

## interparc

by [John D'Errico](#)

01 Feb 2012 (Updated 16 Aug 2012)

Distance based interpolation along a general curve in space

[Watch this File](#)

4.9

4.9 | [22 ratings](#)[Rate this file](#)

220 Downloads (last 30 days)

File Size: 6.81 KB

File ID: #34874

Version: 1.3

[Download Zip](#)[View License](#)

Download apps, toolboxes, and other File Exchange content using Add-On Explorer in MATLAB.

[» Watch video](#)

### Highlights from interparc

[interparc\(t,px,py,varargin\)](#)

interparc: interpolate points along a curve in 2 or more dimensions

[View all files](#)

#### File Information

**Description** A common request is to interpolate a set of points at fixed distances along some curve in space (2 or more dimensions.) The user typically has a set of points along a curve, some of which are closely spaced, others not so close, and they wish to create a new set which is uniformly spaced along the same curve.

When the interpolation is assumed to be piecewise linear, this is easy. However, if the curve is to be a spline, perhaps interpolated as a function of chordal arclength between the points, this gets a bit more difficult. A nice trick is to formulate the problem in terms of differential equations that describe the path along the curve. Then the interpolation can be done using an ODE solver.

As an example of use, I'll pick a random set of points around a circle in the plane, then generate a new set of points that are equally spaced in terms of arc length along the curve, so around the perimeter of the circle.

```
theta = sort(rand(15,1))*2*pi;
theta(end+1) = theta(1);
px = cos(theta);
py = sin(theta);
```

100 equally spaced points, using a spline interpolant.

```
pt = interparc(100,px,py,'spline');
```

```
% Plot the result
plot(px,py,'r*',pt(:,1),pt(:,2),'b-o')
axis([-1.1 1.1 -1.1 1.1])
axis equal
grid on
xlabel X
ylabel Y
title 'Points in blue are uniform in arclength around the circle'
```

You can now also return a function handle to evaluate the curve itself at any point. As well, CSAPE is an option for periodic (closed) curves, as long as it is available in your matlab installation.

```
[~,~,foft] = interparc([],px,py,'spline');
foft(0:0.25:1)
ans =
    0.98319 0.18257
   -0.19064 0.98151
   -0.98493 -0.17486
    0.18634 -0.98406
    0.98319 0.18257
```

**Acknowledgements** This file inspired [Active Figure Zoom For Selecting Points](#), [Riv Map](#), [River Morphodynamics From Analysis Of Planforms](#), [Jdugge/Xy2sn](#), and [Transform Velocity Field Into Wall Coordinates](#).

**Required Products** [MATLAB](#)

**MATLAB release** MATLAB 7.12 (R2011a)

**Other requirements** Earlier releases will work as well.

#### Tags for This File

Please [create a community profile](#) to tag files.

arc length  
cubic  
distance  
equal spacing  
interpolation  
linear  
parametric  
spline






 






Please [create a community profile](#) to add a comment or rating.

**Comments and Ratings** (44)

|             |                                     |   |              |
|-------------|-------------------------------------|---|--------------|
| 02 Sep 2016 | <a href="#">Daniel Moreno</a>       | Your function saved my day! Thank you so much =]  |              |
| 24 Jun 2016 | <a href="#">Alexandr Buyval</a>     | John,<br>Thank you for this function. It exactly what I need.<br>Could you answer me how to get a curvature values along curve?   |              |
| 28 May 2016 | <a href="#">John D'Errico</a>       | :)  | Comment only |
| 28 May 2016 | <a href="#">Toby Wenz</a>           | This is my favorite external Matlab function. I use it all the time in my research to resample datasets in order to make sure I can combine, plot and analyse them together.  |              |
| 12 May 2016 | <a href="#">John D'Errico</a>       | If for some x-values, you may or may not get multiple y values, then the problem becomes complicated.<br><br>You may want to break the problem down into separate sub-curves, where the branch of the curve is chosen so the function is now single valued on that domain. Then you can use any tool.   | Comment only |
| 11 May 2016 | <a href="#">Jianwei Tu</a>          | John,<br><br>Thank you for your reply. I actually have a relationship where some x have multiple y values. I was looking for a function similar to interp1 that does:<br><br><code>yinterp = thefunction(x,y,xgiven);</code><br><br>but I guess it does not make sense and could not be done according to what you just said.<br><br>Thanks anyway!<br><br>Jianwei  | Comment only |
| 11 May 2016 | <a href="#">John D'Errico</a>       | Jianwei,<br><br>If you have a problem with a single valued function of x, then you can just use a call to spline, pchip, or even just interp1. Just pass in the x values you want to interpolate at, and those tools can predict a result nicely.<br><br>If your relationship has multiple y values for any given x, then the above tools won't work. But then it won't make sense to be able to interpolate as a function of x then either.<br><br>So I'm not really sure why you would need interparc here.<br><br>Having said that, I suppose I can come up with a scenario. Consider a path in three dimensions, of the form<br><br>$(x, y(x), z(x))$<br><br>So y and z are both functions of x. Now, you might wish to interpolate points along that curve for given values of x. But that is still easy to do. Again, just use spline models for each of y(x) and z(x).<br><br>So I'm not sure what you might be looking for. | Comment only |
| 10 May 2016 | <a href="#">Jianwei Tu</a>          | John, is it possible to use interparc to generate a series of data points with user defined x values? Thanks, Jianwei   |              |
| 03 May 2016 | <a href="#">Mario Marín</a>         |   |              |
| 28 Mar 2016 | <a href="#">Joe Passman</a>         |   |              |
| 10 Feb 2016 | <a href="#">Jack</a>                | Another terrific and very useful routine from John D'Errico!  |              |
| 26 Jan 2016 | <a href="#">Nadim</a>               | I need a similar function that runs on Fortran. Does anyone know where I can find such a function. Thanks!  | Comment only |
| 03 Nov 2015 | <a href="#">Kevin J. Delaney</a>    | Fantastic! Just what I needed.  |              |
| 09 Sep 2015 | <a href="#">Christopher Thissen</a> | Very helpful - thanks!  |              |
| 23 Jul 2015 | <a href="#">Leonardo</a>            | Dear John,<br>thank you for your code. I would like to ask you..<br>It is in your plan to implement an upgrading also for the interpolating method "cscvn()" for closed curves ?<br>Thank you in advance.   | Comment only |
| 02 Jul 2015 | <a href="#">navid alemi</a>         | Thanks alot.  |              |
| 09 Apr 2015 | <a href="#">Harel</a>               |   |              |
| 05 Sep 2014 | <a href="#">Maximilian</a>          | Works perfect! Thanks!  |              |
| 21 Jul 2014 | <a href="#">Justin</a>              | This is amazing. Basically matlab magic.<br><br>Running on MATLAB 7.8(R2009a)   |              |
| 03 Jul 2014 | <a href="#">Steve</a>               | very nice, works great for my 2D-points!  |              |
| 11 Jun 2014 | <a href="#">Thomas</a>              | Very good code BTW.   |              |
| 03 Jun 2014 | <a href="#">John D'Errico</a>       | It is never obvious, because the error often results from some language construct of which the older MATLAB release has no understanding. So it throws some strange error.  | Comment only |

|             |                               |  |              |
|-------------|-------------------------------|--|--------------|
| 03 Jun 2014 | <a href="#">Blaise</a>        | <p>hey,</p> <p>I have upgraded to 2014A and that fixes it. sorry to bother you, should have checked this myself</p> <p>blaise</p>  | Comment only |
| 02 Jun 2014 | <a href="#">John D'Errico</a> | <p>Every time someone tells me that some code of mine fails to run on their machine with a strange looking error, especially when they have called it in a normal way, my first guess is they are trying to run it with too old of a MATLAB release.</p> <p>INTERPARC was written on MATLAB 2011a. If your release is significantly older than that, I imagine this is the problem. If not, then I can only guess you have some conflicting code on your search path, perhaps you wrote something called spline.m?</p>   | Comment only |
| 02 Jun 2014 | <a href="#">Blaise</a>        | <p>hello,</p> <p>I have just downloaded the code and tried to run it with the code taken from above:</p> <pre>theta = sort(rand(15,1))*2*pi; theta(end+1) = theta(1); px = cos(theta); py = sin(theta); pt = interparc(100,px,py,'spline');</pre> <p>however i get the error message:</p> <p>??? Reference to non-existent field 'coefs'.</p> <pre>Error in ==&gt; interparc at 317 nc = numel(sp{k}.coefs);</pre> <pre>Error in ==&gt; test at 13 pt = interparc(100,px,py,'spline');</pre> <p>Just before executing the line spl is a 1*2 cell array, with the second cell being empty and the first one is a structure (with one field, dim=1).</p> <p>I was surprised since the code looks working well for everyone else.</p> <p>Anybody knows what goes wrong?</p> <p>Best,<br/>Blaise</p>   | Comment only |
| 23 May 2014 | <a href="#">John D'Errico</a> | <p>Alias, sorry. Never published, not by me at least. The idea is a basic one though.</p> <ol style="list-style-type: none"> <li>1. Compute the cumulative linear arclength (t) along the curve. This is simply the distance along the curve using a connect-the-dots (piecewise linear) function.</li> <li>2. Fit an interpolating spline through the points in each dimension, as a function of that cumulative linear arclength. This is the classic approach to fitting a curve through a completely general set of points in any number of dimensions, especially if that curve may be a closed one. Thus I generate the curves {x(t), y(t), z(t),...}. Many tools in MATLAB use this same scheme to fit interpolating splines through arbitrarily shaped curves.</li> <li>3. Differentiate those spline functions, to get {x'(t),y'(t),z'(t), ...}. Since the functions are simply cubic polynomial segments, differentiation is trivial.</li> <li>4. Integrate the cumulative arclength differential element,</li> </ol> $\sqrt{x'^2 + y'^2 + z'^2 + \dots}$ <p>Numerical integration is employed in this step, in this case an ODE solver is used, because ODE45 has the ability to integrate a function, finding where that function crosses a list of points of interest. That is important for interparc, since it must locate points along the curve at specific distances from the start point.</p> | Comment only |
| 23 May 2014 | <a href="#">Alias Merzouk</a> | <p>It's a good work Mr. John, this is exactly what I need. But can you give us some papers/references about the used interpolation methods ?</p>   | Comment only |
| 13 May 2014 | <a href="#">Pete</a>          | <p>Superb.</p> <p>"Extremely" useful, especially in conjunction with distance2curve.m and/or slmengine.m</p> <p>I owe you many beers(!)</p>  |              |
| 06 Feb 2014 | <a href="#">Mayar</a>         | <p>Thank You Mr.John, you found my problem very well.</p> <p>With regards</p>  |              |

|             |               |   |   |
|-------------|---------------|---|---|
| 05 Feb 2014 | John D'Errico | <p>Mayar -</p> <p>I'm sorry, but you need to provide an example that fails. In fact, if I try to emulate what you SAID you did that fails, I have no problems at all.</p> <pre>x = [200 221 221 240]; y = [100 102 107 110];  interparc(10,x,y) ans = 200 100 205.69 97.159 211.63 94.893 217.88 94.532 220.9 99.706 220.98 106.06 222.27 112.2 228.01 114.09 234.15 112.49 240 110</pre> <p>It is true that if there are two consecutive points that are identical for all the variables, then it will fail. That is, this sequence of points will cause an error:</p> <pre>x = [200 221 221 240]; y = [100 102 102 110];</pre> <p>If this is indeed your issue, and NOT what you actually stated, just drop the replicates of that point! The error message indicated exactly that. Learn how to use unique to solve that, or you can find my consolidator tool on the file exchange to help with floating point near replicates.</p>   |  Comment only    |
| 04 Feb 2014 | Mayar         | <p>Dear:</p> <p>this function is not working while we have two value for the same station. for instance i have two value of Z for the same station of x. ex: x=221.0 , z= 102 next value in the data set is x=221.0 , z= 107, my data set contain such data, while i run this function, it give me error massage of " the data sites should be distinct", for t=1, and while i change t to vector suppose t=0:5:556 ( 556 end of data set), it give me again error of " all element of t must be 0&lt;= t &lt;=1 " . even I set the t as t=0:0.5:556.</p> <p>wish you give me some instruction soon</p> <p>with regards</p>   |  Comment only   |
| 24 Dec 2013 | John D'Errico | <p>There is no decoupling of the axes. Basic 2nd semester calc, or something like that. I won't claim it is obvious at all, since so many nice tricks in mathematics are only obvious after you see the trick. The arc length of a parametric curve is an integral of the sqrt of the sum of the squares of the first derivatives.</p> <p><a href="http://en.wikipedia.org/wiki/Arc_length">http://en.wikipedia.org/wiki/Arc_length</a></p> <p>Since interparc wants to locate a specific point along that curve, it simply integrates that function using an ODE solver. A nice thing about the ODE solvers in MATLAB is they can identify the location where the integrated function crosses a point of interest.</p> <p>The only decoupling of the axes is the use of splines on each dimension independently as a function of the linear chordal arc length. This allows me to define a parametric curve in an arbitrary number of dimensions, (x(t),y(t),z(t),...), and then form differentials at any point along that curve.</p> <p>So we have the arc length as:</p> $L = \int \sqrt{x'^2 + y'^2 + \dots}$ <p>where the differentials are with respect to t, the linear chordal arc length.</p> |  Comment only |
| 24 Dec 2013 | Mathew        | <p>This program is really impressive. I wanted to understand the math a bit more since it seems to be unique relative to many other interpolation methods. From what I can understand for the spline case, is it using a cubic spline interpolation over each dimension independently, then using differential equations to solve for the length of the curve for each axis? For a curve in 3 dimensions, xyz, I am confused if it is decoupling each axis or it is solving them together.</p> <p>If anyone understands how this actually works it would be great to hear.</p>  |  Comment only |
| 11 Nov 2013 | Dinesh        | <p>Thanks so much John. It really helped me in my hard time. This program is very nice, it is fast also. Even if I input smaller number of points it adjust them to the required number of equally spaced points.</p> <p>Thank you again.</p> <p>Dinesh</p>   |  Comment only |
| 03 Mar 2013 | Evgeny Pr     |   |   |
| 19 Jan 2013 | Jon           | <p>John,</p> <p>Thanks for the response! I've been playing with the linear interpolation and you're right--it's much faster. It also seems to be accurate enough for my model. Thanks for your input!</p>   |   |

|             |                               |   |   |
|-------------|-------------------------------|---|---|
| 18 Jan 2013 | <a href="#">John D'Errico</a> | <p>Jon - you can gain considerable speed by use of the 'linear' method in the code, if you can afford the loss in accuracy. It will be MUCH faster (as much as 1000 times faster), because I do the interpolation for that method in a different way from the other methods.</p> <p>Those smoother methods in this tool are forced to call an odsolver for the interpolation, so the time taken is much higher for those methods. The only possibility is that I might have offered a different ode solver than ode45, but it is the fastest solver for these problems in my tests.</p> |  Comment only    |
| 18 Jan 2013 | <a href="#">Jon</a>           | <p>Excellent code.</p> <p>I have only one complaint, which is the code takes awhile to run. For some applications this doesn't matter much, but I'm trying to resample a stream centerline after each time step in a 100's of time steps numerical model, and run times have become too large for me to use this (at least with each iteration). Do you have any suggestions for speeding it up? I tried going line-by-line through the code to find the bottleneck, but I was unsuccessful.</p>  |   |
| 16 Aug 2012 | <a href="#">John D'Errico</a> | <p>Iain - You may find pchip a better choice for that class of curve. However, you have a valid point that there is a problem for this curve when a spline is used. I've just sent in a new version that should do better. It should appear in the morning.</p>   |  Comment only   |
| 15 Aug 2012 | <a href="#">Iain</a>          | <p>Hi John. It is only a problem when pushing the spline fit quite hard, i.e. when there is a big gap between data. Then t and ti are not necessarily close. Test data and program at:<br/><a href="http://www.srcf.ucam.org/~icw26/interparc_temp/">http://www.srcf.ucam.org/~icw26/interparc_temp/</a></p> <p>Try with and without change suggested. Thanks again for the code, hope that helps.</p>  |   |
| 15 Aug 2012 | <a href="#">John D'Errico</a> | <p>Iain - correct, there was a problem in recognizing contractions. I accidentally used strcmpi, not strncmpi. Fixed, and uploaded to appear when they process it.</p> <p>As far as derivatives, there is no need to move the derivative evaluation into the loop. ppval is fully vectorized, so no explicit loop is needed for that part. In fact, putting it inside a loop will just slow things down. If you have found a case where this fails please let me know, but the code does run correctly in my tests, as I would expect.</p>  |  Comment only   |
| 15 Aug 2012 | <a href="#">Iain</a>          | <p>Excellent function, just what I needed. Thanks.</p> <p>Slight bug in evaluating dudt - the (nargout &gt; 1) block should be moved into the i=1:nt loop, and evaluated at ti, i.e.</p> <pre>for i=1:nt     %stuff - then     if nargout &gt; 1         for j = 1:ndim             dudt(i,j) = ppval(spld{j},ti);         end     end end</pre> <p>Also does not accept contractions of 'spline' etc. as equivalent arclength function does.</p>   |   |
| 26 Jul 2012 | <a href="#">John D'Errico</a> | <p>Prabhu - I don't know what you are doing to generate nans. Please send me an example that causes it to happen, as none of my tests generated NaNs.</p> <p>With one exception. There is actually one case I can find where interparc returns NaN. That is when I passed in a vector with NaNs in it already. In that case, how would one expect it not to return NaN? What logical action should it do when one asks it to interpolate a curve that is undefined at some set of points?</p>   |  Comment only |
| 26 Jul 2012 | <a href="#">Prabhu</a>        | <p>John:<br/>Thanks for this excellent code.<br/>I'm facing one problem when I use this code. It works fine in most cases. But sometimes its churning NaNs as one of the points. Can you help me out with this issue?<br/>Thanks</p>  |  Comment only |

| Updates     |     |  |
|-------------|-----|--|
| 04 Mar 2012 | 1.1 | Add csape for periodic fits, and return a function handle for external evaluation. |
| 15 Aug 2012 | 1.2 | Fixed bug in recognizing contractions  |
| 16 Aug 2012 | 1.3 | Fix for dudt at the proper points.   |

[Contact us](#)

**MathWorks**

Accelerating the pace of engineering and science

MathWorks is the leading developer of mathematical computing software for engineers and scientists.

Discover...

**Explore Products**

MATLAB

Simulink

Student Software

Hardware Support

File Exchange

**Try or Buy**

Downloads

Trial Software

Contact Sales

Pricing and Licensing

**Learn to Use**

Documentation

Tutorials

Examples

Videos and Webinars

Training

**Get Support**

Installation Help

Answers

Consulting

License Center

**About MathWorks**

Careers

Company Overview

Newsroom

Social Mission

