1986

Realistic PDE Solutions for Non-Rectangular Domains

Calvin J. Riobbens

John R. Rice
Purdue University, jrr@cs.purdue.edu

Report Number:
86-639

REALISTIC PDE SOLUTIONS FOR NON-RECTANGULAR DOMAINS

Calvin J. Ribbens
John R. Rice

CSD-TR-639
December 1986

REALISTIC PDE SOLUTIONS FOR NON-RECTANGULAR DOMAINS

Calvin J. Ribbens and John R. Rice

CSD-TR 639
December 22, 1986

Abstract

We present an algorithm to define exact solutions of realistic elliptic problems on non-rectangular domains. This allows us to create a population of elliptic PDE problems with the following characteristics:

1. The elliptic operators are second order, linear and of general nature.
2. The domains are non-rectangular.
3. The true solutions are known exactly and can have the singularities and boundary layers that one expects in applications.
4. The population allows very large samples to be taken for statistical studies.

We also enlarge the population of domains given in [Rice, 1984] to 25 domains, each with two parameters. These domains and true solutions are incorporated in the ELLPACK system to allow studies of PDE software performance.

1. OBJECTIVES

Our objective is to define a population of elliptic partial differential equation (PDE) problems on a set of non-rectangular domains. These problems must have exactly known solutions and yet be realistic in nature. It is trivial to take the PDE population of [Dyksen, Houstis and Rice, 1981] and simply change the domains. However, one does not obtain realistic problems because these solutions do not have the singularities and boundary layers that are characteristic of real applications. For further background on the use of such populations see [Boisvert, Houstis and Rice, 1979], [Rice, 1979] and [Rice, 1986]. The specific objective is to create a population of elliptic PDE problems with the following characteristics:

1. The elliptic operators are second order, linear and of general nature.
2. The domains are non-rectangular.
3. The true solutions are known exactly.
4. The true solutions have the singularities and/or boundary layers that one expects in applications.
5. The populations have parameters so that very large samples can be taken for statistical studies of PDE software performance.

Our main task is to meet the fourth objective. We construct true solutions that are the sums of smooth functions, functions with singularities at domain corners (or elsewhere) and functions that have boundary layers along domain sides. The latter is the principal difficulty. As with many other geometric problems, it seems easy to define a boundary layer along a curved side which smoothly merges into the neighboring boundary pieces. In fact, this is quite difficult to do in general and the construction we use is rather complicated and not entirely fool-proof. This construction is presented in detail in Section 2.

We also extend the population of domains given by [Rice, 1984] in two ways. First, every

domain now has two parameters. Second, five more domains are added to give a total of 25 domains. These are described in Appendix 2. Actual precise definitions are given in Fortran programs that are part of the ELLPACK system.

2. THE TRUE SOLUTION

We construct true solutions which are linear combinations of three functions — a smooth function, a function with a singularity at a specified point, and a function with a boundary layer along a specified domain side. First and second partial derivatives with respect to x and y are also available. The mixed partial derivative is not available. The generic function $true(x, y)$ may be defined as follows:

$$true(x, y) = smooth(x, y) + singpt(x, y) + blayer(x, y) ,$$

where

$$smooth(x, y) = \left[\cos(b*y) + \sin(b*(x - y))\right] * (1 + a*\sin(b*x/2))$$
$$singpt(x, y) = as * \left[distp(x, y)\right]^{bs}$$
$$blayer(x, y) = ab * w1(distb(x, y)) * w2(param(x, y)) * e^{-bb * distb(x, y)}$$

Two parameters, a and b vary the size and the oscillation frequency, respectively, of the smooth term. If $a = b = 0$, then $smooth(x,y) = 1$.

The parameters as and bs determine the magnitude and exponent, respectively, of the point singularity term. The function $distp(x,y)$ is simply the distance from a point (x, y) to a specified point (xs, ys). The point (xs, ys) is chosen in one of two ways: it may be set by indicating a boundary piece *is* whose first point will be the corner where the singulariry is found, or the user may enter coordinate values for a point (xs, ys) lying anywhere.

The function $blayer(x,y)$ is used to define a boundary layer along a specified domain piece. A parameter ib selects the desired piece. Boundary pieces in the ELLPACK system are numbered in the order they are defined by the user. The parameters ab and bb determine the magnitude and the decay exponent, respectively, of the boundary layer. The function $distb(x,y)$ is

simply the minimum distance from a point (x, y) to the specified boundary piece *ib*. This value is computed numerically using a modified bisection scheme. Boundary curves are defined parametrically in ELLPACK and hence the computation of *distb*(x,y) normally requires many evaluations of the curve parameterization functions. We find that it is best to do this computation in double precision on 32-bit machines.

Two "special cases" complicate the computation of *distb*(x, y), and *blayer*(x, y) as a whole. The first is that the closest point to (x, y) on a boundary piece with large curvature may change dramatically with only a small change in x or y. In fact the closest point might not even be unique. As a simple example, suppose the domain is a circle and we would like *true*(x, y) to have a boundary layer around the entire boundary. At the center of this circle *distb*(x, y) is itself continuous, but its derivatives with respect to x and y are clearly discontinuous. This discontinuity introduces an extra difficulty into *true*(x, y) that we do not intend. We avoid this problem by introducing the weight function $w1$(*distb*(x, y)). The effect of $w1$ is to make sure the boundary layer term goes to zero for points close to or beyond the center of curvature of some part of the selected boundary piece. More specifically, our method is to first estimate r_{min}, the minimum radius of curvature at a point on piece *ib*. This is done only once, when a new value of *ib* is selected. The weight function $w1$ is then defined to be zero if dist$b(x, y) > r_{min}/2$, and is defined by a quintic which rises from 0 to 1 as dist$b(x, y)$ goes from $r_{min}/2$ to 0. In this way, the boundary layer is kept close to the boundary if the piece has large curvature.

The second special case occurs when the closest boundary point to a point (x, y) is an endpoint. Again, this introduces unwanted discontinuities in the derivatives of *distb*(x, y) and hence into *true*(x, y). We handle this case by computing an extension to the boundary piece and then finding the closest point on that extension. The extension is simply a curve extending from the endpoint with the same slope and curvature as we estimate the boundary piece to have at that endpoint. This succeeds in preserving the desired continuity in *distb*(x, y), but it has the

unwanted side-effect in some cases of continuing the boundary layer along the next boundary piece or even out into the domain. A second function, $w2(param(x, y))$ is used to deal with this problem. The weight function $w2$ depends on the value of the boundary parameter at the boundary point closest to (x, y). If the closest boundary point is on the boundary piece, then $w2 = 1$. If the closest point is on an extension, $w2$ is a function which goes from one to zero as the parameter value increases (or decreases) along the extension away from the endpoint. The weight function $w2$ is identically zero for parameter values much less or much greater than the range of parameter values which define the boundary piece. If the boundary parameter p varies between $p_{\min}$ and $p_{\max}$ along piece ib, then $w2 = 0$ if

$$p_{est} < p_{\min} - \frac{1}{10}(p_{\max} - p_{\min})$$

or

$$p_{est} > p_{\max} + \frac{1}{10}(p_{\max} - p_{\min}) ,$$

We estimate p_{est}, the parameter value at the point on the boundary piece extension, by extrapolating from parameter values at points near the end of the piece. This approach succeeds in preventing boundary layers from continuing too far along neighboring pieces. If the boundary extension re-enters the domain (as is the case with a re-entrant corner for example), there will be singularities in $true(x, y)$ for a short distance along this extension.

We give examples of several different true solutions in Appendix 1.

3. Sample ELLPACK Program

We include a sample Interactive ELLPACK [Dyksen and Ribbens, 1986] program which can be used to experiment with the various domains and the solutions described above. The program shown in Figure 3.1 is to be run on a Tektronix 4115 graphics terminal. Menu items are defined which allow the user to choose a domain from the set of domains described in Appendix

```
*
* Interactive ELLPACK program for solving Poisson equation on
* various domains with various realistic true solutions.
*
OPTIOINS. terminal = tek4115
      max x points = 65
      max y points = 65

*
* declare common blocks for parameterized true(x,y)
*
DECLARATIONS.
   common /c9trrv/ r9trrv(11)
   common /c9triv/ i9triv(2)

*
* define simple p.d.e. problem
*
EQUATION. uxx + uyy = f(x,y)

*
* define generic domain with at most 6 sides
*
BOUNDARY. u=true(x,y) on x = r9nrxc(p,1), y = r9nryc(p,1)    &
                  for p = r9nrmn(idom,apar,bpar) to r1brng(2,1)
      u=true(x,y) on x = r9nrxc(p,2), y = r9nryc(p,2)    &
                  for p = r1brng(1,2) to r1brng(2,2)
      u=true(x,y) on x = r9nrxc(p,3), y = r9nryc(p,3)    &
                  for p = r1brng(1,3) to r1brng(2,3)
      u=true(x,y) on x = r9nrxc(p,4), y = r9nryc(p,4)    &
                  for p = r1brng(1,4) to r1brng(2,4)
      u=true(x,y) on x = r9nrxc(p,5), y = r9nryc(p,5)    &
                  for p = r1brng(1,5) to r1brng(2,5)
      u=true(x,y) on x = r9nrxc(p,6), y = r9nryc(p,6)    &
                  for p = r1brng(1,6) to r1brng(2,6)

MENU. 'Problem Solving Menu'
     'cd:change domain'
      fortran.
       print *, ' enter domain number:'
       read *, idom
       l0hvgr = .false.
     'cdp:change domain parameters'
      fortran.
```

Figure 3.1. Sample Interactive ELLPACK program which may be used to experiment with the population of domains and the realistic true solutions.

```
      print *, ' enter domain parameter 1:'
      read *, apar
      print *, ' enter domain parameter 2:'
      read *, bpar
    'cp:change problem parameters'        fort.
                                  call q9trcp
    'ig:interactive grid'              grid. interactive
    'sp:solve problem'                  disc. 5 point star
                               solu. linpack band

MENU.  'Output Menu'
    'pd:plot domain'                    out.  plot domain
    'pt:plot true'                    out.  plot(true)
    'pu:plot u'                       out.  plot(u)
    'tu:table u'                      out.  table(u)
    'pe:plot error'                    out.  plot(error)
    'me:max error'                     out.  max(error)
    'mv:move plot from view to view'       out.  move view
    'cv:copy plot from view to view'       out.  copy view
    'dv:delete plot from view'           out.  delete view
    'ev:enlarge views'                 out.  enlarge views
    'pp:put function plot to  a file'     out.  put plot
    'gp:get function plot from a file'     out.  get plot
    'pg:plot current grid on view'         out.  plot grid

SUBPROGRAMS.
    function f(x,y)
c
c  define right side function f(x,y)
c
    call q9trdv (x, y, true, truex, truey, truexx, trueyy)
    f = truexx + trueyy
    return
    end

    function true(x,y)
c
c  define true solution
c
    true = r9true(x,y)
    return
    end
END.
```

Figure 3.1 continued.

2, modify that domain by changing the domain parameters, define a true solution by selecting values for the parameters described in Section 2, solve the PDE by finite differences and Band Gauss Elimination, and plot the true and computed solutions as well as the error. A subprogram Q9TRCP is used to change the problem (by modifying the true solution parameters). A function R9TRUE and a subroutine Q9TRDV are provided to return values of the true solution and of its derivatives, respectively.

REFERENCES

W.R. Dyksen, E.N. Houstis and J.R. Rice, A population of linear, second order, elliptic partial differential equations on rectangular domains, *Math. Comp.*, 36 (1981), 475–484.

W.R. Dyksen and C.J. Ribbens, An interactive problem solving environment for elliptic partial differential equations, CSD-TR-588, Computer Science, Purdue University, (1986), 21 pages.

J.R. Rice, Methodology for the algorithm selection problem, in *Performance Evaluation of Numerical Software* (L. Fosdick, ed.) North-Holland (1979), 301–307.

R.F. Boisvert, E.N. Houstis and J.R. Rice, A system for performance evaluation of partial differential equations software, *IEEE Trans. Software Engr.*, 5 (1979), 418–425.

J.R. Rice, Design of a tensor product population of PDE problems, CSD-TR 628, Computer Science, Purdue University, (1986), 12 pages.

J.R. Rice, Algorithm 625: A two dimensional domains processor, *ACM Trans. Math. Software*, 10 (1984), 453–462.

APPENDIX 1

We give a set of four figures which illustrate a few of the many realistic PDE problems which may be defined in our system. Figure A.1 shows plots of four true solutions on Domain 1 of the population of test domains. In View 1 of Figure A.1 the default values of all parameters are selected. The true solution consists solely of the smooth term

$$true(x,y) = [cos(y) + \sin(x - y)] * [1 + \sin(\frac{x}{2})].$$

In View 2 we have introduced a boundary layer term along side 2 (the top side). In View 3 the decay exponent of the boundary layer term is increased from 5.0 to 10.0, so that the boundary layer extends a much shorter distance out into the domain. In View 4 the magnitude of the boundary layer is increased from 1.0 to 2.0. Now the boundary layer term appears to dominate the solution to an even greater extent.

Figure A.2 shows contour plots of four true solutions on Domain 11. In View 1 the smooth term is the only one present. In View 2 a boundary layer of size 1.0 and decay exponent 5.0 has been added along side 3 (the curved boundary piece). In Views 3 and 4 the size of the boundary layer term is increased to 5.0 and 10.0 respectively.

In Figure A.3 four true solutions on Domain 24 are shown. The first view contains a contour plot of a true solution consisting of a point singularity term. The singularity is located at (0.5,0.5), the re-entrant corner at the upper right. The default magnitude (1.0) and exponent (1.5) are used here. In View 2 a boundary layer term along piece 4 (the right curved side) has also been introduced. In View 3 the boundary layer is the same, but the singularity has been moved to (0.0,0.6), in the interior of the region. Finally, in View 4 we see a true solution consisting solely of a boundary layer of magnitude 2.0 along the horizontal piece at the bottom of the region. Notice that the boundary layer extends a short way into the domain at the right end of the indicated boundary piece.

In Figure A.4 we give contour plots of four true solutions defined on Domain 7. The function in View 1 has a point singularity at the upper right corner. In View 2 there is both a point singularity at the upper right corner and a boundary layer along piece 4 (the curved piece). The magnitude of the boundary layer term is increased from 1.0 to 2.0 in View3, and the decay exponent is increased from 5.0 to 10.0 in View 4.

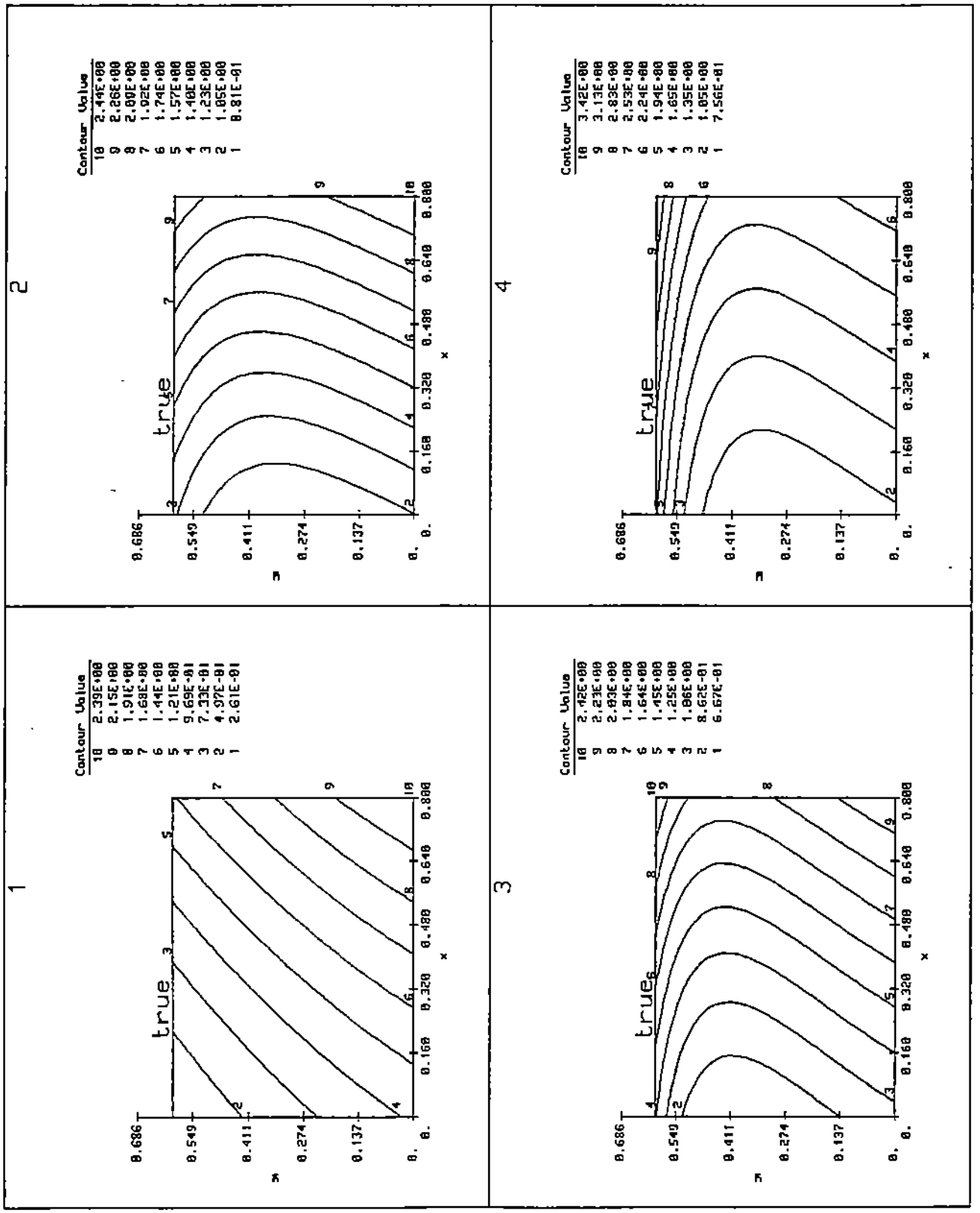


Figure A.1. Contour plots of four true solutions on Domain 1. View 1: default (*smooth*) solution; View 2: *smooth* + *blayer* (magnitude = 1.0, exponent = 5.0); View 3: *smooth* + *blayer* (magnitude = 1.0, exponent = 10.0); View 4: *smooth* + *blayer* (magnitude = 2.0, exponent = 10.0).

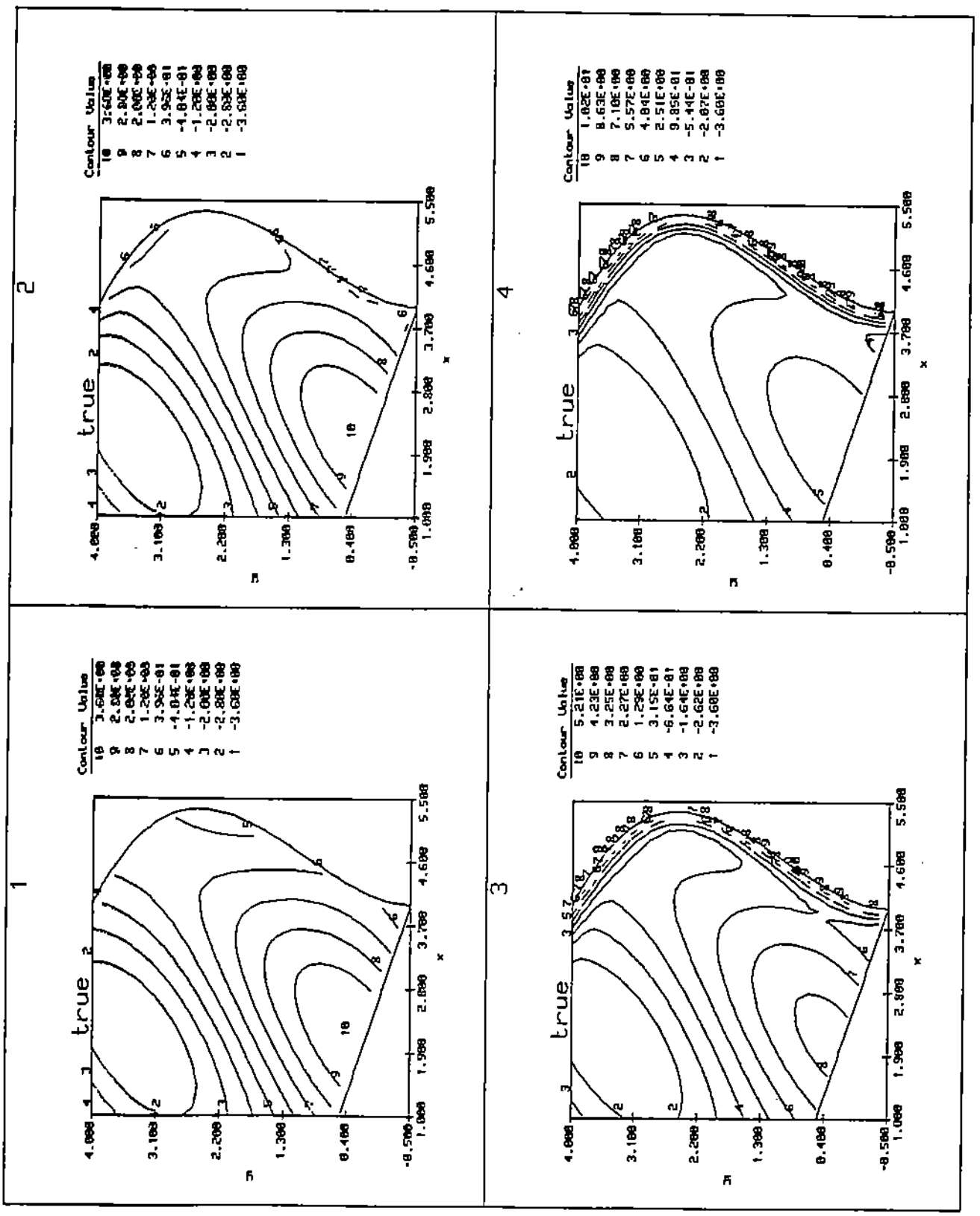


Figure A.2. Contour plots of four true solutions on Domain 11. View 1: *smooth*; View 2: *smooth* + *blayer* (size = 1.0, exponent = 5.0); View 3: *smooth* + *blayer* (size = 5.0, exponent = 5.0); View 4: *smooth* + *blayer* (size = 10.0, exponent = 5.0).

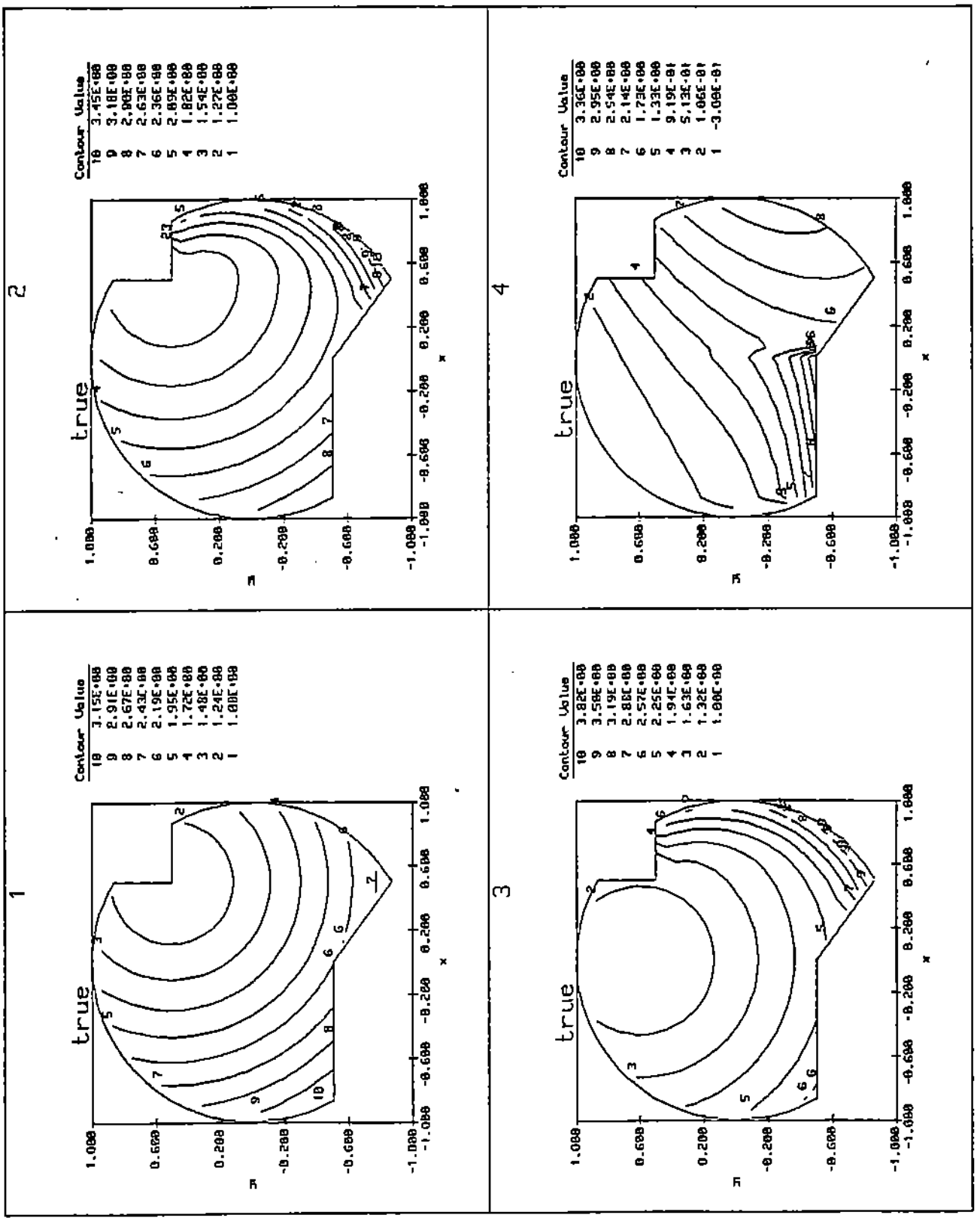


Figure A.3. Contour plots of four true solutions on Domain 24. View 1: *singpt* (size = 1.0, exponent = 1.5); View 2: *singpt* (size = 1.0, exponent = 1.5) + *blayer* (size = 1.0, exponent = 5.0); View 3: *singpt* (size = 1.0, exponent = 1.5) + *blayer* (size = 1.0, exponent = 5.0); View 4: *blayer* (size = 2.0, exponent = 5.0).

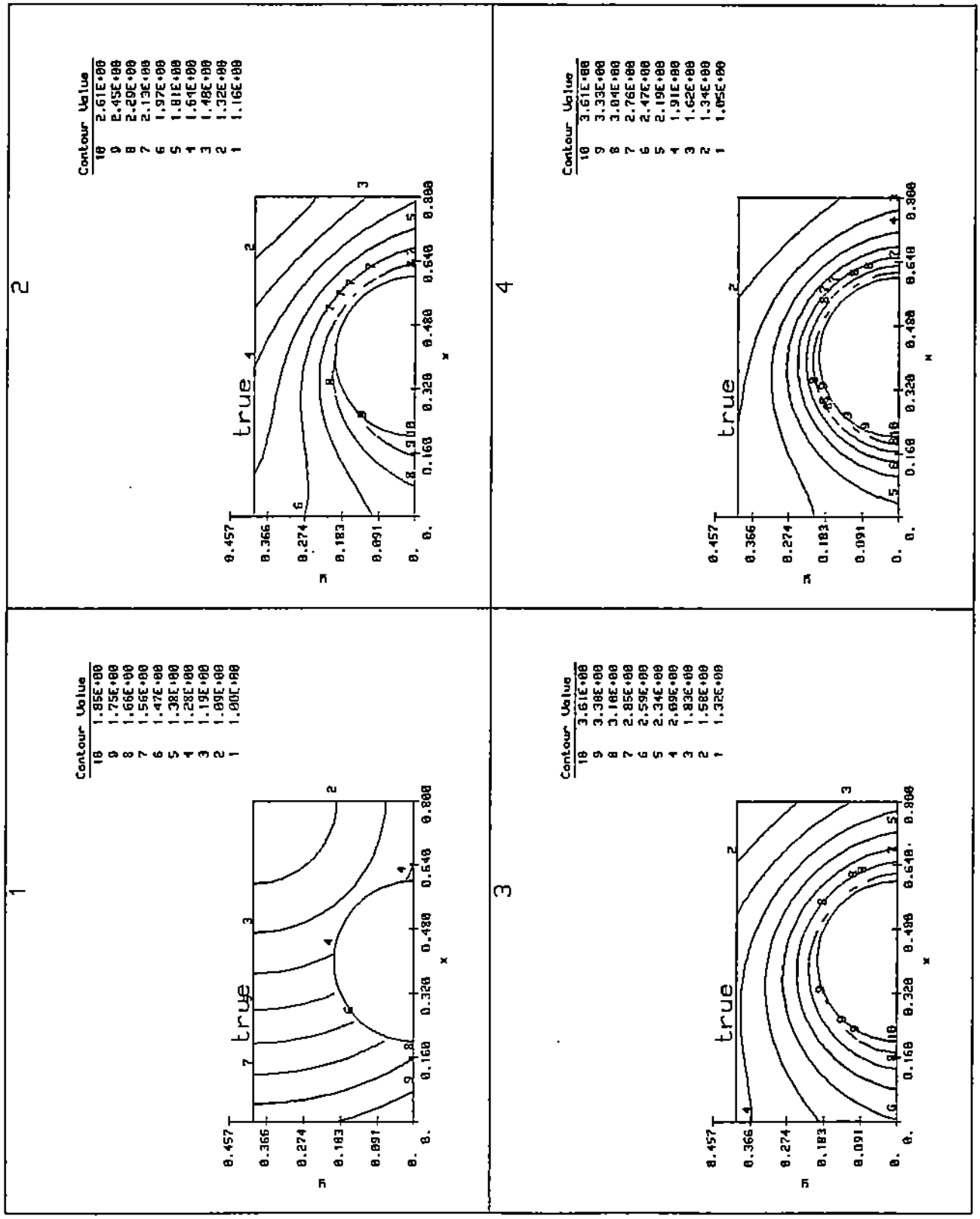


Figure A.4. Contour plots of four true solutions on Domain 7. View 1: *singpt* (size = 1.0, exponent = 1.5); View 2: *singpt* (size = 1.0, exponent = 1.5) + *blayer* (size = 1.0, exponent = 5.0); View 3: *singpt* (size = 1.0, exponent = 1.5) + *blayer* (size = 2.0, exponent = 5.0); View 4: *singpt* (size = 1.0, exponent = 1.5) + *blayer* (size = 2.0, exponent = 10.0).

APPENDIX 2

General Description:

A set of figures is given, 2 or 4 per domain, for the 25 members of the domain population. Each domain has two parameters. The figures given at the end of this appendix illustrate their influence, but in many cases one must experiment to see the effect of varying the parameters. Exact definitions of the domains are provided by the programs Q9NRSR and Q9NRBC discussed below. Table 1 provides some summary information about the domains. We also list other properties as follows:

Smooth: 3, 8

Counterclockwise: 3, 19, 20, 22, 23, 25

Re-entrant corners: 4, 5, 6, 9, 12, 14, 19, 20, 21 ($b < 0$), 22 ($b < -1$), 24

Sharp points, cusps possible: 4, 11, 19, 22, 25

Rapid oscillations, fine variations possible: 9, 13, 19

Unusually complex shape: 6, 9, 12, 14, 19

Precise Definition:

The domains are defined by a set of Fortran subprograms. Since they are incorporateed into the ELLPACK system, they have unusual names which follow the ELLPACK conventions.

Q9NRBC: Contains all the formulas for the boundary pieces, used as

CALL Q9NRBC (P, x, y, *IPIECE*)

when P is the parameter on piece *IPIECE* of the boundary. The coordinate values (x, y) of the corresponding point are returned.

Q9NRSR. Sets the parameter ranges and number of boundary pieces for each domain. The parameters given are included to see if they satisfy constraints designed to eliminate illegal domains.

R9NRMN. A main program (but a function) which must be called to initialize the use of a particular domain as follows:

R9NRMN (INDOMN, PAR1, PAR2)

where INDOMN is the domain number and PAR1, PAR2 are the two parameters. Default parameters are selected by setting PAR1 = PAR2 = –99999.

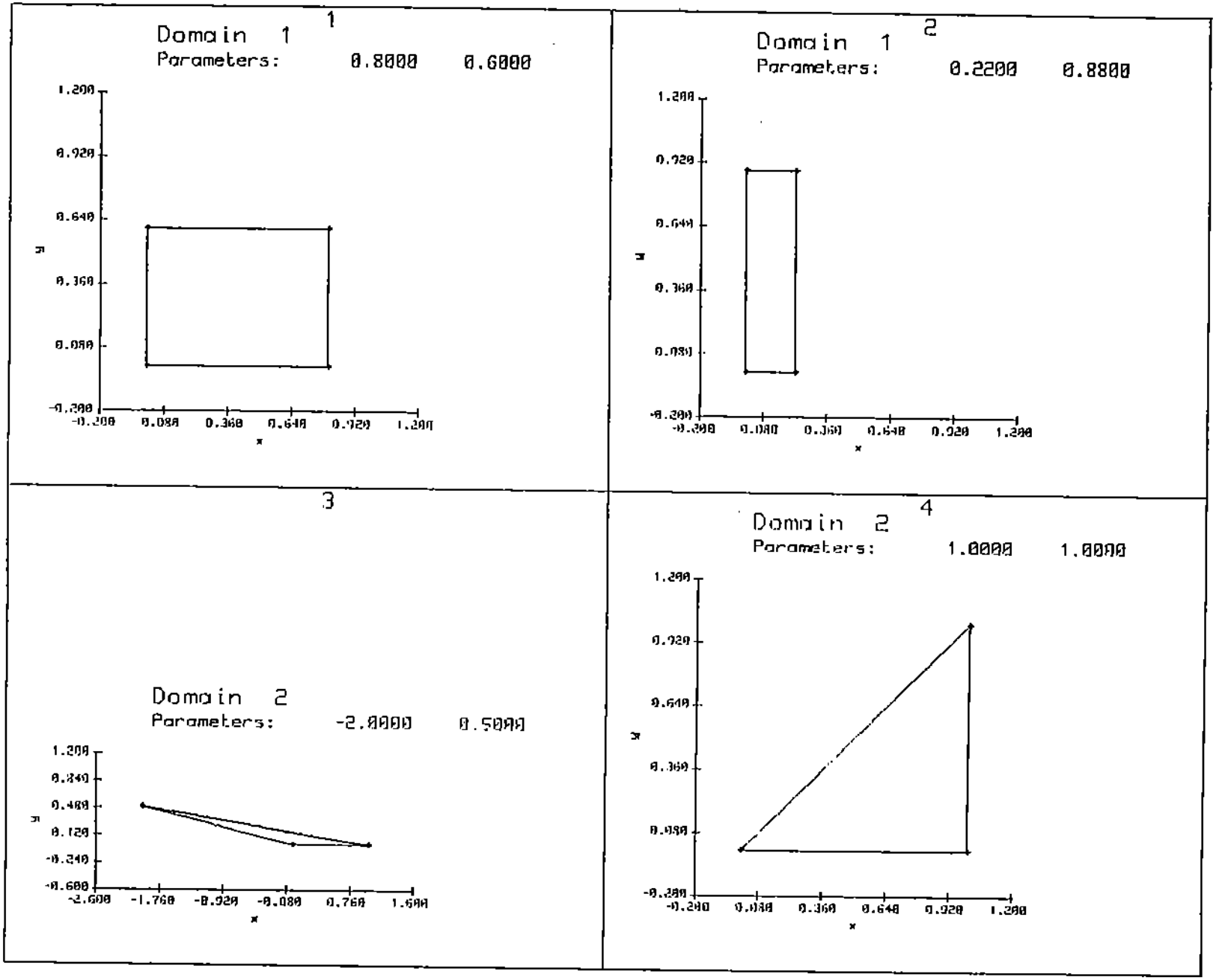
R9NRXC, R9NRYC. Two functions which use Q9NRBC to return the x and y coordinates of a point on the boundary:

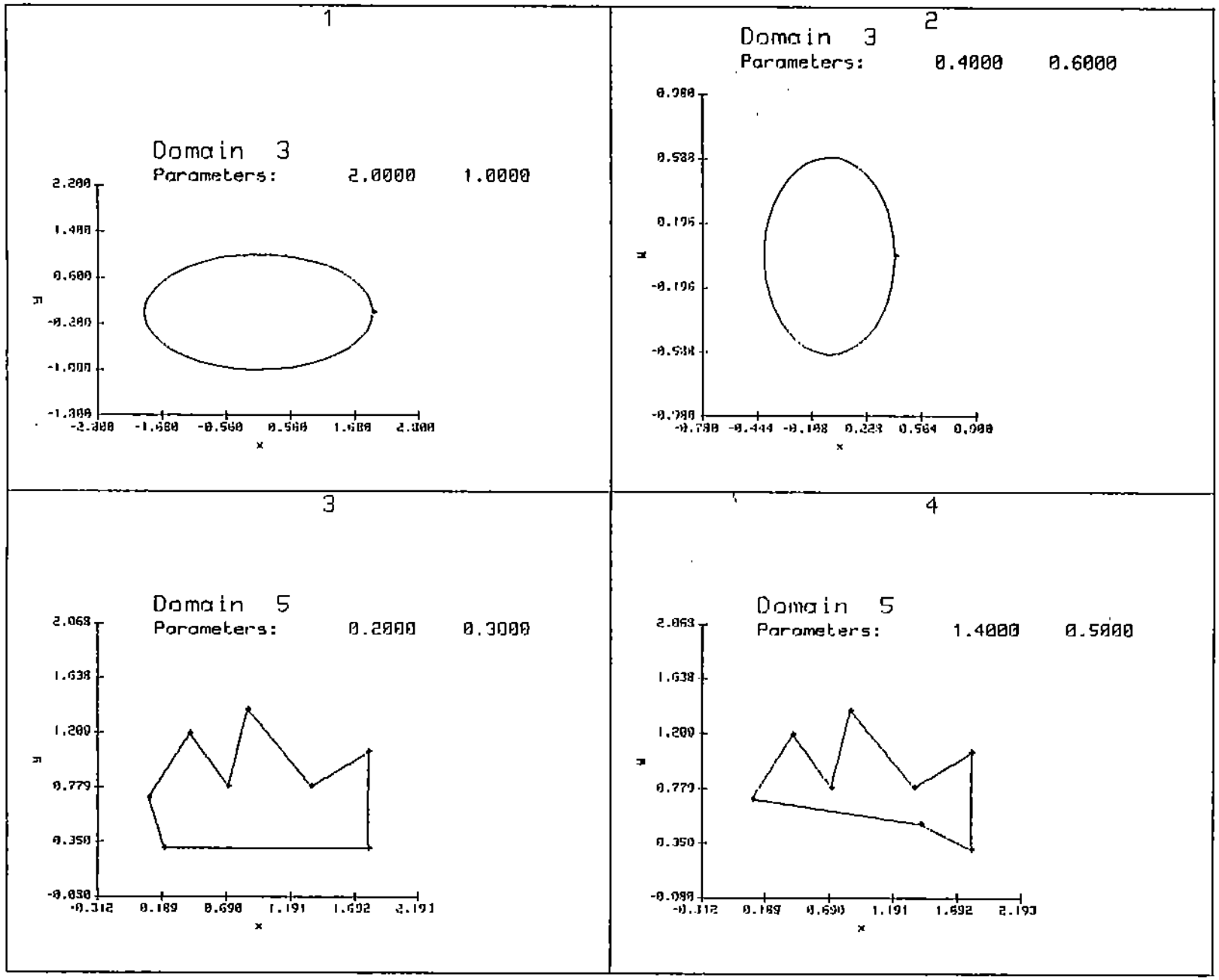
(R9NRXC (*P* , *IPIECE*), R9NRYC (*P* , *IPIECE*))

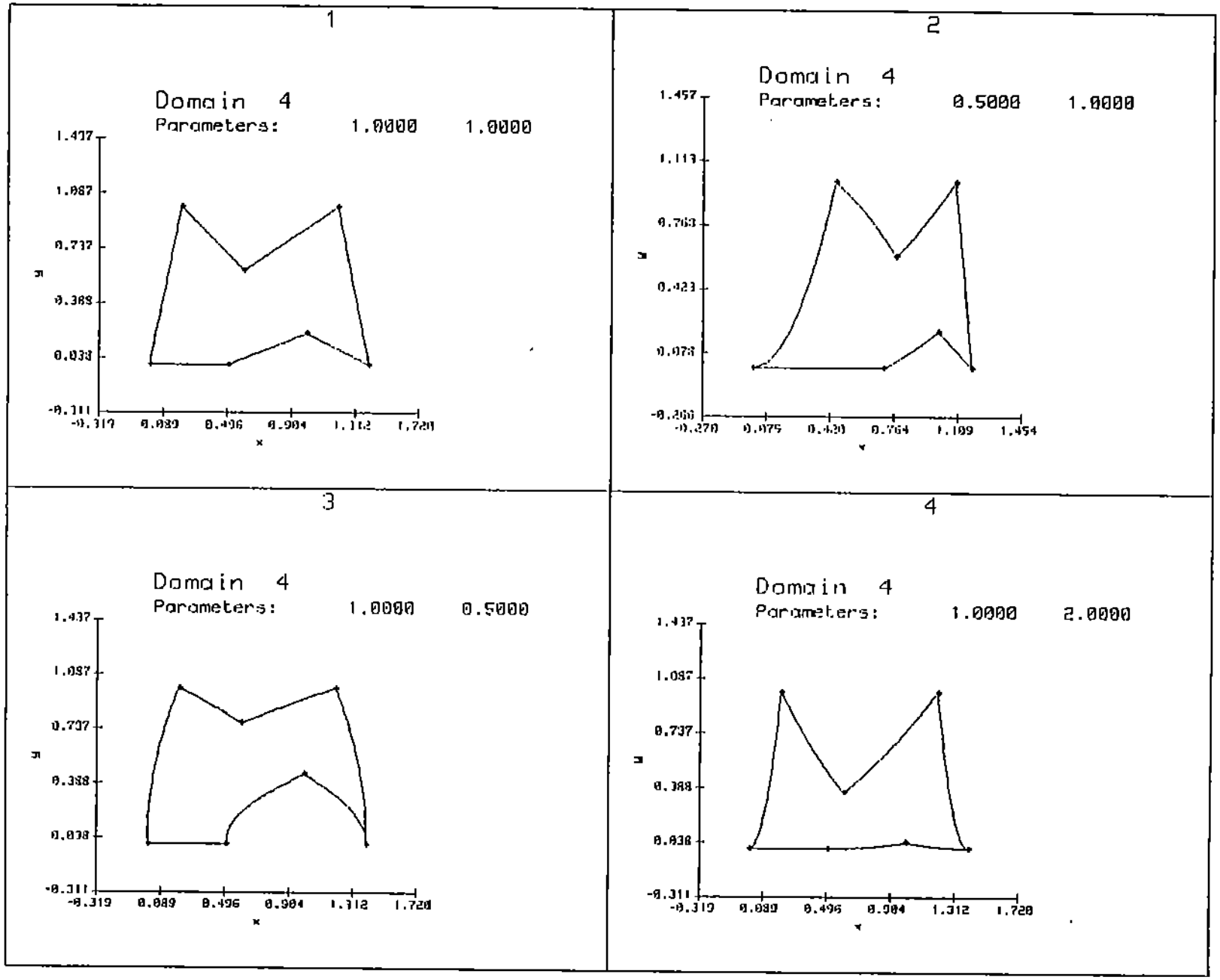
It is more efficient, but sometimes less convenient, to use Q9NRBC to obtain both coordinates at once.

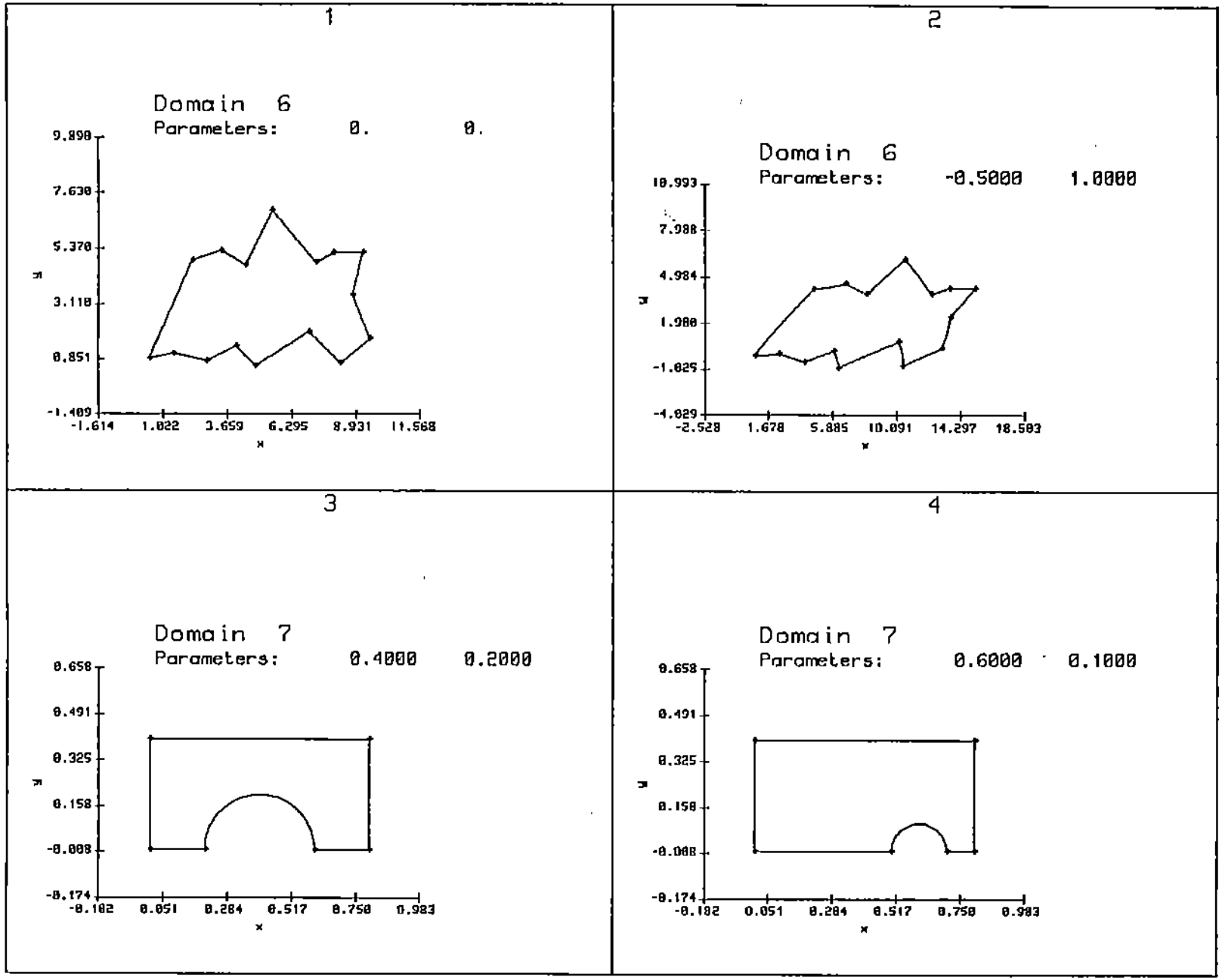
Table 1. Summary information about the 25 domains. The domains are listed in order along with the default values of the parameters, the number **nbound** of boundary pieces and remarks.

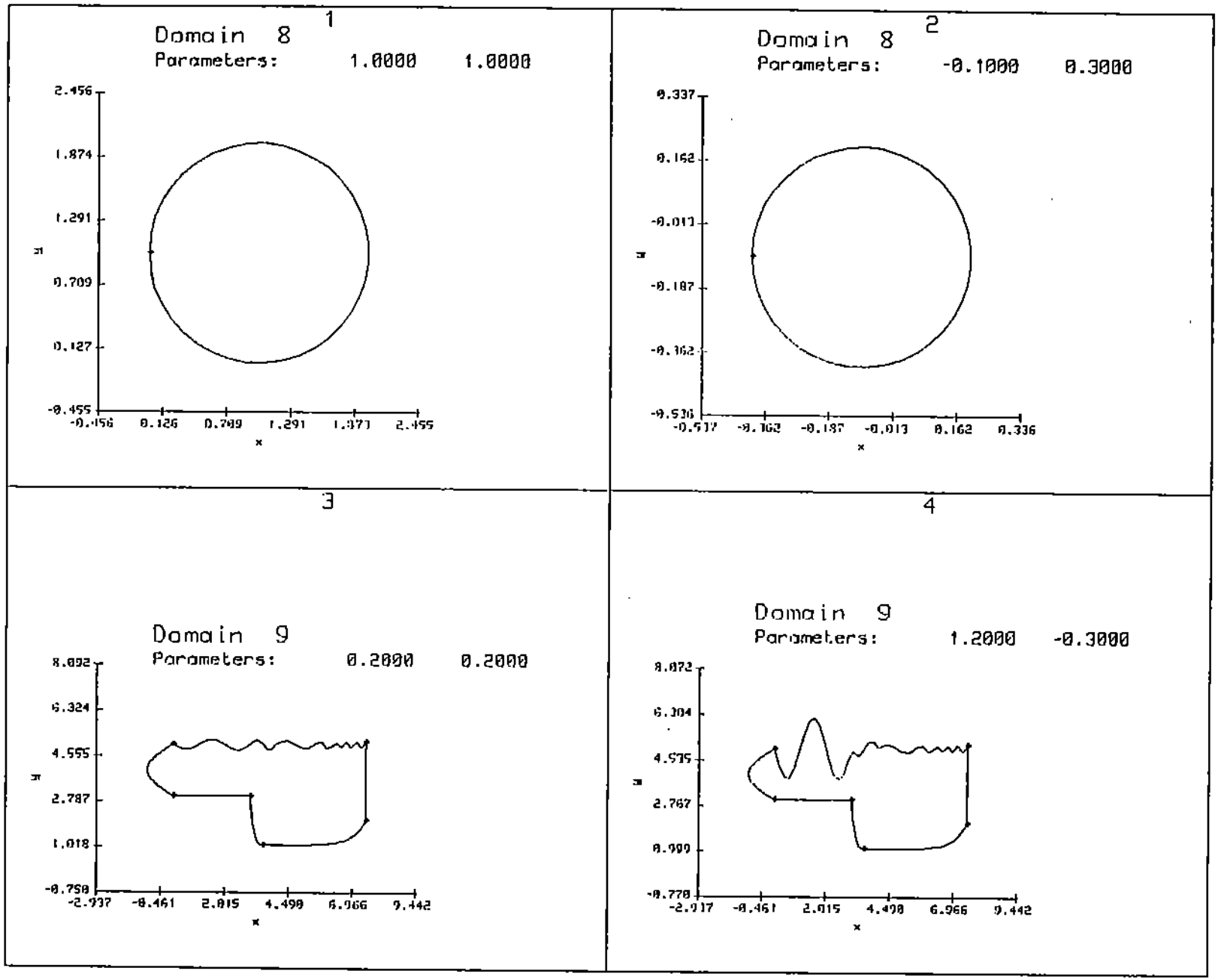
No.	Defaults	nbound	Remarks
1	.8, .6	4	Rectangles
2	1, 1	3	Triangles
3	2, 1	1	Ellipses. Counterclockwise
4	1, 1	7	Polygon mapped by $x = u^a$, $y = v^b$
5	.2, .3	8	Polygon with one vertex variable
6	0, 0	16	Polygon with mapping of (u, v) plane
7	.4, .2	6	Rectangle with semi-circular notch on bottom
8	1, 1	1	Circle
9	.2, .2	6	Meat cleaver shape with wavy top, variable waves
10	1, 0	3	Triangle with one corner smoothed, u variable mapped
11	1, 0	4	Piano top with mapping of (u, v) plane
12	4.4, 3.6	14	Polygon with one part adjustable, one variable hump at the end of a long neck
13	.25, 1	4	Rectangle with wavy top, variable amplitude and frequency
14	0, 0	11	Model of reactor wall with mapping of (u, v) plane
15	.5, 1.5	4	Quarter annulus
16	1.25, .75	5	Circular sector with semi-circular notch on bottom
17	0, 0	6	Spatula shape with rotation and expansion
18	0, 0	7	Default has several horizontal and vertical tangents, with mapping of (u, v) plane
19	0, 0	7	Complex shape with deep, sharp, reentrant boundary segment and with mapping of (u, v) plane. Counterclockwise
20	.5, 1	2	Two intersecting circles. Counterclockwise
21	.2, .5	4	Has one curved side and one movable vertex
22	1, .1	4	Has two curved sides with mapping of (u, v) plane. Counterclockwise
23	0, .5	5	Big wave on one side, with mapping of (u, v) plane. Counterclockwise
24	.5, .5	6	Circle with two notches removed
25	.2, 0	4	Has two curved sides, is three sided when $b = -1$. Counterclockwise

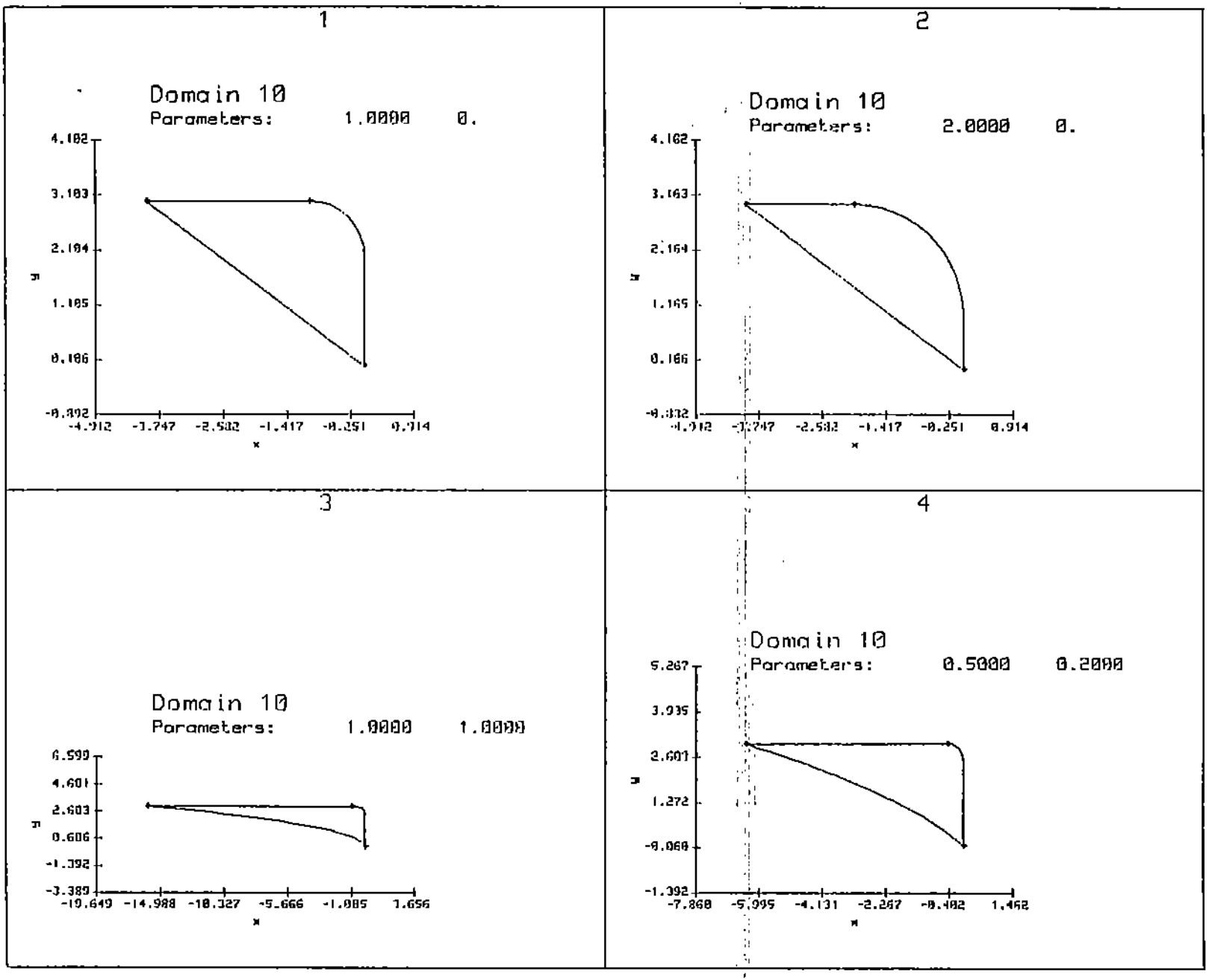

1
Domain 1
Parameters: 0.8000 0.6000
1.200
0.920
0.640
0.360
0.080
-0.200
-0.200 0.080 0.360 0.640 0.920 1.200
y
x
2
Domain 1
Parameters: 0.2200 0.8800
1.200
0.920
0.640
0.360
0.080
-0.200
-0.200 0.080 0.360 0.640 0.920 1.200
y
x
3
Domain 2
Parameters: -2.0000 0.5000
1.200
0.240
0.480
0.720
-0.240
-0.600
-2.600 -1.760 -0.920 -0.080 0.760 1.600
y
x
4
Domain 2
Parameters: 1.0000 1.0000
1.200
0.920
0.640
0.360
0.080
-0.200
-0.200 0.080 0.360 0.640 0.920 1.200
y
x

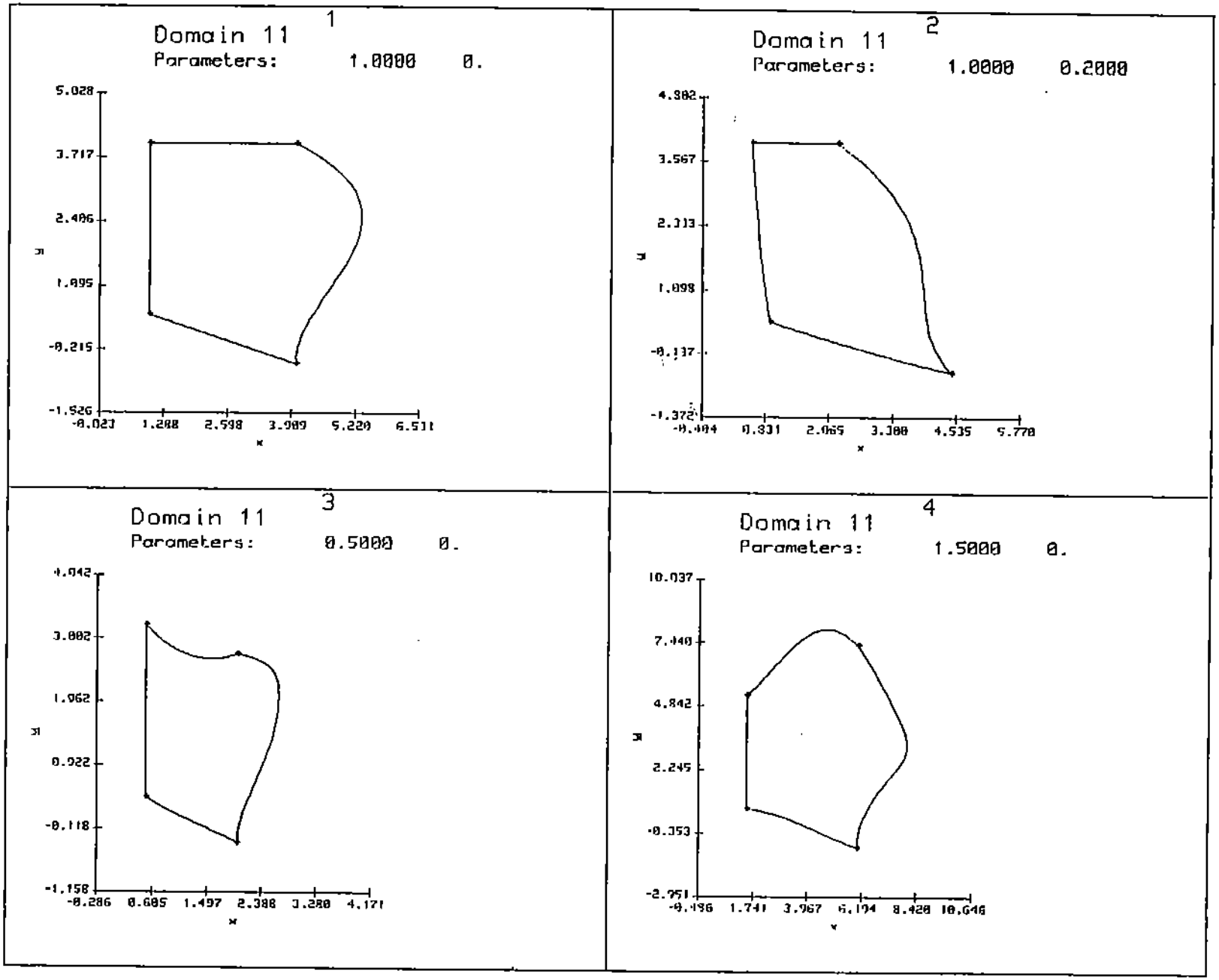

1
Domain 3
Parameters: 2.0000 1.0000
2.200
1.400
0.600
-0.200
-1.000
-1.800
y
-2.300 -1.600 -0.900 -0.200 0.500 1.200 1.900
x
2
Domain 3
Parameters: 0.4000 0.6000
0.980
0.588
0.196
-0.196
-0.588
-0.980
y
-0.780 -0.444 -0.108 0.228 0.564 0.900
x
3
Domain 5
Parameters: 0.2000 0.3000
2.068
1.638
1.200
0.779
0.350
-0.080
y
-0.312 0.189 0.690 1.191 1.692 2.193
x
4
Domain 5
Parameters: 1.4000 0.5000
2.068
1.638
1.200
0.779
0.350
-0.080
y
-0.312 0.189 0.690 1.191 1.692 2.193
x


1
Domain 4
Parameters: 1.0000 1.0000
1.437
1.087
0.737
0.388
0.038
-0.311
-0.319 0.089 0.496 0.904 1.312 1.720
x
y
2
Domain 4
Parameters: 0.5000 1.0000
1.457
1.113
0.769
0.423
0.078
-0.268
-0.270 0.075 0.420 0.764 1.109 1.454
x
y
3
Domain 4
Parameters: 1.0000 0.5000
1.437
1.087
0.737
0.388
0.038
-0.311
-0.319 0.089 0.496 0.904 1.312 1.720
x
y
4
Domain 4
Parameters: 1.0000 2.0000
1.437
1.087
0.737
0.388
0.038
-0.311
-0.319 0.089 0.496 0.904 1.312 1.720
x
y


1
Domain 6
Parameters: 0. 0.
9.890
7.630
5.370
3.110
0.851
-1.409
-1.614
1.022
3.659
6.295
8.931
11.568
2
Domain 6
Parameters: -0.5000 1.0000
10.993
7.988
4.984
1.980
-1.025
-4.029
-2.528
1.678
5.885
10.091
14.297
18.503
3
Domain 7
Parameters: 0.4000 0.2000
0.658
0.491
0.325
0.158
-0.008
-0.174
-0.102
0.051
0.204
0.517
0.750
0.983
4
Domain 7
Parameters: 0.6000 0.1000
0.658
0.491
0.325
0.158
-0.008
-0.174
-0.102
0.051
0.204
0.517
0.750
0.983

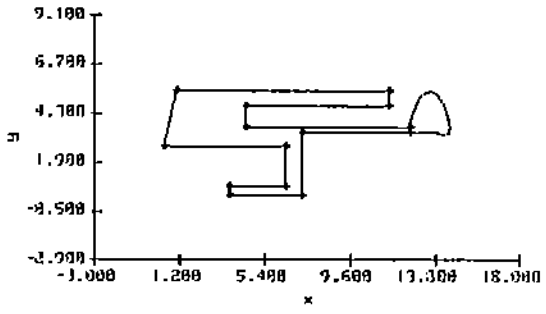

1
Domain 8
Parameters: 1.0000 1.0000
2.456
1.874
1.291
0.709
0.127
-0.455
-0.456 0.126 0.709 1.291 1.873 2.455
x
y
2
Domain 8
Parameters: -0.1000 0.3000
0.337
0.162
-0.013
-0.187
-0.362
-0.536
-0.537 -0.362 -0.187 -0.013 0.162 0.336
x
y
3
Domain 9
Parameters: 0.2000 0.2000
8.092
6.324
4.555
2.787
1.018
-0.750
-2.937 -0.461 2.015 4.490 6.966 9.442
x
y
4
Domain 9
Parameters: 1.2000 -0.3000
8.072
6.304
4.535
2.767
0.999
-0.770
-2.917 -0.461 2.015 4.490 6.966 9.442
x
y


1
Domain 10
Parameters: 1.0000 0.
4.102
3.103
2.104
1.105
0.106
-0.892
-4.912 -3.747 -2.582 -1.417 -0.251 0.914
2
Domain 10
Parameters: 2.0000 0.
4.162
3.163
2.164
1.165
0.166
-0.832
-4.912 -3.747 -2.582 -1.417 -0.251 0.914
3
Domain 10
Parameters: 1.0000 1.0000
6.599
4.601
2.603
0.606
-1.392
-3.389
-19.649 -14.988 -10.327 -5.666 -1.005 3.656
4
Domain 10
Parameters: 0.5000 0.2000
5.267
3.935
2.601
1.272
-0.060
-1.392
-7.860 -5.995 -4.131 -2.267 -0.402 1.462


1
Domain 11
Parameters: 1.0000 0.
5.028
3.717
2.406
1.095
-0.215
-1.526
-0.021 1.288 2.598 3.909 5.220 6.531
2
Domain 11
Parameters: 1.0000 0.2000
4.382
3.567
2.313
1.098
-0.137
-1.372
-0.404 0.831 2.065 3.300 4.535 5.770
3
Domain 11
Parameters: 0.5000 0.
4.042
3.002
1.962
0.922
-0.118
-1.158
-0.286 0.685 1.497 2.308 3.280 4.171
4
Domain 11
Parameters: 1.5000 0.
10.037
7.440
4.842
2.245
-0.353
-2.951
-0.496 1.741 3.967 6.194 8.420 10.646

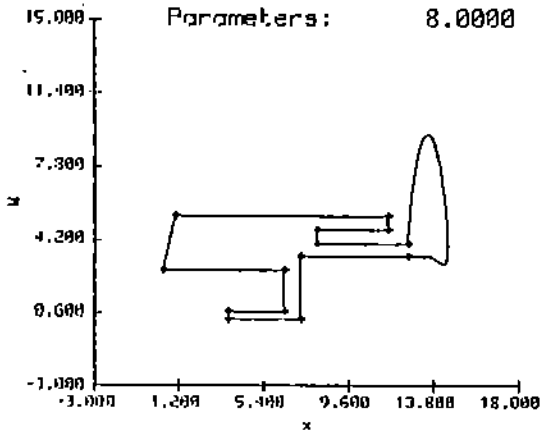
1

Domain 12
Parameters: 4.4000 3.6000



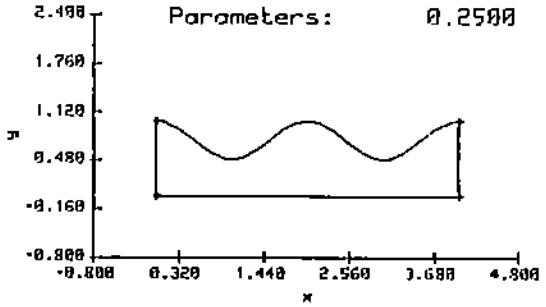
2

Domain 12
Parameters: 8.0000 4.0000



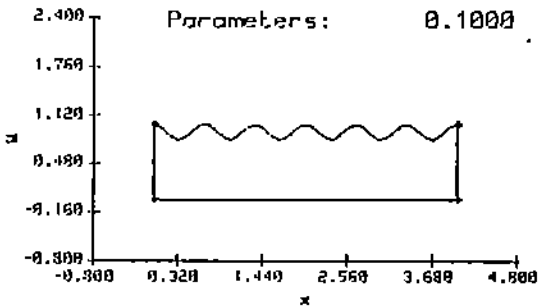
3

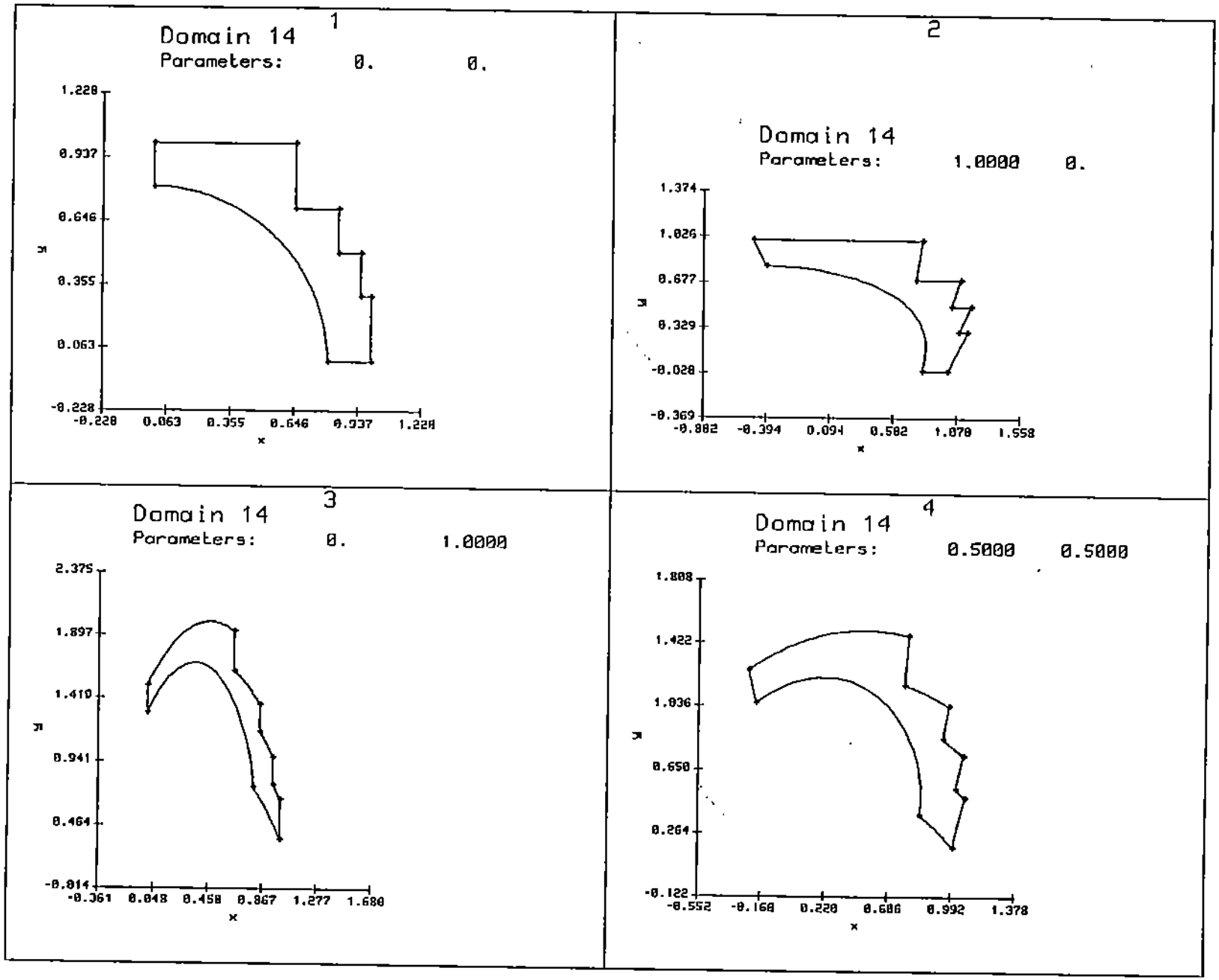
Domain 13
Parameters: 0.2500 1.0000

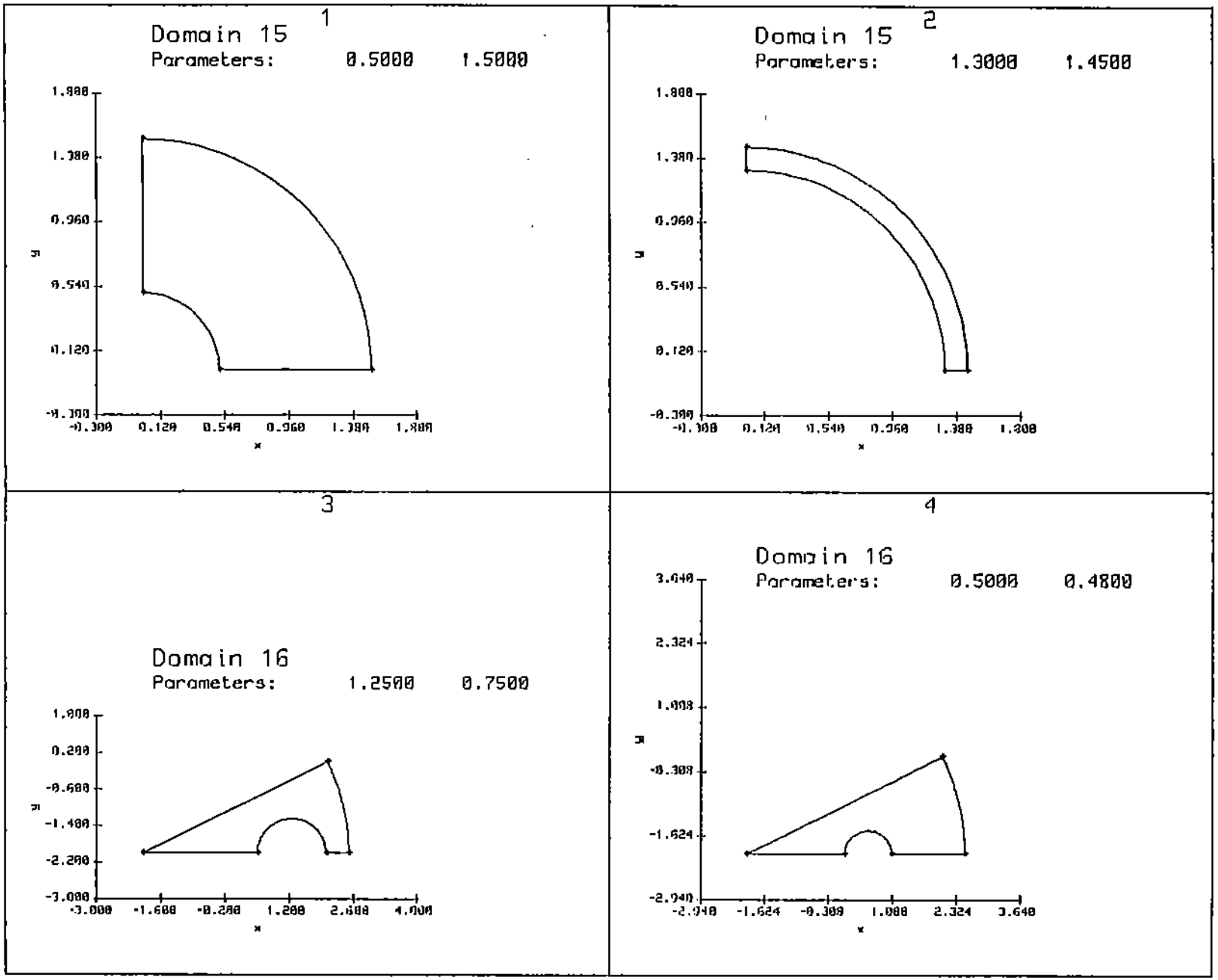


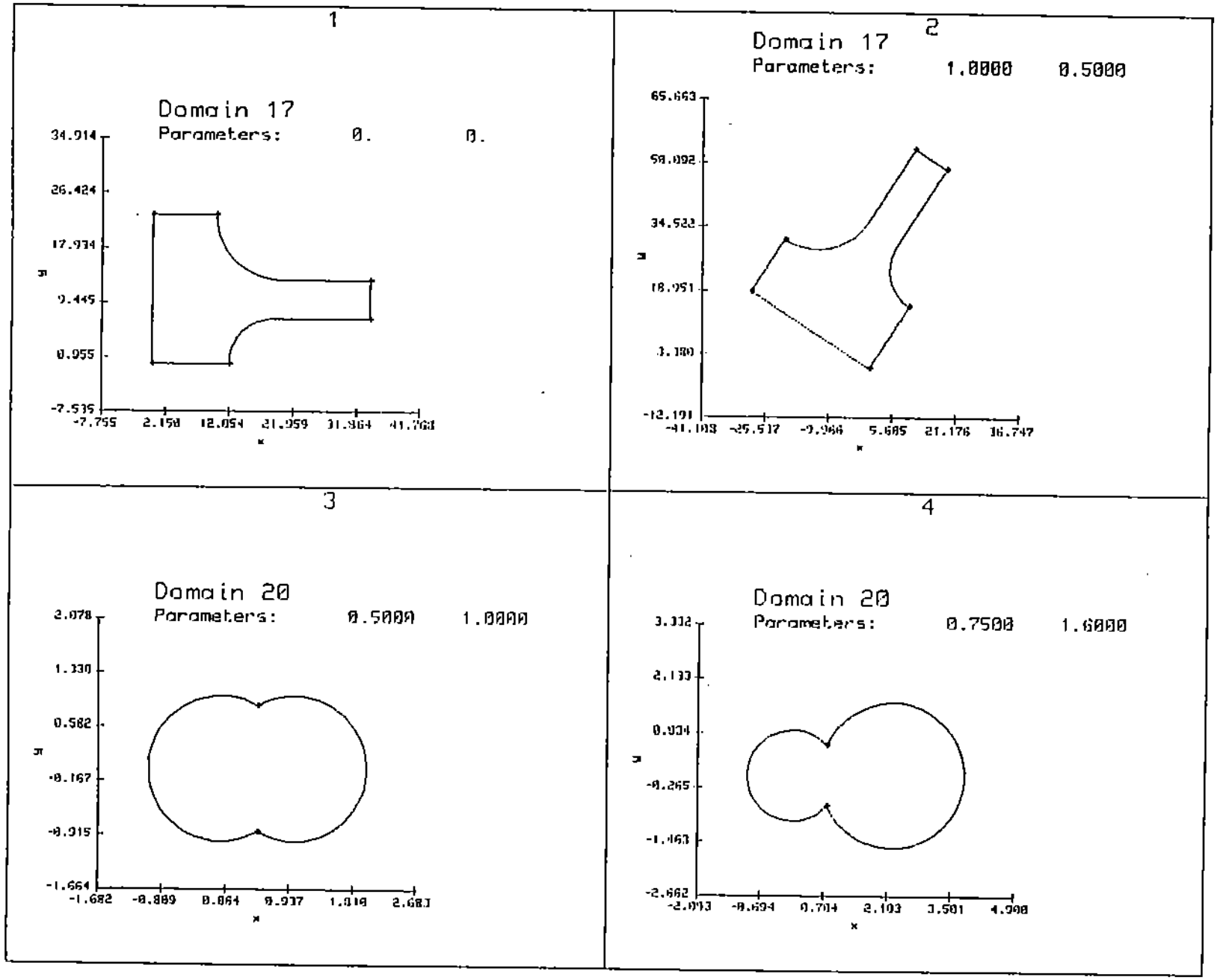
4

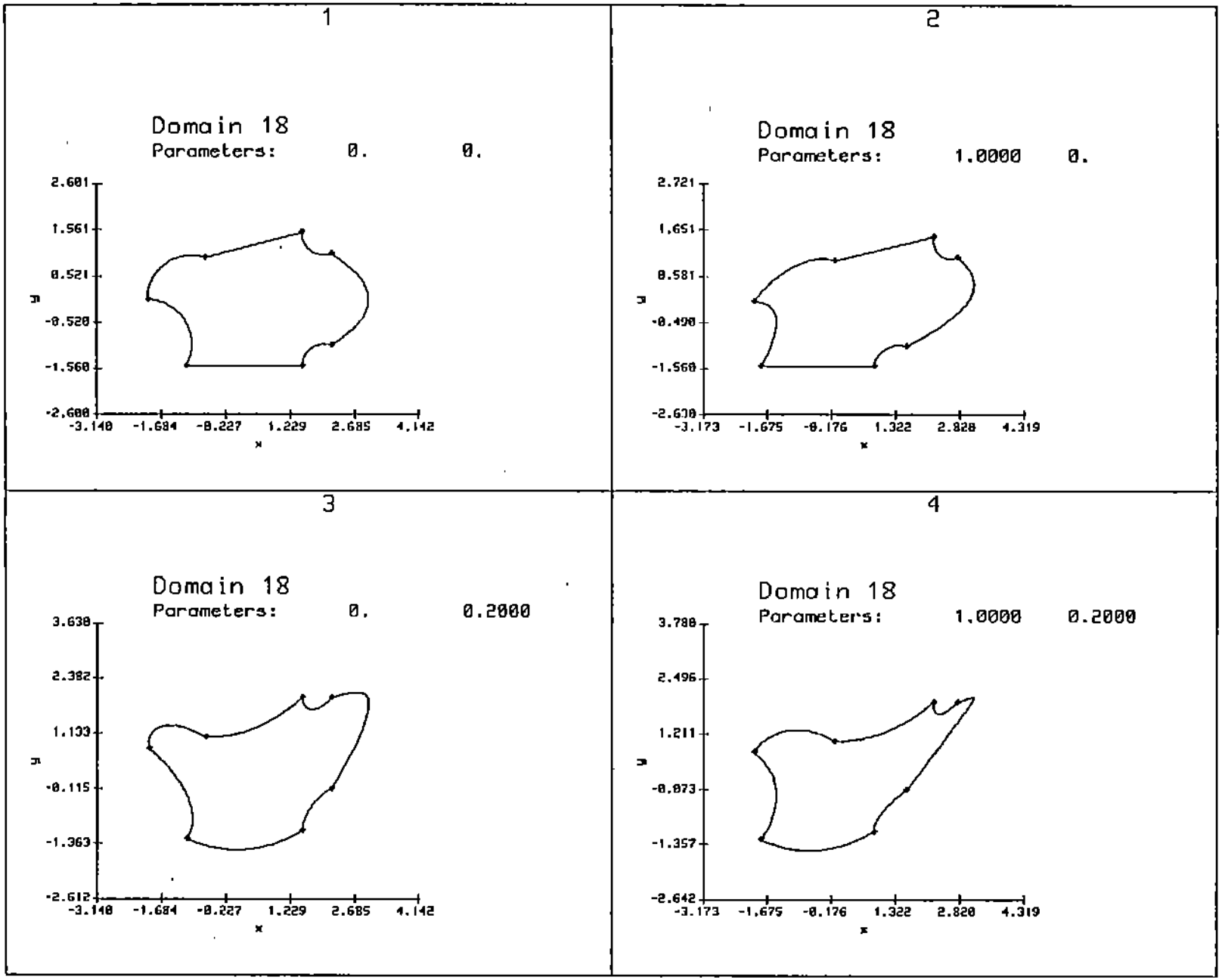
Domain 13
Parameters: 0.1000 3.0000

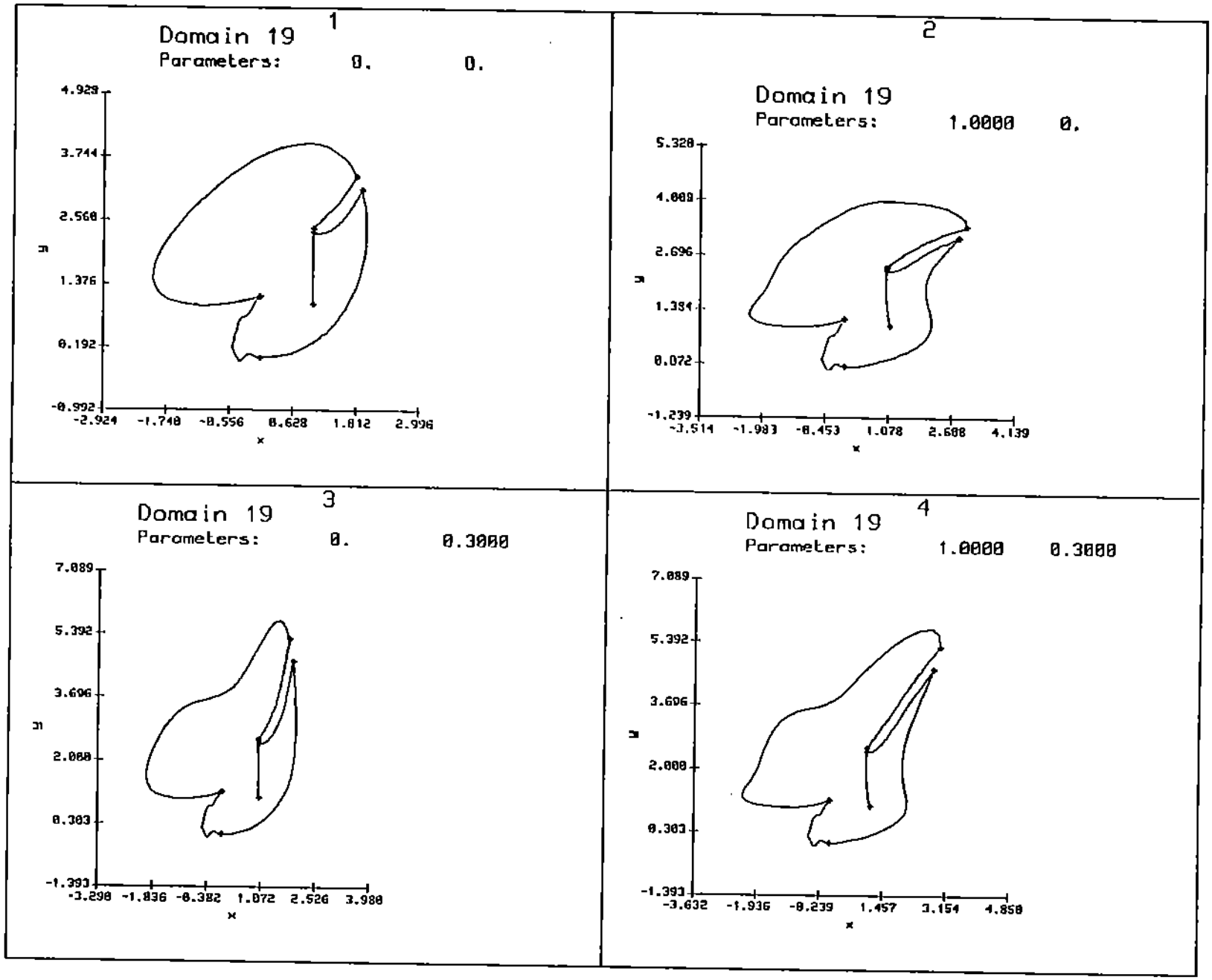


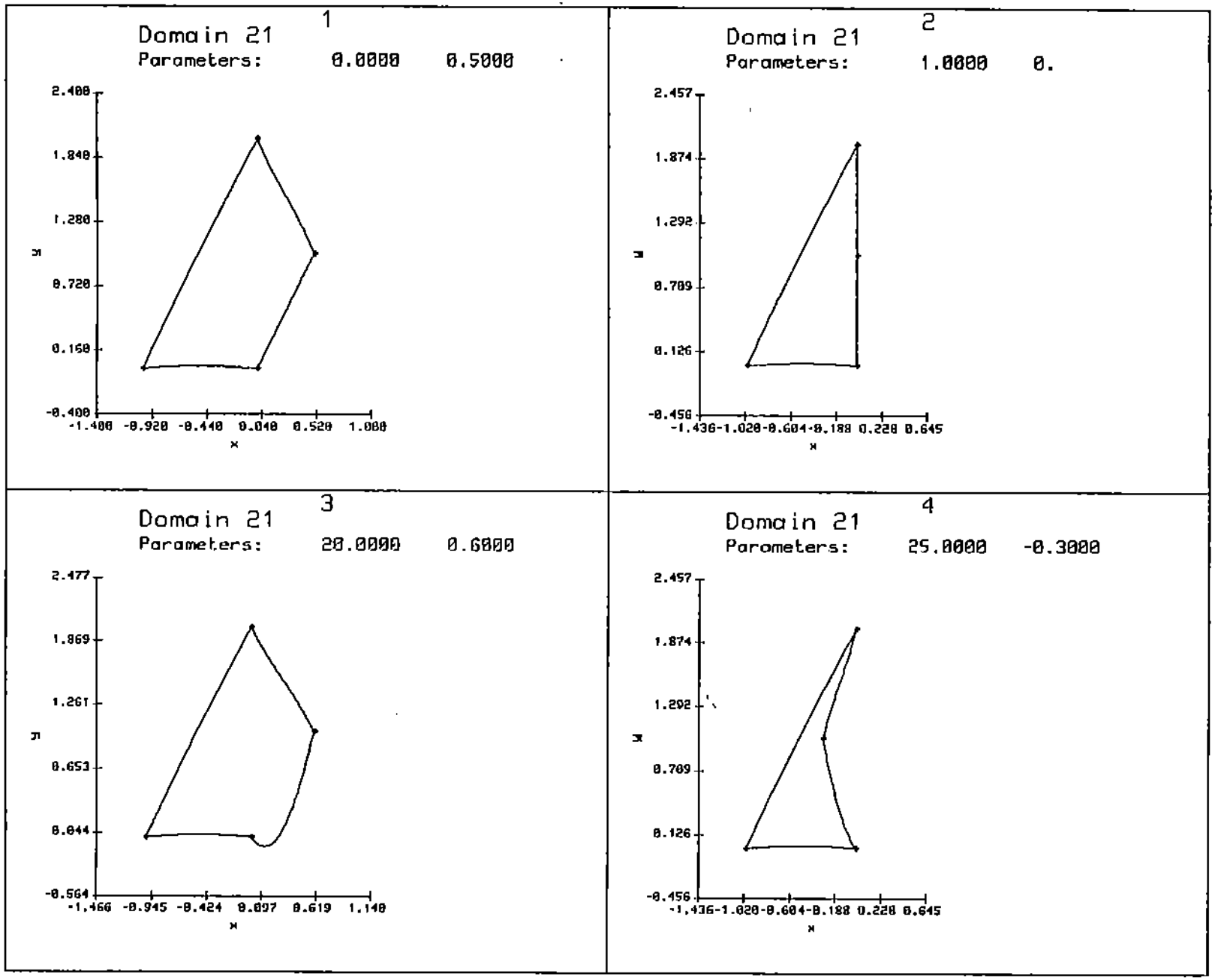

1
Domain 14
Parameters: 0. 0.
1.228
0.937
0.646
0.355
0.063
-0.228
y
-0.228 0.063 0.355 0.646 0.937 1.228
x
2
Domain 14
Parameters: 1.0000 0.
1.374
1.026
0.677
0.329
-0.028
-0.369
y
-0.882 -0.394 0.094 0.582 1.070 1.558
x
3
Domain 14
Parameters: 0. 1.0000
2.375
1.897
1.419
0.941
0.464
-0.014
y
-0.361 0.048 0.458 0.867 1.277 1.686
x
4
Domain 14
Parameters: 0.5000 0.5000
1.808
1.422
1.036
0.650
0.264
-0.122
y
-0.552 -0.166 0.220 0.606 0.992 1.378
x

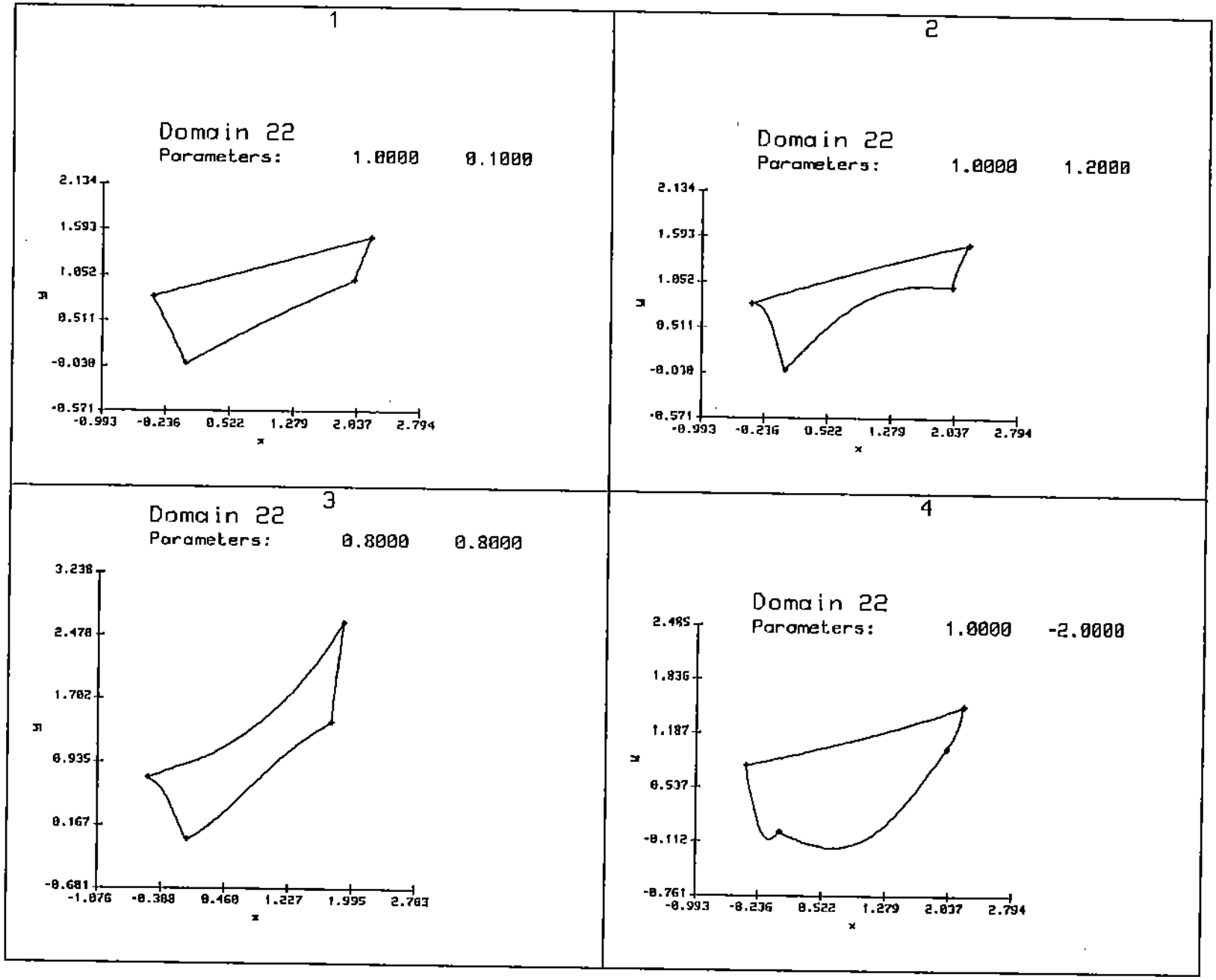

1
Domain 15
Parameters: 0.5000 1.5000
1.800
1.380
0.960
0.540
0.120
-0.300
-0.300 0.120 0.540 0.960 1.380 1.800
y
x
2
Domain 15
Parameters: 1.3000 1.4500
1.800
1.380
0.960
0.540
0.120
-0.300
-0.300 0.120 0.540 0.960 1.380 1.800
y
x
3
Domain 16
Parameters: 1.2500 0.7500
1.000
0.200
-0.600
-1.400
-2.200
-3.000
-3.000 -1.600 -0.200 1.200 2.600 4.000
y
x
4
Domain 16
Parameters: 0.5000 0.4800
3.640
2.324
1.008
-0.308
-1.624
-2.940
-2.940 -1.624 -0.308 1.008 2.324 3.640
y
x

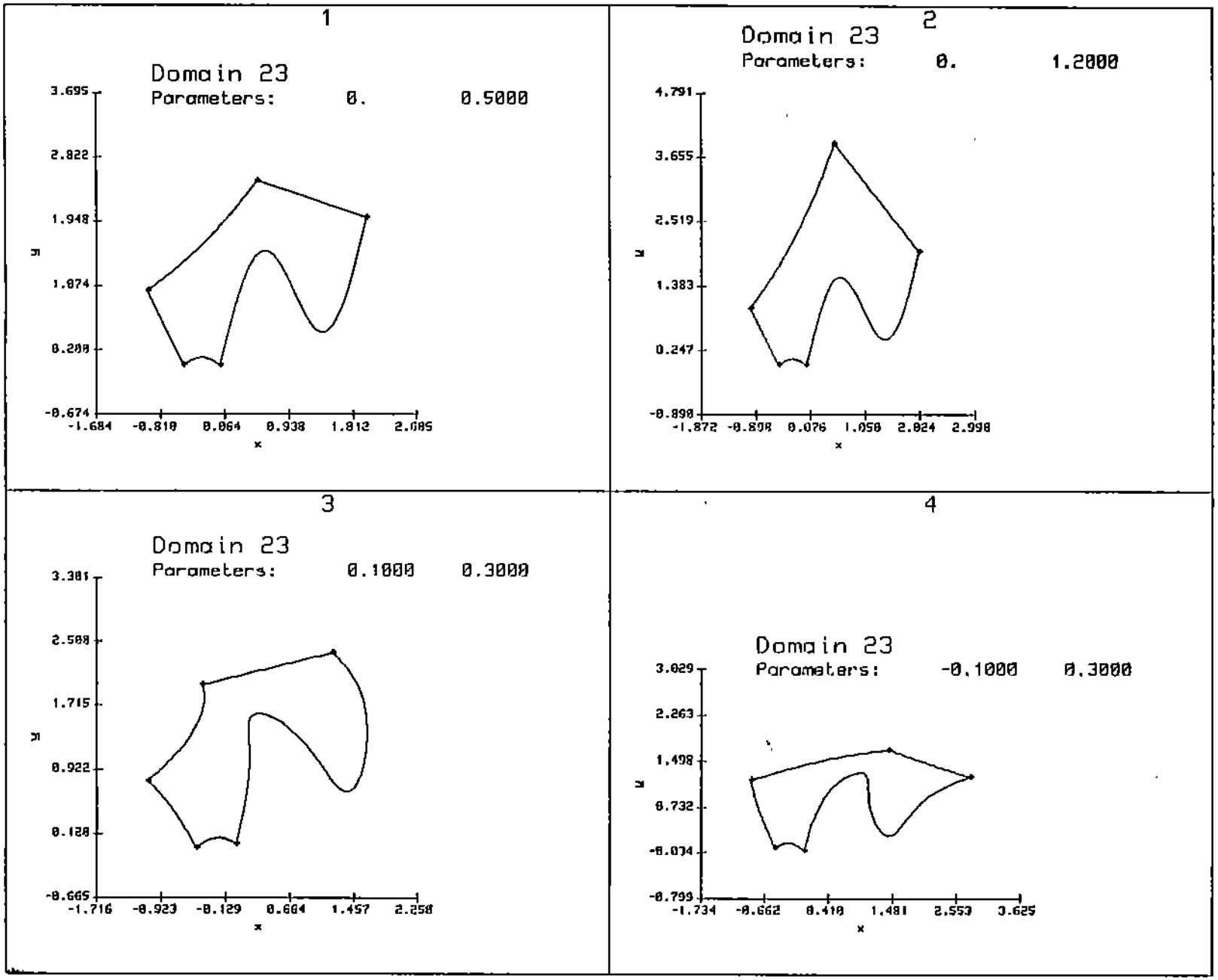

1
Domain 17
Parameters: 0. 0.
34.914
26.424
17.934
9.445
0.955
-7.535
-7.755
2.158
12.054
21.959
31.864
41.768
2
Domain 17
Parameters: 1.0000 0.5000
65.663
58.092
34.522
18.951
3.380
-12.191
-41.103
-25.517
-9.966
5.685
21.176
36.747
3
Domain 20
Parameters: 0.5000 1.0000
2.078
1.330
0.582
-0.167
-0.915
-1.664
-1.682
-0.889
0.864
0.937
1.810
2.683
4
Domain 20
Parameters: 0.7500 1.6000
3.338
2.133
0.934
-0.265
-1.463
-2.662
-2.093
-0.694
0.704
2.103
3.501
4.900

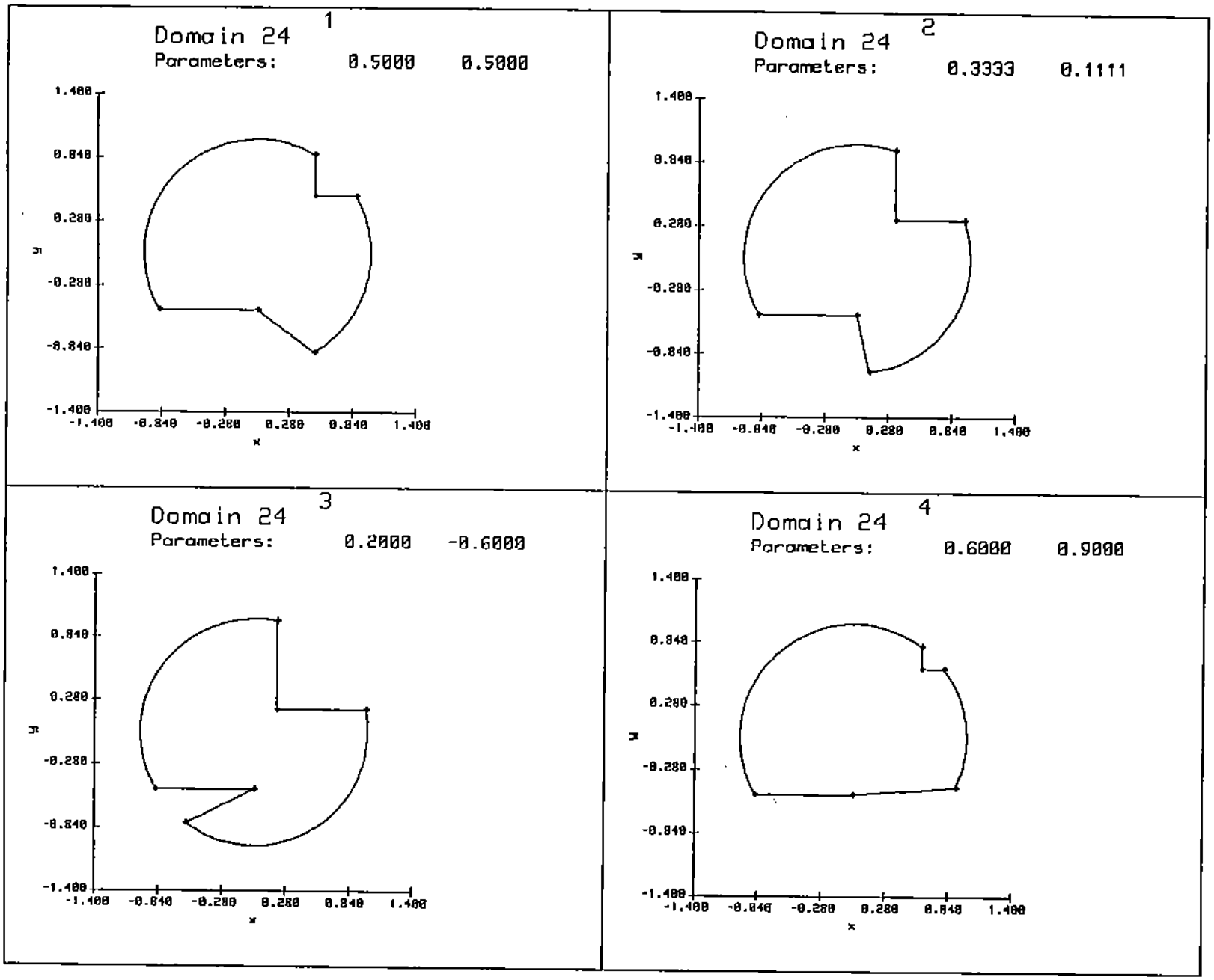

1
Domain 18
Parameters: 0. 0.
2.601
1.561
0.521
-0.520
-1.560
-2.600
-3.140 -1.684 -0.227 1.229 2.685 4.142
2
Domain 18
Parameters: 1.0000 0.
2.721
1.651
0.581
-0.490
-1.560
-2.630
-3.173 -1.675 -0.176 1.322 2.828 4.319
3
Domain 18
Parameters: 0. 0.2000
3.638
2.382
1.133
-0.115
-1.363
-2.612
-3.140 -1.684 -0.227 1.229 2.685 4.142
4
Domain 18
Parameters: 1.0000 0.2000
3.788
2.496
1.211
-0.073
-1.357
-2.642
-3.173 -1.675 -0.176 1.322 2.820 4.319

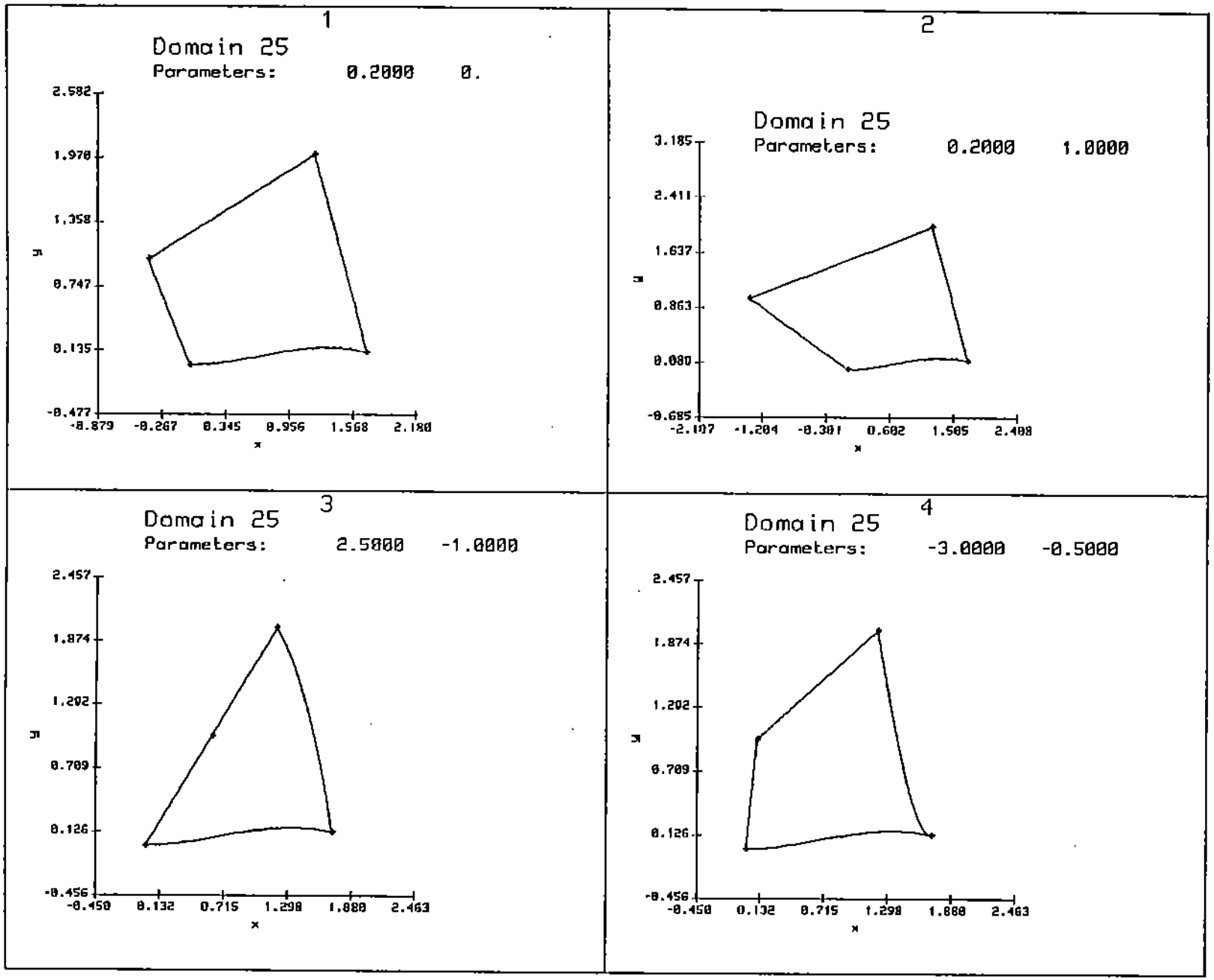

1
Domain 19
Parameters: 0. 0.
4.928
3.744
2.560
1.376
0.192
-0.992
y
-2.924 -1.740 -0.556 0.628 1.812 2.996
x
2
Domain 19
Parameters: 1.0000 0.
5.328
4.088
2.696
1.304
0.072
-1.239
y
-3.514 -1.983 -0.453 1.078 2.608 4.139
x
3
Domain 19
Parameters: 0. 0.3000
7.089
5.392
3.696
2.000
0.303
-1.393
y
-3.298 -1.836 -0.382 1.072 2.526 3.980
x
4
Domain 19
Parameters: 1.0000 0.3000
7.089
5.392
3.696
2.000
0.303
-1.393
y
-3.632 -1.936 -0.239 1.457 3.154 4.858
x


1
Domain 21
Parameters: 0.0000 0.5000
2.408
1.848
1.288
0.728
0.168
-0.400
-1.400 -0.920 -0.440 0.040 0.520 1.000
y
x
2
Domain 21
Parameters: 1.0000 0.
2.457
1.874
1.292
0.709
0.126
-0.456
-1.436-1.020-0.604-0.188 0.228 0.645
y
x
3
Domain 21
Parameters: 20.0000 0.6000
2.477
1.869
1.261
0.653
0.044
-0.564
-1.466 -0.945 -0.424 0.097 0.619 1.140
y
x
4
Domain 21
Parameters: 25.0000 -0.3000
2.457
1.874
1.292
0.709
0.126
-0.456
-1.436-1.020-0.604-0.188 0.228 0.645
y
x


1
Domain 22
Parameters: 1.0000 0.1000
2.134
1.593
1.052
0.511
-0.030
-0.571
-0.993 -0.236 0.522 1.279 2.037 2.794
2
Domain 22
Parameters: 1.0000 1.2000
2.134
1.593
1.052
0.511
-0.010
-0.571
-0.993 -0.236 0.522 1.279 2.037 2.794
3
Domain 22
Parameters: 0.8000 0.8000
3.238
2.478
1.702
0.935
0.167
-0.681
-1.076 -0.388 0.460 1.227 1.995 2.783
4
Domain 22
Parameters: 1.0000 -2.0000
2.485
1.836
1.187
0.537
-0.112
-0.761
-0.993 -0.236 0.522 1.279 2.037 2.794


1
Domain 23
Parameters: 0. 0.5000
3.695
2.822
1.948
1.074
0.200
-0.674
-1.684 -0.810 0.064 0.938 1.812 2.685
y
x
2
Domain 23
Parameters: 0. 1.2000
4.791
3.655
2.519
1.383
0.247
-0.890
-1.872 -0.898 0.076 1.050 2.024 2.998
y
x
3
Domain 23
Parameters: 0.1000 0.3000
3.301
2.508
1.715
0.922
0.128
-0.665
-1.716 -0.923 -0.129 0.664 1.457 2.258
y
x
4
Domain 23
Parameters: -0.1000 0.3000
3.029
2.263
1.498
0.732
-0.034
-0.799
-1.734 -0.662 0.410 1.481 2.553 3.625
y
x


1
Domain 24
Parameters: 0.5000 0.5000
2
Domain 24
Parameters: 0.3333 0.1111
3
Domain 24
Parameters: 0.2000 -0.6000
4
Domain 24
Parameters: 0.6000 0.9000
1.400
0.840
0.280
-0.280
-0.840
-1.400
-1.400 -0.840 -0.280 0.280 0.840 1.400
x
y


1
Domain 25
Parameters: 0.2000 0.
2.582
1.970
1.358
0.747
0.135
-0.477
-0.879 -0.267 0.345 0.956 1.568 2.180
2
Domain 25
Parameters: 0.2000 1.0000
3.185
2.411
1.637
0.863
0.080
-0.685
-2.107 -1.204 -0.301 0.602 1.505 2.408
3
Domain 25
Parameters: 2.5000 -1.0000
2.457
1.874
1.292
0.709
0.126
-0.456
-0.450 0.132 0.715 1.298 1.880 2.463
4
Domain 25
Parameters: -3.0000 -0.5000
2.457
1.874
1.292
0.709
0.126
-0.456
-0.450 0.132 0.715 1.298 1.880 2.463