Improvement in Vacuum Grids of Terpsichore

Guoyong Fu , 8/13/98

(1) Problems with vacuum grids :

Terpsichore treats vacuum as shearless pseudo-plasma: a vacuum grid is needed.

vacuum pseudo-surfaces intersect
due to closeness of the prescribed conducting wall to the plasma boundary at some location.

(2) Solutions:

<u>old conducting wall:</u>

R_wall = R_pvi + (a-1)(R_pvi - R_m=0);

<u>New conducting wall</u>:

R_wall = R_pvi + (a-1)(R_pvi - R_axis);

or

R_wall = R_pvi + (a-1)(R_pvi - R_ave);

or

R_wall = R_pvi + (a-1)(R_n=0 - R_n=0,ave);

(3) Results with the new vacuum grids

For cases where the old method works, the results with new grids are the same as the old ones when the wall is sufficiently far way.

The new code successfully treats the all previous failed cases considered so far, even for a case with extreme boundary shape. The new code is robust and is suitable for use in optimization.

Figure Captions

(1) Old grids for case qas3_h3a;

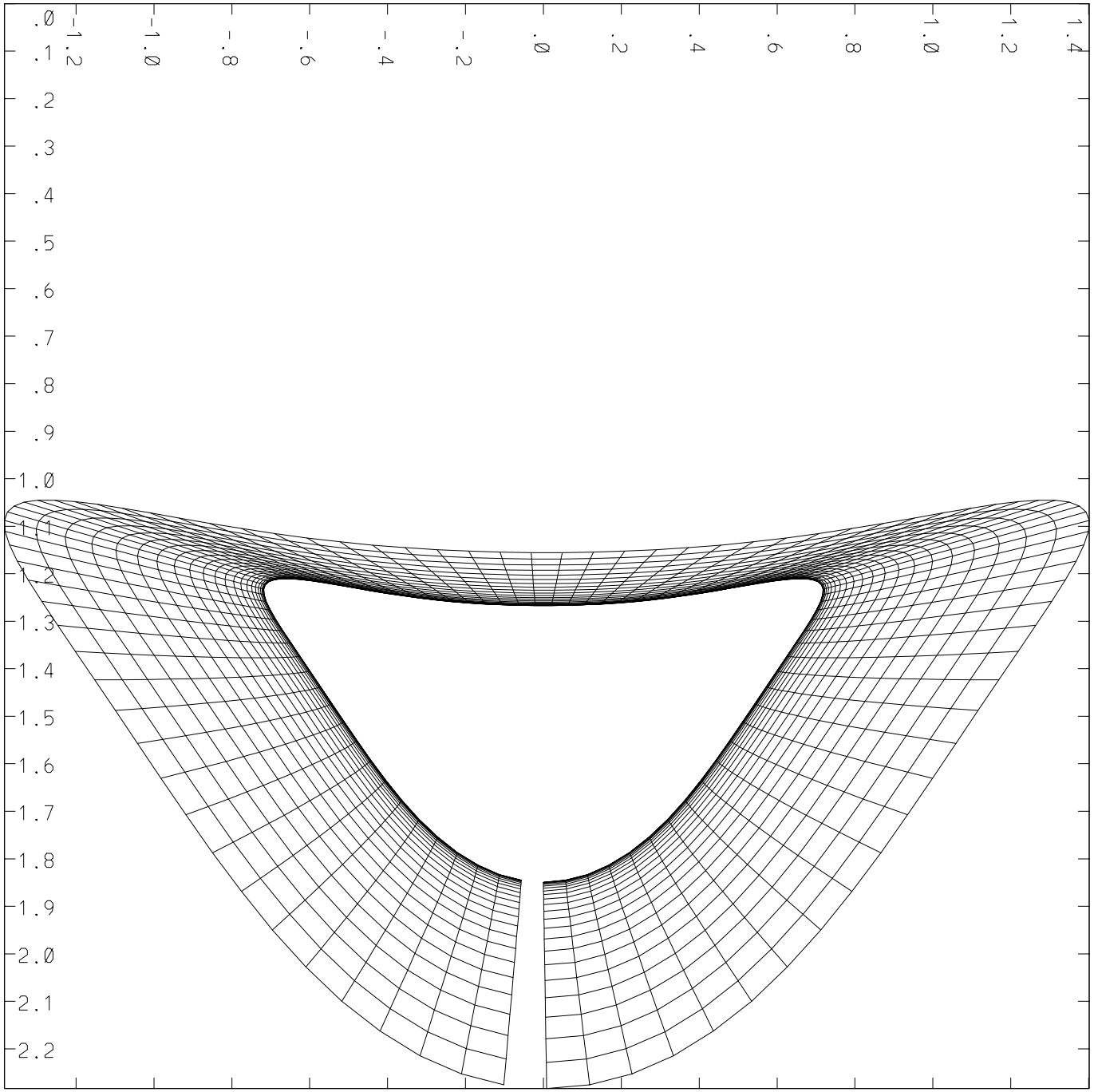
(2) New grids, type II;

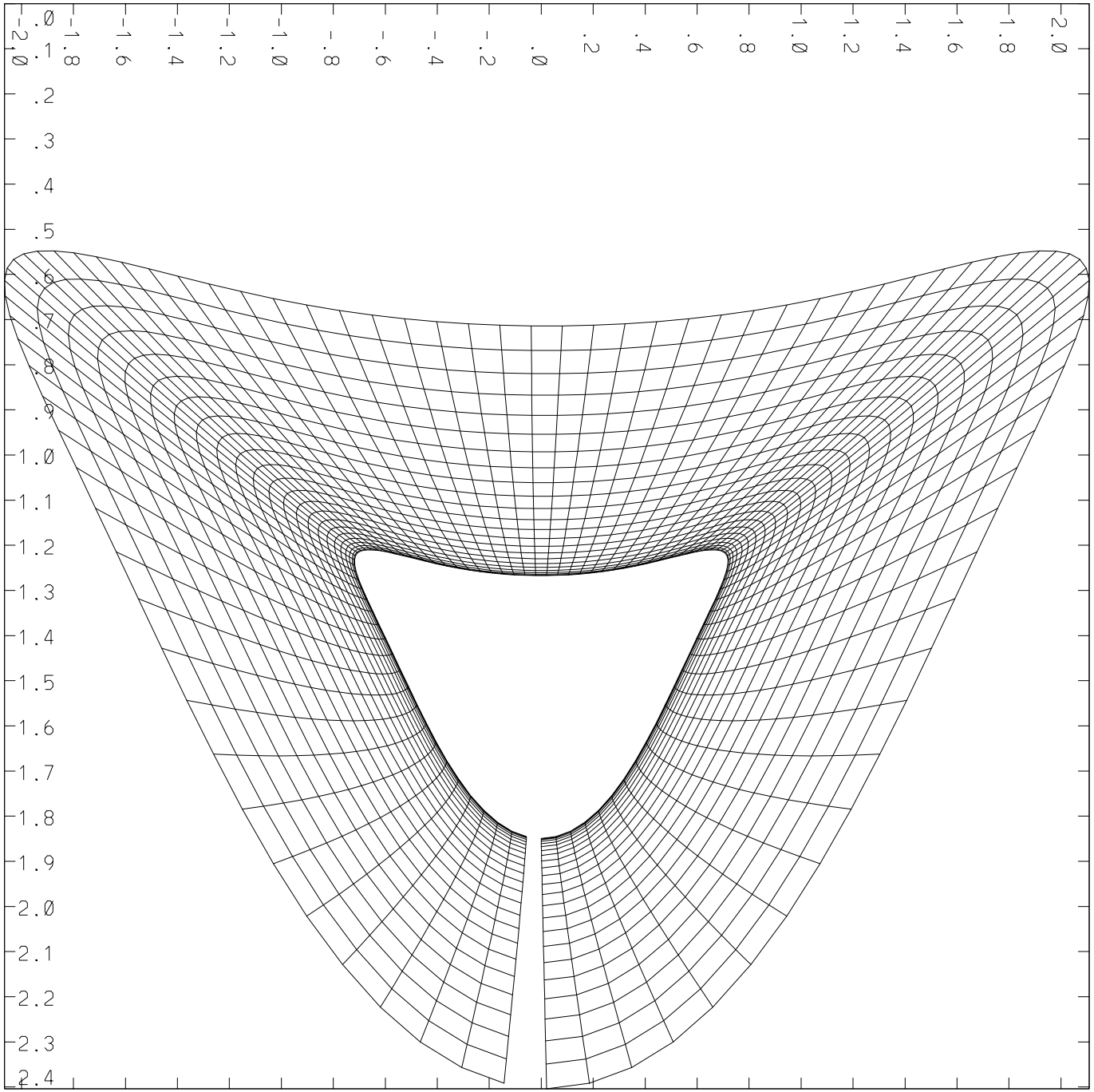
(3) New grids, type III;

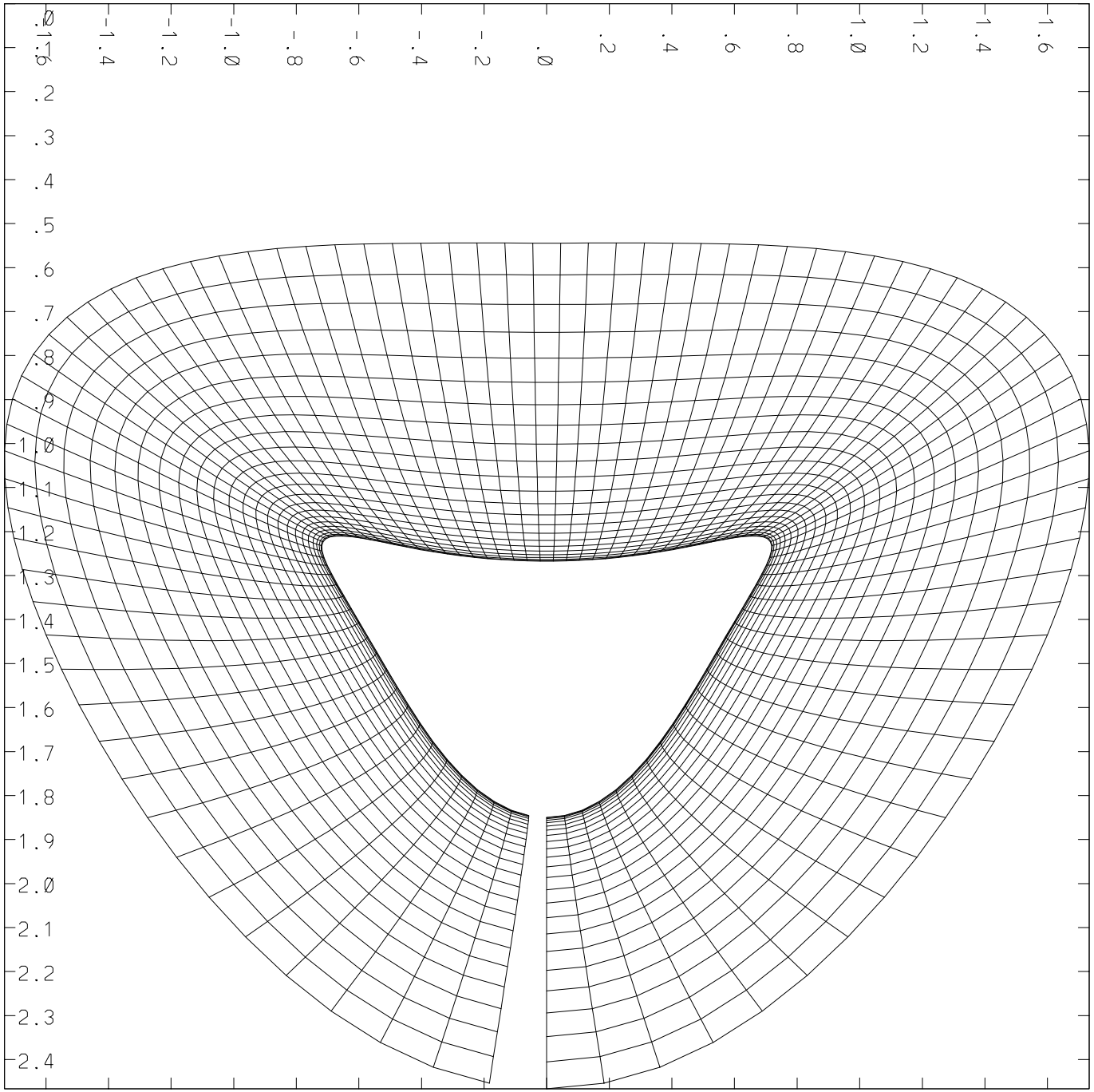
(4) Old eigenfunction of n=1 kink for case qas3_h3a;

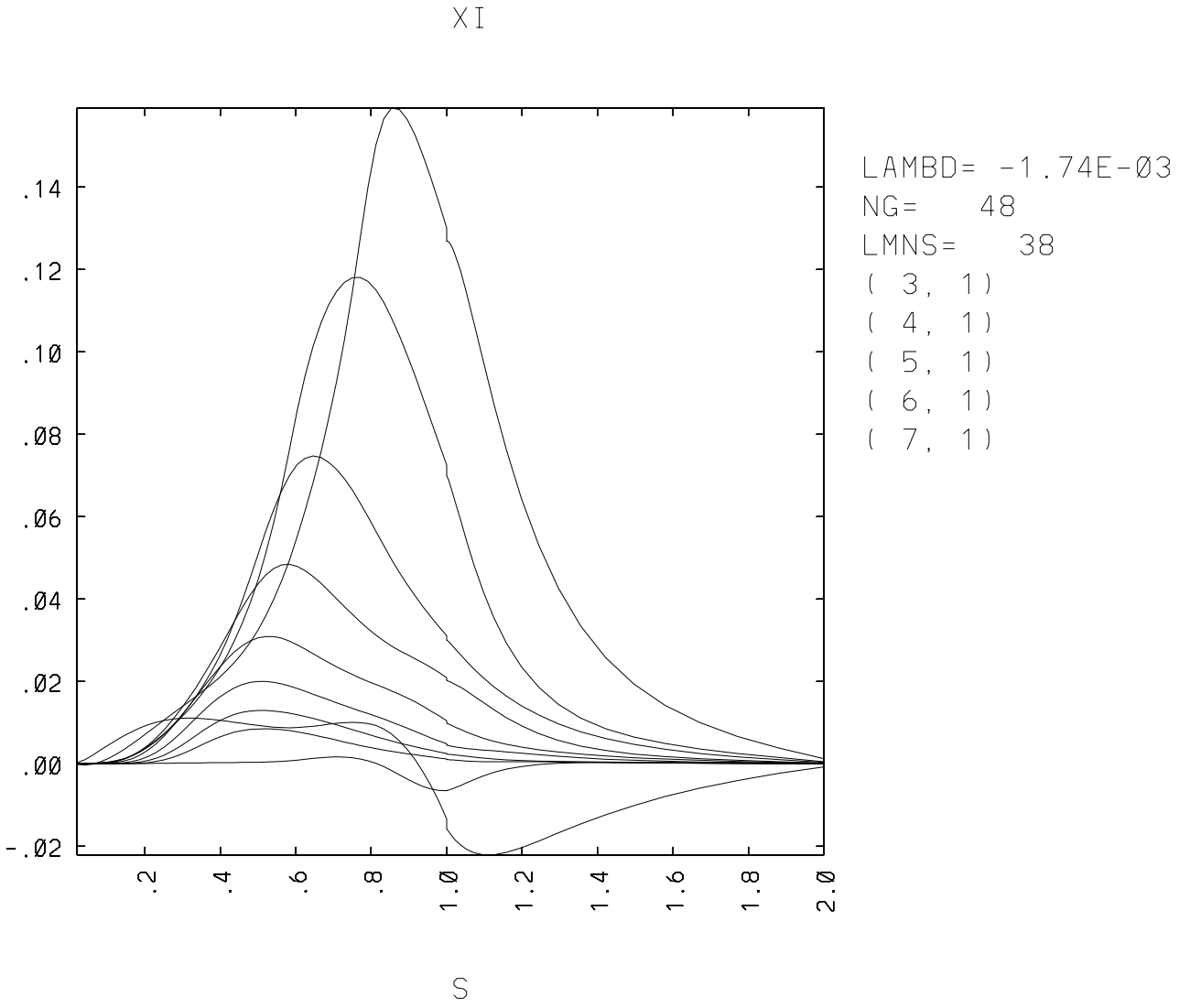
(5) New eigenfunction with new grids, type II;

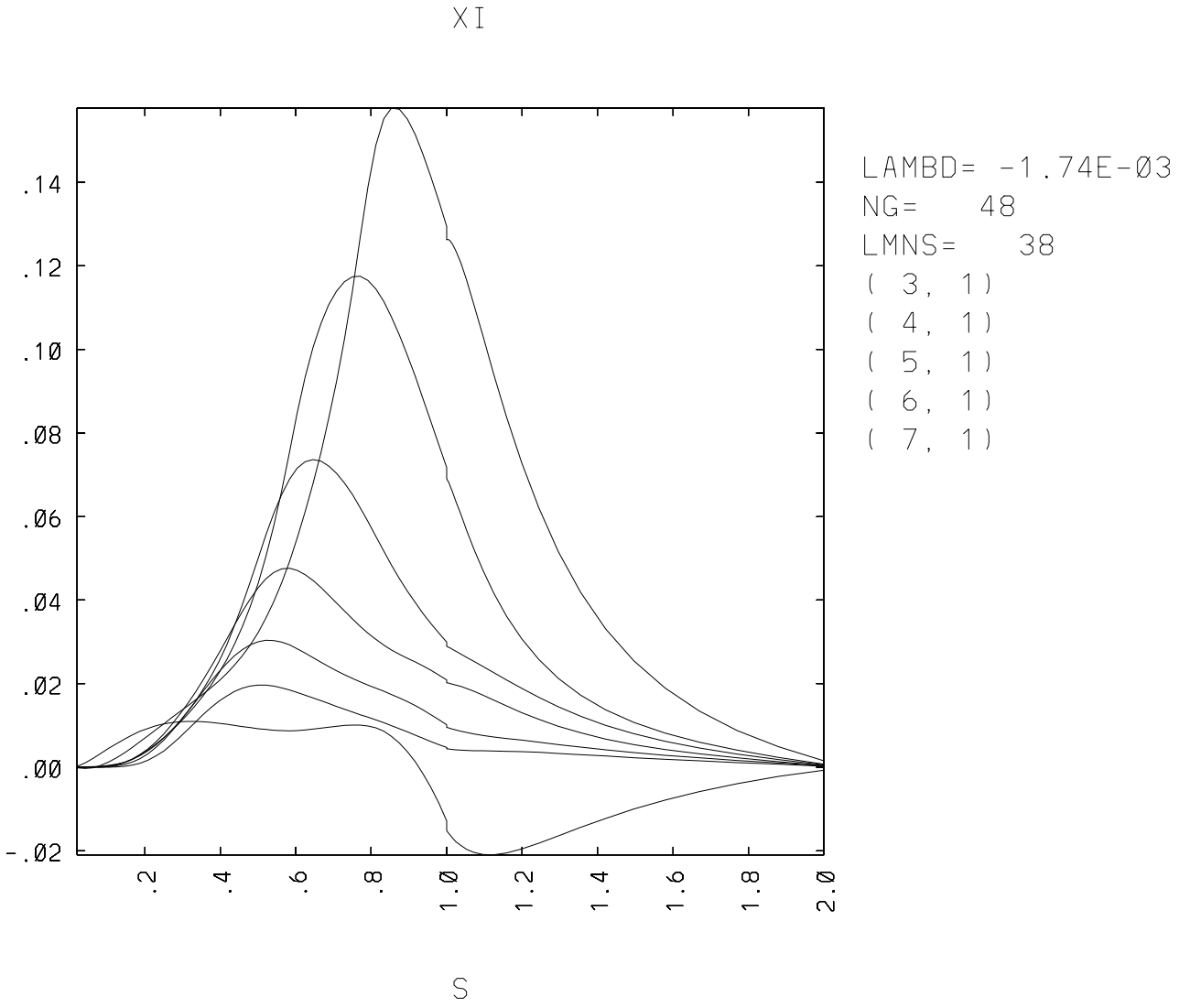
(6) A working new grids for an extreme case (type III).

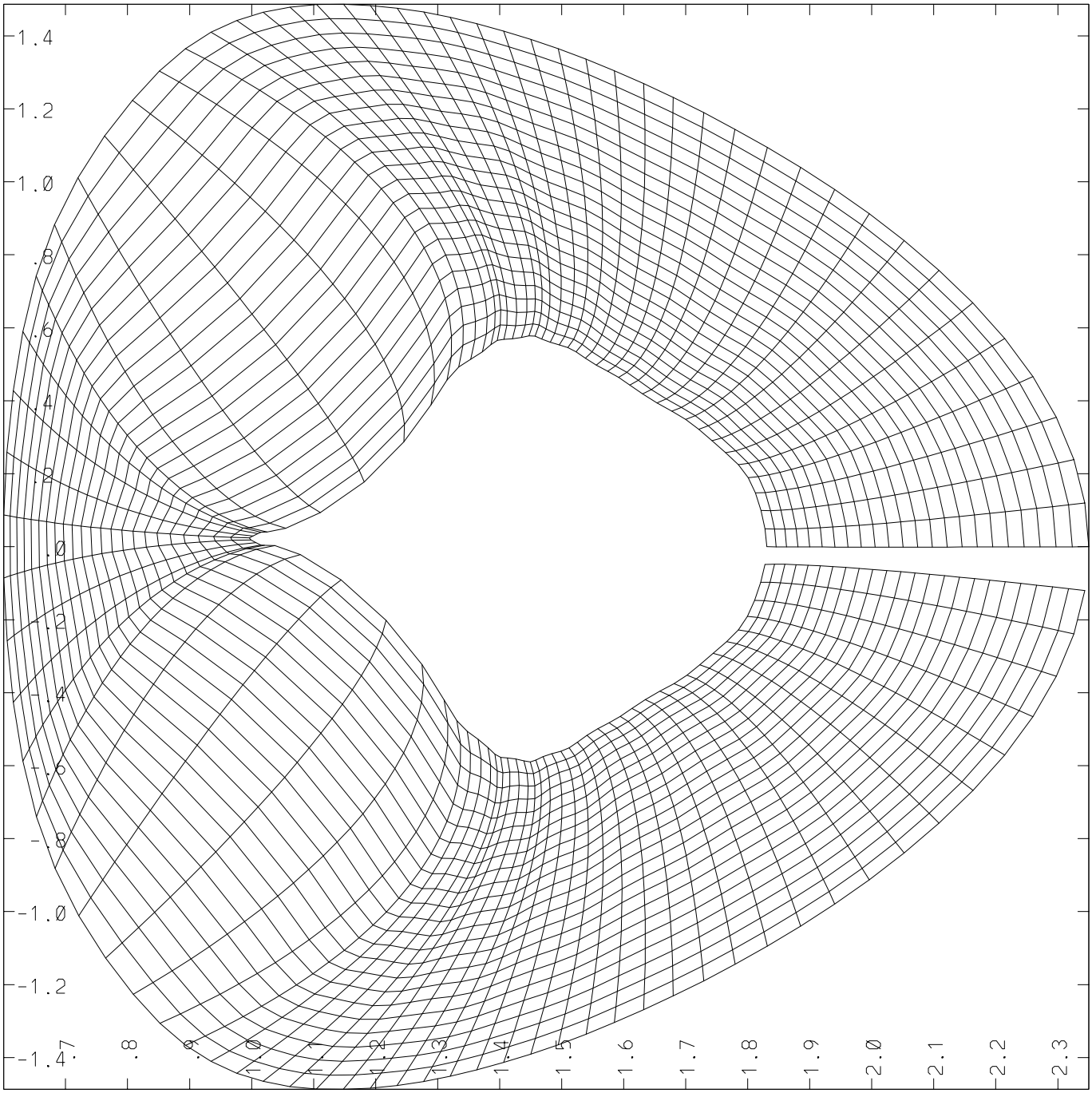

-1.2
-1.0
-.8
-.6
-.4
-.2
.0
.2
.4
.6
.8
1.0
1.2
1.4
.0
.1
.2
.3
.4
.5
.6
.7
.8
.9
1.0
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
1.9
2.0
2.1
2.2


-2.0
-1.8
-1.6
-1.4
-1.2
-1.0
-.8
-.6
-.4
-.2
.0
.2
.4
.6
.8
1.0
1.2
1.4
1.6
1.8
2.0
.0
.1
.2
.3
.4
.5
.6
.7
.8
.9
1.0
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
1.9
2.0
2.1
2.2
2.3
2.4


-1.6
-1.4
-1.2
-1.0
-.8
-.6
-.4
-.2
.0
.2
.4
.6
.8
1.0
1.2
1.4
1.6
.0
.1
.2
.3
.4
.5
.6
.7
.8
.9
1.0
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
1.9
2.0
2.1
2.2
2.3
2.4


XI
.14
.12
.10
.08
.06
.04
.02
.00
-.02
.2
.4
.6
.8
1.0
1.2
1.4
1.6
1.8
2.0
S
LAMBD= -1.74E-03
NG= 48
LMNS= 38
(3, 1)
(4, 1)
(5, 1)
(6, 1)
(7, 1)


XI
.14
.12
.10
.08
.06
.04
.02
.00
-.02
.2
.4
.6
.8
1.0
1.2
1.4
1.6
1.8
2.0
S
LAMBD= -1.74E-03
NG= 48
LMNS= 38
(3, 1)
(4, 1)
(5, 1)
(6, 1)
(7, 1)


1.4
1.2
1.0
.8
.6
.4
.2
.0
-.2
-.4
-.6
-.8
-1.0
-1.2
-1.4
.7
.8
.9
1.0
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
1.9
2.0
2.1
2.2
2.3