

```
function [LnfX,errest,cubature] = pdint(n,xyrange,funct,X,varargin)
```

```
%-----
% USAGE of "pdint".
%
% [LnfX,errest] = pdint(n,xyrange,funct,X)
% [LnfX,errest,cubature] = pdint(n,xyrange,funct,X)
% [LnfX,errest] = pdint(n,xyrange,funct,X1,X2)
% [LnfX,errest,cubature] = pdint(n,xyrange,funct,X1,X2)
% [LnfX,errest] = pdint(n,xyrange,funct,X,[],opt1,opt2,...)
% [LnfX,errest,cubature] = pdint(n,xyrange,funct,X,[],opt1,opt2,...)
% [LnfX,errest] = pdint(n,xyrange,funct,X1,X2,opt1,opt2,...)
% [LnfX,errest,cubature] = pdint(n,xyrange,funct,X1,X2,opt1,opt2,...)
%
% Compute the interpolation polynomial of degree n on the
% Padua points defined in the rectangle
% [xyrange(1),xyrange(2)] x [xyrange(3),xyrange(4)] of the function
% funct, evaluated at the target points X(:,1),X(:,2) or at the
% meshgrid(X1,X2) and the interpolation error estimate.
% Optionally, compute the cubature through the coefficient matrix.
%-----
% INPUT.
%
% n      : interpolation degree
% xyrange : a vector [a,b,c,d] defining the rectangle [a,b] x [c,d]
% fun     : function to be interpolated in the form
%           fun(x,y,opt1,opt2,...), where opt1, opt2, ... are
%           optional arguments for f
% X       : a matrix with the abscissas of the target points in the
%           first column and the ordinates in the second one
% X1,X2   : two vectors defining the (mesh)grid X1 x X2 of the
%           target points
%
% OUTPUT.
%
% LnfX    : interpolation polynomial at X(:,1),X(:,2) or
%           at meshgrid(X1,X2)
% errest  : interpolation error estimate
% cubature: cubature through the coefficient matrix
%-----
% FUNCTIONS CALLED BY THIS CODE:
% 1. pdpts
% 2. pdcfsFFT
% 3. pdval
%-----
% Authors:
%         Marco Caliari      <marco.caliari@univr.it>
%         Stefano De Marchi  <demarchi@math.unipd.it>
%         Alvise Sommariva   <alvise@math.unipd.it>
%         Marco Vianello     <marcov@math.unipd.it>
%
% Date: November 2009.
%-----

% Compute the Padua points in the rectangle defined by xyrange
Pad = pdpts(n,xyrange);
if (nargin == 4)
```

```
% Target points as X(:,1),X(:,2)
if (nargout <= 2)
    [C0f,errest] = pdcfsFFT(Pad,funct,xyrange);
else
    [C0f,errest,cubature] = pdcfsFFT(Pad,funct,xyrange);
end
LnfX = pdval(C0f,xyrange,X);
elseif (nargin >= 5)
    if (nargout <= 2)
        [C0f,errest] = pdcfsFFT(Pad,funct,xyrange,varargin{2:end});
    else
        [C0f,errest,cubature] = pdcfsFFT(Pad,funct,xyrange,varargin{2:end});
    end
    Y = varargin{1};
    if (size(X) == size(Y))
% Target points as meshgrid(X,Y)
        LnfX = pdval(C0f,xyrange,X,Y);
    else
% Target points as X(:,1),X(:,2)
        LnfX = pdval(C0f,xyrange,X);
    end
else
    error('Wrong number of input arguments')
end
```