```
function P = InterX(L1,varargin)
%INTERX Intersection of curves
    %...Argument checks and assignment of L2
    error(nargchk(1,2,nargin));
    if nargin == 1,
        L2 = L1;    hF = @lt;   %...Avoid the inclusion of common
 points
    else
        L2 = varargin{1}; hF = @le;
    end

    %...Preliminary stuff
    x1  = L1(1,:)';  x2 = L2(1,:);
    y1  = L1(2,:)';  y2 = L2(2,:);
    dx1 = diff(x1); dy1 = diff(y1);
    dx2 = diff(x2); dy2 = diff(y2);

    %...Determine 'signed distances'
    S1 = dx1.*y1(1:end-1) - dy1.*x1(1:end-1);
    S2 = dx2.*y2(1:end-1) - dy2.*x2(1:end-1);

    C1 = feval(hF,D(bsxfun(@times,dx1,y2)-
bsxfun(@times,dy1,x2),S1),0);
    C2 = feval(hF,D((bsxfun(@times,y1,dx2)-
bsxfun(@times,x1,dy2))',S2'),0)';

    %...Obtain the segments where an intersection is expected
    [i,j] = find(C1 & C2);
    if isempty(i),P = zeros(2,0);return; end;

    %...Transpose and prepare for output
    i=i'; dx2=dx2'; dy2=dy2'; S2 = S2';
    L = dy2(j).*dx1(i) - dy1(i).*dx2(j);
    i = i(L~=0); j=j(L~=0); L=L(L~=0);  %...Avoid divisions by 0

    %...Solve system of eqs to get the common points
    P = unique([dx2(j).*S1(i) - dx1(i).*S2(j), ...
                dy2(j).*S1(i) - dy1(i).*S2(j)]./[L L],'rows')';

    function u = D(x,y)
        u =
 bsxfun(@minus,x(:,1:end-1),y).*bsxfun(@minus,x(:,2:end),y);
    end
end
```

```
Warning: NARGCHK will be removed in a future release. Use NARGINCHK or
NARGOUTCHK instead.

Error using InterX (line 5)
Not enough input arguments.
```

*Published with MATLAB® R2016b*