tp Software User's Guide, version 2.7[-1-]

List of Contents

Tensor Product Spline Spaces

Let π_n be a partition $a = x_0 < x_1 < \ldots < x_n = b$ of an interval $[a, b]$, and let P_d be the space of univariate polynomials of degree d. Then

$$S_d^r(\pi_n) = \left\{ s \in C^r[a,b] : s|_{[x_{i-1},x_i]} \in P_d \right\}$$

is the univariate spline space of all r times differentiable splines that can be represented as a polynomial of degree d on each subinterval of the underlying partition.

Let S_1, S_2, and S_3 be three such spline spaces, where the values of r_i, d_i, and n_i may be different for each S_i. Then the basic tensor product space analyzed by the software is either the bivariate space $T_2 = S_1 \oplus S_2$ or the trivariate space $T_3 = S_1 \oplus S_2 \oplus S_3$. (Actually, the software treats T_2 as the special case of T_3 where $n_3 = 1$ and $r_3 = d_3 = 0$.)

It's obvious that

$$\dim S_d^r(\pi_n) = d + 1 + (n-1)(d-r) = r + 1 + n(d-r)$$

[-1-] by Peter Alfeld, pa@math.utah.edu, `www.math.utah.edu/~pa/tp`. TEX processed February 10, 2011

and

$$\dim T_3 = \prod_{i=1}^{3} [r_i + 1 + n_i(d_i - r_i)]$$

and it is easy to construct minimal determining sets for these spaces.

The tp software lets you examine subspaces of T_3 that are obtained by forcing splines in the subspace to be r_i times differentiable across a specified set of rectangular faces perpendicular to x_i, in the process effectively eliminating those faces.

Running the Software

The most common way to run the software is to give the command

java tp

This will ordinarily let you continue where you last quit and saved. It is possible to store a state of investigation in a file, say the file *discovery*. To continue work on that saved state give the command

java tp discovery

tp stores the state of an investigation in a file with the extension *.tp*. When you click on the *save and quit* button the software saves the current state in the file *tpkeepit.tp*. When you start it up the next time without specifying a different file name it will read that file and start from there. Always specify just the file name, not the extension.

If the old state cannot be restored (because the specified file does not exist, or because you have a new version of tp that is inconsistent with older .tp files) the software starts from a default state[-2-]. If you invoke the software from the applet it always starts in that state.

For complicated problems you may want to cause your system to dedicate additional memory to your job, by giving a command like

java -ms1024m -mx1024m tp

If you are running java on a PC under Windows, it may be necessary to tell java explicitly where to look for the class files, as in

java -cp . tp

To examine a particular spline space you want to follow these steps in this sequence:

1. Choose the dimension of the domain (bi or trivariate).
2. Chose the partition in each variable. The left end point of the interval in each case is 0, without loss of generality. The x_i must be integer. Usually they are evenly spaced, but you can change that.
3. Decide which rectangular faces (in the 3D case) or edges (in the 2D) should be effectively absent. The software simulates their removal by setting the degree of smoothness across them equal to the polynomial degree in the appropriate direction.
4. Choose the values of r and d. These may be different for each variable.

[-2-] i.e., $n_i = d_i = 2, r_i = 1$.

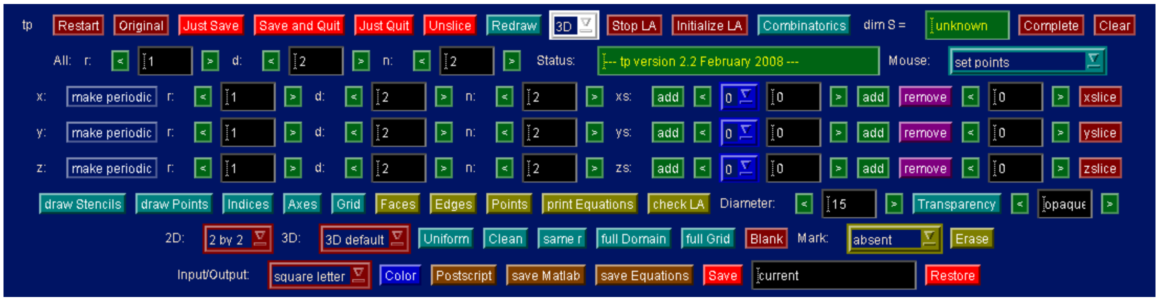
tp Restart Original Just Save Save and Quit Just Quit Unslice Redraw 3D Stop LA Initialize LA Combinatorics dim S = unknown Complete Clear
All: r: < 1 > d: < 2 > n: < 2 > Status: --- tp version 2.2 February 2008 --- Mouse: set points
x: make periodic r: < 1 > d: < 2 > n: < 2 > xs: add < 0 0 > add remove < 0 > xslice
y: make periodic r: < 1 > d: < 2 > n: < 2 > ys: add < 0 0 > add remove < 0 > yslice
z: make periodic r: < 1 > d: < 2 > n: < 2 > zs: add < 0 0 > add remove < 0 > zslice
draw Stencils draw Points Indices Axes Grid Faces Edges Points print Equations check LA Diameter: < 15 > Transparency < opaque >
2D: 2 by 2 3D: 3D default Uniform Clean same r full Domain full Grid Blank Mark: absent Erase
Input/Output: square letter Color Postscript save Matlab save Equations Save current Restore

Figure 1. The Control Panel.

5. The splines are thought of as being represented in tensor product Bernstein Bézier form and several drawing windows display their domain points. You click on those points to include them in a minimal determining set.

The Control Panel

Figure 1 shows a (reduced size) image of the Control Panel. You'll be better off running the software and looking at your actual Control Panel, trying out things as you read this guide, but I'll use the Figure for reference. It shows the full version of the Control Panel as it is used for the examination of a trivariate space using the software directly (rather than through a browser). If you are examining a bivariate space the fifth row (marked with *z-r:* on the left), and some of the controls are absent. If you invoke the software through a browser^{-3-} the last line is missing (and you won't be able to save your work). For better clarity, the control panel also has a different color when you examine bivariate spaces.

Switching between bi- and trivariate

The menu in the top row of the Control Panel (showing **3D** in Figure 1) lets you switch back and forth between bi and trivariate spline spaces. A few settings are preserved when switching, but basically the first thing you want to do is to set that menu to **2D** or **3D** as required.

Quitting and Saving

The button "save and quit" in the top row attempts to save your work and closes all windows. The button "Just Quit" quits without saving. Similarly, the button "just save" saves without quitting. If you subsequently "just quit" you start the next time at the point where you "just saved".

At any stage you can specify a file name in the text field in the last row and click on "Save" in that row to store your work in that named file. Similarly, by choosing the appropriate file name, and clicking "Restore" you can continue where you left off earlier.

For the saving and restoring feature to work you need to be connected to the directory containing the software, and you need to have write access in that directory. Some systems may let you specify other directories as part of the filename.

Saving files have the extension ".tp". So if you specify "current" (the default) as the file name, the saving file will be "current.tp". The saving files are ordinary text files containing integers and boolean values. Most, but not all, settings are saved, as are the current state of the underlying Linear Algebra, and the current choices of the minimal determining set.

If you want to restore the default state click on the button labeled "Original" in the top row.

Saving Equations

Clicking on "save Equations" will write the smoothness conditions to a file whose name is given in the text field in the last row, and whose extension is ".eqs". Clicking on "save Matlab" will do the same thing and also write a file with the extension ".m" that will define the coefficient matrix in Matlab notation and have Matlab compute its rank. The matrix is defined as a full matrix and may be too large to be properly read by Matlab.

Saving Postscript files

The three leftmost items in the Input/Output row can be used to save the space and slice drawings as postscript files. The menu can be used to select the size of the printer paper, or, equivalently, the aspect ratio of the drawing. The choice affects the x and y ranges in the postscript file, with 72 units corresponding to one inch. The following choices are available:

- **full letter:** Use the whole page of American standard writing paper, i.e., 8.5 × 11 inches.
- **screen letter:** Use the same aspect ratio as the screen, but fit it into 8.5 × 11 inches.
- **square letter:** Make the aspect ratio 1 and fit onto standard letter.
- **legal:** Fill American legal format, 8.5 × 11 inches.
- **11 by 17:** Fill 11 × 17 inches.
- **17 by 22:** Fill 17 × 22 inches.
- **A4:** Fill European A4 format (210 × 297 millimeters.)

The button labeled "Color" or "Gray" can be used to choose color or gray scale. The postscript file also contains statements that can be commented or uncommented to switch from one to the other.

Clicking on the button labeled "Postscript" actually saves the file.

Current Status

The dimension text field in the top row shows the dimension of the current spline space (if known), and the large status field in the second row gives additional information. Usually it shows the number of domain points, the number of points that have been assigned (i.e., included in the growing MDS), the number of points that have been implied by previously made assignments, and the number of points that still need to be assigned. During the initialization of the linear algebra it also shows the status of that process.

Choosing the partition

The underlying partition is determined by the values of r, d, and n, and the set of knots, in each of the variables x, and y, (and z in the trivariate case). There are various ways to change these values while maintaining as much as possible the current structure of the domain with respect to faces that have been previously removed. The best way to see how this works is to experiment with the controls. The Button labeled "full Grid" restores all previously removed faces and lets you start over with the full tp space.

If the r, d, and n values are the same for all (two or three) variables you can change them using the items in the second row of the Control Panel. The text fields and associated switches work as they to for the 2D and 3D MDS software.

The next two (or three) rows let you make changes individually for each of the variables. You can choose values of r, d, and n as usual. The obvious restrictions apply:

$$0 \leq r \leq d \quad \text{and} \quad n \geq 1.$$

The following explanation is in terms of the variable x. Similar statements apply to the variable being y or z. By default, $x_i = i$, $i = 0, \ldots, n$. The dimension of the full tensor product space is independent of the spacing. However, you can change the spacing with the remaining items at the end of each of rows 3, 4, and 5. The menu selects i, and the text field shows the value of x_i. The green buttons let you scroll through the list, and you can also choose an x_i by using the menu next to the text field. The "add" buttons add a knot to the left or right of the currently displayed x_i. "Remove" removes x_i from the partition. Entering a numerical value in the text field changes the value of x_i. In all cases, all other values are adjusted appropriately, and the left point is always zero. The "slice" buttons and their associated coefficient text fields are present only in the 3D case. Clicking on them displays a slice showing the bivariate spline space at the appropriate value of x_i and a coefficient $j = 0, \ldots, d-1$. To see how the slicing works it is best to try out the feature after clicking on the button "Indices". This will show the internal reference number of the domain points, which will be the same in every window in which a point is drawn.

The Space Window

Figure 2 shows the default (3D) space drawing for reference. Of course you want to experiment with your own as you read this.

This shows a cube. Each partition has two subintervals, and the polynomial degree is $d = 2$ in all three directions. The value of smoothness is $r = 1$ in each case, but this is not indicated by the drawing. Solid black lines indicate the partition, thin blue lines indicated the Bernstein Bézier grid, and colored circles indicate the domain points. The three faces containing the points 32, 56, and 38, respectively, have been removed. Nine coefficients have been set corresponding to domain points at the bottom of the cube. Some coefficients (for example, that corresponding to the point 123) are implied and not drawn.

You can see the three dimensional structure of the picture by rotating the cube. To do so drag the mouse with the right button depressed. (Alternatively, you can hold down the Control key and drag with the left mouse button.)

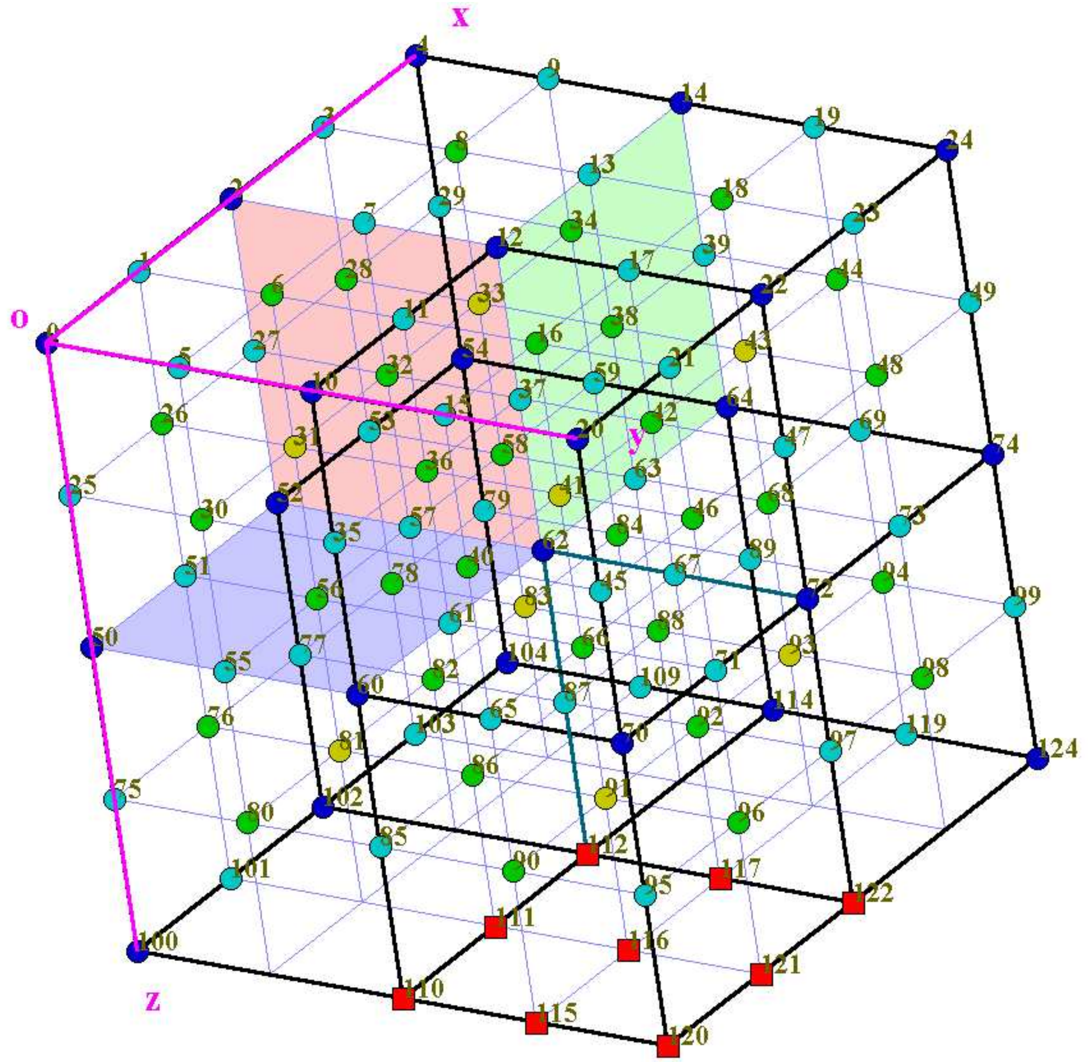

Figure 2. The Space Drawing.

There are a total of 125 control points and the dimension of the full tensor product space is 64. The colors of the control points indicate their location with the respect to one of the 8 small cubes. Blue means vertex of such a cube, cyan means interior of an edge, green means interior of a face, and yellow means interior of a cube. If a domain point is chosen to be in the MDS it is drawn as a red square box. If it is implied by points already included in the MDS it is not drawn at all.

Constructing a Minimal Determining Set

To construct a minimal determining set select “set points” or “set facets: in the mouse menu.

To include a point in the MDS, or remove it from the MDS, just click on it with the left mouse button. The first time you do this the underlying Linear Algebra will be initialized. This may take a significant amount of time, depending on the complexity of the underlying spline space.

If you have selected "set facets" then clicking on a point P causes every possible point in the facet (vertex, edge, face, or cube) of which P is an interior point to be set or removed, according to the status of P

If you have selected "set globs" then clicking on a point P causes every point in the vertex, edge, or face glob containing P, or, if P does not enter any smoothness conditions, then all similar points in the same rectangle or cube, to be set or removed, according to the status of P.

Note that if the mouse action is set to removal of edges or domain parts, after initialization of the linear algebra (by clicking on "Initialize LA") the mouse action is automatically set to *set points*. The purpose of this feature is to avoid wasting the possibly significant initialization effort by accidentally changing the domain, degree of smoothness, or polynomial degree.

2D Slices

2D slices of the three dimensional domain look similar, and act similarly. Indeed, in the 2D case there is only one available drawing window, corresponding to the slice $z = 0$.

Figure 3 shows a slice obtained in the 2D mode. In this case, $r = 1$ and $d = 3$ in both directions. There are a total of 256 domain points, and the dimension of the spline space is 96. There are a total of 25 squares (5 subintervals in each direction). 8 edges have been removed to form a two dimensional ring. Thus there a total of 18 regions, and the spline is a polynomial (of degree 5 in each variable) on each of those regions. The regions are colored so that neighboring regions have different colors. (Two regions are neighbors if they share at least one point.) The red squares indicate the default minimal determining set (for $d = 3$ and $r = 1$). (To obtain that MDS, form the partition as indicated, and click on "complete" in the Control Panel.) All other domain points are implied, and are not shown. Clicking on any of the red points will make that point become round and assume its original color, and it will cause several other round points to reappear as well.

If you are in 3D mode, the background color of the slice indicates its direction. The direction and level are also stated in the title of the slice.

Drawing Stencils

For better clarity, the stencils of the equations can be shown in the slice drawings. These are diamond shaped polygons covering the points entering a specific equation. They can be drawn either in a solid or a transparent color. Transparent stencils may take a significant amount of time to be drawn, depending on your particular implementation of java.

Marking Vertices

In slices, vertices of a given category can be marked, by using the Mark Menu. Selecting a vertex type will circle all such vertices in each slice. The mark will disappear when the slice is redrawn, and it can be erased by clicking on the erase button. The different types

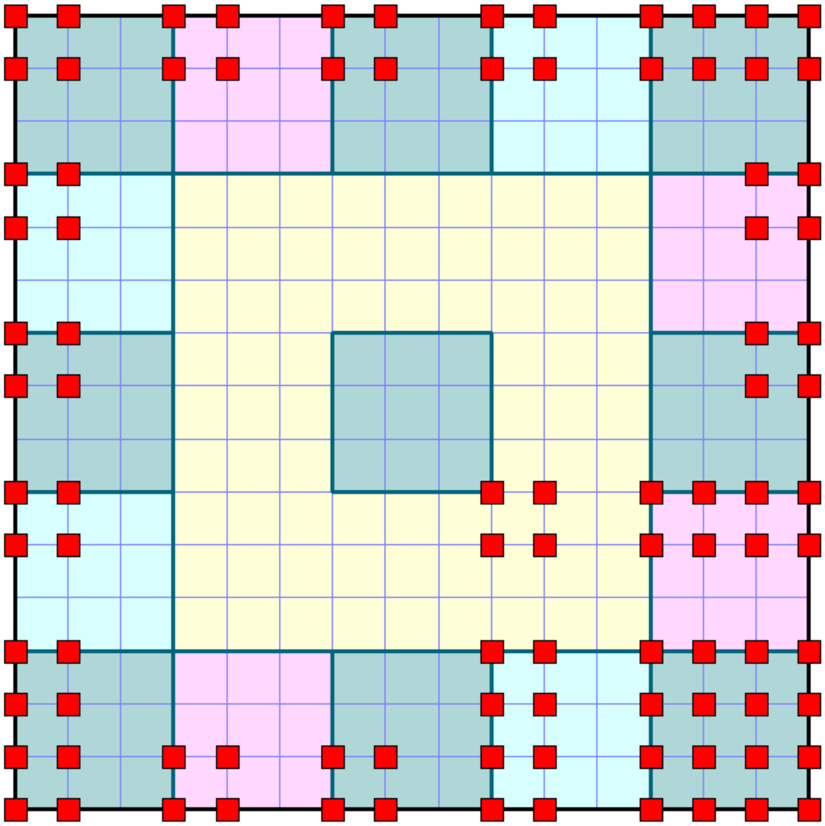

Figure 3. A slice.

of vertices are illustrated in Figure 4. They are:

— **absent** vertices, e.g., the upper left corner.

— **1 region vertices** that are vertices of 1 region only, such as vertices 1, 16, and 29.

— **bdry vertex** that lie on the boundary and share two regions, such as vertices 43 or 27.

— **butterfly** vertices where two regions and two outsides meet, such as vertex 8.

— **touch** vertices where 2 regions and one outside meet, such as vertex 24.

— **hook** vertices where one outside and two regions meet, as at vertex 29.

— **reentrant** vertices where three regions and one outside meet, as at vertex 36 or 37.

— **L** vertices where 2 regions meet, such as at vertex 18 or 25. Vertex 18 is special in that its two edges are each the stem of a T-vertex. Such L vertices are marked with a square box instead of a circle.

— **T** vertices where three regions meet as at vertex 11 or 19.

— **cross** vertices where four regions meet, as at vertex 12.

— **pinch** vertices where two regions meet as at vertex 32.

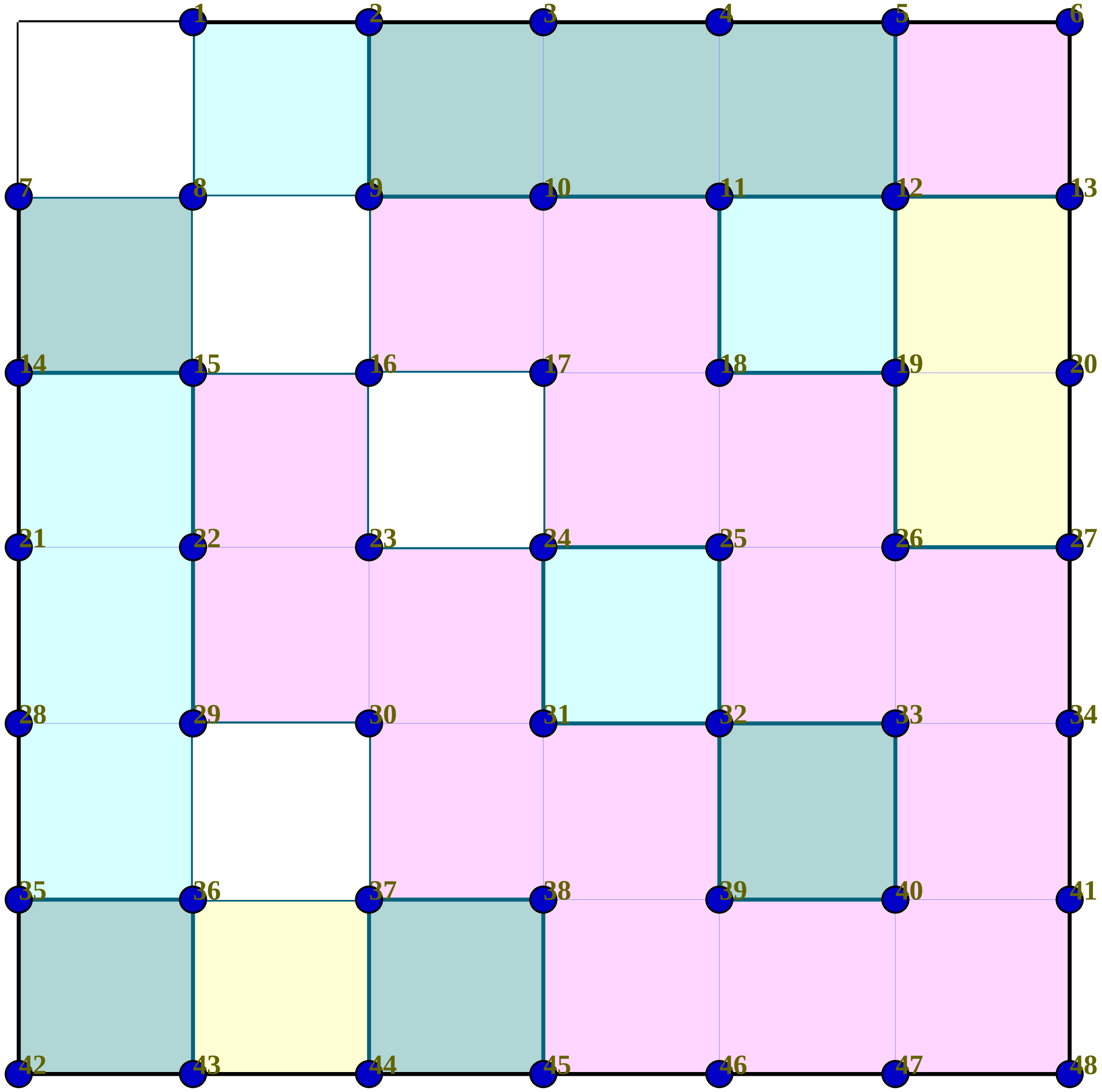

Figure 4. All Vertex Types.

— **in line** vertices that lie on a straight line segment separating two regions, such vertices 10 and 22.

Slice Combinatorics

Clicking on the combinatorics button will compute the combinatorics of each displayed slice. For example, the combinatorics of the domain in Figure 4 are as follows:

```
Combinatorics:  z slice, Level = 0 z = 0 coefficient = 0
```

```
13 regions
 V0  = 1 removed vertices:     0
 V1  = 18 one region vertices:  1 3 4 6 7 16 17 20 21 23 28 30 34 41 42 46 47
48
 L   = 2 inline vertices:      10 22

 28 actual vertices:
 VB  = 9 boundary vertices:    2 5 13 14 27 35 43 44 45
 B2  = 1 butterfly vertices:   8
 B2^ = 1 touch vertices:       24
 B2~ = 1 hook vertices:        29
 B3  = 4 reentrant vertices:   9 15 36 37
 VL  = 8 L vertices:           18 25 26 31 33 38 39 40
 VT  = 2 T vertices:           11 19
 V+  = 1 cross vertices:       12
 VP  = 1 pinch vertices:       32
 horizontal line segments:
[9,11] [11,12] [12,13] [14,15] [18,19] [24,25] [26,27] [31,32] [32,33] [35,36]
[37,38] [39,40]
 EI_h = 12
 vertical line segments:
[15,29] [36,43] [2,9] [37,44] [24,31] [38,45] [11,18] [25,32] [32,39] [5,12]
[12,19] [19,26] [33,40]
 EI_v = 13
 EI = 25
```

If $d > 2r$ the following lower bound is also computed and displayed:

$$\begin{aligned}\dim S_d^r \geq (d+1)^2 R - (r+1)(d-2r-1)E_I - (r+1)^2\big(V_B + \hat{B}_2 + \\ + \tilde{B}_2 + 2B_3 + V_L + 2V_T + 3V_+ + 2V_P\big) - (2r+1)B_2.\end{aligned} \tag{1}$$

Removing Faces or Edges

You can remove faces in 3D mode, or edges in 2D mode. To do so, select “remove faces” (or “remove edges”) in the menu in the next to last row of the Control Panel, and then click on a domain point. This will toggle the status (absent or present) of every interior face that contains the point.

Clicking again undoes your choice. The effect is independent of the drawing window in which you click on the point. Faces that have been removed are indicated by the absence of edges in the 2D windows, and by color in the 3D window.

It is possible to leave isolated faces in a region that has just one polynomial associated with it. You can remove these faces by clicking on the “Clean” button. Otherwise, they are removed by the software when the Linear Algebra is initialized. This enhances clarity of the drawings, but of course it does not affect the dimension of the space.

Refining Regions

You can refine regions without affecting neighboring regions. To do so select *refine regions* on the mouse menu and click on suitable points. Selecting *add empty layer* and clicking on a point adds new x, y, and z layers containing no faces. (Those new layers can then be activated by adding faces as described above.)

Varying the smoothness degree

Associated with each of the three coordinated directions is a degree of differentiability that can depend on the direction as, described above. You can also vary the degree of smoothness across individual faces. To do so select “increase smoothness” or decrease smoothness in the Mouse menu. The click on a point to increase or decrease the smoothness across all interior faces sharing that point. The various degrees of smoothness are indicated by color and edge thickness in the slice drawings, but not in the space window. The lowest possible degree of smoothness is 0, i.e., just continuity. It is indicated in a slice drawing by a dashed black line.

To reset all smoothness values to their defaults click on the Button labeled “same r”.

The absence of faces is modeled by having the degree of smoothness equal to the relevant polynomial degree. However, missing faces are treated as non-existent and are not affected by smoothness changes. To make them susceptible to smoothness changes you first need to make them present again, by selecting “add/remove faces” and clicking on them.

When “increase” or “decrease” smoothness is selected, clicking with the middle or right Mouse Button will have the opposite effect, of decreasing or increasing the smoothness, respectively.

Removing Parts of the Domain

A connected region can be removed entirely from the domain by clicking on one of its interior points. Clicking on the point again will recreate that region. Thus you can model holes and other forms of non rectangular regions. Missing boxes are indicated by Gray or Background Color. You can switch from one mode to the other by clicking on the “Blank” Button.

Periodic Splines

It is possible to require periodicity in any or all of the three coordinate directions. To make a spline periodic click on the appropriate button labeled “make periodic”. Periodicity is indicated by that button turning yellow, and by vertical bars in the slice drawings. For non-periodic splines there are no smoothness c conditions across boundary edges.

Precomputed Configurations

The two buttons labeled “tp” let you set up an ordinary 2D or 3D tensor product space. Applicable current settings are used.

The two menus labeled “2D” and “3D” let you select some precomputed configurations. The lists are short at present but may grow in the future. When selecting an item on one of those two menus the software attempts to read the relevant information from a file[−4−].

[−4−] The files come with the “message” that you downloaded. Their names are “2Di.tp” and “3Di.tp” where $i = 0, 1, 2, \ldots$.

If that's not possible it tires to read the information over the net. If that fails it resorts to the 3D default stage.

The following configurations are available:

2D:

2D default A 2×2 grid.

T A standard T-spline, actually the default 2×2 grid, with one edge removed.

Ring A ring embedded in a $5x5$ grid.

2stair Two steps of a stair case

3stair Three steps of a stair case

3 by 3 A 3 by 3 example with 5 regions

5 by 5 A 5 by 5 example showing some counterintuitive behavior.

3 by 4 A more compact example showing similar behavior. S_2^1 on the center horizontal line is polynomial.

hole A simple domain containing a hole.

all A domain exhibiting all vertex types, shown in Figure 4.

puzzle A domain where the lower bound described in (1) fails.

horseshoe A domain with a disconnected composite edge.

3D:

2x2x The $2 \times 2 \times 2$ default cube.

twist A tour of the $3 \times 3 \times 3$ cube starting in the top left front corner, walking along the edges of the cube, and returning to that corner.

Torus A torus embedded in a $5 \times 5 \times 3$ box.

Cave A simple domain containing a cave that is inaccessible from the outside.

T Tensor product of an interval and a 2D T vertex.

L Tensor product of an interval and a 2D L vertex.

tripod A region that's a cube with three cubes attached, embedded in a cube.

sandwich Four layers, the top and bottom of which are polynomials.

stair Four 3×3 layers, with a a 2×2 region spiraling up.

2stair Two layers that capture part of the problem in the preceding configuration.

2stairtp Obtained from the previous domain by removing two faces. This actually complicates the lower bound calculation.

mushroom Obtained from a $2 \times 2 \times 2$ cube by making the upper layer into one box. The central vertex on the floor of that box is in the interior of a two-dimensional face.

hangar Similar to the mushroom, except there are only three rooms downstairs.

Miscellaneous Commands

Following is a list of items available on the Control Panel, including all those that have not yet been mentioned. The list is ordered alphabetically.

Axes Toggles on and off the drawing of coordinate axes. Note that by java conventions the origin is in the upper left corner of a drawing.

Check LA Computes and displays information about the current state of the Linear Algebra, specifically the dimensions of the current matrix and the percentage of non-zero entries. It also checks that all rows add to zero, as they must. (If they don't please let me know.)

Clean Deletes isolate faces or edges in a region that has just one polynomial associated with it. These faces are also removed by the software when the Linear Algebra is initialized. This enhances clarity, but it does not affect the dimension of the space.

Clear Remove all points so far chosen from the current minimal determining set.

Complete Show the current internally maintained minimal determining set.

Diameter is the diameter of the domain points measured in pixels. It can be adjusted up or down as necessary.

Draw Points toggles the drawing of domain points. Not drawing them may be useful during the design of the domain.

Edges causes a similar report edges and their properties printed to standard output.

print Equations toggles on or off the printing of the underlying equations to standard output during the initialization of the linear algebra.

Faces causes report on existing faces and their properties to standard output.

full Grid restores all removed faces.

Grid toggles drawing of the B-form grid.

Indices toggles drawing the internally used indices of the domain points. This maybe useful for noting correspondences between slices and the 3D window.

Initialize LA Starts the preprocessing. This is also caused, if necessary, by other actions, such as clicking on a domain point in the set coefficient mode.

Original Restores the default state, identical to the starting date for the web version.

Points prints a list of domain points and their properties to standard output.

Redraw causes all windows to be redrawn. This should never be necessary. You can also cause any particular drawing window to be redrawn by focusing on it and pressing any key on the keyboard.

Restart restarts at the state at which you first invoked the software in this session.

set points If selected in the menu in the bottom row, will cause the selected point to be added to, or removed from, the current minimal determining set.

set facets If selected in the menu in the bottom row, will cause all points to be processed that lie in the same edge or face as the selected point.

Stop LA Stops the initial processing of the linear system (which may be time consuming.)

Transparency Faces that are removed in the 3D mode are indicated by color according to their direction. These color can be chosen to be transparent, and their opaqueness can be adjusted. (Transparency ranges from 0 (invisible) to 255 (opaque)).

Uniform makes all slices spacings equal.

Unslice makes all slices invisible in the 3D mode.

Dimension Formulas

Suppose that the smoothness r and the polynomial degree $d > 2r$ are the same in all directions, and let S_d^r be the usual space of C^r functions that are polynomial of degree d in each variable.

Then we expect in the 2D case that

$$\dim S_d^r = (r+1)^2 \hat{F}_0 + (r+1)(d-2r-1)\hat{F}_1 + (d-2r-1)^2 \hat{F}_2 \tag{2}$$

and in the 3D case

$$\dim S_d^r = (r+1)^3 \hat{F}_0 + (r+1)^2(d-2r-1)\hat{F}_1 + (r+1)(d-2r-1)^2 F_2 + (d-2r-1)^3 \hat{F}_3 \tag{3}$$

where $\hat{F}_0$ and $\hat{F}_1$ are suitable numbers of vertices and edges, $\hat{F}_2$ is the number of regions in the 2D case and the number of faces in the 3D case, and $\hat{F}_3$ is the number of 3-dimensional regions in the 3D case.

Proving this, and obtaining geometric definitions of those hat quantities is an ongoing project.

However, assuming that (2) and (3) are in fact true, the software can compute the quantities $\hat{F}_i$, $i = 0, 1, 2, 3$. To obtain them, simply click on the red button labeled “Hats”. This will compute the dimensions of S_d^0, for $d = 1, 2, 3$ in the 2D case, or $d = 1, 2, 3, 4$ in the 3D case, and then use those data to compute the $\hat{F}_i$.

Before so doing it stores the current state of the configuration, and restores it after the computation of the $\hat{F}_i$.

Following are two Tables giving the relevant $\hat{F}_i$ for the configurations that have been built into the software.

configuration	$\hat{F}_0$	$\hat{F}_1$	$\hat{F}_2$
2 by 2	9	12	4
T	7	9	3
ring	24	40	18
2stair	8	11	4
3stair	14	22	9
3 by 3	9	13	5
5 by 5	20	35	16
3 by 4	12	19	8
hole	16	24	8
all vs	16	27	13
puzzle	9	12	5
horseshoe	6	7	3
L	5	6	2

$\hat{F}_i$ **in 2D case**

configuration	$\hat{F}_0$	$\hat{F}_1$	$\hat{F}_2$	$\hat{F}_3$
2x2x2	27	54	36	8
twist	24	60	54	16
torus	120	296	246	68
cave	100	234	180	44
T	14	25	15	3
L	10	17	10	2
tripod	15	30	21	5
sandwich	32	80	69	20
small cave	64	144	108	26
stair	64	109	88	24
2stair	18	36	25	6
2stairtp	16	30	19	4
mushroom	17	33	22	5
hangar	15	28	18	4

$\hat{F}_i$ **in 3D case**

Technical Comments

- The software sets up an integer linear system of smoothness conditions and processes that system in residual arithmetic modulo the Mersenne prime number

  $$P = 2^{31} - 1 = 2,147,483,647$$

  This is the largest integer that can be expressed as an *int* in java.

- The relevant matrix is kept in an integer array if possible, or a hash table if necessary because of memory requirements. Hashing slows down the Linear Algebra, but does make it possible to handle larger problems.

- Removal of edges and faces is simulated by enforcing high order smoothness conditions. This increases the number of linear equations that must be analyzed, but it allows you to study which Bézier coefficients can be traded for others.

- The software has been tested on Sun workstation running Unix, PCs running Windows XP, and Macs running Mac OS X (Darwin).