro segtest
x1=[0.1,0.2,0.6,0.7]
x2=[0.5,0.4,0.5,0.3]
y1=[1,2,3,4]
y2=y1
window
plot,x1,y1,psym=-2
oplot,x2,y2,psym=-2
n=n_elements(x2)
y2_1=interpol(y1,x1,x2)
b=y2_1 gt y2
interval=where(b[0:n-2]-b[1:*],ct)
if ct ne 0 then begin
xy=fltarr(2,ct)
for i=0,ct-1 do begin
j=interval[i]
 L2x=x2[j:j+1]
 L2y=y2[i:j+1]
 j=value_locate(x1,L2x)
 k=0
 repeat begin
 L1x=x1[j[k]:j[k]+1]
 L1y=y1[j[k]:j[k]+1]
 code=segmentintersect(L1x,L1y,L2x,L2y,xy=tmp)
 b=code eq 1
 if b then begin
  xy[*,i]=tmp
 plots,[tmp[0],tmp[0]],[0,tmp[1]],/data
 endif
 endrep until b or (k eq 2)
endfor
endif
```

end;pro segtest

Subject: Re: Point of intersection

Posted by Wox on Thu, 31 Jul 2008 08:54:04 GMT

View Forum Message <> Reply to Message

On Thu, 31 Jul 2008 10:51:15 +0200, Wox <nomail@hotmail.com> wrote:

> On Wed, 30 Jul 2008 08:46:30 -0700 (PDT), kishore1818@gmail.com wrote:

>

>> ...how to find out that

>> particular interesection x and y value.

>

>

> The could below works for any x and y values but might be simplified

> in your case (special y values).

<snip>

Note that xy will contain all intersection points (i.e. 1 point in your case).

Subject: Re: Point of intersection

Posted by Wox on Thu, 31 Jul 2008 08:58:26 GMT

View Forum Message <> Reply to Message

On Thu, 31 Jul 2008 10:51:15 +0200, Wox <nomail@hotmail.com> wrote:

> On Wed, 30 Jul 2008 08:46:30 -0700 (PDT), kishore1818@gmail.com wrote:

>

>> ...how to find out that

>> particular interesection x and y value.

> >

> The could below works for any x and y values but might be simplified

> in your case (special y values).

could = code ;-)

I knew I posted this before:

http://objectmix.com/idl-pvwave/343059-there-quick-way-find-

intersection-two-lines.html#post1247556

View Forum Message <> Reply to Message

```
On Jul 30, 1:11 pm, Bennett < juggernau...@gmail.com> wrote:
> On Jul 30, 11:55 am, Bennett < juggernau...@gmail.com> wrote:
>
>
>
>> On Jul 30, 11:46 am, kishore1...@gmail.com wrote:
>>> Hello.
>
>>> I hope this is simple question for experienced guys.
>>> How to find out perfect point of intersection of x value and
>>> corresponding y value.
>>> For example:
>>> x1=[0.1,0.2,0.6,0.7]
>>> x2=[0.5,0.4,0.5,0.3]
>>> y=[1,2,3,4]
>>> plot,x1,y,xran=[0.,0.8]
>>> oplot,x2,y
>>> In this, two plots are intersection at one point, how to find out that
>>> particular interesection x and y value.
>>> Thanking you,
>
>>> Kishore
>> Well if the v values are always equal like you have there then they
>> will intersect where the x values are equal. Unless I'm thinking the
>> wrong way? Which is entirely possible....
>
> Think I may have jumped the gun there...what you really want to do
> since your dataset does not have any x that match (what I neglected to
> see because I'm not the brightest bulb) is to interpolate both your
> x's over a longer range and then find where they match within some
> error...like the following
>
> x1_{int} = interpol(x1, 100)
> x2 int = interpol(x2, 100)
> location = where(abs(x1 int-x2 int) LT 0.001); - Where you can set
> your error to whatever it is that you want..which I assume will depend
> on the degree to which you interpolate
>
 For your case this gives the intersection to be 0.4666
>
> Hope this helps get you in the right direction....as well as myself
```

Thanks for nice suggestion.

Kishore

```
Subject: Re: Point of intersection
Posted by kishore1818 on Thu, 31 Jul 2008 13:44:09 GMT
View Forum Message <> Reply to Message
On Jul 31, 5:51 am, Wox <nom...@hotmail.com> wrote:
> On Wed, 30 Jul 2008 08:46:30 -0700 (PDT), kishore1...@gmail.com wrote:
>> ...how to find out that
>> particular interesection x and y value.
> The could below works for any x and y values but might be simplified
> in your case (special y values).
 function segmentintersect, L1x, L1y, L2x, L2y, xy=xy
>
> ; code:
> ; 0: no intersecting
> ; 1: intersect in 1 point
> ; 2: parallel
> ; 3: coincident
>
 denom=float(L2y[1]-L2y[0])*(L1x[1]-L1x[0])-(L2x[1]-L2x[0])*( L1y[1]-L1y[0])
  numa = (L2x[1]-L2x[0])*(L1y[0]-L2y[0])-(L2y[1]-L2y[0])*(L1x[0]-L2x[0])
  numb = (L1x[1]-L1x[0])*(L1y[0]-L2y[0])-(L1y[1]-L1y[0])*(L1x[0]-L2x[0])
>
> if denom eq 0 then code= (numa eq 0 and numb eq 0)+2$
> else begin
      ua = numa / denom
>
      ub = numb / denom
>
      code= ua ge 0 and ua le 1 and ub ge 0 and ub le 1
>
      if code then $
> xy=[L1x[0]+ua*(L1x[1]-L1x[0]),L1y[0]+ua*(L1y[1]-L1y[0])]
> endelse
> return,code
> end;function segmentintersect
```

> pro segtest

```
> x1=[0.1,0.2,0.6,0.7]
> x2=[0.5,0.4,0.5,0.3]
> y1=[1,2,3,4]
> y2=y1
> window
> plot,x1,y1,psym=-2
> oplot,x2,y2,psym=-2
>
> n=n_elements(x2)
> y2_1=interpol(y1,x1,x2)
> b=y2_1 gt y2
> interval=where(b[0:n-2]-b[1:*],ct)
 if ct ne 0 then begin
      xy=fltarr(2,ct)
>
      for i=0,ct-1 do begin
>
          j=interval[i]
>
          L2x=x2[j:j+1]
>
          L2y=y2[j:j+1]
>
          j=value_locate(x1,L2x)
>
          k=0
>
          repeat begin
>
              L1x=x1[j[k]:j[k]+1]
>
              L1y=y1[j[k]:j[k]+1]
>
              code=segmentintersect(L1x,L1y,L2x,L2y,xy=tmp)
>
              b=code eq 1
>
              if b then begin
>
                   xy[*,i]=tmp
>
                   plots,[tmp[0],tmp[0]],[0,tmp[1]],/data
>
              endif
>
              k++
>
          endrep until b or (k eq 2)
>
      endfor
>
> endif
 end;pro segtest
>
```

Hi,

Thanks for your program. This is useful for multiple intersection of two lines. It is very interesting.

Kishore