J. Approx. Softw., Vol. W, No. X (YYYY) Z

Journal of Approximation Software

SPLISURF
Reference Manual

C. Dagnino[1] and P. Lamberti[1,⋆]

[1]*Dipartimento di Matematica "Giuseppe Peano", Università di Torino*

Received: dd/mm/yyyy – Published: DD/MM/YYYY
Communicated by: ...

Abstract

This manual contains a description of the main commands in *SPLISURF*. It refers to the manuscript *SPLISURF: a software for* C^1 *quasi-interpolating splines on criss-cross triangulations*, presented for possible publication in the Journal of Approximation Software.

It begins with a listing of entries grouped by subject area and continues with the reference entries in alphabetical order.

Keywords: Bivariate spline, Quasi-interpolation, Criss-cross triangulation, Approximation order (MSC2020: 65D07, 65D10, 41A25)

⋆ Corresponding author: paola.lamberti@unito.it

1 General scheme

B-splines	
$-$ *uniform case* $-$	
`bij`	evaluation of a $B-$spline B_{ij}
`bijplt`	display a $B-$spline B_{ij} in S
`oij`	display $Q_{ij} \cap S$
$-$ *non uniform case* $-$	
`nubij`	evaluation of a $B-$spline B^*_{ij}
`nubijplt`	display a $B-$spline B^*_{ij} either on Q^*_{ij} or in R
`nuoij`	display either the support Q^*_{ij} or $Q^*_{ij} \cap R$

Q-i splines	
`amn`	evaluation of the spline $\mathscr{A} f$
`nuamn`	evaluation of the spline $\mathscr{A}^* f$

Utilities	
`lv, lw`	generation of the functionals matrix for V_{mn}, W_{mn}
`nulv, nulw`	generation of the functionals matrix for V^*_{mn}, W^*_{mn}

Demonstrations	
`oijall`	display $Q_{ij} \cap S$
`bijall`	display B_{ij} in S
`nuoijall`	display $Q^*_{ij} \cap R$
`nubijall`	display B^*_{ij} in R
`amnf, nuamnf`	display some examples of the splines $\mathscr{A} f$ and $\mathscr{A}^* f$, respectively

Table 1: Table of functions

2 Functions

amn

Purpose Quadratic C^1 q-i spline on a uniform type-2 triangulation.

Syntax `z=amn(u,v,a,b,c,d,l)`

Description A quasi-interpolating (q-i) spline with a uniform type-2 triangulation of $[a,b]\times[c,d]$ to given data `u, v` is constructed according to the definition

$$\mathscr{A} f(x,y) = \sum_{ij} \Lambda_{ij} f B_{ij}\left(\frac{x-a}{b-a},\frac{y-c}{d-c}\right), \quad (x,y)\in[a,b]\times[c,d]$$

where $\{\Lambda_{ij} f\}$ is a given set of functionals provided by the $(m+2)(n+2)$ matrix `l`, i.e. `l(i,j)`$=\Lambda_{i-1,j-1}f,\ i=1,\dots,m+2,\ j=1,\dots,n+2$, with m the subdivisions number of $[a,b]$ and n the subdivisions number of $[c,d]$.
Therefore `amn(u,v,a,b,c,d,l)` provides the matrix `z` of size `[length(u),length(v)]` containing the values of the spline $\mathscr{A} f$ at `u`×`v`, for two real vectors `u` and `v`.
`bij` is also used to construct the spline $\mathscr{A} f$.
We should check for the seven $B-$splines B_{ij} that are non zero on one of the four triangles $T_{\bar\imath,}^{(h)}$, $h=1,2,3,4$ of the rectangle

$$R_{\bar\imath,} = \left[\frac{\bar\imath-1}{m},\frac{\bar\imath}{m}\right]\times\left[\frac{-1}{n},\frac{}{n}\right],\ \bar\imath=1,\dots,m,\ \ =1,\dots,n$$

where the point (x,y) lies, according to the definition of $\mathscr{A} f$ given in §2.1 *On uniform spline construction*, but for convenience `amn` computes the q-i spline at the point (x,y), using the nine $B-$spline B_{ij} that are non zero on $R_{\bar\imath,}$:

$B_{\bar\imath-1,-1},\ B_{\bar\imath-1,},\ B_{\bar\imath-1,+1},\ B_{\bar\imath,-1},\ B_{\bar\imath,},\ B_{\bar\imath,+1},$
$B_{\bar\imath+1,-1},\ B_{\bar\imath+1,},\ B_{\bar\imath+1,+1}.$

Examples The statements

```
l=lv(25,42,0,1,0,1,'sqrt(abs(x-y))');
u=linspace(0,1,3); v=linspace(0,1,4);
```

```
z=amn(u,v,0,1,0,1,l)
```

return

```
z =
    1.3430e-001  5.7700e-001  8.1637e-001  9.9993e-001
    7.0691e-001  4.0723e-001  4.0723e-001  7.0691e-001
    9.9993e-001  8.1637e-001  5.7700e-001  1.3430e-001
```

i.e. $V_{25,42}f$ at u×v with $f(x,y) = \sqrt{|\,x-y\,|}$ and the statements

```
l=lw(15,20,-1,2,-1,1,'max(0,sin(pi.*(x)).*sin(pi.*(y)))');
z=amn(1.5,0.7,-1,2,-1,1,l)
```

return

```
z =
    0
```

i.e. $W_{15,20}f(1.5,0.7)$ with $f(x,y) = \max\{0, \sin(\pi x)\sin(\pi y)\}$.
The demo `amnf` shows some examples of q-i splines, approximating test functions in different domains for the uniform case.

See also `bij, lv, lw`

bij

Purpose Evaluate either the $B-$spline B_{ij} or the ZP-element (uniform case).

Syntax `b=bij(u,v,m,n,i,j)`

Description `bij(u,v,m,n,i,j)` and `bij(u,v)` provide the matrix `b` of size `[length(u),length(v)]`, containing the B_{ij} and the ZP-element values at `u`$\times$`v`, respectively, where `u` and `v` are two real vectors.
B_{ij} is so defined:

$$B_{ij}(x,y) = B(mx - i + \frac{1}{2}, ny - j + \frac{1}{2}),$$

$i = 0,\ldots,m+1,\ j = 0,\ldots,n+1$, where B is the ZP-element and m and n are the subdivisions numbers of $[0,1]$ on the x-axis and on the y-axis, respectively.
The ZP-element is the B-spline function with support Q, the octagon with center at the origin of the real plane and vertices at the points $(3/2,1/2),(1/2,3/2),(-1/2,3/2),(-3/2,1/2),(-3/2,-1/2),$ $(-1/2,-3/2),(1/2,-3/2),(3/2,-1/2)$, while the $B-$spline B_{ij} has the octagon Q_{ij} as its support with centre at the point $\left(\frac{2i-1}{2m}, \frac{2j-1}{2n}\right)$ and partitioned into 25 cells.
See also §2.1 *On uniform spline construction*.

Examples The statement

```
b=bij([.6 .2],[.2 .34],4,5,3,2)
```

generates the output

```
b =
    0.3700    0.4750
    0         0
```

that is

$$\begin{matrix} B_{32}(.6,.2) & B_{32}(.6,.34) \\ B_{32}(.2,.2) & B_{32}(.2,.34) \end{matrix}$$

while the statement

```
b=bij(0,0)
```

returns

```
b =
  0.5000
```

and

```
b=bij([1 0 -1 -.75],[1 1.5 -.5])
```

returns

```
b =
     0   0   0.0625
0.1250   0   0.3750
     0   0   0.0625
0.0156   0   0.1406
```

See Fig. 2 and 3 in §2.1 *On uniform spline construction.*

See also `bijplt, oij`

bijplt

Purpose Plot the $B-$spline B_{ij} (uniform case).

Syntax `bijplt(m,n,i,j)`

Description `bijplt(m,n,i,j)` plots the $B-$spline B_{ij} in $[0,1] \times [0,1]$ for fixed values of m, n, i and j.
The $B-$spline B_{ij} is evaluated on a 30×30 uniform grid of $[0,1] \times [0,1]$ by `bij`.
If no input arguments are provided, `bijplt` plots the ZP-element in the domain $\left[-\frac{3}{2}, \frac{3}{2}\right] \times \left[-\frac{3}{2}, \frac{3}{2}\right]$.

Examples The commands

```
bijplt(3,3,3,1), pause, bijplt(3,3,0,4)
```

generate the graphs of Fig. 1 and 2, respectively.
The demo `bijall(m,n)` shows all B_{ij}'s in $[0,1] \times [0,1]$ for fixed values of m and n.

See also `bij, oij`

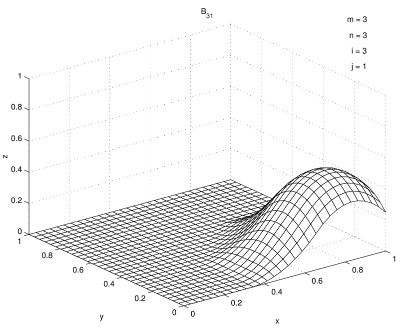

Figure 1: B_{31} in $[0,1] \times [0,1]$

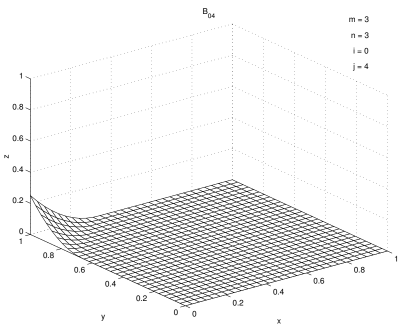

Figure 2: B_{04} in $[0,1] \times [0,1]$

lv, lw

Purpose Generate a linear functionals $\{\Lambda_{ij} f\}$ matrix (uniform case).

Syntax

```
l=lv(m,n,a,b,c,d,sf)
l=lw(m,n,a,b,c,d,sf)
```

Description `lv` and `lw` are utilities providing a matrix `l` of size $(m+2)(n+2)$ that can be used as input argument for `amn`.

`lv(m,n,a,b,c,d,sf)` generates the matrix `l` of the linear functionals $\Lambda_{ij} f$, defining the bivariate q-i spline $\mathscr{A} f = V_{mn} f$ with a uniform type-2 triangulation on $[a,b] \times [c,d]$, for a given test function f, i.e.

`l(i,j)`$=\Lambda_{i-1,j-1} f = f\left((b-a)\frac{2i-3}{2m} + a, (d-c)\frac{2j-3}{2n} + c\right),$

$i = 1,\dots,m+2,\ j = 1,\dots,n+2$, where m is the subdivisions number of $[a,b]$ and n is the subdivisions number of $[c,d]$.
`sf` is the string containing either the function f or the name of the M-file with '.m' extension in which the function f is defined.

`lw(m,n,a,b,c,d,sf)` generates the matrix `l` of the $\Lambda_{ij} f$'s defining the bivariate q-i spline $\mathscr{A} f = W_{mn} f$:

`l(i,j)`$=\Lambda_{i-1,j-1} f = 2f\left((b-a)\frac{2i-3}{2m} + a, (d-c)\frac{2j-3}{2n} + c\right)$
$\qquad -\frac{1}{4}\sum_{r=i-1}^{i}\sum_{s=j-1}^{j} f\left((b-a)\frac{r}{m} + a, (d-c)\frac{s}{n} + c\right),$

$i = 1,\dots,m+2,\ j = 1,\dots,n+2.$

Examples The statement

```
l=lv(3,3,-1,1,-1,1,'abs(x.^2+y.^2-0.25)')
```

returns

```
l =
  3.3056  1.9722  1.5278  1.9722  3.3056
  1.9722  0.6389  0.1944  0.6389  1.9722
  1.5278  0.1944  0.2500  0.1944  1.5278
  1.9722  0.6389  0.1944  0.6389  1.9722
  3.3056  1.9722  1.5278  1.9722  3.3056
```

while the command

```
l=lw(3,3,0,1,0,1,'max(0,sin(pi.*x)*sin(pi.*y))')
```

provides

```
l =
   1.3750      0        0        0      1.3750
      0     1.3750  2.7500  1.3750  -0.0000
      0     2.7500  5.5000  2.7500  -0.0000
      0     1.3750  2.7500  1.3750  -0.0000
   1.3750 -0.0000 -0.0000 -0.0000   1.3750
```

See also amn

nuamn

Purpose Quadratic C^1 q-i spline on a non uniform type-2 triangulation.

Syntax `z=nuamn(u,v,p,q,a,b,c,d,l)`

Description A quasi-interpolating (q-i) spline with a non uniform type-2 triangulation of $[a,b]\times[c,d]$ to given data `u, v` is constructed according to the definition

$\mathscr{A}^* f(x,y) = \sum_{ij} \Lambda^*_{ij} f B^*_{ij}(x,y), \quad (x,y)\in[a,b]\times[c,d]$

where $\{\Lambda^*_{ij} f\}$ is a given set of functionals provided by the $(m+2)(n+2)$ matrix `l`, i.e. `l(i,j)`$= \Lambda^*_{i-2,j-2} f,\ i=1,\dots,$ $m+2,\ j=1,\dots,n+2$, with m the subdivisions number of $[a,b]$ and n the subdivisions number of $[c,d]$.
`p` and `q` are two real vectors so defined:

`p`: $x_{-2} < x_{-1} < a = x_0 < x_1 < \ldots < x_m = b < x_{m+1} < x_{m+2}$
`q`: $y_{-2} < y_{-1} < c = y_0 < y_1 < \ldots < y_n = d < y_{n+1} < y_{n+2}$

Therefore `nuamn(u,v,p,q,a,b,c,d,l)` provides the matrix `z` of size `[length(u),length(v)]` containing the values of the spline $\mathscr{A}^* f$ at `u`$\times$`v`, for two real vectors `u` and `v`.
`nubij` is also used to construct the spline $\mathscr{A}^* f$.
We should check for the seven $B-$splines B^*_{ij} that are non zero on one of the four triangles $T^{(h)}_{\bar{\imath},}$, $h=1,2,3,4$ of the rectangle

$R_{\bar{\imath},} = [x_{\bar{\imath}}, x_{\bar{\imath}+1}]\times[y_{,} y_{+1}],\ \bar{\imath}=0,\dots,m-1,\ \ =0,\dots,n-1$

where the point (x,y) lies, according to the definition of $\mathscr{A}^* f$ given in §2.3 *On non uniform spline construction*, but for convenience `nuamn` computes the q-i spline at the point (x,y), using the nine $B-$spline B^*_{ij} that are non zero on $R_{\bar{\imath},}$:

$B^*_{\bar{\imath}-1,-1},\ B^*_{\bar{\imath}-1,},\ B^*_{\bar{\imath}-1,+1},\ B^*_{\bar{\imath},-1},\ B^*_{\bar{\imath},},\ B^*_{\bar{\imath},+1},$
$B^*_{\bar{\imath}+1,-1},\ B^*_{\bar{\imath}+1,},\ B^*_{\bar{\imath}+1,+1}.$

Examples The statements

```
p=[-0.2 -0.1 0 0.5 0.75 1 1.1 1.2];
q=[-0.2 -0.1 0 0.6 0.8 1 1.1 1.2];
l=nulv(p,q,0,1,0,1, ...
     '(1+2*exp((-3)*(10*sqrt(x.^2+y.^2)-6.7))).^(-1/2)');
u=linspace(0,1,3); v=linspace(0,1,4);
z=nuamn(u,v,p,q,0,1,0,1,l)
```

return

```
z =
    8.6539e-004  1.8244e-001  7.0349e-001  9.9971e-001
    2.6709e-001  5.4837e-001  9.1507e-001  9.9996e-001
    9.9947e-001  9.9982e-001  9.9998e-001  1.0000e+000
```

i.e. $V_{33}^{*}f$ at u$\times$v, where $f(x,y) = \left(1+2e^{-3(10r-6.7)}\right)^{-\frac{1}{2}}$ with $r = \sqrt{x^2+y^2}$ and the statements

```
p=[-1.5 -1.2 -1 0.5 1.3 2 2.1 2.5];
q=[-1.3 -1.1 -1 0.3 0.8 1 1.1 1.2];
l=nulw(p,q,-1,2,-1,1,'max(0,sin(pi.*(x)).*sin(pi.*(y)))');
z=nuamn(1.5,0.7,p,q,-1,2,-1,1,l)
```

return

```
z =
    7.4367e-002
```

i.e. $W_{33}^{*}f(1.5,0.7)$ with $f(x,y) = \max\{0, \sin(\pi x)\sin(\pi y)\}$.
The demo `nuamnf` shows some examples of q-i splines, approximating test functions in different domains for the non uniform case.

See also `nubij, nulv, nulw`

nubij

Purpose Evaluate the $B-$spline B^*_{ij} (non uniform case).

Syntax `nub=nubij(u,v,p,q,i,j)`

Description `nubij(u,v,p,q,i,j)` provides the matrix `nub` of size `[length(u), length(v)]`, containing the B^*_{ij} values at `u`×`v`, where `u` and `v` are two real vectors. `p` and `q` are two real vectors so defined:

`p`: $x_{-2} < x_{-1} < a = x_0 < x_1 < ... < x_m = b < x_{m+1} < x_{m+2}$
`q`: $y_{-2} < y_{-1} < c = y_0 < y_1 < ... < y_n = d < y_{n+1} < y_{n+2}$

The B^*_{ij} has support Q^*_{ij}, the octagon with centre at the point

$$\left(\frac{x_i+x_{i+1}}{2}, \frac{y_j+y_{j+1}}{2}\right) = \left(\frac{\mathtt{p(i+3)+p(i+4)}}{2}, \frac{\mathtt{q(j+3)+q(j+4)}}{2}\right),$$

$i = -1,\ldots,m,\ j = -1,\ldots,n$ and partitioned into 28 cells.
See also §2.3 *On non uniform spline construction.*

Examples The statements

```
p=[-1.5 -1.2 -1 0.5 1.3 2 2.1 2.5];
q=[-1.3 -1.1 -1 0.3 0.8 1 1.1 1.2];
nub=nubij([.5 -.5],[.3 -.4],p,q,1,1)
```

generate the output

```
nub =
    4.7101e-001 1.0033e-001
    5.2335e-002       0
```

that is

$$\begin{matrix} B^*_{11}(.5,.3) & B^*_{11}(.5,-.4) \\ B^*_{11}(-.5,.3) & B^*_{11}(-.5,-.4) \end{matrix}$$

See also `nuoij, nubijplt`

nubijplt

Purpose Plot the $B-$spline B^*_{ij} (non uniform case).

Syntax `nubijplt(p,q,i,j[,flag])`

Description `nubijplt(p,q,i,j,flag)` plots the $B-$spline B^*_{ij} on its support Q^*_{ij}, for any value of $flag$, while `nubijplt(p,q,i,j)` plots the B $-$spline B^*_{ij} in $[a,b]\times[c,d]$, for fixed values of p, q, i and j, where

p: $x_{-2} < x_{-1} < a = x_0 < x_1 < \ldots < x_m = b < x_{m+1} < x_{m+2}$
q: $y_{-2} < y_{-1} < c = y_0 < y_1 < \ldots < y_n = d < y_{n+1} < y_{n+2}$

The $B-$spline B^*_{ij} is evaluated on a 30×30 uniform grid of $[a,b]\times[c,d]$ by `nubij`.

Examples The commands

```
p=[-1.23 -0.5 1 4.7 7 10 10.5 11.2];
q=[0 1.7 2 3.3 5.1 6 7 8.1 9];
nubijplt(p,q,1,4), pause, nubijplt(p,q,0,1,0)
```

generate the graphs of Fig. 3 and 4, respectively.
The demo `nubijall(p,q)` shows all B^*_{ij}'s in $[a,b]\times[c,d]$ for fixed values of p and q.

See also `nubij, nuoij`

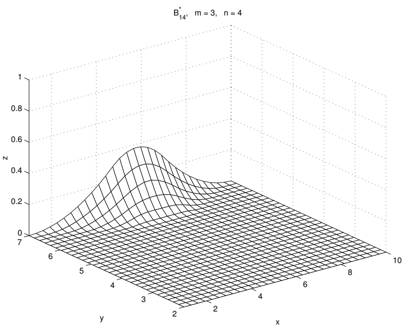

Figure 3: B^*_{14} in $[0,1]\times[0,1]$

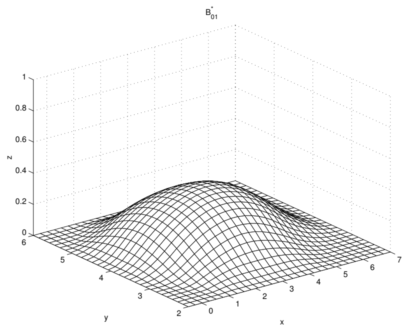

Figure 4: B^*_{01} on Q^*_{01}

nulv, nulw

Purpose Generate a linear functionals $\{\Lambda_{ij}^* f\}$ matrix (non uniform case).

Syntax

```
l=nulv(p,q,a,b,c,d,sf)
l=nulw(p,q,a,b,c,d,sf)
```

Description `nulv` and `nulw` are utilities providing a matrix `l` of size $(m+2)(n+2)$ that can be used as input argument for `nuamn`.

`nulv(p,q,a,b,c,d,sf)` generates the matrix `l` of the linear functionals $\Lambda_{ij}^* f$ defining the bivariate q-i spline $\mathscr{A}^* f = V_{mn}^* f$ with a non uniform type-2 triangulation on $[a,b] \times [c,d]$, for a given test function f, i.e.

$$\texttt{l(i,j)=}\Lambda_{i-2,j-2}^* f = f\left(\frac{x_{i-2}+x_{i-1}}{2}, \frac{y_{j-2}+y_{j-1}}{2}\right) = f\left(\frac{\texttt{p(i+1)+p(i+2)}}{2}, \frac{\texttt{q(j+1)+q(j+2)}}{2}\right),$$

$i = 1, \ldots, m+2,\ j = 1, \ldots, n+2$, where m is the subdivisions number of $[a,b]$ and n is the subdivisions number of $[c,d]$ and `p` and `q` are two real vectors so defined:

`p`: $x_{-2} < x_{-1} < a = x_0 < x_1 < \ldots < x_m = b < x_{m+1} < x_{m+2}$
`q`: $y_{-2} < y_{-1} < c = y_0 < y_1 < \ldots < y_n = d < y_{n+1} < y_{n+2}$

`sf` is the string containing either the function f or the name of the M-file with '.m' extension in which the function f is defined.

`nulw(p,q,a,b,c,d,sf)` generates the matrix `l` of the $\Lambda_{ij}^* f$'s defining the bivariate q-i spline $\mathscr{A}^* f = W_{mn}^* f$:

$$\texttt{l(i,j)=}\Lambda_{i-2,j-2} f = 2f\left(\frac{x_{i-2}+x_{i-1}}{2}, \frac{y_{j-2}+y_{j-1}}{2}\right) - \frac{1}{4}\sum_{r=i-2}^{i-1}\sum_{s=j-2}^{j-1} f(x_r, y_s),$$

$i = 1, \ldots, m+2,\ j = 1, \ldots, n+2.$

Examples The statements

```
p=[-1.3902 -1.2091 -1.0000 -0.5000 0.5000 1.0000 ...
1.2091 1.3902];
q=[-1.3902 -1.2091 -1.0000 -0.5000 0.5000 1.0000 ...
1.2091 1.3902];
l=nulv(p,q,-1,1,-1,1,'abs(x.^2+y.^2-0.25)')
```

return

```
l =
  2.1900  1.5325  0.9700  1.5325  2.1900
  1.5325  0.8750  0.3125  0.8750  1.5325
  0.9700  0.3125  0.2500  0.3125  0.9700
  1.5325  0.8750  0.3125  0.8750  1.5325
  2.1900  1.5325  0.9700  1.5325  2.1900
```

while the commands

```
p=[-0.1564 -0.0785 0 0.3536 0.5000 0.6464 ...
1.0000 1.0785 1.156];
q=[-0.2588 -0.1564 0 0.5000 1.0000 1.1564 1.2588];
l=nulw(p,q,0,1,0,1, ...
  '1/3*exp(-81/16*((x-1/2).^2+(y-1/2).^2))')
```

provide

```
l =
   0.0128  0.0617  0.0617  0.0128
   0.0335  0.1603  0.1603  0.0335
   0.0570  0.2701  0.2701  0.0570
   0.0570  0.2701  0.2701  0.0570
   0.0335  0.1603  0.1603  0.0335
   0.0128  0.0617  0.0617  0.0128
```

See also `nuamn`

nuoij

Purpose Plot either $Q^*_{ij} \cap [a,b] \times [c,d]$ or Q^*_{ij} (non uniform case).

Syntax `nuoij(p,q,i,j[,flag])`

Description `nuoij(p,q,i,j)` plots the intersection of Q^*_{ij}, support of the $B-$spline B^*_{ij}, with $[a,b] \times [c,d]$, while `nuoij(p,q,i,j,flag)` plots all the support Q^*_{ij}, for fixed values of p, q, i and j and for any value of $flag$.
Every cell of Q^*_{ij} is numbered using the 'text' function.
See §2.3 *On non uniform spline construction.*

Examples The statements

```
p=[-1.23 -0.5 1 4.7 7 10 10.5 11.2];
q=[0 1.7 2 3.3 5.1 6 7 8.1 9];
nuoij(p,q,1,4), pause, clf, nuoij(p,q,0,1,0)
```

generate the graphs of Fig. 5 and 6, respectively.
The demo `nuoijall(p,q)` shows all Q^*_{ij}'s in $[a,b] \times [c,d]$ for fixed values of p and q.

See also `nubij, nubijplt`

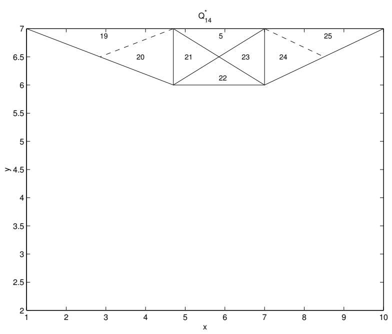

Figure 5: $Q^*_{14} \cap [a,b] \times [c,d]$

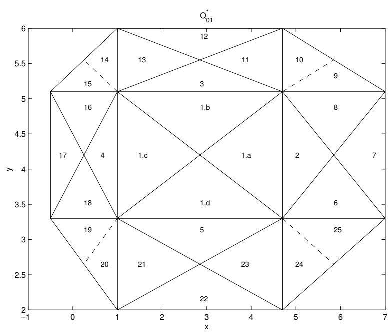

Figure 6: Q^*_{01}

oij

Purpose Plot either $Q_{ij} \cap [0,1] \times [0,1]$ or Q (uniform case).

Syntax `oij(m,n,i,j)`

Description `oij(m,n,i,j)` plots the intersection of Q_{ij}, support of the $B-$spline B_{ij}, with $[0,1] \times [0,1]$ for fixed values of m, n, i and j.
Every cell $\Delta_k^{(ij)}$ of Q_{ij} is labelled by k, using the 'text' function.
See §2.1 *On uniform spline construction.*
If no input arguments are provided, `oij` plots Q, support of the ZP-element.

Examples The statements
`oij(3,3,3,1)`
and
`oij(3,3,0,4)`
generate the graphs of Fig. 7 and 8, respectively.
The demo `oijall(m,n)` shows all Q_{ij}'s in $[0,1] \times [0,1]$ for fixed values of m and n.

See also `bij, bijplt`

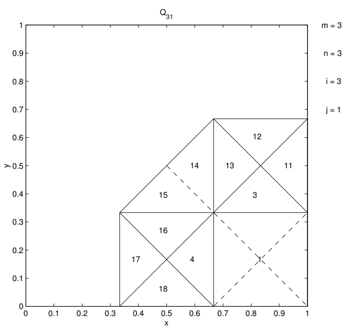

Figure 7: $Q_{31} \cap [0,1] \times [0,1]$

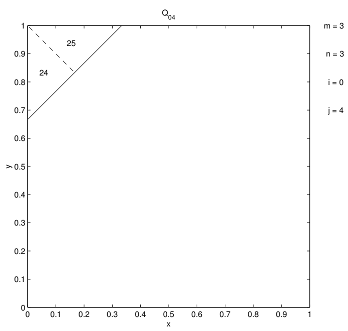

Figure 8: Q_{04}