Lawrence Berkeley National Laboratory
Recent Work

Title

Implementation of a New Algorithm for Density Equalizing Map Projections (DEMP)

Permalink

https://escholarship.org/uc/item/23s897w3

Authors

Close, E.R.
Merrill, D.W.
Holmes, H.H.

Publication Date

1995-07-01

eScholarship.org Powered by the California Digital Library
University of California

LBL-35738
UC-405

Lawrence Berkeley Laboratory
UNIVERSITY OF CALIFORNIA

Information and Computing Sciences Division

Implementation of a New Algorithm for Density Equalizing Map Projections (DEMP)

E.R. Close, D.W. Merrill, and H.H. Holmes

July 1995

REFERENCE COPY
Does Not
Circulate
Bldg. 50 Library.
LBL-35738
Copy 1

Prepared for the U.S. Department of Energy under Contract Number DE-AC03-76SF00098

DISCLAIMER

This document was prepared as an account of work sponsored by the United States Government. While this document is believed to contain correct information, neither the United States Government nor any agency thereof, nor the Regents of the University of California, nor any of their employees, makes any warranty, express or implied, or assumes any legal responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by its trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or the Regents of the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof or the Regents of the University of California.

LBL-35738
UC-405

Implementation of a New Algorithm for Density Equalizing Map Projections (DEMP)

E.R. Close, D.W. Merrill and H.H. Holmes

Information and Computing Sciences Division,
Ernest Orlando Lawrence
Berkeley National Laboratory
University of California
Berkeley, CA 94720

July 1995

Copyright © 1995 The Regents of the University of California. Address all correspondence to Deane W. Merrill, Information and Computing Sciences Division, 50B-2239, Lawrence Berkeley National Laboratory, 1 Cyclotron Road, Berkeley CA 94720. Tel: (510) 486-5063. Fax: (510) 486-6363. Internet: dwmerrill@lbl.gov.

This work was supported by the Director, Office of Epidemiologic Studies; Office of Health; Office of Environment, Safety and Health; U.S Department of Energy under Contract No. DE-AC03-76SF00098.

Notice

The Government is granted for itself and others acting on its behalf a paid-up, nonexclusive, irrevocable worldwide license in this data to reproduce, prepare derivative works, and perform publicly and display publicly.

Beginning five (5) years after June 8, 1995, subject to two possible five year renewals, the Government is granted for itself and others acting on its behalf a paid-up, nonexclusive, irrevocable worldwide license in this data to reproduce, prepare derivative works, distribute copies to the public, perform publicly and display publicly, and to permit others to do so.

NEITHER THE UNITED STATES NOR THE UNITED STATES DEPARTMENT OF ENERGY, NOR ANY OF THEIR EMPLOYEES, MAKES ANY WARRANTY EXPRESS OR IMPLIED, OR ASSUMES ANY LEGAL LIABILITY OR RESPONSIBILITY FOR THE ACCURACY, COMPLETENESS, OR USEFULNESS OF ANY INFORMATION, APPARATUS, PRODUCT, OR PROCESS DISCLOSED, OR REPRESENTS THAT ITS USE WOULD NOT INFRINGE PRIVATELY OWNED RIGHTS.

recycled paper

Request for: **Loan** **Order No. -27051**

Date of Request: 6/26/2003 **Not Needed After** **Maxcost** Please contact us if you charge more

CALL NO. LBL 35738

Interlibrary Loans
Davidson Library - Bldg 525
University of California
Santa Barbara, CA 93106

Authorized By **Gary Johnson**

Title: **Interlibrary Loans, Head**

Shipped: 7/1/03
Charges: $ 0
Received:
Due Date: 8/1/03
Renewed:
Returned:
Ship Via:

Tel: **(805) 893-3436** Fax: **(805) 893-5290** ARIEL: **128.111.96.251**
E-Mail: **ill@library.ucsb.edu** WWW: **www.library.ucsb.edu**

For use of: TOBLER,WALDO/F Special Notes:

Author

Author: Close, E.R.;Merrill, D.W.;Holmes, H.H.

Title, Edition, Imprint

Title: Implementation of a New Algorithm for Density Equalizing Map Projections (DEMP) / **RN# LBL-35738**
Ed: REPORT # LBL-35738 Imprint: 1995

Verified in:

LAWRENCE BERKELEY LABORATORY
BLDG 50B ROOM 4206
LIBRARY - ILL
BERKELEY, CA 94720

Not sent because:

- ☐ Not on shelf
- ☐ Not found as cited
- ☐ Not owned
- ☐ Non-circulating
- ☐ Other

Contents

Figures iii

Tables iv

1. Introduction and Task Description 1
   - 1.1 PAREP project 1
   - 1.2 Prototype Small Area Study 1
   - 1.3 Density Equalizing Map Projections (DEMP) 2
     - 1.3.1 1988 LBNL Algorithm 2
     - 1.3.2 1991 LBNL Algorithm 3
     - 1.3.3 1993 Russian Algorithm 3
2. Description of Russian Algorithm 5
   - 2.1 Basic Features 5
   - 2.2 Summary of the Algorithm 5
3. Preparation of Input files 9
   - 3.1 Definitions and conventions 9
   - 3.2 Recommendations 10
   - 3.3 Map file construction 11
4. Pathological Regions 13
   - 4.1 Regions With Negative Initial Area 13
     - 4.1.1 External Boundary 13
     - 4.1.2 Holes 13
   - 4.2 Zero Population Land Areas 14
   - 4.3 Water Areas (target area unspecified) 14
   - 4.4 Areas With Present Area Negative 14
   - 4.5 Islands and Inlets 15
5. Illegal Boundary Crossings and Negative Magnifications 17
   - 5.1 The Problem 17
   - 5.2 A Modified Algorithm 17
6. The LBNL Implementation of the Russian Algorithm 19
   - 6.1 Overview of the LBNL Implementation 19
   - 6.2 Logical Structure of Program RLInt 20
   - 6.3 Program Options 21
     - 6.3.1 Transformation Iterations: 22
     - 6.3.2 Controlling the Type of Transformation Step 22
     - 6.3.3 Type of Integral Scaling 23
     - 6.3.4 Map Scaling 23
     - 6.3.5 Transformation Reset 23
     - 6.3.6 Output/Reports 24
       - Summary of icheck, idisk, isum, inow 24
       - List of I/O parameters and their function: 24
   - 6.4 Convergence Criteria 27
     - 6.4.1 Relationship Between the RMS and Hsum Functions 27
     - 6.4.2 Discussion of Machine Precision 29
7. Summary of Results 31
   - 7.1 Circle Expansion With and Without Scaling 31

7.2 Vermont Maps 34
7.2.1 The 44-Region Vermont Triangle Map 35
7.2.2 The 44-Region Vermont Hexagon Map 36
Comparison of Full-step vs. Equal Interval Transformations 36
Comparison of Equal Interval vs. 1/2 Step Transformations 36
Comparison of Russian Scaling vs. HH Scaling 37
Region Maps 38
7.3 Runs on 44- and 104-Region Vermont Hexagon Maps 42
7.3.1 RMS Convergence 43
7.3.2 Area Scatter Plot Convergence 48
7.3.3 Initial and Final Region Maps 55
7.4 Timing Comparisons 64
7.4.1 Timing the Line Integral Mapping Algorithm 65
7.4.2 Number of Iterations Required for Convergence 69
8. Future Work Required 71
8.1 Input Maps for the 4-County Area 71
8.2 Handling of Pathological Conditions 71
8.2.1 Correction Within DEMP program 71
8.2.2 Correction Outside the DEMP Program 71
8.3 Program Improvements 72
8.3.1 Elimination of Duplicate Boundary Points 72
8.3.2 Illegal Boundary Crossings and Poor Convergence 73
Illegal Boundary Crossings 73
Non-Boundary Points on Boundary Line Segments 74
Convergence Monitoring 74
Summary 74
References 77
Appendix A Integral Transformation and Correction Factors 79
A.1 Introduction 79
A.2 Derivation of the Mapping Integrals 79
A.3 Line Integral Evaluations 84
A.3.1 Point z not on Line 84
A.3.2 Point z at the Left End of the Line Segment 86
A.3.3 Point z at the Right End of the Line Segment 87
A.3.4 Point z Between the Left and Right End of the Line Segment 87
A.4 Special Case: Domain is a Circle of Radius R 88
A.4.1 Some Examples 88
A.4.2 The Expected Translation Vector 89
A.4.3 The Calculated Translation Vector 89
Evaluation of the Area Integral for a Circle 89
The Point z is Outside the Circle on the X Axis 89
The X-component of the Translation Vector: 90
The Y-component of the Translation Vector: 91
The Point z is Inside the Circle on the X Axis 91
The X-component of the Translation Vector: 92
The Y-component of the Translation Vector: 92
The *Calculated* Results for a Circle 92
A.4.4 The Correction factor 93
A.5. Calculation of Region Areas 94

Appendix B Description of RLInt I/O Files 97
B.1 Parameter and cl3 Data Files 97
B.1.1 Map Data File `RLInt.dat` (cl3 format) 97
B.1.2 Comments on the Region Data 98
B.2 Output Files 101
B.2.1 Description of Output files 101
B.2.2 Basic Output Files 102
B.2.3 Internally Stored Output Files 103
B.2.4 Intermediate Mapping Files 104
Appendix C. Example of Complete Run 107
C.1 Description of Example Run 107
C.2 Data input Files 107
C.2.1 File `RLInt.par` 107
C.2.2 File `RLInt.dat` 107
C.3 Data output Files 111
C.3.1 File `RLInt.std` 111
C.3.2 File `RLInt.msg` 112
C.3.3 File `RLInt.out.sum` 112
C.3.4 File `RLInt.out.plot` 125
C.3.5 File `RLInt.sum.0007` 133
C.3.6 Output Files Not Included in this Appendix 134

Figures

Figure 1. Illegal Region 17
Figure 2. Circle or Radius 1 Mappings 33
Figure 3. Triangle Convergence Plots 35
Figure 4. Hexagon Convergence Plots 36
Figure 5. Comparison of HH Scaling to Russian Scaling 37
Figure 6. Mapping 44 Hexagonal Regions 39
Figure 7. Mapping 44 Triangular Regions 40
Figure 8. Scatter Plots of Current Area and Target Area 41
Figure 9. Convergence of the Hexagon Mapping 42
Figure 10. RMS Function vs. Iteration: Pop80, White, Other, & Spanish 44
Figure 11. RMS Function vs. Iteration: Asianpi, Indiana, & Black 45
Figure 12. Convergence of the Asianpi 104-Region Map 46
Figure 13. Asianpi Convergence vs. Iteration Count 48
Figure 14. Area Scatter Plots for Vermont Pop80 49
Figure 15. Area Scatter Plots for Vermont White 50
Figure 16. Area Scatter Plots for Vermont Other 51
Figure 17. Area Scatter Plots for Vermont Spanish 52
Figure 18. Area Scatter Plots for Vermont Asianpi 53
Figure 19. Area Scatter Plots for Vermont Indian 54
Figure 20. Area Scatter Plots for Vermont Black 55
Figure 21. Initial 44- and 104-Region Vermont Map 56
Figure 22. Final Mapping 44 Regions pop80 57
Figure 23. Final Mapping 104 Regions pop80 57

Figure 24. Final Mapping 44 Regions White 58
Figure 25. Final Mapping 104 Regions White 58
Figure 26. Final Mapping 44 Regions Other 59
Figure 27. Final Mapping 104 Regions Other 59
Figure 28. Final Mapping 44 Regions Spanish 60
Figure 29. Final Mapping 104 Regions Spanish 60
Figure 30. Final Mapping 44 Regions Asianpi 61
Figure 31. Final Mapping 104 Regions Asianpi 61
Figure 32. Final Mapping 44 Regions Indian 62
Figure 33. Final Mapping 104 Regions Indian 62
Figure 34. Final Mapping 44 Regions Black 63
Figure 35. Final Mapping 104 Regions Black 64
Figure 36. Timing for Hexagon Regions 65
Figure 37. Timing of 4-County Map 66
Figure 38. Number of Points in Each Region for the 4-County Map 67
Figure 39. Cumulative Number of Points for the 4-County Map 68
Figure 40. Shared Region Boundary 72
Figure 41. Input Data File `RLInt.dat` 99
Figure 42. How to Use a Cut for Generating a Doughnut Region 101

Tables

Table 1. Graph Labels 34
Table 2. Vermont Hexagon Map RMS and Hsum Function Values 43
Table 3. Mapping a Circle 88

1. Introduction and Task Description

1.1 PAREP project

The purpose of the PAREP (Populations at Risk to Environmental Pollution) Project at Lawrence Berkeley National Laboratory (LBNL), an ongoing Department of Energy (DOE) project since 1978, is to develop resources (data, computing techniques, and biostatistical methodology) applicable to DOE's needs. Specifically, the PAREP project has developed techniques for statistically analyzing disease distributions in the vicinity of supposed environmental hazards. Such techniques can be applied to assess the health risks in populations residing near DOE installations, provided adequate small-area health data are available.

The FY 1994 task descriptions for the PAREP project were determined in discussions at LBNL on 11/2/93. The FY94 PAREP Work Authorization specified three major tasks: a prototype small area study, a feasibility study for obtaining small-area data, and preservation of the PAREP data archive. The complete FY94 work plan, and the subtasks accomplished to date, were included in the Cumulative FY94 progress report (reference 1).

1.2 Prototype Small Area Study

Task 1, the prototype small area study, demonstrated the use of PAREP data and methodology for similar studies around DOE sites. This task consisted of the following milestones:

- Obtain, document, and organize the small-area data needed for study - task report by 12/15/93 (reference 2).
- Manipulate map files to make them useful - task report by 1/15/94 (reference 3).
- Implement new Russian algorithm for density equalizing map projections (DEMP) - technical report by 4/1/94 (reference 4).
- Integrate data inputs (numerators, denominators, maps) - task report by 5/94.
- Develop nearest neighbor techniques for geographic data analysis - task report by 5/94.

A draft project report on the prototype four-county small-area study (reference 10) was due 8/31/94. The final DOE report was published on 1/2/95 (reference 13). A condensed version was presented at a CDC/ATSDR Symposium on 1/25/95 (reference 11) and submitted for publication on 4/1/95 (reference 14).

The present report (reference 4) is the users' manual and the task completion report for the third subtask: implemention of the new Russian algorithm for density equalizing map projections (DEMP). The latest version is available in the World Wide Web (WWW) at the URL listed in reference 4. Additional program modifications are described in Appendix E of reference 13.

For the prototype small area study, a four-county area in California (Fresno, Kings, Kern and Tulare) was selected because of the availability at LBNL of small-area health data. The data used were 401 childhood cancer cases reported between 1980 and 1988, from a study conducted by the California State Department of Health Services (DHS) (references 5 and 6). The study was initiated to investigate questions raised by a reported childhood cancer cluster in the community of McFarland, California.

The first draft interim report (reference 5) examined childhood cancer rates by cancer site, age, sex, race/ethnicity (Anglo, Hispanic and other), county, and land

use (rural vs. urban, and agricultural vs. non-agricultural). The calculation of population at risk is described in detail. Observed rates were found to be consistent with rates reported in other studies; the only significant departure from uniformity was that rates among children in urban non-agricultural areas are slightly higher than those in rural non-agricultural areas. The urban non-agricultural rates are comparable to urban rates elsewhere in California. Rates in agricultural areas are not elevated.

Interim report #2 (reference 6) examined the differences among geographic areas in greater detail. The four-county study area was subdivided into 101 communities. For each community the observed number of cases was compared with the number expected, assuming the underlying cancer rate to be uniform. Based on the Poisson distribution, six of the 101 communities had rates that fell outside 95% confidence limits (three with more cases than expected and three with fewer cases than expected). The result is consistent with the null hypothesis of uniform underlying rates. One community (McFarland) had an elevated rate outside the 99% confidence limit, exactly what would have been expected from chance alone.

The purpose of the prototype LBNL four-county study (references 10, 11, 13, 14) was to test the applicability of the technique of density equalizing map projections (DEMP) for analyzing small area disease distributions near DOE installations or other potential environmental hazards. Although the four-county study area is not near a DOE installation, the DEMP technique can be similarly applied in any area where adequate small-area health data exist. Valuable insight was gained by applying the technique to a real data set with its specific problems, rather than to a set of imaginary computer-generated cases.

Regarding the significance of the McFarland childhood cancer cluster, a statistically significant effect was reported in references 13 and 14, based on the results from a k'th nearest neighbor analysis. Due to uncertainties in the population data used, no epidemiologic conclusions can be drawn at the present time. A revised analysis is in progress (reference 15). Regardless of the final conclusions of the LBNL study, comparing the results from the LBNL analysis and the earlier DHS analysis will provide valuable insight for future studies of reported disease clusters around supposed environmental hazards.

1.3 Density Equalizing Map Projections (DEMP)

Density equalizing map projections (DEMP) have been used at LBNL to investigate geographic disease distributions in the vicinity of supposed environmental hazards. In a DEMP transformation, boundaries of geographic subareas (for example census tracts) are transformed so that population density is uniform over the entire transformed map. On the transformed map, the statistical analysis of the distribution of disease cases is simplified because the confounding effect of population density has been removed.

1.3.1 1988 LBNL Algorithm

The first LBNL algorithm, published in 1988 (reference 7), employed a radial expansion or contraction relative to the centroid of each subarea in the map. The radial transformation changed the area but not the shape of the particular subarea in question, while changing the shape but not the area of all other subareas. The resulting map depended on the arbitrary order in which the subareas were transformed; in addition, it was possible for subarea boundaries to overlap after the transformation. Case locations were transformed along with subarea boundaries during the DEMP transformation.

1.3.2 1991 LBNL Algorithm

A second LBNL algorithm, completed in 1991 (reference 8), subdivided the map into triangles. As a function of all the coordinates of all the triangle vertices, we defined (1) a constraint function H which vanishes only when each triangle reaches its desired target area, and (2) an objective function G which measures overall distortion relative to the original map. A minimization program adjusted all the vertex coordinates so as to minimize G subject to the constraint H=0. The final solution defined a linear transformation for each triangle, which was applied to all the case locations within that particular triangle. With the 1991 LBNL algorithm, solutions were uniquely defined and overlapping boundaries were avoided; however, the time required for computation was prohibitive. To limit computation time, considerable geographic detail had to be sacrificed.

1.3.3 1993 Russian Algorithm

In 1993 a new algorithm was published by Gusein-Zade and Tikunov (reference 9), in which the vector translation of each geographic map coordinate is calculated from the expansion or contraction of each infinitesimal area in the entire map. The translation due to a given subarea is calculated as a line integral around the boundary of that subarea. Convergence is achieved in a small number of iterations. Case locations are transformed along with subarea boundaries during the DEMP transformation. The present report describes the LBNL implementation of the algorithm described in reference 9.

2. Description of Russian Algorithm

2.1 Basic Features

The LBNL program RLInt (Russian Line Integral) is a direct implementation of the constructive algorithm described in Ref 9. A full description of this implementation is given below in Section 6, "**The LBNL Implementation of the Russian Algorithm.**" A derivation of the line integral which is the basis of this algorithm is given in Appendix A, "**Integral Transformation and Correction Factors.**" The basic features of this algorithm are that it is global in nature, the transformation applied at each iteration step depends on all regions in the map, and it is quadratic in the number of points used to represent the region boundaries. This latter statement means that if we double the number of boundary points, or equivalently the number of regions, the amount of computational work increases by about four times. It also appears to converge rather quickly to the target values, say in 10 to 20 iterations. The exact value is dependent on the initial map and the required accuracy of the final map.

2.2 Summary of the Algorithm

The authors of reference 9 describe the general problem of constructing cartograms and then describe, in a heuristic manner, the construction of an integral mapping algorithm. We include in this LBNL report a derivation of that algorithm and also the derivation of some special cases (see Appendix A).

In the present section we present an overview of the reasoning that leads to the line integral representation of the mapping.

We note that the mapping as initially conceived is continuous, but that it is applied as a discrete mapping. The discrete mapping has generated the results presented in this report and has led to the conclusions obtained. This is noted here because some of the results that we present and some of the conclusions that we derive may appear to be at variance with those that would be expected of the continuous mapping. We believe that there are no real contradictions implied by these differences.

The Russian authors start out with an infinitesimal transformation of a small region. The global region D has a density function defined on it, $\rho(x',y')$. The transformation makes the final density in the infinitesimal region equal to the target value, $\rho_f(x',y')=\rho_f$ (a constant). An arbitrary reference point $z=(x,y)$ exterior to the infinitesimal region is chosen. The point $z=(x,y)$ is any point in the region. The density at this point, $\rho(x,y)$, does not change. To summarize, a transformation is defined that maps the global region so that for all points inside the infinitesimal region, the density becomes constant, and for all points outside the infinitesimal region, the density does not change. However, the region does change in shape; the region points move.

If we observe the point z, the point moves due to the transformation that maps the infinitesimal region centered at (x',y'). If we now center the infinitesimal region at a new point in the plane, a new (x',y'), and reapply the transformation, the point z will again be displaced. If this is done for all points in the plane, (except for point z), we will in the end have a resultant position for z that corresponds to applying all the transformations to all the infinitesimal regions.

Another, and possibly better, way of understanding this is the following. Cover the region D with non-overlapping infinitesimal regions $\Delta x\Delta y$. Select a point z in the region for reference. Exclude that point's region from all the others. In each of the other infinitesimal regions apply the above described transformation. The global region will change in shape and the point z will move to a position determined by the sum of all the translations applied to each infinitesimal region. In each of the small regions the density will be the chosen constant target density ρ_f.

The constructive algorithm that the authors describe particularizes the transformation to that of a dilation of an infinitesimal circle that has the same area as $\Delta x\Delta y$. The principal interest is in the effect of this transformation on the point z. The authors show that the effect on the point z of transforming one infinitesimal region is given by $\left(\frac{\rho}{\rho_f}-1\right)\frac{\vec{r}\Delta x\Delta y}{2\pi r^2}$ where $\vec{r}$ is the vector from any point in the infinitesimal region $\Delta x\Delta y$ to the reference point z, $\rho=\rho(x',y')$ is the density at (x',y'), and ρ_f is the target density in the region $\Delta x\Delta y$ after the mapping. The transformation is applied to all the infinitesimal regions that cover D and the resultant push on the point z is the vector sum of the individual pushes from each region. When the limit is taken as the regions shrink to zero and their number becomes infinite, this sum becomes $\int_D\left(\frac{\rho}{\rho_f}-1\right)\frac{\vec{r}dxdy}{2\pi r^2}$. This integral is the basic formula used to define the mapping. The region D is assumed to be the union of a collection of subregions D_i. In this case the integral can be written as the sum of integrals over each of the subregions. Within each subregion the density is assumed to be a constant ρ_i. and the constant target density is ρ_{f_i}. This allows the density coefficient to be placed outside the integral. The transformation then becomes $\sum_{D_i}\left(\frac{\rho_i}{\rho_{f_i}}-1\right)\int_{D_i}\frac{\vec{r}dxdy}{2\pi r^2}$.

The authors then recognize that the area integrals that define the x and the y components of the transformation vector can be replaced by line integrals around the region boundaries. This application of Stoke's theorem results in the transformation vector being defined by

$$\Delta z=(\Delta x,\Delta y)=\sum_{D_i}\left(\frac{\rho_i}{\rho_{f_i}}-1\right)\left(\frac{1}{4\pi}\right)\left(-\oint_{D_i}\ln\left[(x-x')^2+(y-y')^2\right]dy',\oint_{D_i}\ln\left[(x-x')^2+(y-y')^2\right]dx'\right)$$

A detailed derivation of this result is given in Section A.2 of Appendix A.

Having derived this integral transformation the authors then proceed to use it to complete the one-parameter family of transformations $h_t(z)$ which transform the density $\rho(z)$ to the constant density $\rho_f(z)$ as the parameter t goes from 0 to 1, continuously. It is this discussion that leads to the discrete, iterative transformation. In that realization of the transformation the above interval from 0 to 1 is subdivided into N equal subintervals, *e.g.*, 10, and the above transformation is applied in 10 equal steps. This sequence of transformation can be written as

$$z_i = z_{i-1} + c_i \Delta z_{i-1};$$

$$z_0 = z;$$

$$\Delta z_{i-1} = (\Delta x_{i-1}, \Delta y_{i-1});$$

$$c_i = 1/(N - i + 1);$$

$$i = 1,2,\ldots N$$

At each step of the transformation the resultant transformation vector is constructed from the current values of the regions' boundaries and densities. In this report we refer to the iterative transformation, with c_i as defined above, as the Russian algorithm.

We noted at the beginning of this section that this iterative sequence, which is a discrete realization of the continuous transformation, does not necessarily map as expected. For example it does not give the best convergence rate when applied to the test maps presented in this report. However, there is no question that the transformation must be applied repeatedly to generate a sequence of resultant maps that converge to the desired target densities in the subregions of the final map.

3. Preparation of Input files

The input files required by the Russian Line Integral program RLInt are described in Section B.1 of Appendix B. Two input files are required: a map data file `RLInt.dat`, and a parameter file `RLInt.par`.

Construction of the parameter file `RLInt.par` is a trivial task. Examples are given in Section C.2.1 of Appendix C.

The map file `RLInt.dat` is considerably more complex and is normally constructed with a computer program. The method chosen will depend on the format of the input data available. Three basic inputs are required:

boundary points (x-y coordinates of polygon boundaries);
population data corresponding to the same polygon boundaries;
non-boundary points (x-y coordinates).

3.1 Definitions and conventions

The boundary points should be in a right-handed Cartesian coordinate system (x increasing left to right, y increasing upward). The x- and y-coordinates should have equal size (for example, kilometers, *not* degrees of longitude and latitude). For small regions of the earth's surface, most common projections are suitable.

Simple polygons, either land and water polygons, are described counter-clockwise. We define these polygons as having positive area. Polygons which are traced clockwise have negative area.

A doughnut-shaped region (for example a county with a lake, or a lake with an island) can be described in one of two ways, as illustrated in Figure 42 (Section B.2.1).

(a) One can describe the doughnut "hole" (unshaded) with an additional clockwise polygon. This representation corresponds to the left portion of Figure 42, if the diagonal "cut" is ignored. The lake within a county (unshaded) has positive area and is described counter-clockwise. The county (shaded) is described by two polygons: the counter-clockwise outer boundary, and the clockwise lake boundary. Its net land area is equal to the area of the clockwise outer boundary, minus that of the counterclockwise inner boundary.

(b) One can "slice" the doughnut as shown in the right portion of Figure 42 so that the doughnut region (shaded) is simply connected.

The two representations are computationally equivalent. The net contribution from the "slice" is zero because it is traversed in both directions.

Optionally, for completeness, the entire region may be described by a clockwise external boundary. In this case, every line segment in the entire map is traced in both directions, and the net area of the entire map is zero. This representation corresponds to the DIME (Dual Independent Map Encoding) format developed by the Census Bureau, in which every segment is identified by its two adjacent polygons. The clockwise external boundary (the boundary of the surrounding "ocean") is optional because it is explicitly ignored in the RLInt calculation.

The entire map area should be covered by non-overlapping polygons. To ensure that this condition is met, it is sufficient to require that:

- every polygon must have net positive area;
- no polygon has boundary segments that intersect each other; and
- the net sum of all the polygon areas (including lakes) is equal and opposite to the (negative) area of the exterior boundary.

The target areas of the polygons should be specified in the same units as the coordinates of the polygon boundaries. For example, if coordinates are given in kilometers, target areas must be given in square kilometers.

Target areas must be positive or zero. A negative value indicates that the area is free to float during the density equalization process.

Boundary points and non-boundary points must be expressed in the same units.

3.2 Recommendations

In principle any map file satisfying the criteria above is a valid input file for RLInt. In practice, however, the following additional guidelines are useful.

Include no more detail than necessary. Computation time increases approximately as as the square of the number of boundary points in the map file.

On the other hand, density equalization cannot be achieved if the polygons have too few boundary points. For example, a map composed entirely of triangles cannot be density equalized as easily as the corresponding hexagon map obtained by bisecting all the triangle boundaries. An example is given in Section 7.

As described in Section 8, the algorithm breaks down (*i.e.*, the computation is invalid and will not converge) if polygon boundaries begin to overlap during the transformation. To avoid this occurrence, provide enough iterations and provide enough geographic detail in the troublesome areas. Sparsely populated areas, which shrink to narrow slivers between more populous areas, tend to give the greatest problems.

If water regions with unspecified target area become negative, it may be necessary to constrain their target areas to be exactly zero. If islands or other coastline areas overlap each other, it may be necessary to fill the intervening regions with water polygons whose target area is zero.

Numerical precision may be a problem if the map coordinates have large numeric values. Use a coordinate system whose origin (0,0) is not too far from the center of the map. If coordinate values are large compared with 1, use the map scaling parameter `iara` described in section 6.3.4.

Convergence is improved and distortion is minimized, if all the target areas are normalized so that total target land area is equal to the initial land area.

For some applications only the total (adjusted) area of each polygon is of importance. This is true for the mapping applications described in reference 9. The authors of

that paper included considerable geographic detail in each polygon boundary, and did not check for local variations in magnification.

For other applications, notably case mapping in public health applications, it is important that all portions of a given polygon receive the same magnification. Nonuniformities of magnification are automatically avoided in a map composed entirely of triangles, as was done in reference 8 and in the Vermont triangle map of Section 7.2.1. However, maps composed entirely of triangles do not converge easily. Hexagon maps are a convenient compromise in applications where uniformity of magnification is required. Regions that are highly distorted by density equalization can be further triangulated as necessary.

3.3 Map file construction

The physical format of the map file `RLInt.dat` is defined in Section B.1.1 of Appendix B. A complete example is provided in Section C.2.2 of Appendix C.

How the map file is constructed will depend on the format of the input data available. That discussion is outside the scope of this report. Construction of map files at LBNL is described in detail in reference 3. Programs are described which convert map files among a variety of formats, including:

- SEEDIS MAPEDIT format (polygon)
- National Planning Data Corporation (polygons)
- NICKEL (segments of polygons)
- DIME (inter-polygon segments)
- EDIME (linked geocodes, points, and segments)
- FDIME (enhanced EDIME, with calculated areas)
- TIGER (linked nodes and chains)
- SPlus (linked geocodes, chains, and polygons)
- CL3 (polygon: the format of `RLInt.dat`)

The CL3 output subroutines can be easily integrated with input routines that read the user's own map files and population data.

In addition, utility programs are described which:

- Obtain population data from SEEDIS or from 1990 Census CD-ROMs
- convert the population and case data for use with the map files
- correct certain types of map errors
- remove "slivers" due to boundary mismatching
- remove unnecessary geographic detail
- perform a Delaunay triangulation
- subdivide segments to produce hexagon maps or provide additional detail;
- display map files with the use of SPlus.

Numerous examples are provided via WWW.

4. Pathological Regions

In general the maps to which the transformation is applied have regions for which the target area is positive. However, there are regions for which this is not the case. There are also regions that we want to let float; *i.e.*, we don't care what their areas are, or what their shapes become as long as they don't "get into trouble." We say that these regions are pathological; there is something about them that is different, non-standard, and which we must be able to handle. We discuss some of the pathologies in this section.

4.1 Regions With Negative Initial Area

These regions are special in that their boundaries are never used to construct the translation vector. Their boundaries could be entered as non-boundary points, and they would be mapped to wherever the transformation takes them. However, it is more natural to leave them as regions in the map because they are used by the associated mapping programs described in Section 3.3. For the maps that we are interested in there are two regions that fall into this classification.

4.1.1 External Boundary

For all the examples presented in this report, the first region appearing in the file `RLInt.dat` is really the sum of all the subregions. This region furnishes a total region area and an external boundary. The total region area is preserved as the map is transformed. This is true only if the target areas in the input file have been scaled to preserve total area. The maps that we consider in this report have that feature. This scaling is a way of controlling the map size and ensuring that the spread in target areas is not too great; some areas shrink and some grow in size.

As long as the transformation is proceeding correctly, the external boundary will be identical to that obtained from the outer subregions. If the map is badly distorted this may not be the case, and this discrepancy will be visible in the final map. Also, the total area of the subregions should be the same (in absolute value) as that of the external boundary. If the map deteriorates in quality the sum of the subregion areas will begin to differ from that of the external boundary. Describing the boundary of this region in clockwise order implies that the area is negative.

4.1.2 Holes

Holes include both holes in land regions (lakes) and holes in water regions, which are islands. Holes can be nested, for example an island in a lake, or a lake in an island.

A hole in a region can be represented as a separate region with negative area (with boundary described clockwise). However, this means that to get the proper region area this negative area must be subtracted from the total region area. Presently, this is not done by the program; so there would be a problem if the hole were furnished this way.

One way to place a hole in a region is to define a region with a cut. We have shown, how this can be done in Figure 42, "How to Use a Cut for Generating a Doughnut Region," which is in Appendix B, Section B.2.1 of this report. When this is done the line integral along the cut cancels, the area of the regions is correctly represented, and the hole has boundary points that represent the inner boundary of the region. In principle, there is no problem with furnishing a number of holes. In practice, we have to be

conscious of the fact that these holes have the possibility of drifting on top of, or into, one another. The holes have separate boundary points which are the inner boundary points of the region. There is no coding in the current version of RLInt to ensure that these boundaries do not overlap as they are transformed.

4.2 Zero Population Land Areas

The algorithm described in reference 9 requires a magnification coefficient calculated as

$$(M-1)=\left(\frac{\text{current region density}}{\text{target region density}}-1\right)$$

In the maps we need to consider there are land areas for which the target region area (*i.e.* the population) is zero and the target region density is infinite. In order to avoid this problem we have defined instead

$$(M-1)=\left(\frac{\text{target region area}}{\text{current region area}}-1\right)$$

Zero target area is accepted as valid input to the program. If the current region area happens to go negative while shrinking to zero we penalize it in the HSum function by including it in the error result. The magnification coefficient $(M-1)=\left(\frac{\text{target region area}}{\text{current region area}}-1\right)$ is undefined for regions with current area and target area both zero. Problems are avoided by using a small positive value instead of zero in the denominator if the current region area is zero.

4.3 Water Areas (target area unspecified)

Since our maps are of real geographical areas, we have to be able to handle regions such as lakes, or those which behave in the mapping as lakes. We call these regions water areas. If they are small, one may choose to specify the target area as zero, exactly like unpopulated land areas. For larger water areas it is preferable to leave the area free to float, which is done by providing a negative target area. Such areas are mapped, but they do not enter into the calculations. They are simply pushed around and fall where they fall. Their areas are calculated, locally, and have whatever values they have. However, as long as they do not "get into trouble," they are ignored. They can have a problem if their boundaries intersect or if they turn upside down; in the latter case, their calculated area changes from positive to negative.

Note that when the region areas are added up to get the total area, no account whatsoever is taken of the water areas. If a water area is not included in the region, or total area, it will stay not included. If it was included in the original region area data, it will not be taken out. These areas are ignored, except for being mapped. They could alternately be mapped as non-boundary points. Again, as with holes, there is the possibility of them going partially or completely out of their region, or overlapping with other water areas.

4.4 Areas With Present Area Negative

Normal regions (those other than holes and the external boundary) are described counterclockwise in the initial map, so that their initial area is positive. It is possible, in the course of mapping, that their area becomes negative. If this should happen, they continue to be mapped, but the present version of RLInt no longer uses them in constructing the translation vector. . They do not enter into the calculation of the RMS function; however, they are included (penalized) in the Hsum function. Should these regions again become positive in area, they are re-included in the construction

of the translation vector. As we discuss in Section 8, "**Future Work Required**" the program should be modified later to prevent normal regions from going negative. Additionally, regions with either positive or negative area can have intersecting boundaries, which are illegal. Currently no check is made to trap and correct this illegal mapping.

4.5 Islands and Inlets

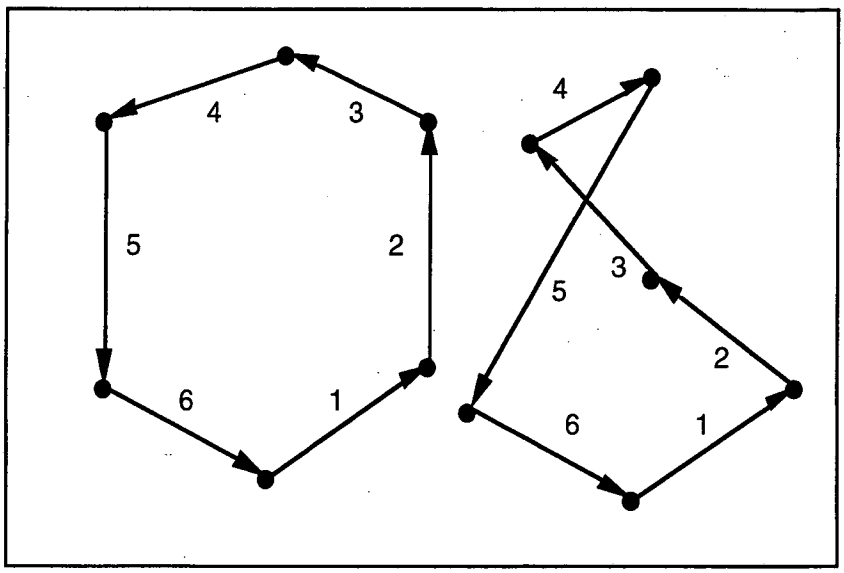
In the course of mapping, it is possible for islands to overlap each other, or for land areas on two sides of a narrow inlet to overlap each other. If this occurs, it can be prevented by filling the intervening water area with triangles and requiring that no triangle have negative area.

5. Illegal Boundary Crossings and Negative Magnifications

5.1 The Problem

The line integral algorithm that we have implemented in program RLInt has the potential to distort maps to the point that regions have boundaries that cross upon themselves, or in the worst case the regions fold over completely and the areas are definitely negative. In the maps that we transform, regions will always have target areas that are positive, or possibly zero; there are no negative target areas. Thus, a region for which the magnification $M = \left(\frac{targetArea}{currentArea} \right)$ becomes negative is not properly defined. It once had a positive magnification, but during the mapping process it was mapped into a region with a negative area. Also, before reaching that stage it is possible that a region has boundaries that cross. We show such a region in Figure 1, "Illegal Region."

Figure 1. Illegal Region



In this figure on the left we show a region that is legal and on the right the same region after it has been pushed into an illegal configuration. We note that the region has a boundary segment, 5, that crosses another boundary segment, 3. The total area is positive, but the small triangular area at the top is traversed in the clockwise direction and the area contribution from line segment 4 subtracts from the total area. In effect the region's area is the algebraic sum of a positive and a negative area. We can see from the above picture that the present area (and hence the magnification) can go negative, even without the whole boundary of the region folding over.

We discuss below in Section 8.3.1, "**Illegal Boundary Crossings**," what one might to do to prevent this type of mapping. Presently illegal boundary crossings are not detected, except by visual examination of the resultant map. The examples shown in this report have some regions in some of the maps that do in fact have such illegal crossings.

5.2 A Modified Algorithm

The problem of illegal boundary crossings and negative areas arises from two different aspects of the mapping. The first, and probably the most important, is the fact that this is a discrete representation of a continuous mapping. It is discrete in that the boundaries are represented by a finite number of boundary points. We ignore the missing boundary points and assume that these points are connected by straight lines. This representation permits the evaluation of the line integral in

closed form, but it does not really represent the boundaries of the regions. One solution is to furnish more boundary points, which greatly would increase the calculation time.

Another aspect of the mapping that we have discretized is the mapping step. In principle, the transformation from the initial map to the final map is continuous. This continuous transformation has been replaced by a finite number of steps. In the original Russian mapping algorithm the interval was sudivided into equal steps, 10 or so. As is shown in Section 7.2 of this report, it turns out in practice that the full step, fixed point, mapping converges more quickly than the Russian algorithm in those cases where it can be applied without leading to boundary crossings or negative areas.

Our investigation of the properties of this integral mapping has shown that when applied to a simple circular region, it generates too strong a push for points near the boundary, inside and outside. For the case of a circle, the correct push can be simply calculated by knowing what the final area of the circle must be. We have tried scaling the displacement vector so that the push inside the circle, and on the boundary points, is exactly correct in the case of a circle. This scaling is discussed in Section 7.1 of this report.

The push from a particular region depends primarily upon its area. One can construct a circle having the same area and centroid as the polygon. We define R to be the radius of the circle, and r to be the radial distance of the point being pushed. Inside the circle the correct scaling factor is $\frac{\left(2\sqrt{M}-1\right)}{(M-1)}$., which has been verified both theoretically and experimentally. For points inside the circle (r less than R) the scaling factor does not depend on r or R. For points outside, the correct scaling factor is

$$\frac{\sqrt{\left(r^2+R^2(M-1)\right)}-r}{R^2(M-1)/(2r)}$$

See Section 7.1, "**Circle Expansion With and Without Scaling,**" for a discussion of these scale factors.

We have not investigated the effect of using this modified algorithm. The present version of RLInt could easily implement this scaling. Should the input data structure be changed from polygons to boundary line segments (DIME format) it might require a little more work to apply this type of scaling.

For real problems with a large number of polygons and a large number of iterations the scaling does not make much difference. In most of our work we choose to ignore it for reasons of comparability with the implementation of reference 9.

6. The LBNL Implementation of the Russian Algorithm

6.1 Overview of the LBNL Implementation

At LBNL the Fortran program RLInt (Russian Line Integral) is a straightforward implementation of the cartogram mapping described in reference 9. The derivation of the area integral that generates the mapping is described in Appendix A, "**Integral Transformation and Correction Factors**," of this report. Appendix A also shows how this area integral can be replaced by line integrals around the boundaries of the regions that make up the map. This line integral representation is what has been implemented in RLInt.

For the purpose of understanding the LBNL implementation of the mapping algorithm it is convenient to consider the original map to be a region D that is composed of N subregions D_k. We thus have that $D = \sum_k^N D_k$. We assume that the regions do not overlap. In each region we have a current quantity and a target quantity. The object of the transformation is to change the regions D_k so that after the completion of the mapping the current quantities are the same as the target quantities. In the original Russian algorithm the quantities used were the current and target densities of the subregion, respectively ρ_k and ρ. We choose to represent the mapping coefficient (M-1) as $(A - A_k)/A_k$ rather than $(\rho_k - \rho)/\rho$ because the target density ρ is undefined for a region of zero target area, *i.e.* zero population.

In principle the transformation that takes the map from its original configuration to its final configuration is continuous. This is not the case for the numerical implementation of the mapping, however. The discrete transformation, as given in reference 9, divides the transformation into equal steps. For example 10 steps might be used. In that case the transformation is generated and only 1/10-th of the transformation is applied on the first step, then the image from that application is used to generate a new mapping and for that mapping 1/9-th of the transformation is applied. This is repeated until the full number of steps, 10 for this example, has been taken. After that, fixed point iterations are applied until the difference between the current region quantities and the target quantities is less than a prescribed error. By fixed point or full step iterations we mean that the full transformation is applied. In this report we refer to a mapping where the step sizes are (for example) 1/10, 1/9, ..., 1/2, 1, as the Russian algorithm.

A close examination of the results obtained when this algorithm is applied to test regions has caused us to incorporate some optional modifications to the Russian algorithm. In particular, we have the option of scaling the transformation so that it correctly maps (in a single full step) the interior of a circular region. See Section 7.1 "**Circle Expansion With and Without Scaling.**" We have also found it useful to choose an optional fixed size subdivision of the mapping step. We have chosen 1/2 as a convenient, suitable value. For this type of subdivision and 10 transformation steps, we would take for each step exactly 1/2 of the generated transformation. In this report we refer to a mapping where the step sizes are 1/2, 1/2, ..., as the binary algorithm.

The results of applying these different mappings are described in Section 7.2.2, "**The 44-Region Vermont Hexagon Map,**" of this report.

The transformation that we have implemented is global in nature, meaning that the transformation applied to any point in the map depends on all the other points in the map. In principle, one would have to do the area integral for each region and apply the result to all the points in the map. The fact that the area integral is represented by a line integral around the region boundaries means that one needs to map only the boundaries. Furthermore the boundaries are represented by line segments, so in practice only the line segments need be mapped. In practice, what we really transform are the end points of the line segments. We calculate the mapping transformation from the line segments that have these endpoints, then we map only the end points to new positions. The simplifying assumption is made that straight line segments between these new end points represent the correct new boundary; these are used to generate the transformation for the next step in the discrete mapping.

We have put into RLInt the ability to map arbitrary points; *i.e.*, non-boundary points. If these points coincide with the end points of the boundary segments, they are mapped exactly as the boundary points. If they are anywhere else in the map, they will move to wherever the transformation takes them. The maps to be transformed have such non-boundary points that we wish to keep track of. The only difference of these points from the boundary points is that they do not generate any of the map transformations; they do not push on other points.

The mapping algorithm as described in Appendix A has some numerical limitations. The basic problem is overflow during the calculation of the analytic function that gives the line integral along one of the boundary segments. The length L of the segment, and also the distance of the mapped point z from this line segment, cannot become too large. The limit on these values depends on the computer that is being used. To avoid these problems the map is scaled to the unit square (|x|<=1, |y|<=1). This scaling is done automatically by the program and it can be applied to all the input maps. Thus, large maps are shrunk and small maps expanded. This allows the program to work within a predefined range of numerical values.

6.2 Logical Structure of Program RLInt

The program RLInt consists of a main program, RLInt, and a mapping subroutine `rtrans`. It also has two housekeeping subroutines: one that writes files and one that deletes files.

The main program RLInt reads parameter data and map data from disk files. It initializes program variables and then calls the subroutine `rtrans` to perform the transformation that takes the initial map to the final map. Upon completion of this mapping task it writes requested reports and then exits.

The subroutine `rtrans` is the LBNL implementation of the Russian line integral algorithm. The basic inputs to `rtrans` are: the number of regions, the end points of the segments that define the boundaries of these regions, the number of such points for each region, the total number of non-boundary points, the coordinates of these non-boundary points, the current areas of the regions (calculated from the boundary points), and the target areas of the regions. In addition, we specify the desired error to be met, the number of transformation steps to apply, the type of transformation step (Russian or binary), and the number of fixed point full step iterations to perform.

The data are read and `rtrans` performs the requested mapping. Upon completion of the mapping, the program returns the transformed map. In principle, the returned

map is suitable as an input map that can be used to continue the mapping. This, of course, is not done if the map has converged. However, it is possible to stop after a specified number of steps, then use the intermediate map as a starting point for continuing the mapping.

As it performs the mapping sequence, `rtrans` can write reports about the intermediate maps. It also keeps track of badly distorted regions and can generate a history of the discrete mapping steps. These reports and history are controlled by parameters that are specified in a parameter file, as are the number and type of iterations and the scaling of the mapping function.

Calculating the full step transformation is straightforward. The k-th region, with N_k boundary points, is D_k (k = 1, 2,..., N). The i-th boundary point of D_k is z_i^k (i = 1, 2,..., N_k). The transformation vector Δz_i^k for this point is the sum of the line integral contributions from all the segments that make up the boundaries of all the regions. See Appendix A.3, "**Line Integral Evaluations**," which describes the evaluation of the line integral along these line segments. In an identical way, the subroutine also constructs transformation vectors for all the non-boundary points, if there are any.

At this point, we have the original map points z_i^k and the transformation vector Δz_i^k for each of these points. At each step of the iteration, the present area of each region is available. Before applying the transformation, all current region areas are compared with the target region areas. If the map has converged (*i.e.*, if the global error criterion for the final map has been met) then the transformation is not applied; the subroutine considers the current map to be the last map in the sequence so it exits, returning control the calling program.

The calculation of each transformation vector Δz_i^k requires the value of the line integral around all the boundaries of all the regions. The current area of each of the local regions is also calculated as described in Appendix A.5, "**Calculation of Region Areas**." These areas are used to check whether the error criterion has been met. At this stage, we have the current map and the transformation vectors that produced the map from the preceding map. It is thus possible to go back one full step any time during the mapping process. This can be useful, as we will see below, in the discussion of the results.

This subroutine's primary task is to construct the transformation vectors and perform the discrete transformation mapping of the furnished maps. It also writes intermediate reports as requested, and checks on the convergence criterion. Optionally (if scaling is requested) it scales the input map to the unit square, and upon exit unscales the resultant map so that it will have the correct values for the boundary point coordinates and transformation vectors, along with the correct areas.

6.3 Program Options

The program RLInt has a number of parameters that allow the user to select different options for a given run. RLInt has two input files, `RLInt.par` and `RLInt.dat`. If desired, there can be some input from the terminal to control the program calculation. The file `RLInt.par` is a parameter file that allows the user to set the available program options. In the following sections we describe these option parameters. Appendix B, "**Description of RLInt I/O Files**," includes complete examples of the files `RLInt.par` and `RLInt.dat`. Both files are read using FORTRAN list directed

read statements. A fuller explanation of how that effects the data input is given below in Appendix B.

We will now list and describe the available parameters.

Additional program options implemented in late 1994 are described in Reference 13.

6.3.1 Transformation Iterations:

There are two iteration loops in the discrete mapping. The outer loop is the number of fixed point iterations. The inner loop is the number of mapping steps in the transformation. In real time the inner loop is completed first: the transformation is divided into a specific number of steps; then at the end of these steps there are applied a specific number of fixed point iterations.

The two parameters that control these loops are:

`maxit0`	number of fixed point iterations	(`maxit0` ≥ 0)
`maxit`	number of transformation steps	(`maxit` ≥ 0)
`itstp`	maximum total number (`it0+it`) of iterations to be taken	

As an example let `maxit` = 5; then the transformation is divided into 5 equal steps. This means that the transformation vector is applied 5 times to transform (the initial) map 0 into map 1 (0 $\rightarrow$ 1); then 1 $\rightarrow$ 2, 2 $\rightarrow$ 3, 3 $\rightarrow$ 4, and 4 $\rightarrow$ 5. The fraction of the full transformation vector that is applied at each step depends on another flag which is described below. The parameter `maxit` simply tells how many times the inner transformation loop is executed.

If `maxit0` = 3, then after applying 5 (`maxit`) transformations, 3 fixed point iterations are applied. For each of these iterations the full transformation vector is applied to get maps 6, 7, and 8. These three fixed point full step mappings perform the steps 5 $\rightarrow$ 6, 6 $\rightarrow$ 7, and 7 $\rightarrow$ 8.

The total number of iterations taken is the sum (`maxit + maxit0`). We assume in this example that nothing has caused the mapping to stop before completion of both loops. The mapping will stop prior to this limit if convergence is achieved. If `maxit0` = `maxit` = 0, then the final map is the initial input map; no mapping is performed. This is useful for checking the input map data.

The parameter `itstp` can be used to stop the program before the completion of the two loops. Setting `itstp=1` will force a stop after exactly one iteration, in this example one transformation iteration. If instead `itstp` = 7, exactly 7 iterations will be taken: 5 transformations followed by 2 fixed point iterations.

6.3.2 Controlling the Type of Transformation Step

The transformation mapping can be done in two different ways. One method (used in reference 9) is to divide the transformation into `maxit` equal steps. In this case the first mapping applies `1/(maxit)` of the transformation vector. In the next step, it uses `1/(maxit-1)`, and then `1/(maxit-2)` until on the (`maxit`)th step it uses `1/1` to arrive at the final point in the sequence of mappings. The step size `ci` for iteration `it` is `1/(maxit-it+1)`.

The other choice is to again perform the transformation mapping in N steps, but at each step take exactly 1/2 of the current transformation vector to arrive at the next map in the sequence.

These choices are controlled by the parameter

iciset	0	step size = 1/(maxit - it + 1), (it = 1,2,...maxit)
	1	step size = 1/2, (it = 1,2,...maxit)

Note that this choice only applies during the execution of the inner transformation loop. After `maxit` transformations have been taken, `maxit0` fixed point iterations will be taken with step size `ci = 1/1`.

6.3.3 Type of Integral Scaling

Two types of scaling are available when evaluating the line integral contribution from the boundary segments. See Section 7.1, "**Circle Expansion With and Without Scaling**," of this report. The original algorithm, as given in reference 9, we call unscaled; the LBNL version of the algorithm, with the scaling coefficient $2\left(\sqrt{M}-1\right)/(M-1)$ we call scaled or HH. M is the current subregion magnification, equal to (target area/current area).

The parameter that controls this scaling is:

iscale	-0	line integral unscaled
	1	line integral scaled

6.3.4 Map Scaling

Calculation of the line integral along a boundary line segment can produce results that exceed the numerical precision of the computer being used. This happens in calculating a quantity that depends on (a) the length of the current boundary line segment and (b) the distance of the mapped point from the line segment. The problem can be avoided by optionally scaling the map to the unit square (|x|<=1, |y|<=1). The scaling is with respect to the origin (x = 0, y = 0) rather than the map centroid; this is not a problem for maps that are approximately centered about the origin. The scaling is internal only and is not reflected in input or output files.

The parameter that controls this scaling is:

iara	0	no internal map scaling, possibility of overflow failure
	1	map scaled internally so that |x|<=1, |y|<=1.

6.3.5 Transformation Reset

As the transformation is applied there is the potential of creating badly distorted regions: in particular, regions with negative areas or with boundaries that cross upon themselves. This has led us to include a parameter that allows us to optionally back up one complete step and continue again from that point in the mapping sequence. The criterion for applying this reset has not been selected, and the algorithm that is to be applied after resetting the transformation has not been decided upon. Thus, in the file `RLInt.par` the parameter `ireset` should not be changed from its default value of 0. The present version simply excludes the regions that have a negative area when calculating the transformation vector and does not include them in the RMS function error calculation. (The Hsum function never includes regions that have negative (unspecified) original area. However, it does include the regions for which the area goes negative; *i.e.*, it penalizes these areas. See Section 6.4, "**Convergence Criteria.**"

The parameter that controls the reset is:

ireset	0	Negative areas are skipped and do not push.
	1	Negative areas cause a resetting to the top of the iteration loop, acting as a new initial value of the map points. This sets the points back to the values they had at the previous iteration and then starts the iteration outer loop exactly like the initial entry. NOTE: the value

of itstp used to stop the program is the total number of iterations done. If a reset happens, the iteration counter in the program, it0it, is *not* reset and the program will stop when itstp iterations have been completed. Thus, setting itstp to a positive value is a way to ensure that the program will not forever reset and continue to iterate.

6.3.6 Output/Reports

There are a number of print options for this program. Some of them control output to the standard output device, input from standard input, and the creation of files during and at the end of the current run. The output files will be described in the next section. We give a short summary of the I/O parameters and then list the parameters with a short summary of their function. A detailed description of these files is given in Appendix B, "**Description of RLInt I/O Files.**" A complete example is given in Appendix C, "**Example of Complete Run I/O.**" In the following summary, the symbol * refers to files associated with the program flags. It is not the symbol for any text string. When files are deleted, only the files written by the program RLInt are deleted; no other files are deleted.

Summary of icheck, idisk, isum, inow

icheck — Information for tables is stored internally; files containing tables are written at the end of the run. Files: RLInt.out.sum, RLInt.out.plot

idisk — If idisk > 0, one set of files RLInt.*.NNNN is written after every idisk iterations. NNNN is the iteration number. Files: RLInt.out.NNNN, RLInt.sum.NNNN, RLInt.plot.NNNN. If idisk = 0, files RLInt.out and RLInt.plot are written after the last iteration.

isum — If idisk ≠ 0, a summary file is written which contains information for each region. The information in this file is for the same iterations selected by idisk. It is written after the last iteration. File: RLInt.sum

inow — If inow > 0, after every inow iterations two files are updated that contain the current iteration information. Each update overwrites the previous information. Files: RLInt.out.now, RLInt.plot.now

Parameters icheck, idisk, inow are independent; isum is a summary for the files selected with idisk, and is written only when idisk is set to write files.

List of I/O parameters and their function:

iprint — controls the amount of information written to standard output. iprint is basically a simple tool to see what is happening.

iprint = 0 — no output is written.

> 0 — print information from the top and bottom of the fixed point and transformation loops. Helps to track the mapping iterations,

> 1 — print some loop information

> 5 — print a list of skipped regions on stdout.

`itable` = 0 no tables printed (tables are from internally saved results)

= 1 print tables and all input data points; all mapped points after the mapping is done.

= 2 print tables, but no input points, no mapped points

>= 3 print tables and an additional plot table; no input points, no mapped points

`icheck` 0 in `rtrans`, do not save results for printing `itable` tables

1 do save results

This saving of results takes space and time. For small maps it is convenient to save and see the convergence, *etc.* For large maps the space required can be excessive. When `icheck = 0` there are no results to print in the corresponding tables, so the program will not print these tables.

If `itable >= 1` the information written will be on files:

```
RLInt.out.sum
RLInt.out.plot
```

To get these files, the results must be saved with `icheck;` then the files must be printed with `itable>=1`.

NOTE: To save space and time set `icheck = 0` and `idisk=1`. The files written to disk at each iteration can be used to construct tables and plot maps. Alternatively, the `inow` flag can be used to get a dynamic set of files, updated at selected iteration steps.

If this is done, the parameter statements that reserve storage

```
parameter (jjmax = 60)      ! save up to jjmax iterations
parameter (ddmax = 280)     ! save up to ddmax regions
parameter (aamax = 3000)    ! save up to aamax boundary points
parameter (bbmax = 10 )     ! save up to bbmax boundary points
```

can have minimum values when the program is compiled. If `icheck>0` it is necessary that the above statements define values large enough to save the results. When saving data the values of `ddmax, aamax, bbmax` cannot be exceeded. However, the value of `jjmax` can be exceeded; the program will simply start writing over the already stored information. In the example above with `jjmax=60`, if 70 iterations were taken, only iteration 0 and iterations 11-70 would be available for the tables.

`idisk` 0 After completing the call to `rtrans` the following files are written:

`RLInt.out` can be used as an input file `RLInt.dat` for continuing the mapping.

`RLInt.plot` a comma delimited file that can be used to plot mapped points.

NO intermediate disk files *.NNNN are written; no final summary file RLInt.sum (isum = 1) is written.

idisk >0 If iteration NNNN is a multiple of idisk, write files *.NNNN:

RLInt.out.NNNN current state of map, suitable as RLInt.dat for a subsequent run

RLInt.plot.NNNN comma delimited file suitable for plotting maps.

RLInt.sum.NNNN summary file of convergence, areas, etc.

RLInt.sum summary file for complete run (isum = 1)

NNNN is the iteration count. the sum of it+it0. For example,

NNNN = 0000 initial values, same as input data

NNNN = 0001 after 1 complete iteration

There are 3 files:

RLInt.out.NNNN

RLInt.plot.NNNN

RLInt.sum.NNNN

for each iteration. The information in these files can be used to construct maps or tables.

idisk	Write after these iterations
1	0, 1, 2, 3,...
2	0, 2, 4, 6,...
3	0, 3, 6, 9,...
-1	iteration 1 only
-2	iteration 2 only
-3	iteration 3 only

These files are opened, written, and closed within the program. Thus, they are available after each iteration for use by other applications running in parallel with RLInt.

isum 0 no output summary file is written.

1 summary file RLInt.sum is written.

For each region the file RLInt.sum has information for all iterations. The files RLInt.sum.NNNN have information only for iteration NNNN. The isum=1 option is useful if it is desired to see what happens to the regions as the iteration proceeds. If there are a lot of regions, and a lot of iterations for which output is requested, this file can be rather large. The iterations recorded are the same iterations NNNN as for the idisk flag. This file is written only when idisk output is written.

inow 0 no dynamic output files written

>0 write files

RLInt.sum.now

RLInt.out.now

RLInt.plot.now

These files contain the current map information, the same as on the RLInt.*.NNNN files. However, these files are rewritten after every inow iterations, starting with iteration 0. The writing of these files is completely independent of the files written with idisk. The idisk files can be saved for future use. The inow files are updated as

		the run progresses; at the completion of the run they contain information from only the last iteration.
nfdel	<0	NO files are deleted
	NNNN	Delete all output files written from a previous run, including files *.0000 through *.NNNN. This was put in because it can be confusing to know what files the program has created. If the *.NNNN files are meant to be saved, they should be moved somewhere else or renamed. This is an inclusive delete of all files in the range. Only files written by program RLInt are deleted. There is no problem if the files do not exist.
iprmpt	0	no interactive prompting during the run
	≠0	Interactive prompting to check data before doing the transformation

6.4 Convergence Criteria

6.4.1 Relationship Between the RMS and Hsum Functions

The current version of the program RLInt uses a root-mean-square error criterion for determining when the mapping has converged. In the output and elsewhere in this report we refer to this quantity as the RMS function. The RMS function is defined to be

$$RMS = \sqrt{\sum_{k}^{N} \frac{(M_k - 1)^2}{N}}$$

where N is the number of contributing regions and M_k is the subregion magnification which is defined as (target area)/(current area) of the region k. Regions that have negative magnification do not contribute to RMS function, nor do permanently skipped regions. The only regions that contribute are regions that are used to construct the transformation vector.

The mapping coefficient for the line integral that generates the transformation vector contains the quantity $(M_k - 1)$. The size of the transformation vector generated from a particular region depends on this coefficient and also on the area of the region. The current definition of the RMS error function ignores the size of the region.

A more suitable quantity that includes the size of the region is what we refer to as the h function: or Hsum in the program output and the text of this report. This Hsum function was used as a measure of convergence in reference 8. At present it is not used as a convergence criterion in RLInt. However, it is calculated and printed along with the value of the RMS function. For the well-behaved maps that we have used as examples, the behavior of Hsum as a function of the number of iterations is qualitatively the same as that of the RMS function. The value of the RMS function after convergence reflects the numerical precision of the calculation. The Hsum function is approximately equal to the square of the RMS function. Thus, the Hsum function will indicate $\approx 10^{-10}$ when the RMS function gives $\approx 10^{-5}$.

The following definition of Hsum is used in the present version of RLInt. There are three cases that determine the evaluation of the local value h_k for the region D_k that

contributes to HSum. The function $HSum = \sum_k^N h_k$, where this sum is over all contributing regions.

Case 1. The final target area $A_f^k > 0$ (land areas, with population > 0)

$$h_k = \left(\frac{A_f^k}{A_{tot}}\right)\left(\frac{A^k - A_f^k}{A_f^k}\right)^2$$

where A^k is the region current area, and A_{tot} is the total initial land area, the sum of all the land subregion areas. In the input data, we normalize the target areas A_f^k so that $\sum_k^N A_f^k = A_{f,tot} = A_{tot}$ (total target area = total initial area)

Case 2. The final target area $A_f^k = 0$ (land with zero population; *i.e.*, with target area fixed at zero)

$$h_k = \left(\frac{A_f^{\min}}{A_{tot}}\right)\left(\frac{A^k}{A_f^{\min}}\right)^2$$

where $A_f^{\min}$ is the minimum nonzero target area A_f^k, among all regions k. (Specified target areas A_f^k are never negative.)

Case 3. The target area $A_f^k < 0$ (water polygons of unspecified target area)

$$h_k = \left(\frac{A_f^{\min}}{A_{tot}}\right)\left(\frac{\min(0, A^k)}{A_f^{\min}}\right)^2$$

In each case, the error h_k is the square of a relative error weighted by a fractional area. In the first case, the squared relative error is $\left(\frac{A^k - A_f^k}{A_f^k}\right)^2$ and the weight is $\left(\frac{A_f^k}{A_{tot}}\right)$. In the second case, where the target area is zero, the squared relative error is $\left(\frac{A^k}{A_f^{\min}}\right)^2$ and the weight is $\left(\frac{A_f^{\min}}{A_{tot}}\right)$. In the third case, the squared relative error is $\left(\frac{\min(0, A^k)}{A_f^{\min}}\right)^2$ and the weight is $\left(\frac{A_f^{\min}}{A_{tot}}\right)$. There is no contribution to Hsum unless the present area A^k is negative.

To summarize, the Hsum function calculates the squared relative error of the current region area, as compared to the target area, and then weights the result with respect to the fraction of the total area that the region occupies. The basic differences between the Hsum function and the RMS function are: (a) Hsum treats in a

reasonable manner regions that have a target area of zero; (b) Hsum penalizes regions of unspecified target area when their area goes negative; and (c) Hsum weights the region contribution by its area (relative to total area).

We observed above that the square of RMS function is of the same order of magnitude as the Hsum function. This follows from the above definition; there is no square root in the Hsum function. Since neither function is used for minimization, we could probably use absolute values or square roots instead of squares. However, the time spent in the calculation of these quantities is not great and they are probably better understood in their present form. Even in the domain of illegal configurations (negative areas) Hsum and its derivatives are continuous as a function of the boundary point coordinates.

6.4.2 Discussion of Machine Precision

The subroutine `rtrans` that constructs the mapping transformation vector is written in single precision Fortran. When run on a binary 32 bit computer such as a Sun Sparc station, the numerical precision is ≈ 7 significant figures. This is completely adequate for construction of a transformed map. However, this precision should be kept in mind.

As noted in Appendix A.3, "**Line Integral Evaluations,**" there can be a problem with numerical overflow when calculating the line integral. This problem has been fixed by scaling the map to a unit square (|x|<=1, |y|<=1).

This scaling implies that the largest possible distance between any two map points is $2\sqrt{2}$. In the calculation there are times when we must use a numerical value that is not zero but which for all practical purposes is equivalent to zero. Since our coordinates range in magnitude from 0 to 1 and since the calculations are single precision, points closer to each other than $\approx 10^{-5}$, are assumed to concide, and areas less than $\approx 10^{-10}$ are assumed to be zero.

There is no problem in having points at the origin, with zero values. The above distance limit is on the distances *between* points. Target areas can be specified as zero, but to avoid numerical problems, we use the cutoff described above.

The transformation is constructed by integrating around boundaries of individual regions. When transforming a point, it is necessary to determine whether it lies on a boundary segment. This is done by measuring its distance from the boundary points that define the ends of the current line segment. Presently we check only the end points of the segment. When the distance is less than 10^{-5} the points are assumed coincident and the line integral is evaluated as an improper integral.

Two adjacent regions defined in `RLInt.dat` by two separate sets of points actually have coincident points, the ones along the common boundary. If for some reason the above check fails to identify the points as coincident, the regions will come "unglued" and can drift apart or overlap as they are mapped. This should be kept in mind if it appears that the mapping does not correctly map adjacent regions. Also, non-boundary points that coincide with line segment end points should remain coincident with these points. If they do not, then the above check is failing.

The point of this discussion is that there can be problems numerically. For the maps that we have run, the limits chosen appear to be suitable. If the precision of the calculation is other than what we have assumed, then the above limits should be

modified. These limits can be found in parameter statements in the subroutine `rtrans` that constructs and applies the mapping. The statements are:

```
parameter (zeroa = 1.0e-10)     ! single precision area zero
parameter (zero  = 1.0e-5 )     ! 4-5 significant figures
```

There is another reason for keeping the calculation precision in mind. If the maximum coordinate (scaled) has magnitude 1 and the minimum non-zero coordinate is $\approx 10^{-5}$, then the coordinate range of integers is from 1 to 100,000. If the actual coordinate range is 1 to 1,000,000, the scaling will mean that all coordinates that are less than 10 will effectively be zero by the program. In that case a point, that had coordinates (5,5) would be effectively coincident with a point that originally had coordinates (0,0), or with a point that originally had coordinates (10,10). If a relative precision of $\approx 10^{-5}$in coordinate distances is inadequate, it will be necessary to use a computer with an extended exponent range, or to use double precision and change the above numerical definitions of zero.

This precision also enters into the limit set for convergence of the mapping. The basic quantity in the mapping is (M-1) where M is the magnification. In single precision, there is no way that this can get much smaller than $\approx 10^{-5}$. To ask for more precision is not reasonable and will cause the mapping to generate meaningless numerical noise. The quality of the final map will simply deteriorate. One indication of this deterioration is that the current total area begins to disagree with the initial total area. This, of course, applies only to maps for which the areas are scaled to preserve the total map area.

7. Summary of Results

7.1 Circle Expansion With and Without Scaling

The integral mapping algorithm as given in reference 9 can develop problems when applied to regions composed of polygons. Polygon areas do not always converge to the right values, and they can even become negative (polygon completely or partially upside down). "Partially upside down" means that a polygon has become so distorted that its boundary crosses itself.

To better understand the behavior of this mapping, we applied it to a simple circular region of radius R for region magnifications M that covered a wide range of values. A close examination of the resulting images shows that the displacement is too great for points that are on or near the circle boundary.

The original derivation of the integral mapping is based on a mapping that is exactly correct for a circle. For this mapping, the mapped circle is required to have a radius R_f that gives the correct target area, while the exterior points are mapped in a way that preserves local areas. Following a suggestion by Harvard Holmes, we have scaled the integral formula so that it produces this same linear mapping inside the circle; that is, the mapped circle has the specified target area. We refer to this scaling as HH scaling. We then applied the scaled and unscaled integral mapping to a circle for comparison with the expected mapping.

The three mappings (expected, unscaled integral, and scaled integral) have the following form.

(1) $r_f = \left(R_f/R\right)r$ $\quad r \le R$ within circle, expected

$r_f = \sqrt{r^2 + \left(R_f^2 + R^2\right)}$ $\quad r \ge R$ outside circle, expected

(2) $r_f = r + \frac{1}{2\pi}(M-1)\int \frac{\vec{\xi}}{\xi^2} dS$ unscaled integral calculation

(3) $r_f = r + \frac{\left(2\sqrt{M}-1\right)}{(M-1)} \frac{1}{2\pi}(M-1)\int \frac{\vec{\xi}}{\xi^2} dS$ scaled (HH) integral calculation

where $r = \sqrt{x^2 + y^2}$ is the distance from the center of the circle of the point to be mapped and

$$\frac{\vec{\xi}}{\xi^2} = \left(\frac{(x-x')}{(x-x')^2 + (y-y')^2}, \frac{(y-y')}{(x-x')^2 + (y-y')^2} \right); \quad dS = dx'dy'$$

These mappings are illustrated in Figure 2, "Circle of Radius 1 Mappings" for magnifications of M=16 and M=1/8. For each magnification we show the displacement (push) generated by the circle on the points along the x axis and also the final position of the points. In each graph are three cases which are labeled expected, unscaled, and scaled. "Expected" refers to (1) above; namely the displacement that expands or contracts the circle so as to yield the correct target area. "Unscaled" refers to (2) above (the original Russian integral as given in reference 9). "Scaled"

refers to (3) above: namely the modification of (2) that is necessary to obtain agreement with (1) for points inside the circle.

From these graphs, we see that for magnifications less than unity and points inside the circle, the unscaled mapping undershoots and the scaled mapping is correct. For points outside the circle, both formulas approach asymptotically the expected mapping. In other words, the integral mapping does not shrink the circle enough, whereas the scaled mapping shrinks the circle to exactly the right size for a given magnification. The problem is not serious for points that are far from the boundary of the circle.

Even more interesting is the behavior of the mapping when the magnification M is greater than 1. (A magnification of 1 means that there is no expansion or contraction of the circle; the image point is the same as the original point.)

For a magnification M>1, we see that the integral mapping pushes too hard: the displacements for points inside the circle are too large. Again, the scaled mapping is exactly right for points inside the circle. For both mappings, points outside the circle have, displacements that approach asymptotically the exact mapping. An examination of the graphs of the final radius shows that there is a problem with both unscaled and scaled integral mappings.

The problem with these mappings is that they fold the outer region into the inner region. This is evident from the M = 16 graph. The curves show that for points outside the boundary of the circle there is a region that maps to positions interior to the image circle, and vice versa. The mapping has caused the region to fold upon itself.

The authors of reference 9 present an algorithm that divides the mapping into discrete steps. Instead of taking a full step, as was done for the cases shown in these graphs, a fraction of a full step is taken. For example, the mapping can be divided into 10 equal steps. Then 1/10 of the distance is taken for the first step, 1/9 for the second, and so on. For large M, this ameliorates but does not really "cure" the problem. The subdivision that is necessary to avoid folding is simply too small.

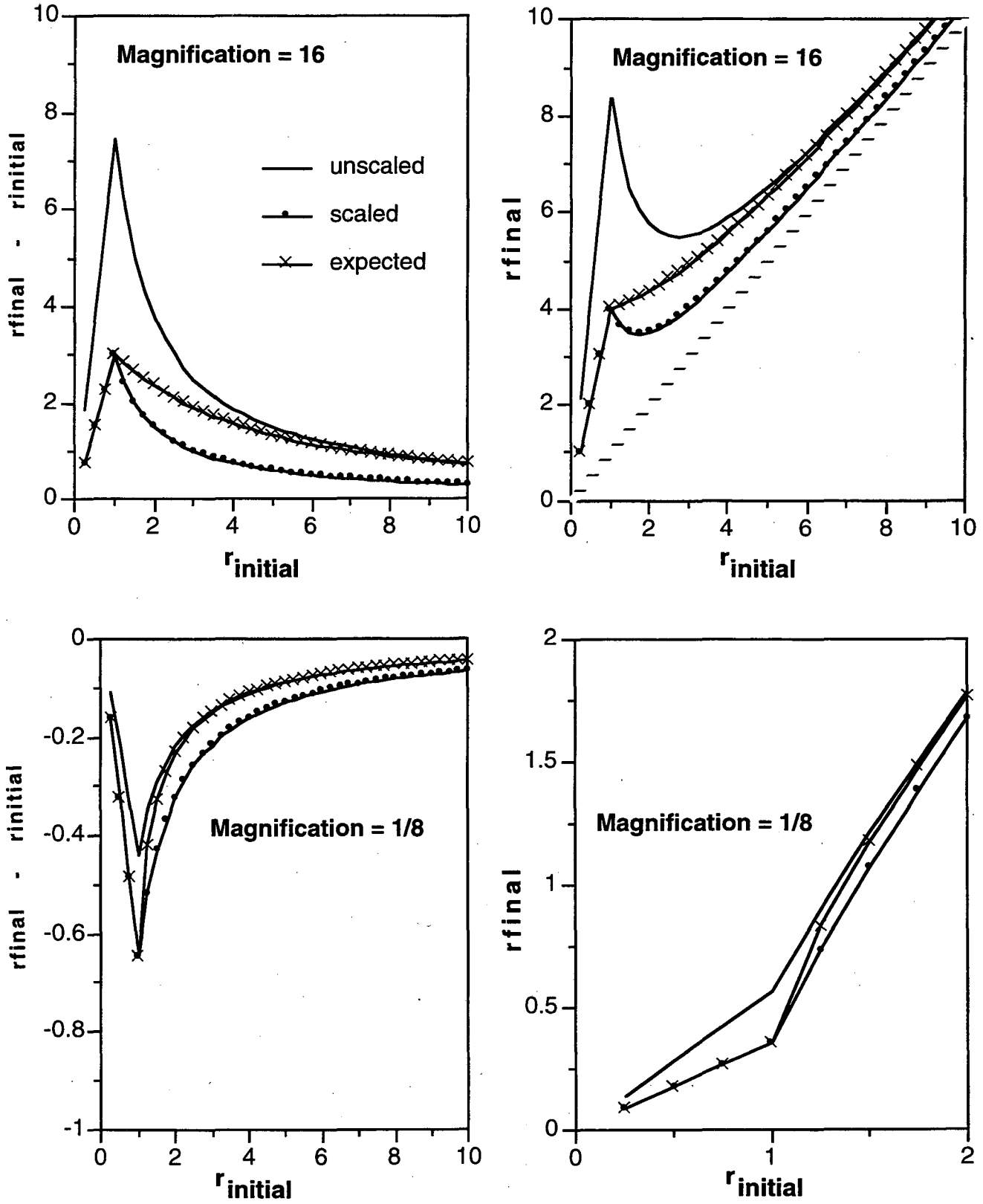
We have found that applying the scaled integral mapping to maps composed of many polygons does occasionally eliminate the extreme boundary distortions that lead to negative area, or boundaries that cross upon themselves.

We note that for points outside the circle a better correction would be an r-dependent correction which gives exactly the expected displacements, for areal magnification M of a circle with radius R. One method would be to use the local polygon area A, a quantity readily available, to calculate an effective polygon radius $R=\sqrt{A/\pi}$. This would produce an r-dependent scale factor of the form

$$\frac{(r_{final}-r_{initial})_{expected}}{(r_{final}-r_{initial})_{unscaled}}=\frac{\sqrt{\left(r^2+R^2(M-1)\right)}-r}{R^2(M-1)/(2r)}.$$

In this case r is the present distance from the center of the equivalent circle to the point being pushed that is outside the local polygon. So far we have not tried a dependent correction factor that depends upon r.

Figure 2. Circle of Radius 1 Mappings



7.2 Vermont Maps

Vermont maps have been used as input to the program RLInt in order to understand the behavior of the mapping algorithm. The Vermont maps have been created at two different levels of detail: with 44 regions and 104 regions.

In each case, the regions are triangular or hexagonal subregions of the 14 counties of Vermont. In the triangular maps, each subregion is a triangle. In the hexagonal maps, each subregion is a hexagon, created by subdividing the sides of the triangle.

We discuss first the 44-region maps. Before density equalization, the (initial) map configuration is identical for the triangle and hexagon maps. The only difference is the addition of more points in the hexagon maps. In each map the entire state is subdivided into 43 smaller regions. These are identified as regions 2-44 in the example of Appendix C.

The first region in both the triangle and hexagon map is the external state boundary. This region, region 1 in Appendix C, is flagged so that it does not contribute to the generation of the transformation vector. The region 1 boundary coincides with the outer boundary of the 43 subregions. The area of the global region (the entire state) is equal to the sum of the areas of the subregions. There are no zero-area regions, no water regions, nothing that should cause any problems with either the triangle or hexagon map.

We wished to compare the convergence behavior of the triangle map to that of the hexagon map. In addition, using the hexagon map, we wanted to study the behavior of several variations of the mapping algorithm. In our tests, the number of transformations and the number of fixed point iterations were varied, while the total number of iterations was fixed at 20.

The object of the mapping is to change the area of each subregion from its present value to a final target value. In the input data file, the target areas were first scaled so that the total target area of the final map was the same as that of the initial map.

The labeling conventions of the graphs is summarized in Table 1.

Table 1. Graph Labels

Label	Scaling	Step size	number of transformation steps	no. of fixed point iterations
HH0.5.1010	HH	1/2	10	10
HH0.5.2000	HH	1/2	20	0
HH1d1.20	HH	1	0	20
HH1d10.10	HH	1/10	10	10
HH1d20.00	HH	1/20	20	0

HH scaling was used throughout, as described in Section 7.1.

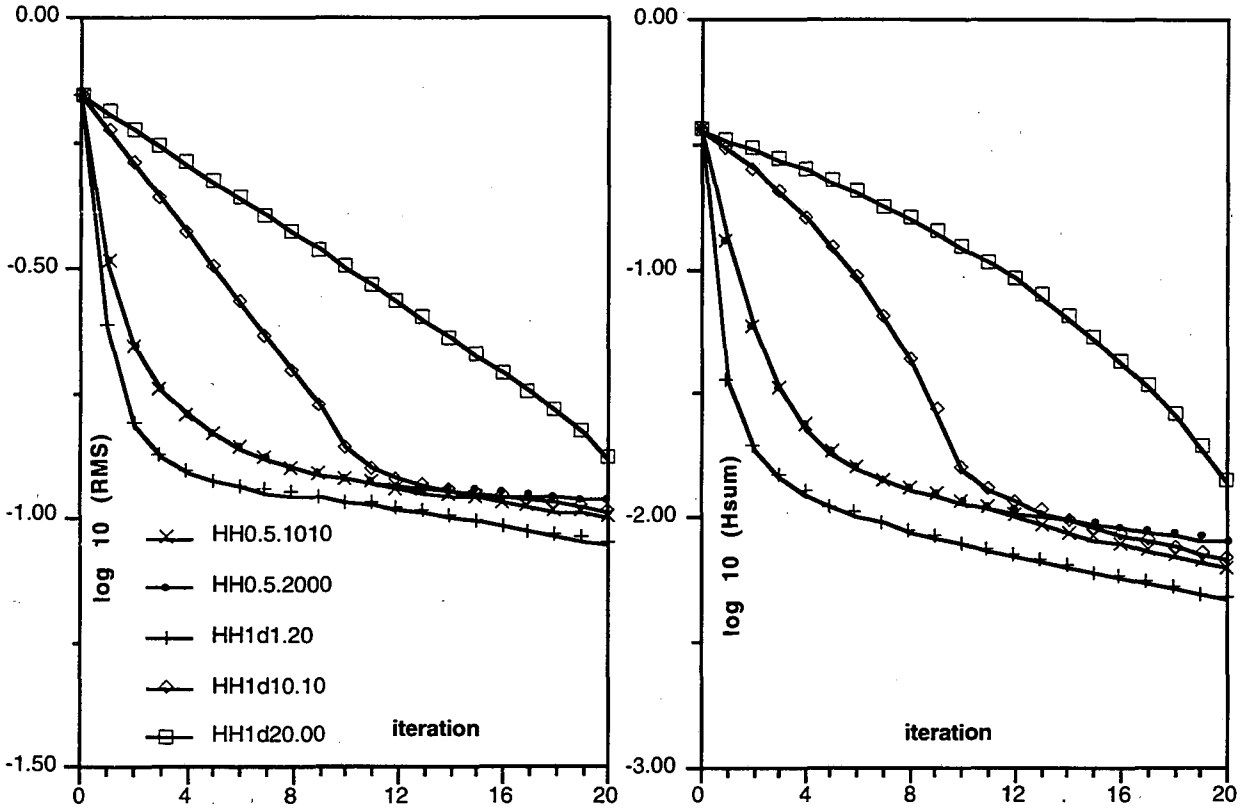
The step size, or transformation subdivision, was 1/2 in the first two runs. In the first run, 10 transformation steps were taken, followed by 10 fixed point iterations. In the second run, 20 transformation steps were taken. The third run had no transformation subdivision; 20 fixed point iterations, or full steps, were made.

In the fourth and fifth runs, the transformation was divided into 10 and 20 equal steps, respectively. In the fourth run, 10 transformations were applied, in which the step size was 1/10, 1/9,...,1. The remaining iterations were fixed point iterations that applied the full transformation vector. In the fifth run, 20 steps were taken, with the step size being 1/20, 1/19,....,1.

7.2.1 The 44-Region Vermont Triangle Map

The 44-region triangle map has a total of 126 subregion boundary points. Each point is shared by at least two triangles. In Figure 3, "Triangle Convergence Plots," we show the values of the RMS function and of the Hsum function, from the initial map at iteration 0 to the final map at iteration 20. The legend in the graph identifies the five runs.

Figure 3. Triangle Convergence Plots



We see from Figure 3 that convergence was worst for the fifth run where the transformation was subdivided into 20 equal steps and no fixed point iterations were taken. The convergence was best for the third run, where the interval was not subdivided, and 20 full steps or fixed point iterations were taken. The other cases fall between these two.

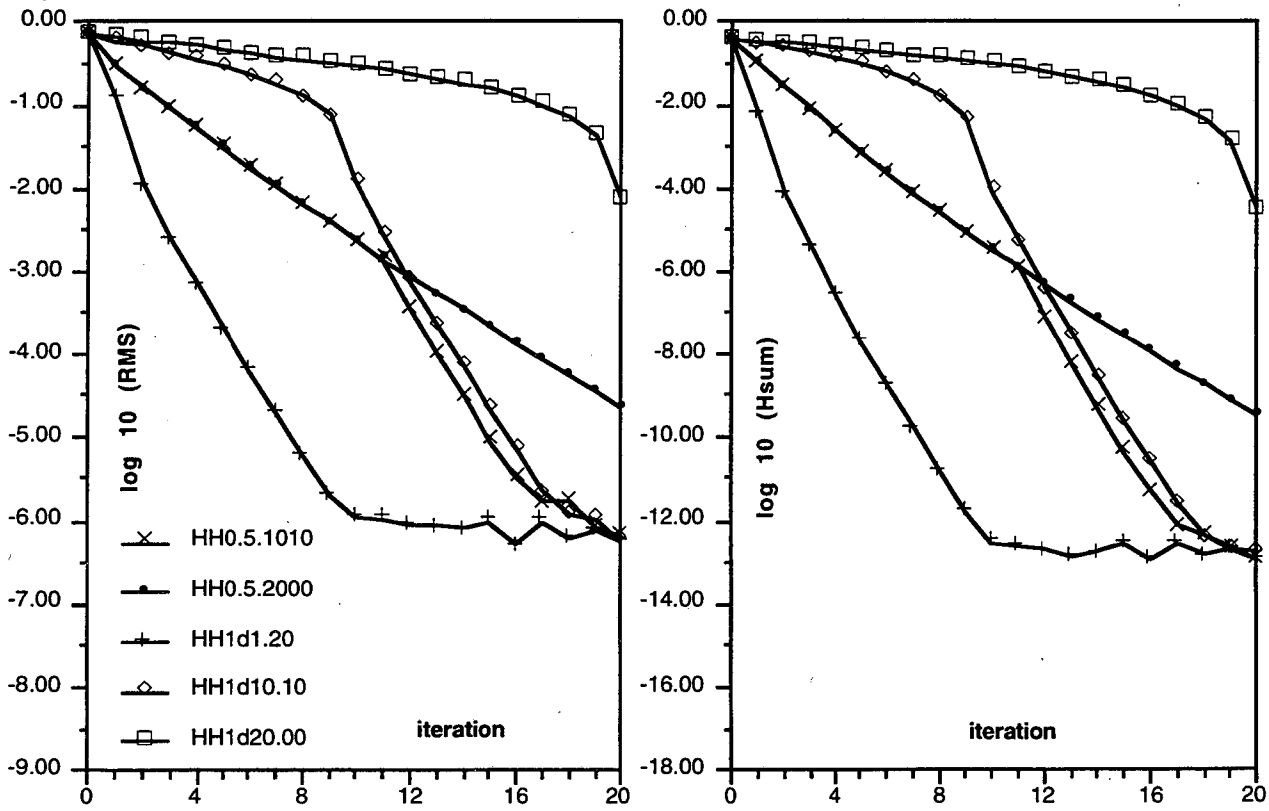
Improvement is slow once the RMS. error reaches about 10^{-1}. Also, for fixed point iterations (the third run), most of the improvement occurred in the first 8 or so iterations. The RMS function and the Hsum function have similar behavior; the minor differences are due to the way they are defined.

7.2.2 The 44-Region Vermont Hexagon Map

In Figure 4, "Hexagon Convergence Plots," we show results for the 44-region hexagon map. The only difference between the triangle runs and the hexagon runs is the initial map. The graphs show the effect of supplying more boundary points. The triangle map is a tightly constrained problem; in the hexagon map, there are more points and therefore more degrees of freedom.

The mapping has achieved full single-precision convergence in 10 iterations. As with the triangle maps, fixed point iterations (run 3) had the best convergence, and equal subdivisions (runs 4 and 5) had the worst. Mapping with step size 1/2 (runs 1 and 2) is intermediate, as for the triangle maps. In every run, convergence improves as soon as fixed point mapping, with step size equal to 1, begins. We note that the last step in the 20-step transformation (the fifth run) is really a full step or fixed point mapping.

Figure 4. Hexagon Convergence Plots



Comparison of Full-step vs. Equal Interval Transformations

The original Russian algorithm (reference 9) gives a heuristic argument for dividing the transformation into a sequence of partial step transformations. They make a choice of equal intervals. However, we have shown that for the Vermont maps convergence is much faster when the full transformation is applied at each step. For the simple and well behaved maps that we have used, there seems to be no reason to not apply the full step mapping; *i.e.*. to use the fixed point algorithm.

Comparison of Equal Interval vs. 1/2 Step Transformations

We have also shown the result of using a constant step size equal to 1/2. This choice produces faster convergence than the equal interval subdivision. Of course, for the maps in question the only real benefit of 1/2 over equal-interval subdivision is that the 1/2 interval is closer to a full 1/1 step than is the equal-interval mapping.

However, in Section 7.3 we shall describe other runs where it is not possible to take the full 1/1 mapping on the first step. Taking that step causes some areas to become negative as a result of illegal crossing of boundaries. Where this problem occurs, it generally occurs during first few iterations. This can be avoided by subdividing the interval. This information has led us to use step size 1/2 for the first 2 or 3 iterations, and then continue with full step mappings.

In summary, it appears that the mapping should be applied with as large a step size as possible that does not cause areas to become negative.

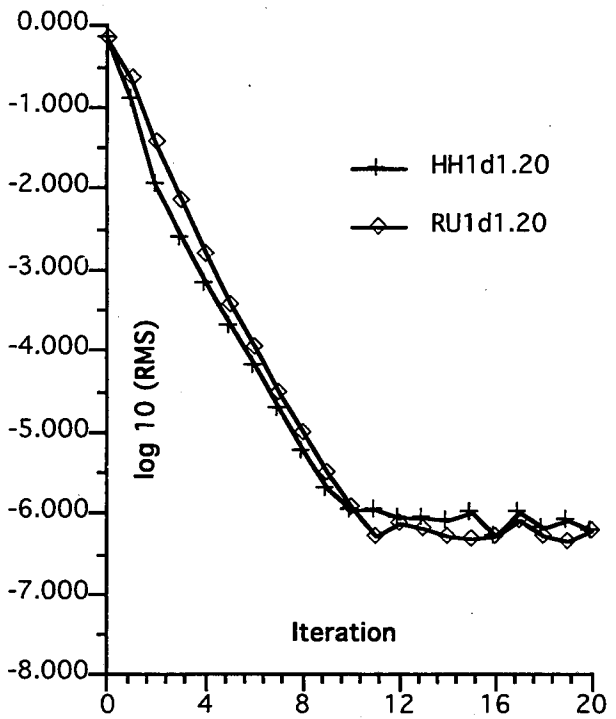
Comparison of Russian Scaling vs. HH Scaling

In Figure 5 below, "Comparison of HH Scaling to Russian Scaling ," we compare the results with and without scaling (HH and RU respectively) for runs with 20 fixed point iterations. No significant differences are noted.

In runs with step size 1/2 (not shown here), the HH scaling has helped prevent areas from becoming negative, and it has improved convergence somewhat. The HH scaling appears to be slightly beneficial in most of the cases we have tested, and we have used it rather consistently.

In larger problems, most points are far from the boundary of any individual region. There is one in the asymptotic region of Figure 2, where the unscaled algorithm is closer to the expected than is the HH-scaled algorithm. Unless one applies an *r*-dependent scaling factor, which we have not done, the HH scaling should probably be avoided. In any event, the choice does not strongly affect the final solution, but only the rate of convergence.

Figure 5. Comparison of HH Scaling to Russian Scaling

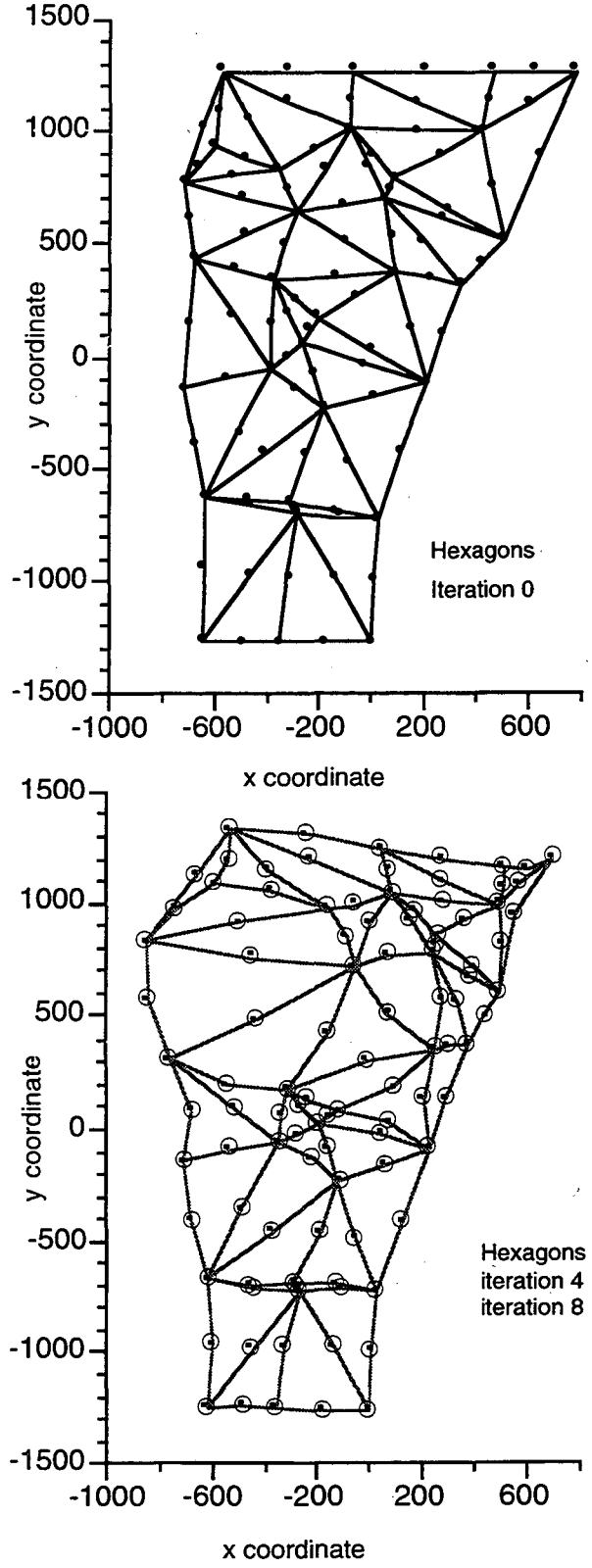


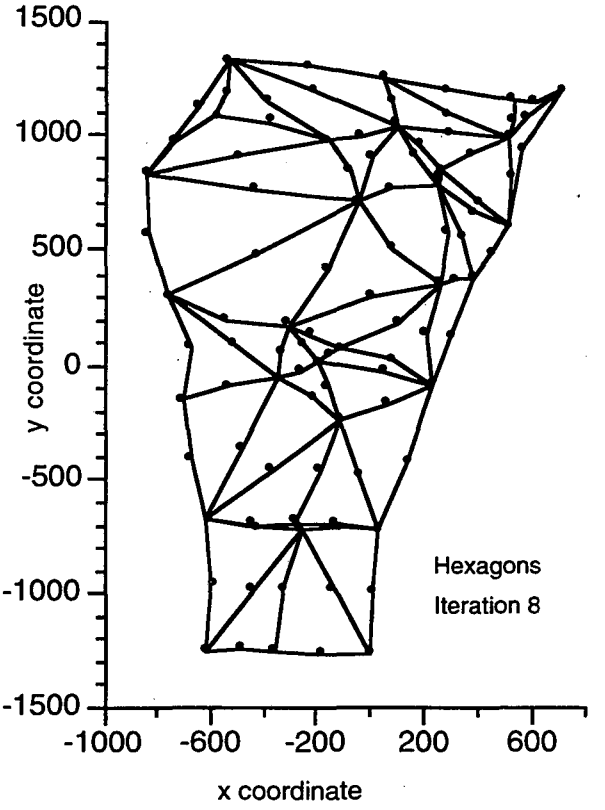
Region Maps

The following pages display the resulting maps for the 44-region Vermont example. The upper left inset of Figure 6, "Mapping 44 Hexagonal Regions," shows the initial boundaries of the 44 regions. There are boundary points at all the triangle vertices and at the midpoints of the triangle sides. Superposed on the map is the region 1 boundary, the boundary of the entire state.

The upper right inset shows the map after application of 8 fixed point iterations, with HH scaling. This map has fully converged, within the limits of single precision computation. In the lower inset, we show the final map after eight iterations. Superimposed on that map is the result after four iterations. For all practical purposes, the two maps are identical. The fact that the dots and circles do not exactly coincide is due to the resolution of the printed graph. Expanded views show agreement of the coordinate values to approximately three significant figures.

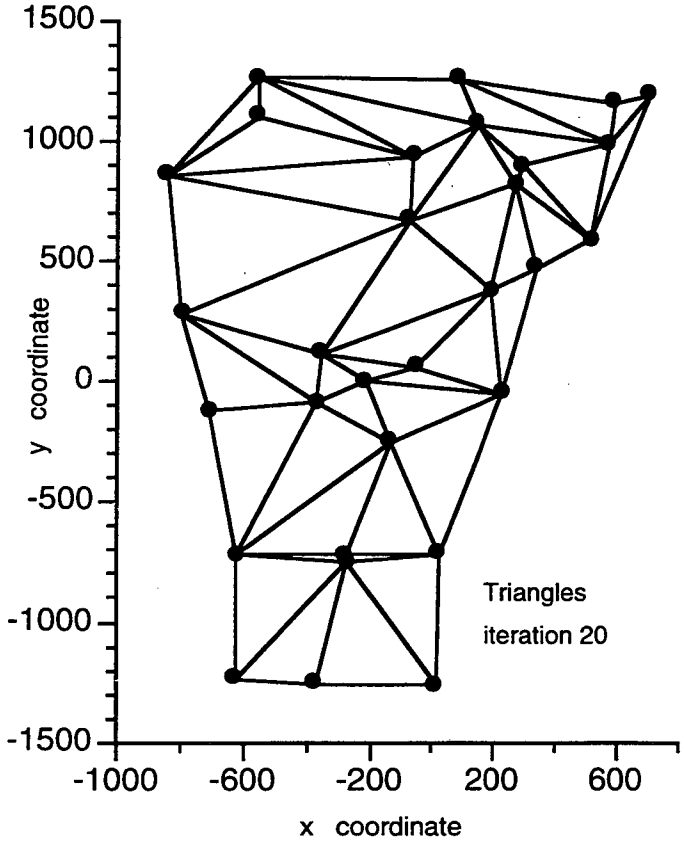
Figure 6. Mapping 44 Hexagonal Regions





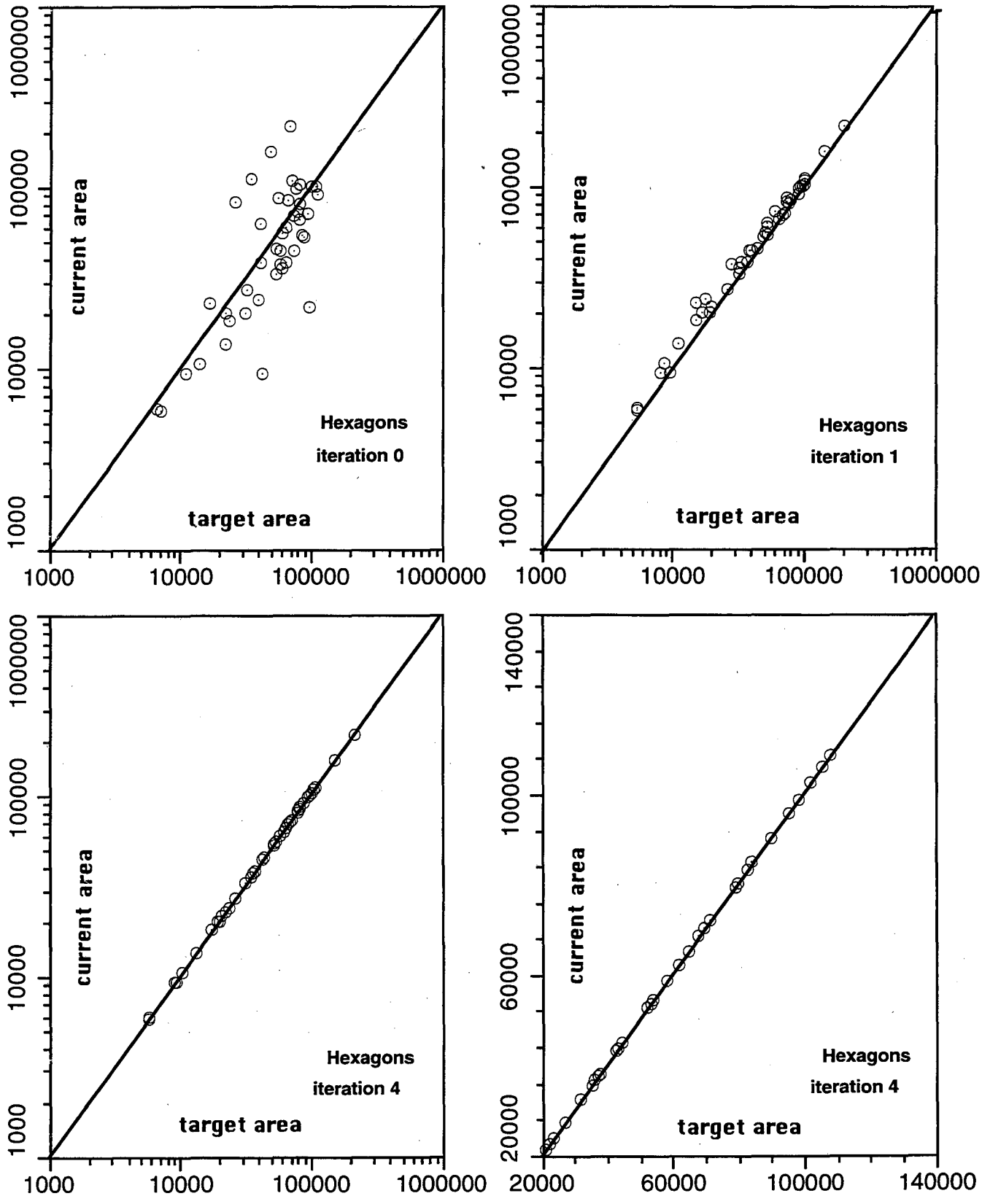
In Figure 7, "Mapping 44 Triangular Regions,," we show the same 44-region Vermont map after 20 fixed point iterations. The only difference between the hexagon maps of Figure 6 and the triangle map of Figure 7 is the initial map; the points in Figure 7 do not include the midpoints of the triangle sides. As we saw in Figure 3, the triangle map does not fully converge; final values of RMS are around 0.1. In Figure 7 the region boundaries are, by necessity, straight lines. The hexagon map of Figure 6 has more degrees of freedom and converges to an RMS value that is zero within single precision machine accuracy.

Figure 7. Mapping 44 Triangular Regions



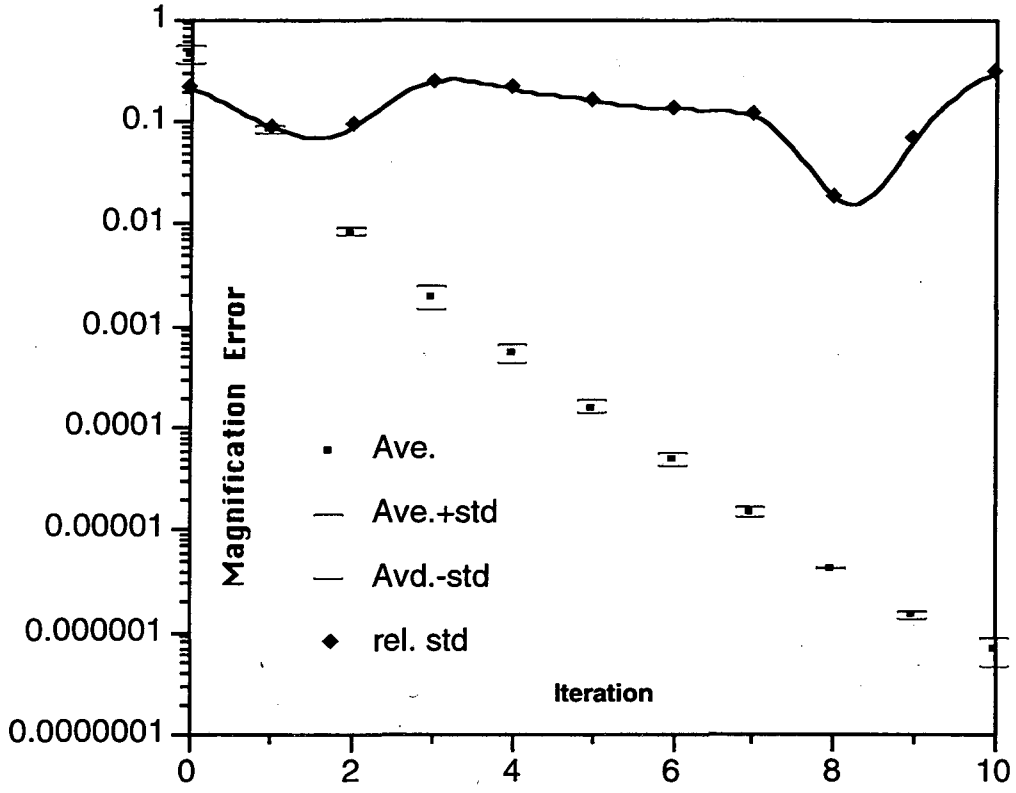
In the next graph, Figure 8,"Scatter Plots of Current Area and Final Area," we plot as a scatter plot the target area along the x axis and the current area along the y axis. The top two graphs are for the initial map, iteration 0, and the map after exactly one fixed point mapping. The lower left graph shows the results after four fixed point mappings. We show in the lower right graph and expanded range of the map at four iterations. The fourth iteration is quite linear in the plot, the points do lie on the line but the printing resolution shows them slightly off of the line. The RMS function at four iterations is $\approx 10^{-3}$, which gives about 3 significant figures for the region subareas.

Figure 8. Scatter Plots of Current Area and Target Area



The final graphs of this section, Figure 9, "Convergence of the Hexagon Mapping," show that the map was fully converged at the 8th iteration. We have plotted here the average, over all 44 regions, of the absolute value of the magnification error for each step in the iteration. The standard deviation of that value is also plotted. In the top graph, we plot the standard deviation, as scaled by the size of the average. We see that there is a distinct dip at 8 iterations. Further iterations cause a spread in this value. The iterations past 8, on this particular map, generate noise. This can also be seen in the above plots of the RMS function and the Hsum function. Further convergence also tends to destroy the accuracy of the total area, as obtained by adding up all the subregions. All of this indicates a deterioration of the quality of the map. To summarize: it is detrimental to the quality of the map to attempt to converge past the inherent accuracy of the algorithm.

Figure 9. Convergence of the Hexagon Mapping



7.3 Runs on 44- and 104-Region Vermont Hexagon Maps

We present in this section results from RLInt runs on Vermont hexagon population maps. There are two basic maps; a 44-region map and a 104-region map. These two maps are essentially the same, except for the different number of regions. They are referred to as hexagon maps because each of the subregion triangles has 6 boundary points, 3 vertices and each side has a midpoint.

For each map there are seven population categories. These are referred to as:

1. pop80 = total population: white + black + asianpi + indian + other
2. white = white population
3. other = pop80 - white - black - asianpi - indian
4. spanish = hispanic

(includes some members of all 5 races: white, black, asianpi, indian, other)

5. asianpi = asian and pacific islanders
6. indian = american indian, eskimo and aleut
7. black = black population

The initial map in each of these seven categories is the same basic 104-region, or 44-region, map. The target areas are adjusted to reflect the map category. The maps, as listed above, are in the order of increasing non-uniformity of the initial population density, *i.e.*, deviations of initial area from the target area. This is reflected in the value of the Hsum function and can be seen in the scatter plots shown below. In he Table 2 below the initial values of the RMS and Hsum functions for these seven cases are shown.

Table 2. Vermont Hexagon Map RMS and Hsum Function Values.

Category	$\log_{10}$(RMS) 44 Regions	$\log_{10}$(Hsum) 44 Regions	$\log_{10}$(RMS) 104 Regions	$\log_{10}$(Hsum) 104 Regions
pop80	-0.153	-0.446	-0.191	-0.324
white	-0.156	-0.450	-0.194	-0.327
other	0.053	-0.040	0.009	-0.034
spanish	0.061	0.064	0.049	0.140
asianpi	0.228	0.121	0.161	0.196
indian	0.161	0.164	0.086	0.201
black	0.164	0.207	0.104	0.286

The program RLInt has been run on each of these categories for the 44-region and the 104-region maps. The results of these runs are presented in the form of graphs and pictures in the sections that follow. The quantities of interest are the rate of convergence, and the final maps. We also are interested in the behavior of the mapping as the final map becomes further from the initial map. We shall look first at the convergence of the mappings.

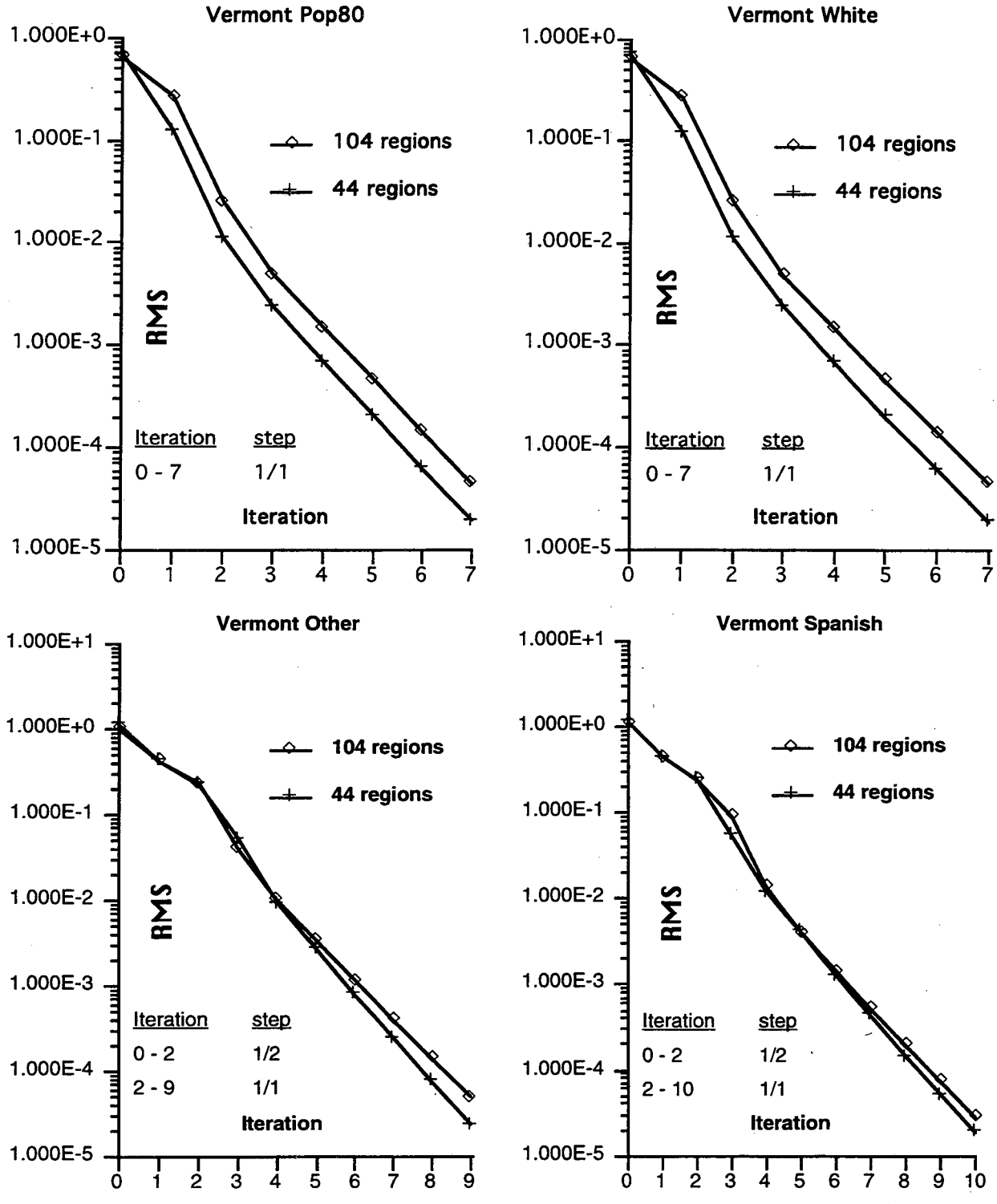
7.3.1 RMS Convergence

In the graphs below, the values of the RMS function are plotted on a logarithmic scale as the map is transformed. For each of the seven population categories there is a graph showing the convergence of the 44-region and the 104-region mappings. In each case the run is interupted when the value of the RMS function is less than the requested converged value of 5×10^{-5}. The maps that meet this error criterion are considered to be fully converged.

The pop80 and White mappings have been run with all iteration steps taken as fixed point iterations; the full step is taken at each transformation. The step size in these graphs is noted as 1/1. This was done because the fastest convergence is obtained when the mapping step is as large as is practical.

The mappings Other and Spanish have been run with the first 2 iterations set to the binary option, transformation step size of 1/2, and the rest of the mappings as fixed point, full mapping step 1/1, mappings. For these maps, the first two steps were changed to the binary 1/2 option because the fixed point mappings caused regions to be lost due to negative areas. In the runs shown no regions were lost; the mappings proceeded with no noticeable anomalies or difficulties. Figure 10, " RMS Function vs. Iteration, Pop80, White, Other, & Spanish," shows the value of the RMS function for these mappings.

Figure 10. RMS Function vs. Iteration: Pop80, White, Other, & Spanish



The Asianpi, Indian, and Black mappings were done with the transformation interval fixed at 1/2 for all the iterations The reason for this is that the previous (1/2, 1/2, 1,1) subdivision produced lost regions due to negative triangles. Running the program RLInt with the binary 1/2 mapping option for the full set of mappings produced good convergence and no regions were lost. The RMS function values for these categories are shown in Figure 11,"RMS Function vs. Iteration, Asianpi, Indian, & Black."

Figure 11. RMS Function vs. Iteration: Asianpi, Indian, & Black

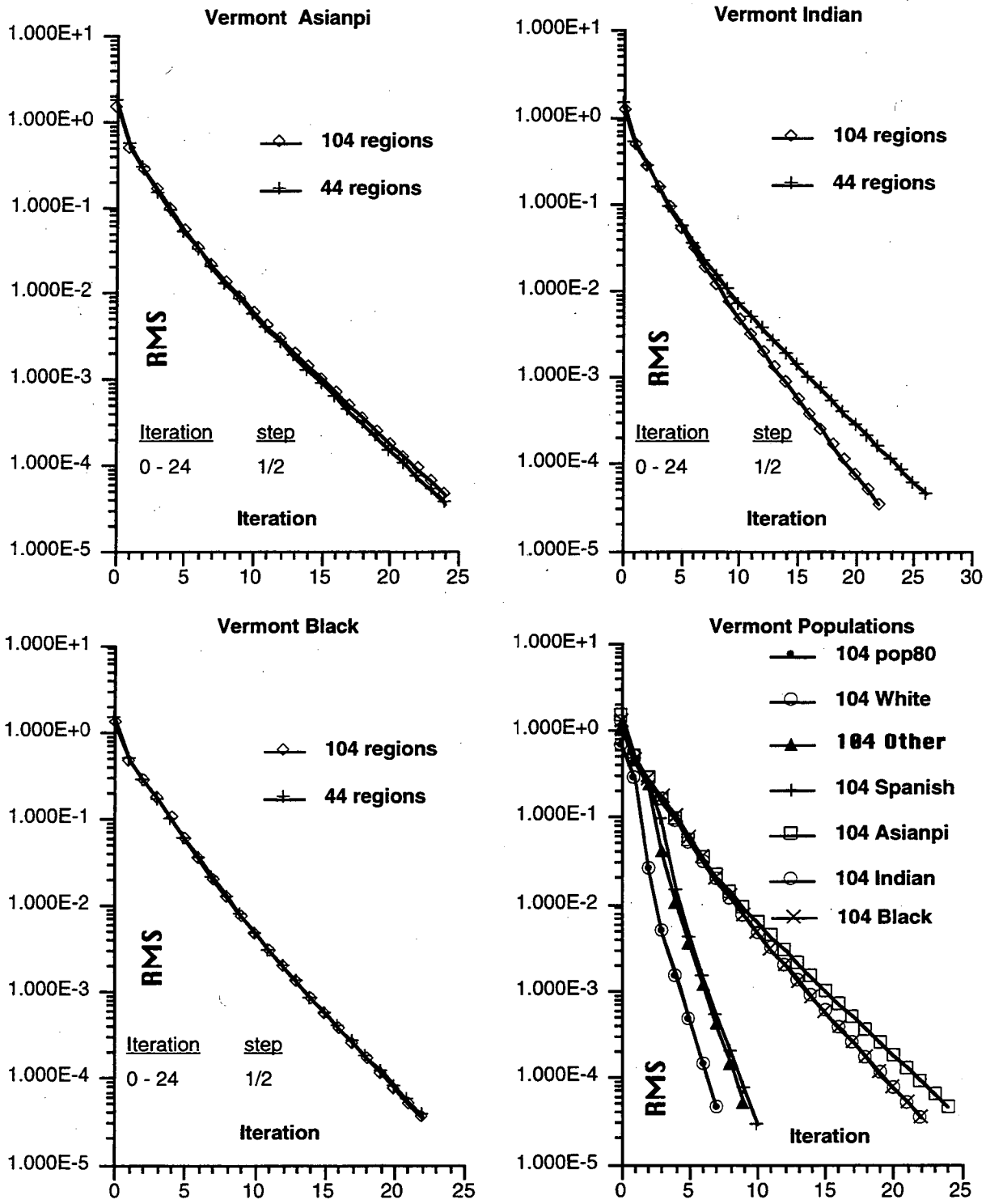
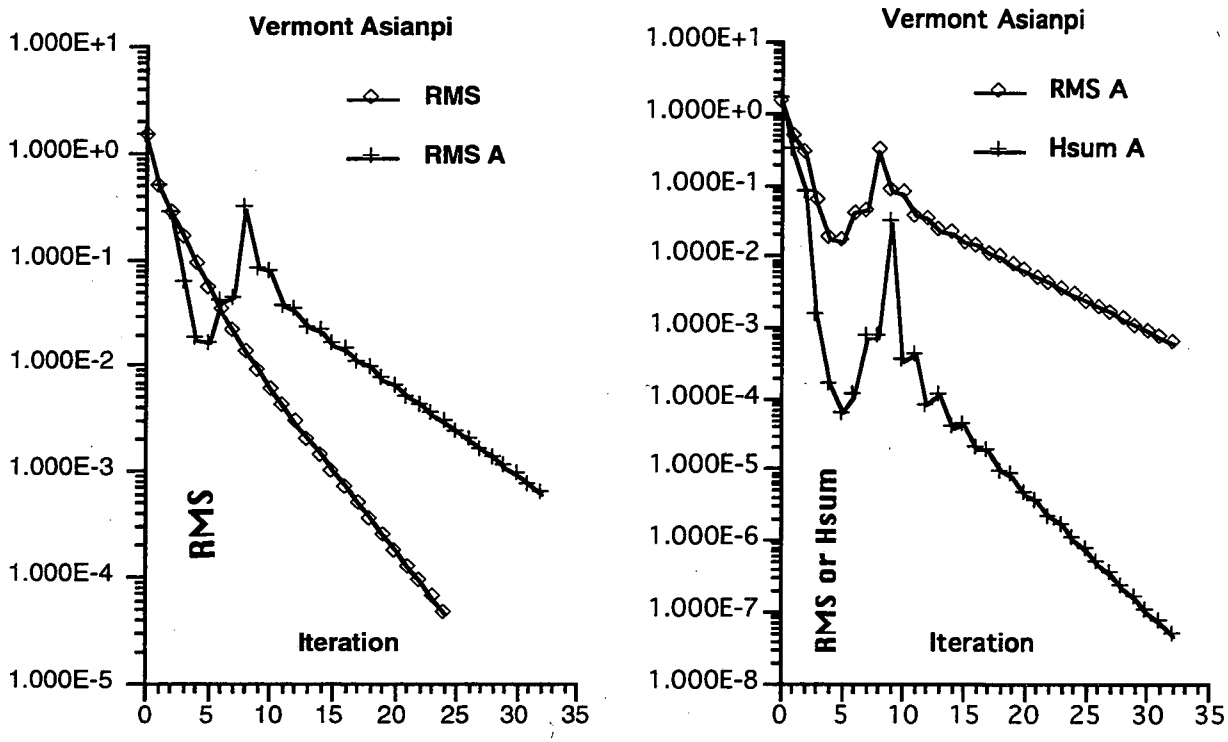


Figure 11 also shows a graph that summarizes all seven categories for the 104-region map. It is interesting to observe from this graph that the convergence rate increases significantly when fixed point iterations are taken. If the first three categories had been run using the binary 1/2 option for all the iterations, their convergence curves would be much closer to the last three categories. No attempt was made in

running the Asianpi, Indian, and Black maps to determine if the mapping could be switched to fixed point after a suitable number of binary steps, for example 10 steps.

In Figure 12, "Convergence of the Asianpi 104 Region Map," we show why we changed the mapping to the binary 1/2 option, and we also compare the RMS and Hsum functions. The graph labeled RMS is the above Asianpi run in which all the mapping steps use the binary 1/2 option. The graph RMS A is the same initial map; however, only the first two steps used the binary 1/2 option, the rest used the full fixed point iteration. The RMS A graph shows that during iterations 5-10 there were some problems. The resultant map has missing regions. The other graph in Figure 12 shows that the RMS and the Hsum functions both indicate that there is a problem in this region.

Figure 12. Convergence of the Asianpi 104-Region Map



The question arises as to what the behavior of the original Russian mapping would be on the Asianpi map. We have run this case and used a subdivision of ten intervals (1/10, 1/9, ..., 1/2, 1/1, then 1/1 ten times) before applying the fixed point mapping. These convergence results are shown in Figure 13,"Asianpi Convergence vs. Iteration Count."

In this graph, we have shown the convergence for different choices of the mapping parameters. For the first map, labeled RU 10@1/10, the original Russian algorithm, as implemented in program RLInt, was applied to the initial map. The subdivision interval was 1/10 which gives 10 equal divisions for the transformation. After that, 21 fixed point mappings are applied. This mapping appears to have converged to $\approx 4 \times 10^{-3}$ after about 13 iterations. The remaining iterations do no significantly improve the error and in fact lead to a deterioration of the quality of the map. This deterioration shows up in the total area; the calculated area begins to disgree with the original total area. The target areas are scaled so that the total area is always the same as the starting total area.

The next mapping, labeled HH 10@1/10, is the same except that the HH scaling was used. This finally began to converge at about iteration 15, and it appears that the

final error criteria of 5.0 10^{-5} could have eventually been met. All these mappings have been stopped at exactly 32 iterations. We note that no regions were lost due to negative areas.

The map labeled HH 10@1/2 has 10 binary transformations, instead of 10 equal interval transformations, and has HH scaling. It converges through iteration 15 and then begins to diverge until iteration 25 at which time it again starts to converge. Its final value is the worst of the maps. However, iteration 15 would have been the best converged map. No regions were lost during this run.

We have included the runs HH 2@1/2 and HH all @ 1/2 for comparison. The HH 2@1/2 run was discussed above as the RMS A run. This run lost regions. It had the first two iterations with the binary 1/2 mapping option and the rest using the fixed point mapping. The HH all @ 1/2 is the mapping that we discussed above, all the iterations were with the binary 1/2 option. The convergence is smooth and steady.

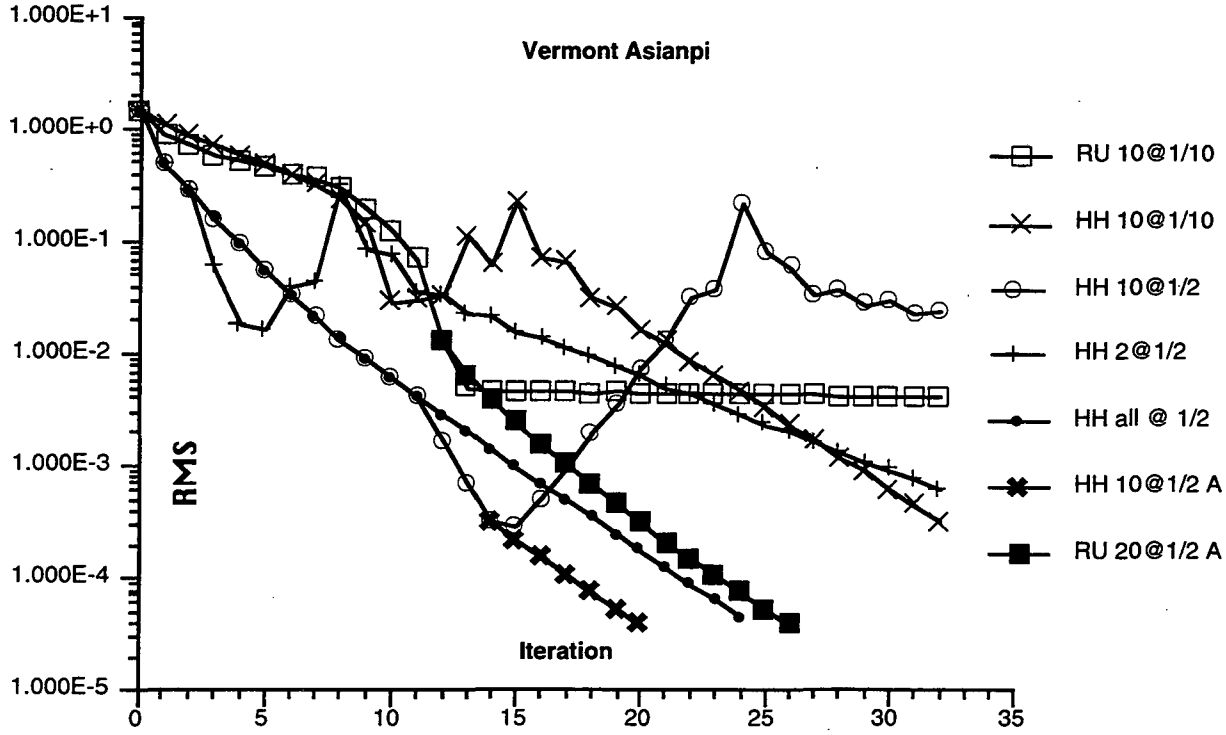
The RLInt program has the ability to restart from any previous mapping step, should that be desired. For example it is interesting to restart the HH 10@1/2 mapping at step 15 and use the binary 1/2 option. The output for this run had been requested every two iterations, so the closest output file available was `RLInt.out.0014`. This file was copied to the input file `RLInt.dat` and the mapping continued for exactly 10 more binary 1/2 mappings. These results are labeled as HH 10@1/2 A. This continued run quickly converged to less than the requested error.

We have also restarted the RU 10@1/10 mapping at iteration 12. The binary 1/2 mapping option was set and the iteration continued with only this type of transformation. This mapping is labeled RU 20@1/2 A. We see that this mapping also continues to converge until it meets the convergence criterion.

We note that for this continued mapping, the slope of the binary 1/2 mappings is the about same as that of the original mapping that took all the steps as binary 1/2 mappings. Also, the slope of the full step mapping, when converging well, is about the same for the maps with different starting options after they shift to the full step, fixed point iteration.

The Asianpi map has been used to illustrate how the RLInt parameters can generate different results. At the present time we have not developed a convergence strategy that would automatically determine what to do when regions are lost, or when the mapping began to diverge after a period of convergence. It would appear that the binary 1/2 option is a good choice when it is not possible to apply the full fixed point iteration.

Figure 13. Asianpi Convergence vs. Iteration Count



7.3.2 Area Scatter Plot Convergence

In the previous section we presented and discussed convergence results using the RMS function as an indicator as to how the mapping converged as the successive transformed maps were generated. In the present section we show scatter plots of the region areas. Each region has a current area and a target area. The initial map, referred to here as iteration 0, has the initial region areas. The final map has the current areas, which are the converged region areas. In principal a fully converged map has, to machine precision, each current region area equal to the corresponding target area. The target areas have been scaled so that the total area of the map is constant; the initial map and the final converged map have the same total area, as do approximately all the intermediate maps. Each of the seven population categories has the same starting map; only the target areas are different.

For each of the seven population categories of the 104-region map we show, in Figures 14-20, four area scatter plots: the initial map, two intermediate maps, and the fully converged map. We note that an intermediate map that has converged to $\approx 10^{-3}$ has an area scatter plot that is all but indistinguishable from the fully converged map. All the region areas for such a map are correct to about 3 significant figures of accuracy. Their area scatter plots and their region maps appear visually identical to those of the fully converged map.

Figure 14. Area Scatter Plots for Vermont Pop80

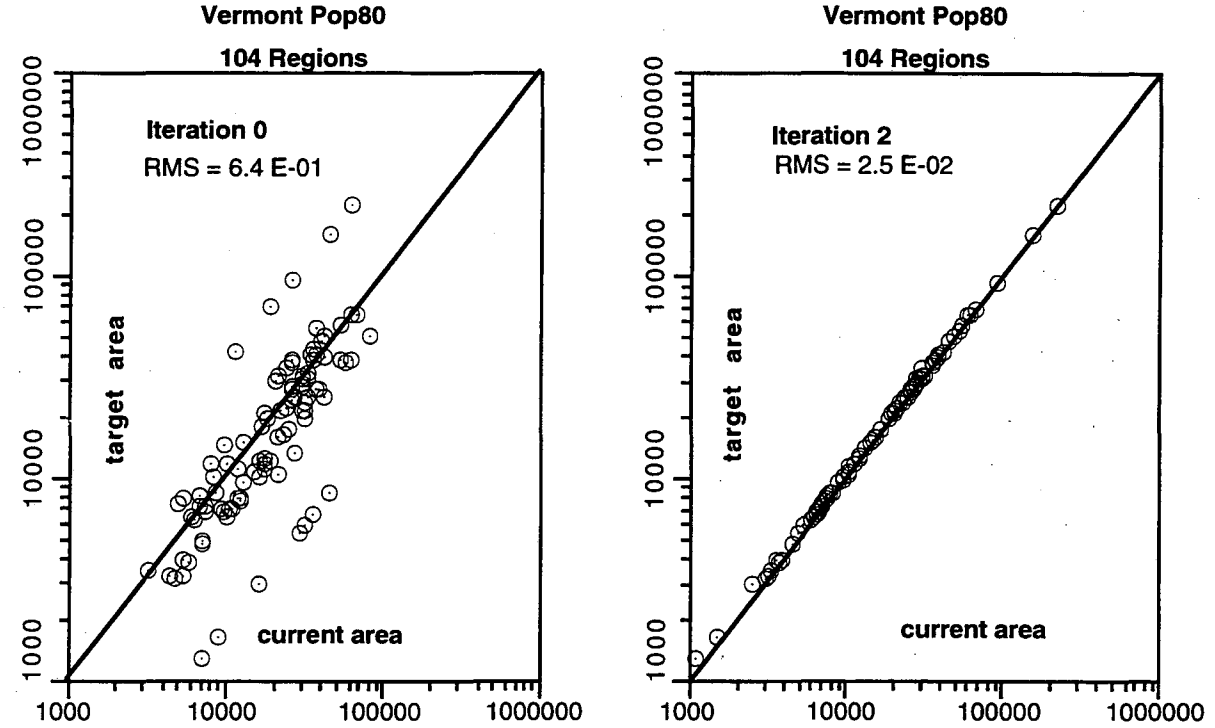


Figure 14. Area Scatter Plots for Vermont Pop80

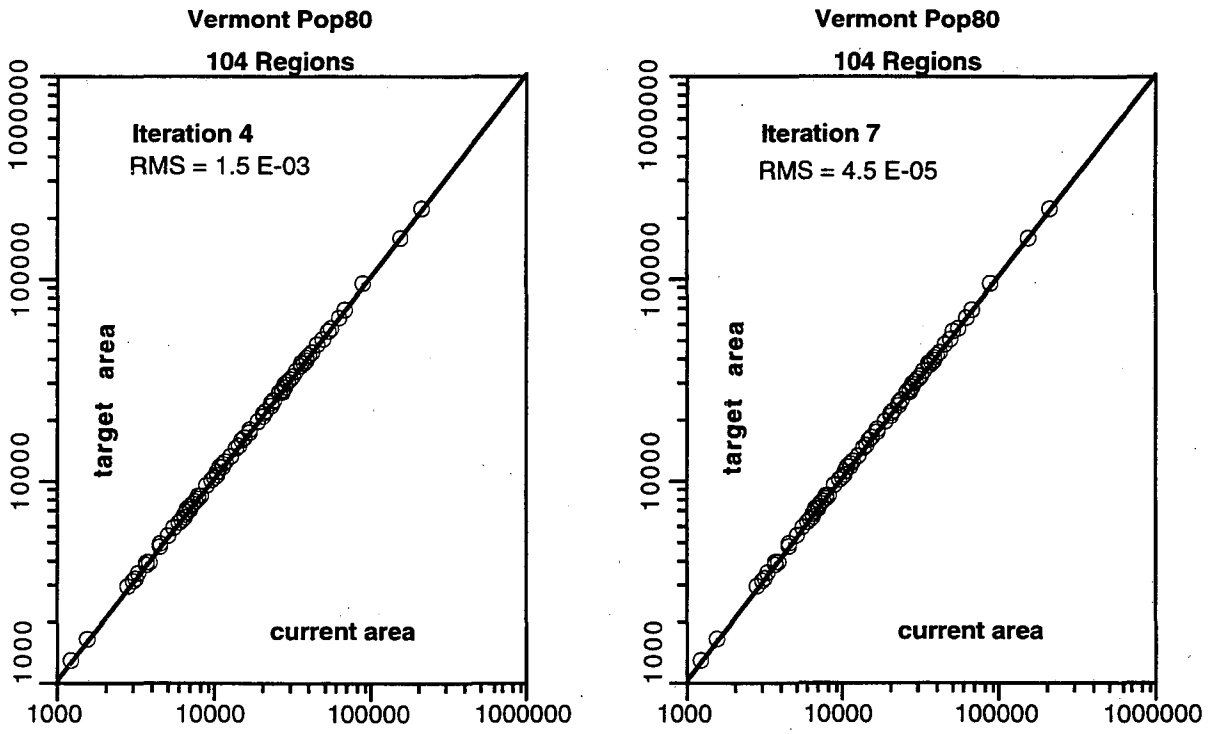


Figure 15. Area Scatter Plots for Vermont White

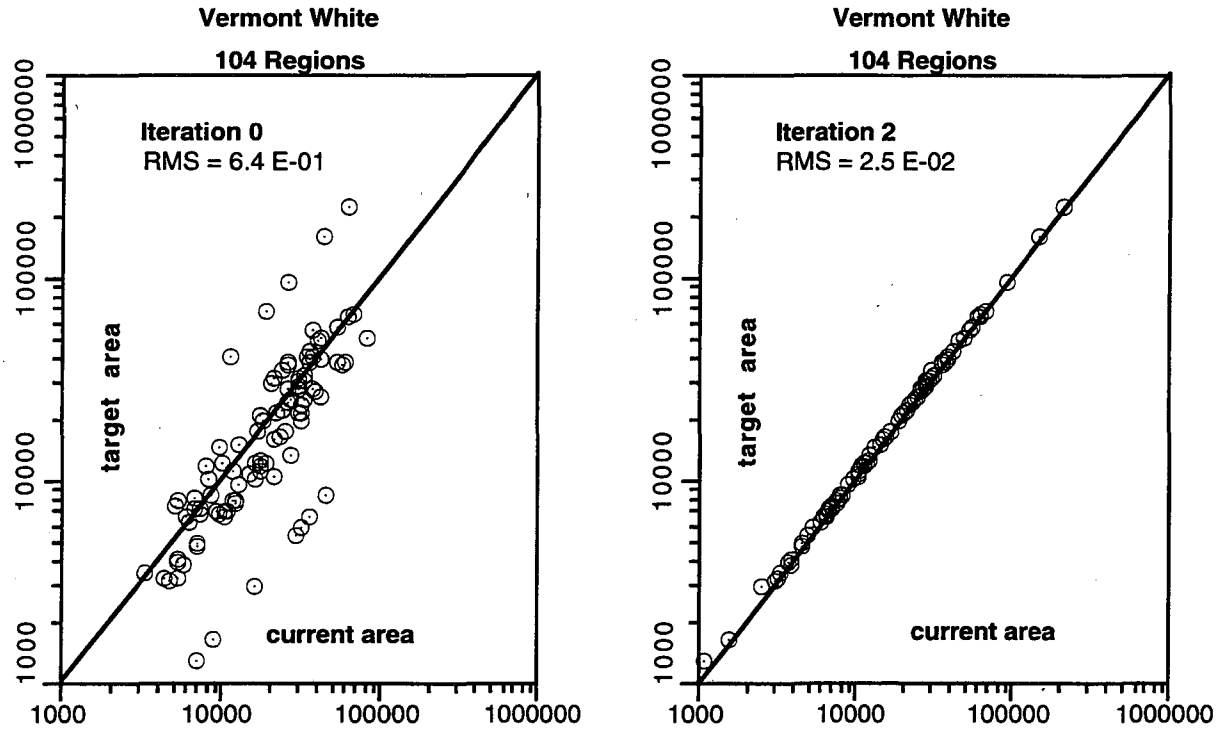


Figure15. Area Scatter Plots for Vermont White

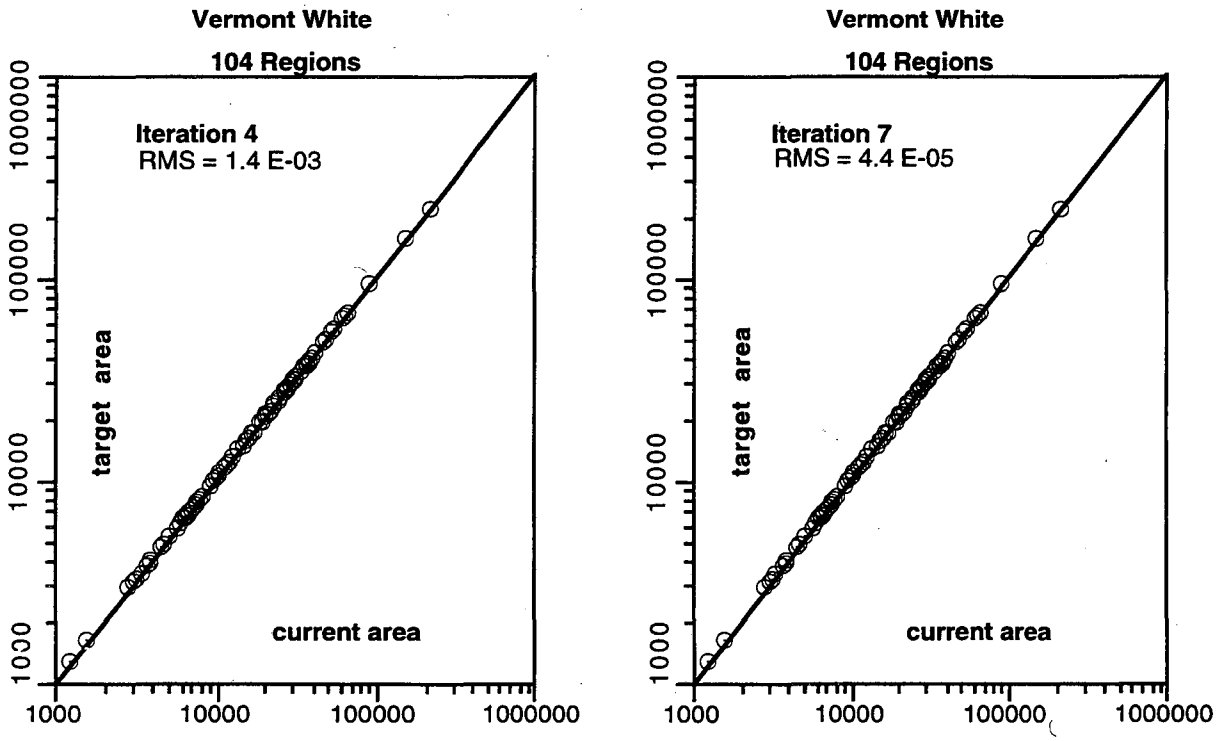


Figure 16. Area Scatter Plots for Vermont Other

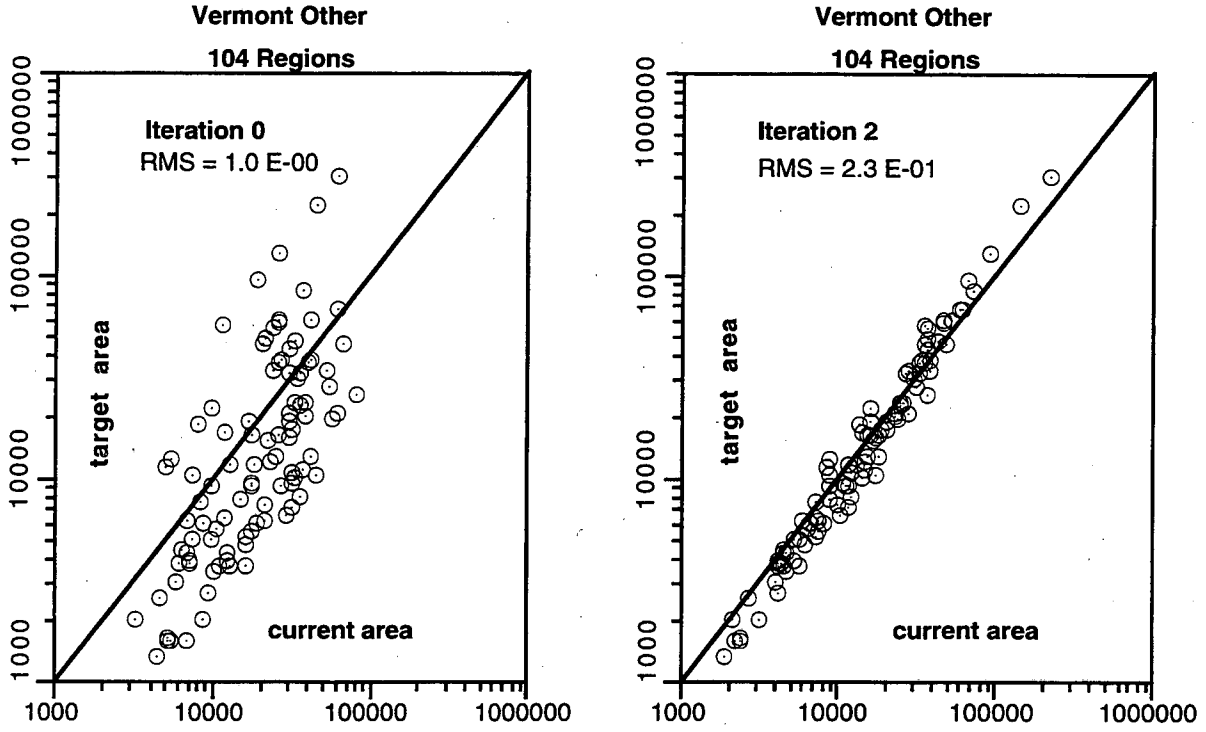


Figure 16. Area Scatter Plots for Vermont Other

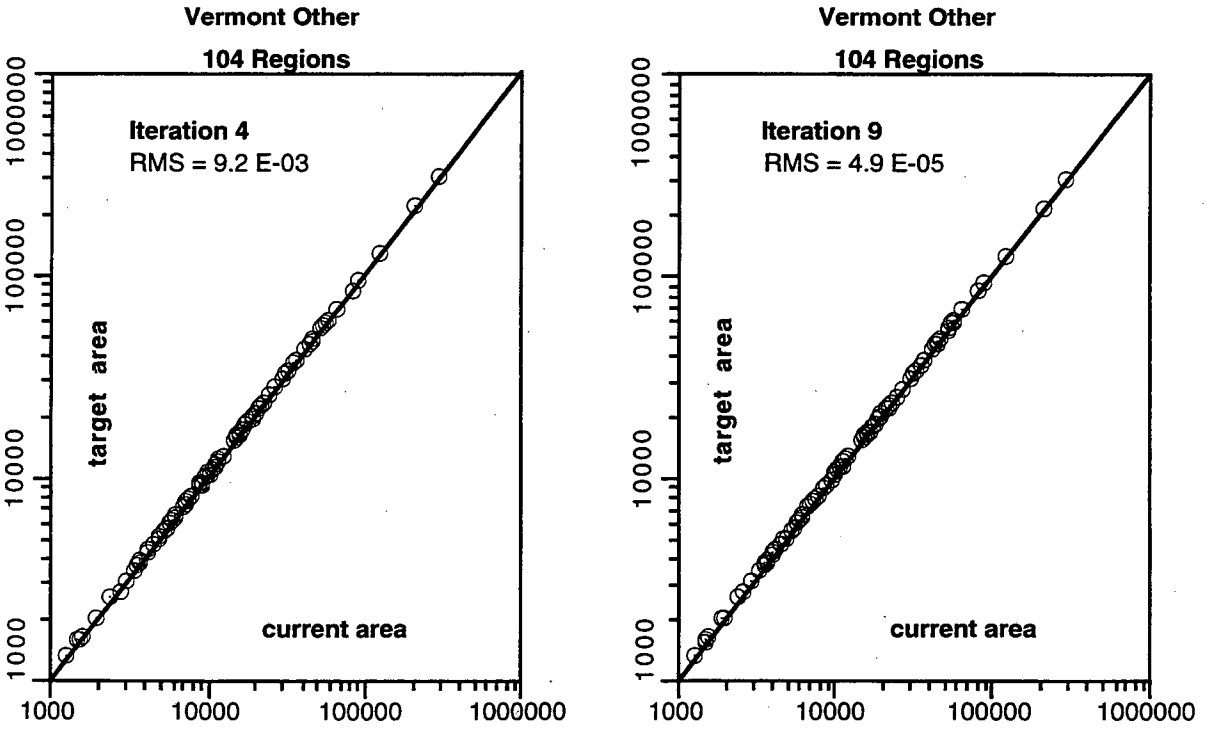


Figure 17. Area Scatter Plots for Vermont Spanish

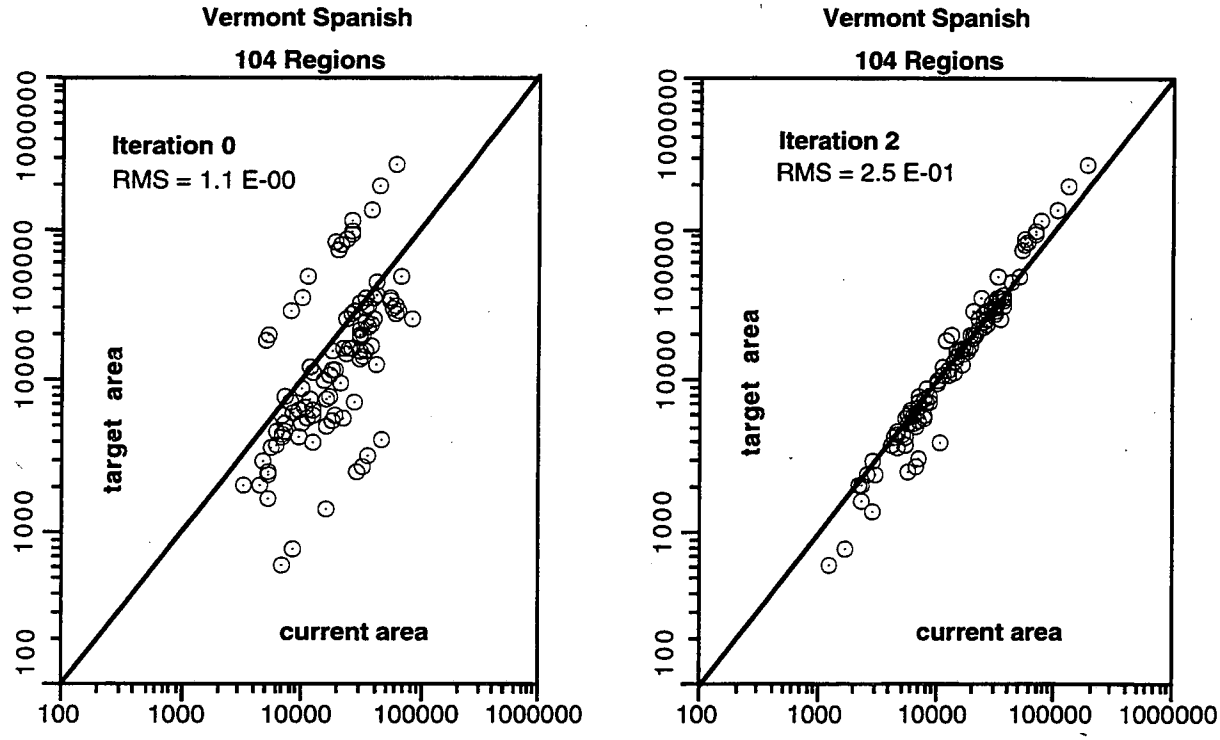


Figure 17. Area Scatter Plots for Vermont Spanish

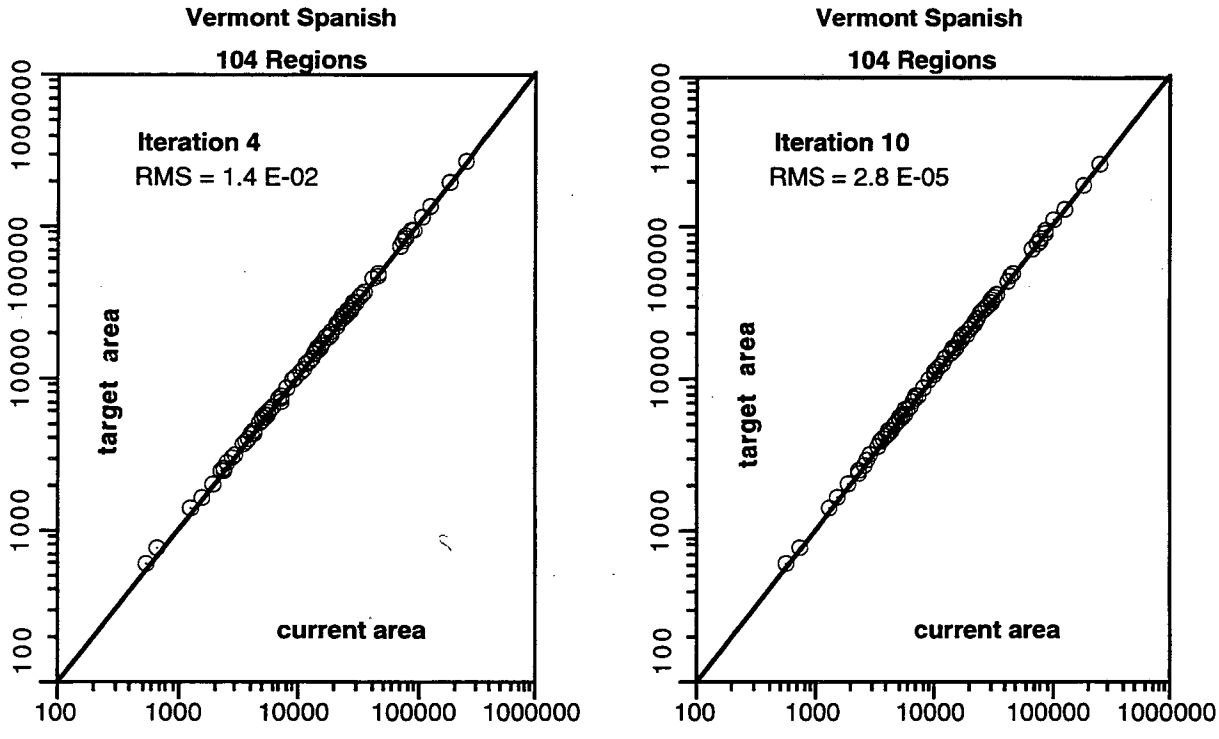


Figure 18. Area Scatter Plots for Vermont Asianpi

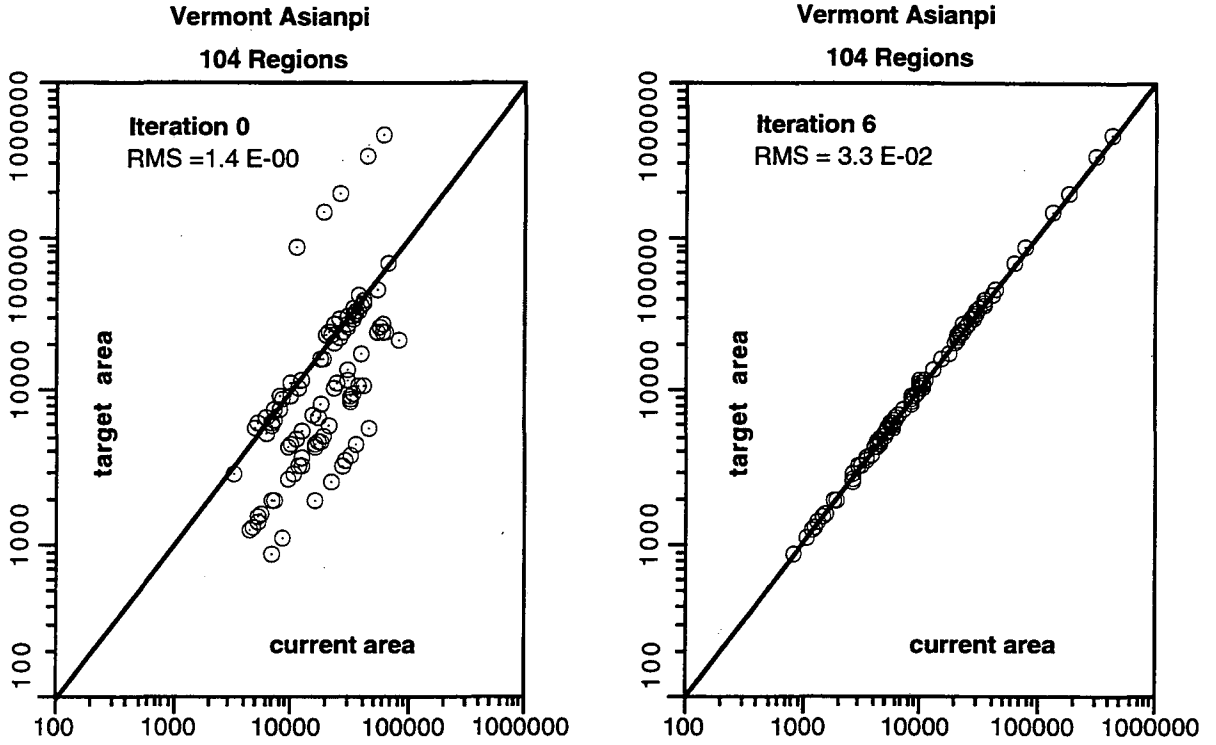


Figure 18. Area Scatter Plots for Vermont Asianpi

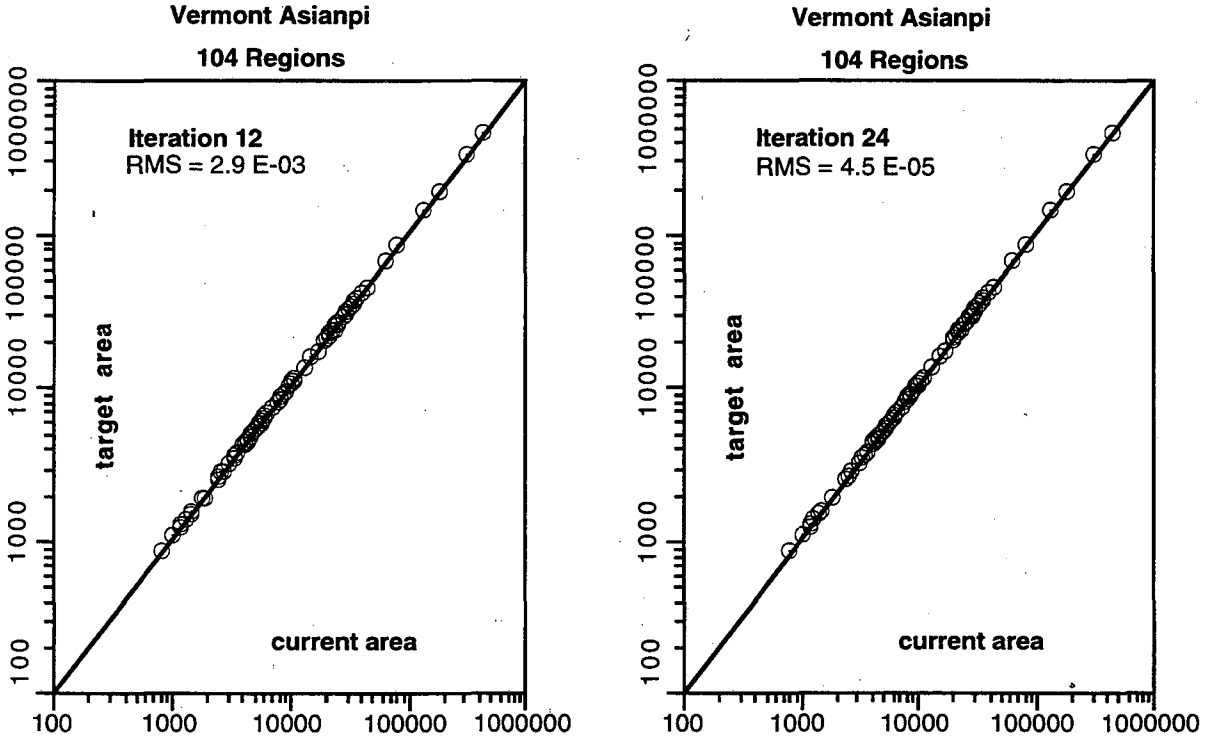


Figure 19. Area Scatter Plots for Vermont Indian

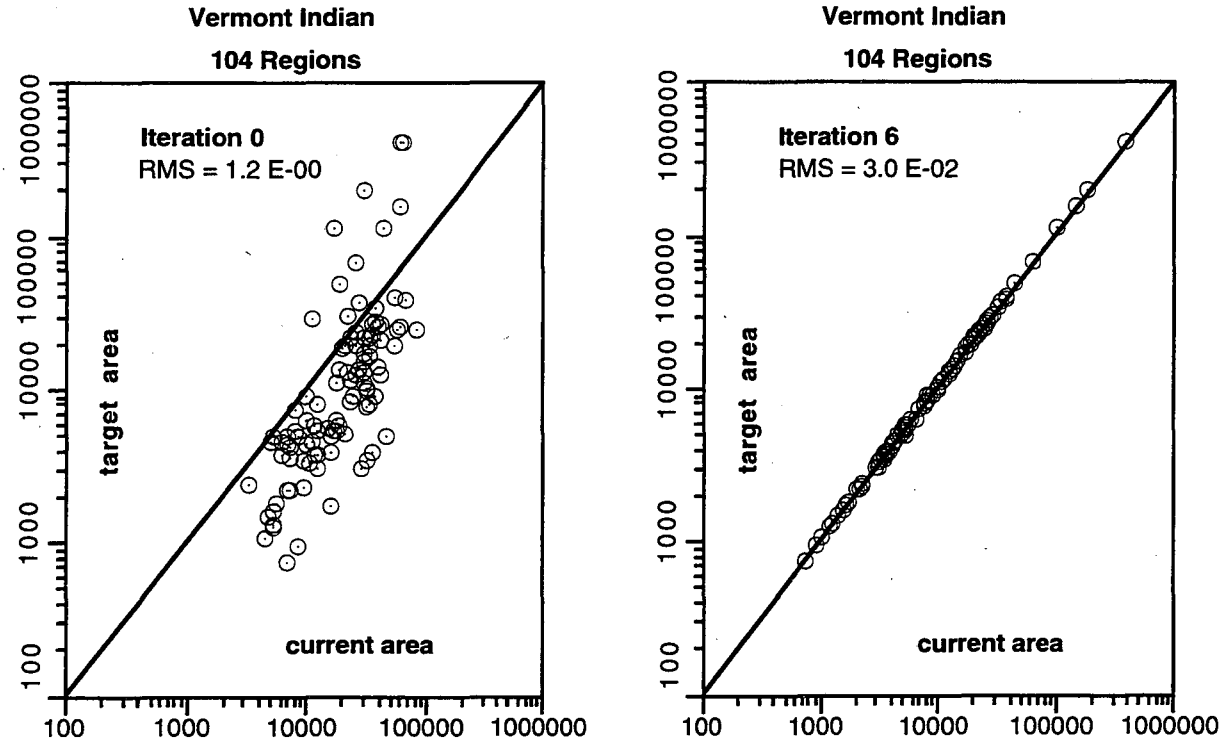


Figure 19. Area Scatter Plots for Vermont Indian

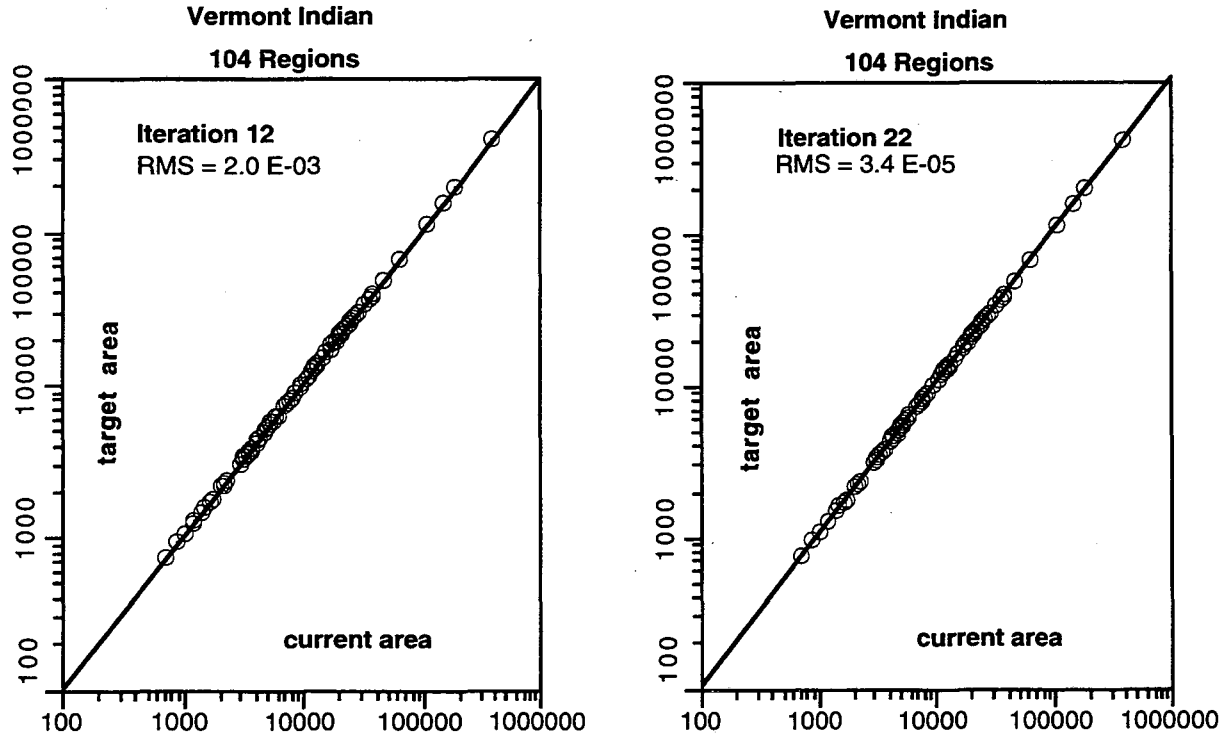


Figure 20. Area Scatter Plots for Vermont Black

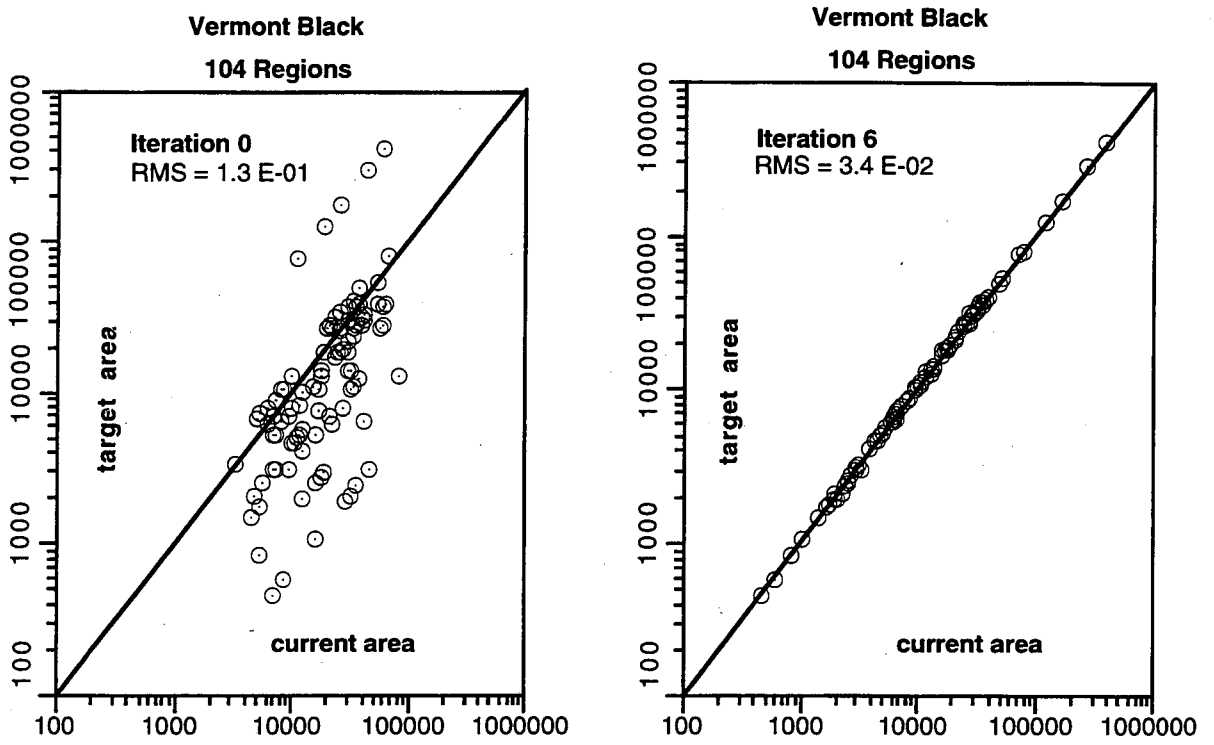
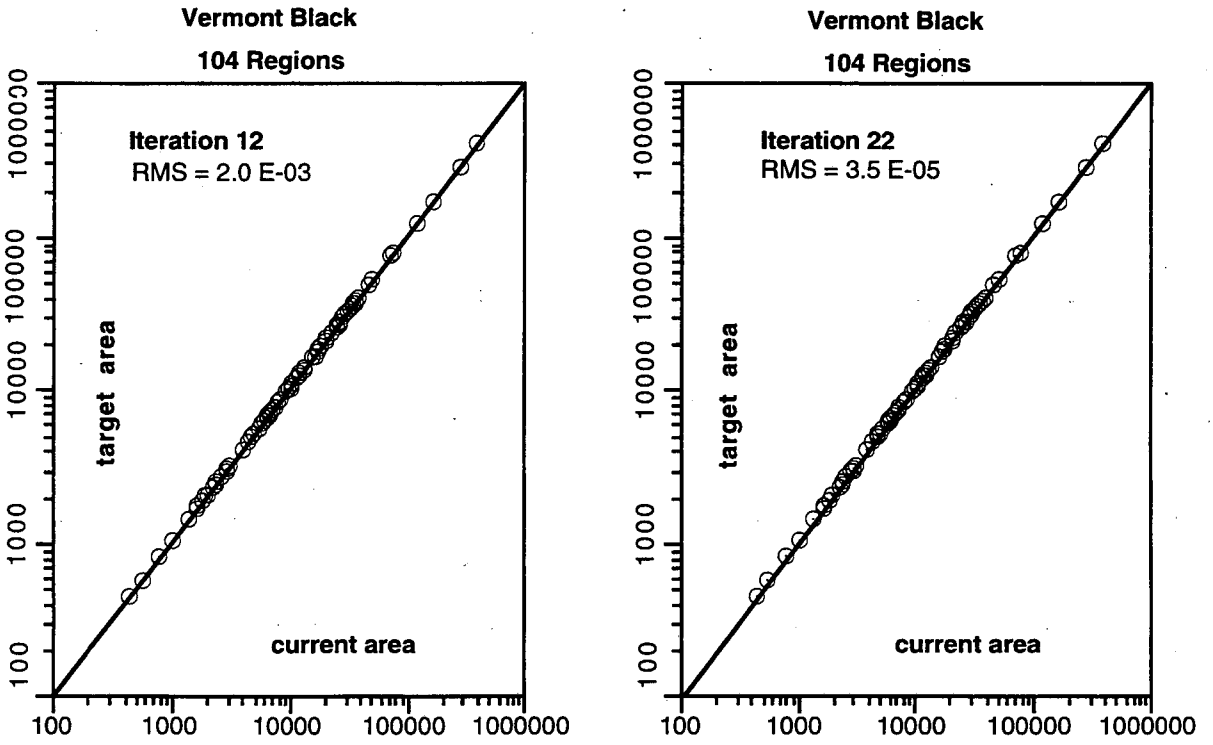


Figure 20. Area Scatter Plots for Vermont Black



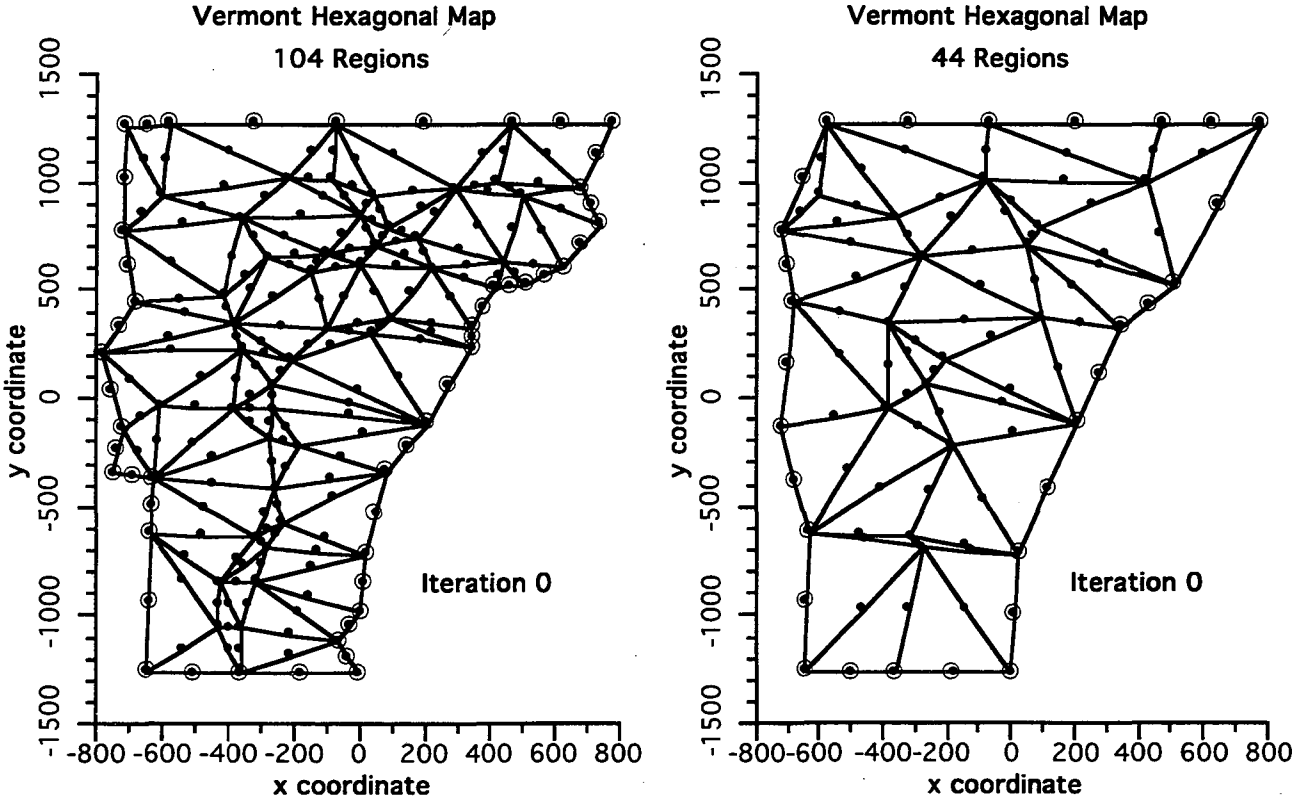
7.3.3 Initial and Final Region Maps

We have in the above sections discussed the convergence of the mappings. In this section we show the initial and final maps for the 44- and 104-region Vermont

hexagon maps. Our principal interest is the quality of the maps. The runs that we show here do not have any regions lost due to negative areas. This does not, however, mean that all the regions in the final map are satisfactory. In the present version of the program RLInt there is no check for folded boundaries; *i.e.*, boundaries that interesect themselves. Such regions could have positive net areas, but such a region has been improperly mapped. The region plots that we present will show if there are any such cases in the final maps.

We first show the initial maps. All seven population categories start with the same initial map. The difference for these maps is the target population. We repeat that the target areas have been scaled so that the final total area of the map is the same as the initial area. Figure 21 shows the initial 44- and 104-region maps. There are actually 43 and 103 distinct subregions; the first region is the outer boundary of all the subregions. We have included this special region in the maps shown below to verify that the outer boundary is correctly represented. The mapping translation vector is applied to the outer boundary, but the boundary points of this region never contribute to the construction of mapping vector.

Figure 21. Initial 44- and 104-Region Vermont Map

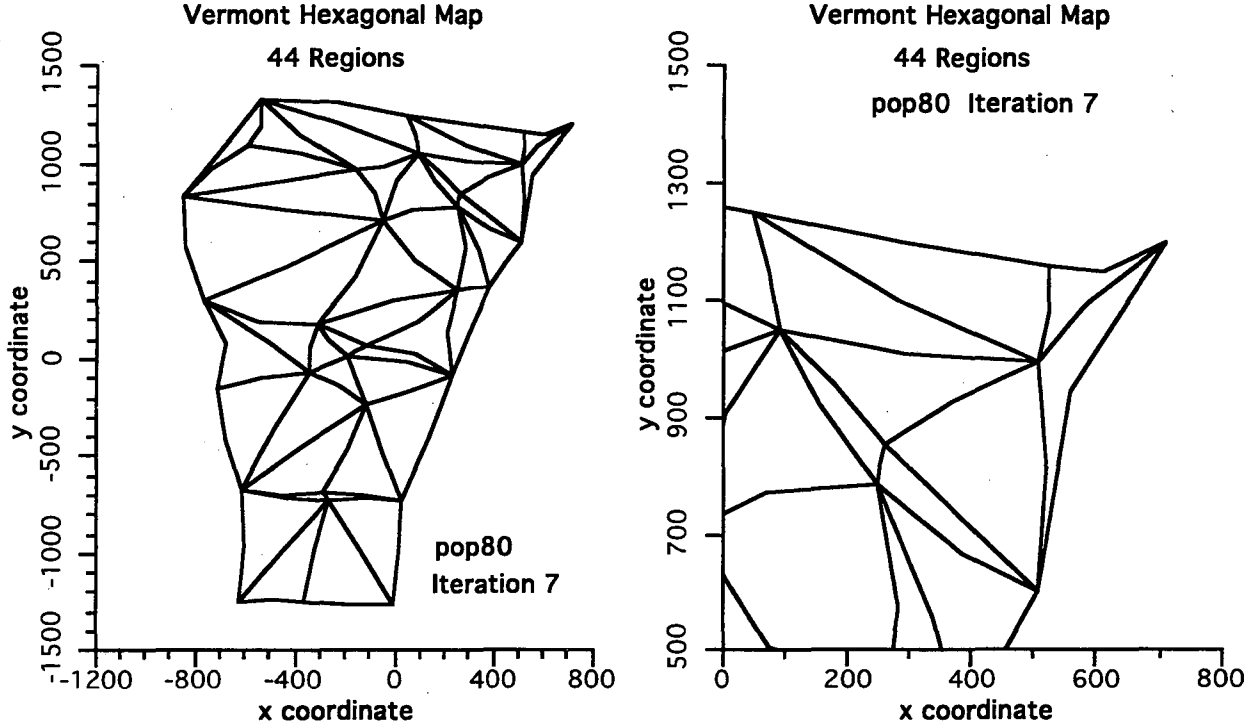


A close examination of these initial maps shows that they are not identical, but they are quite close to being the same. The outer boundary of the 104-region map has more detail and more correctly reflects the true outer boundary of the original Vermont map.

In Figures 22-35 below, the final map and an expanded view for each of the seven population categories are shown. For each category, we present the 44-region and the 104-region final maps along with an expanded view. The expanded view is in a region that could cause problems. We have noted that none of the maps had negative area regions. However, there are only six boundary points for each triangular

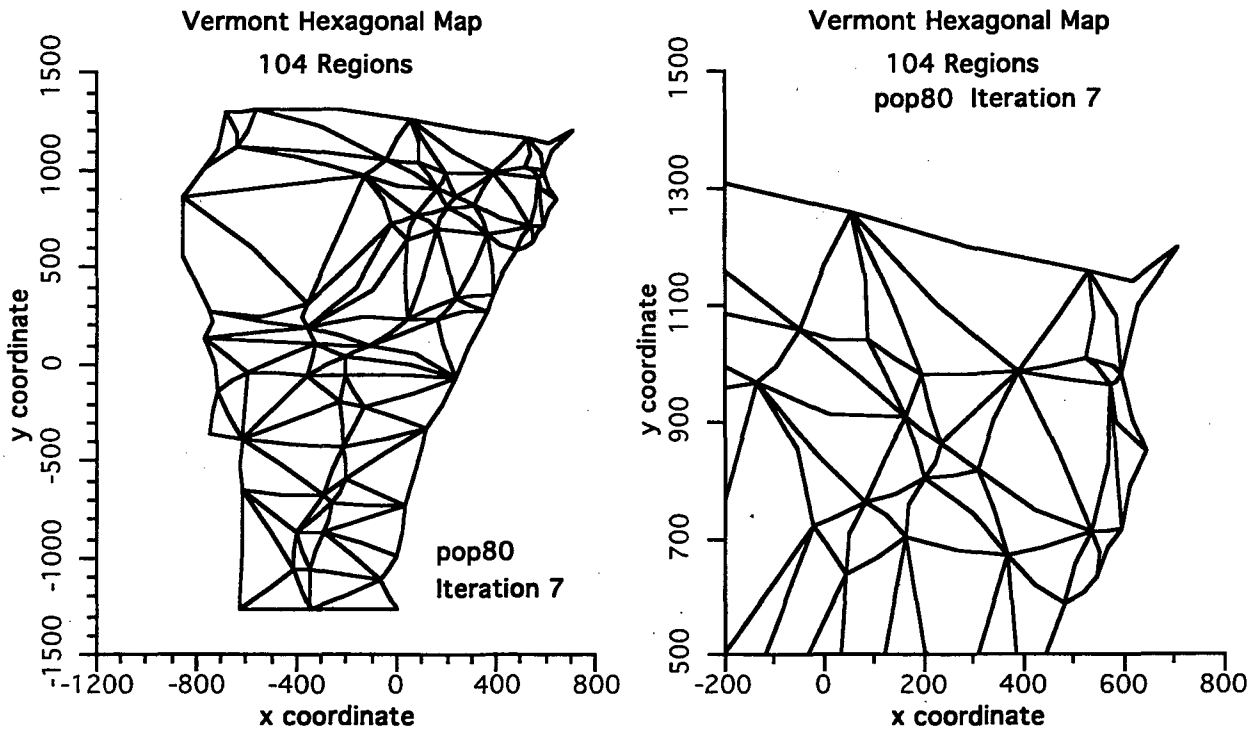
region and as the regions are transformed, it is possible for a boundary to cross itself. In the final maps, the boundary points are not shown. These points are at the ends of the straight lines that represent the regions boundaries.

Figure 22. Final Mapping 44 Regions pop80



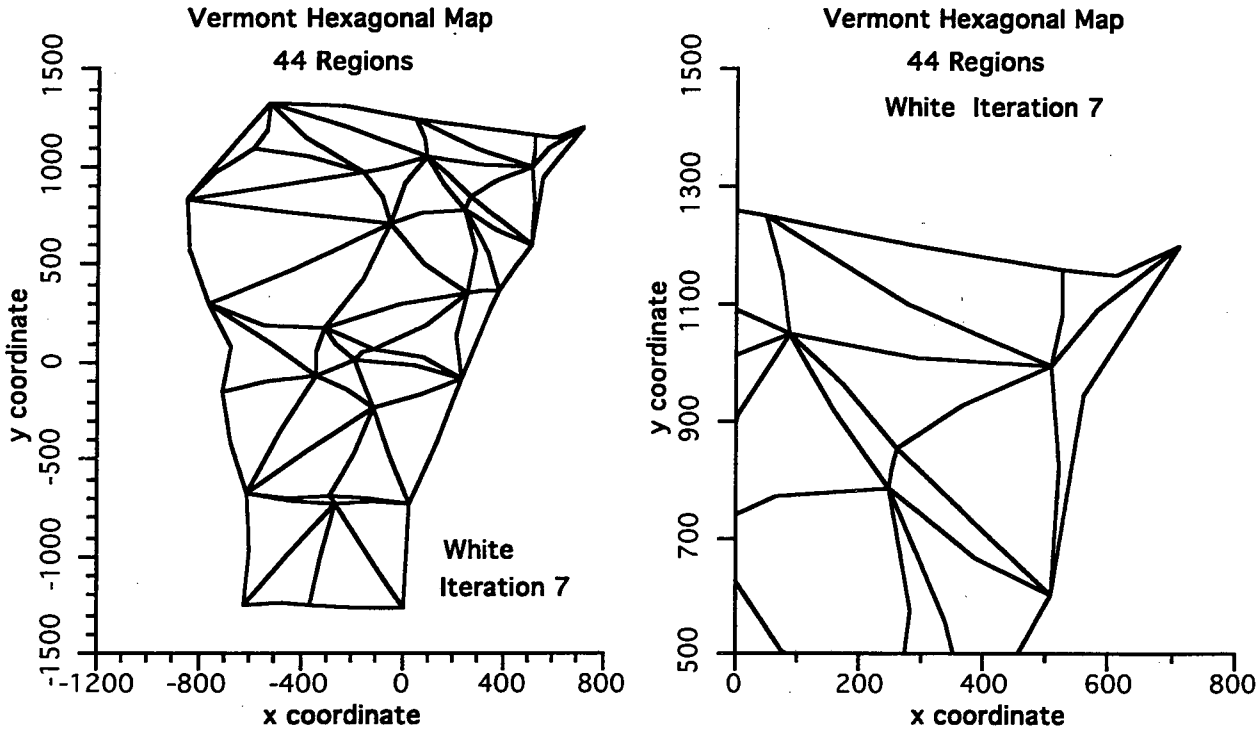
No regions are lost, and there are no boundary crossings.

Figure 23. Final Mapping 104 Regions pop80



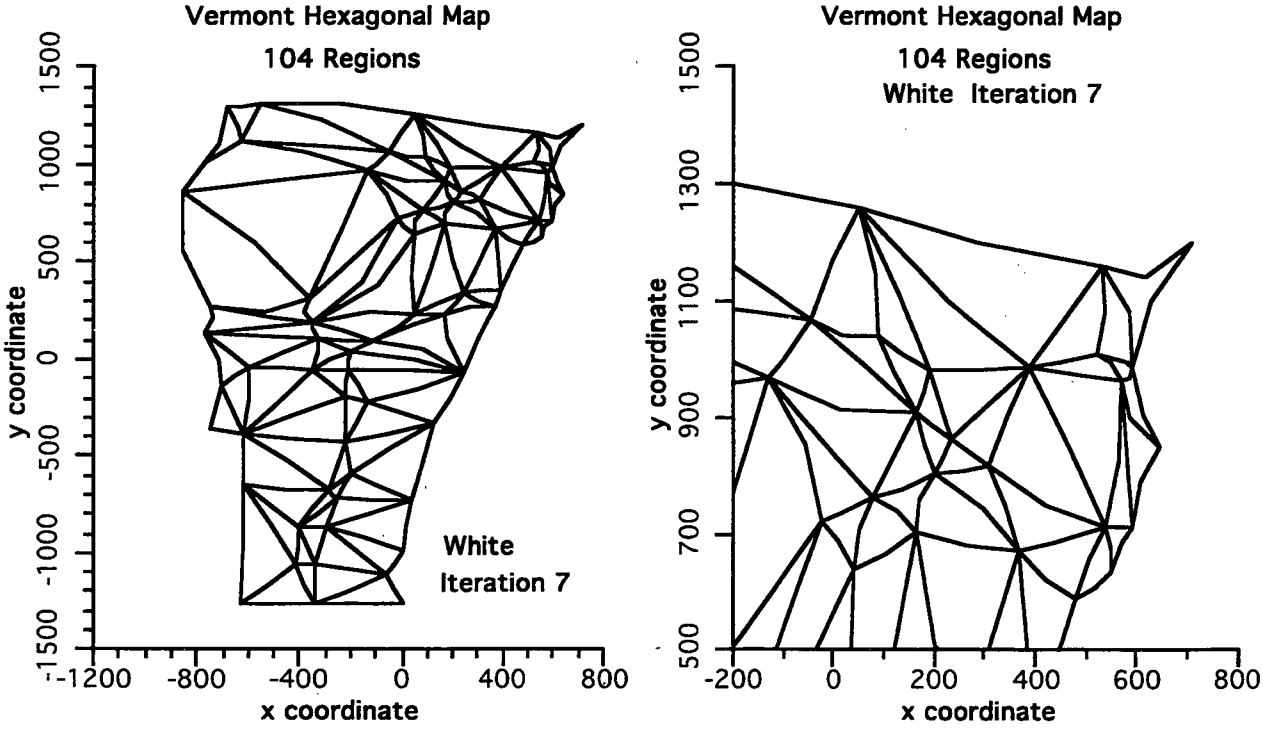
No regions are lost, and there are no boundary crossings.

Figure 24. Final Mapping 44 Regions White



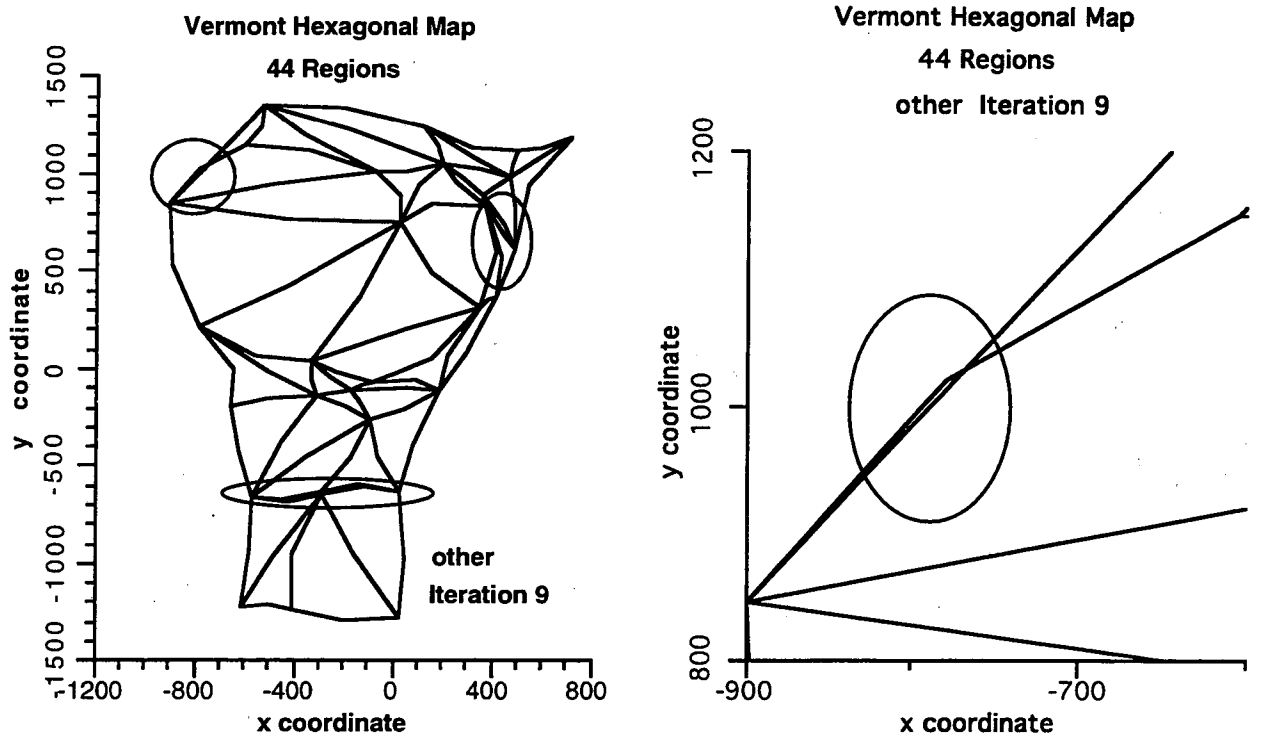
No regions are lost, and there are no boundary crossings.

Figure 25. Final Mapping 104 Regions White



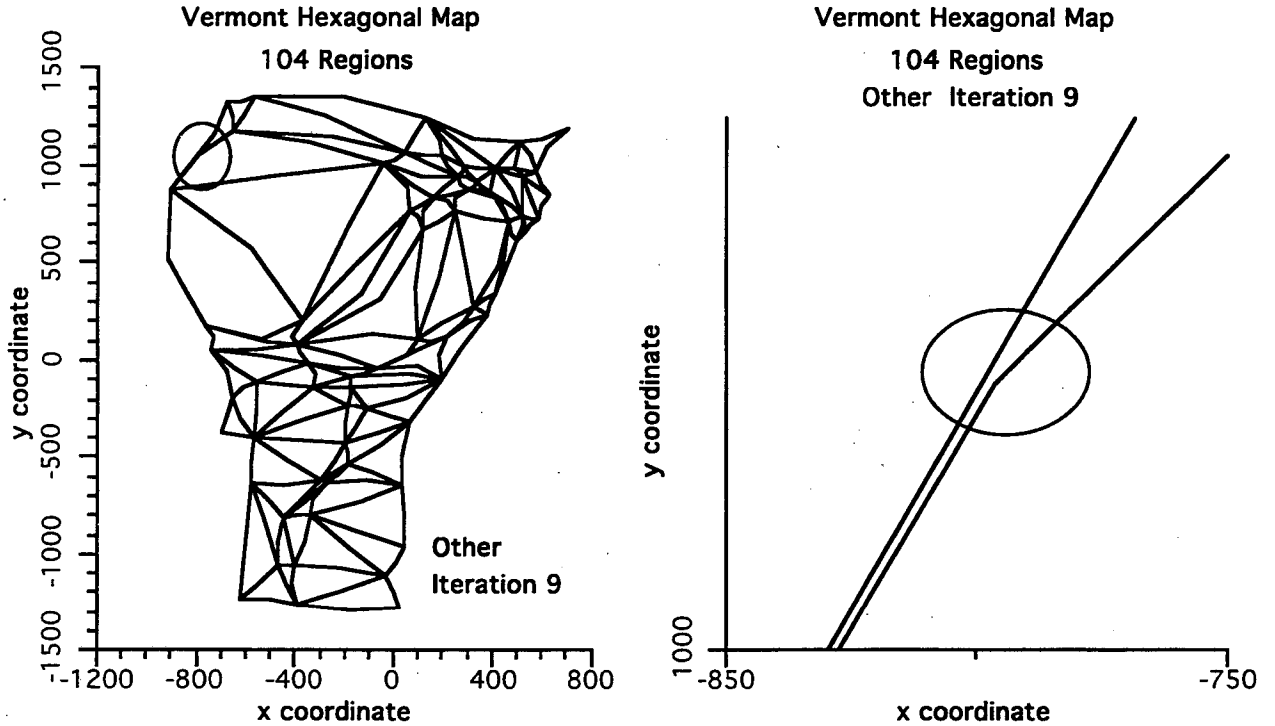
No regions are lost, and there are no boundary crossings.

Figure 26. Final Mapping 44 Regions Other



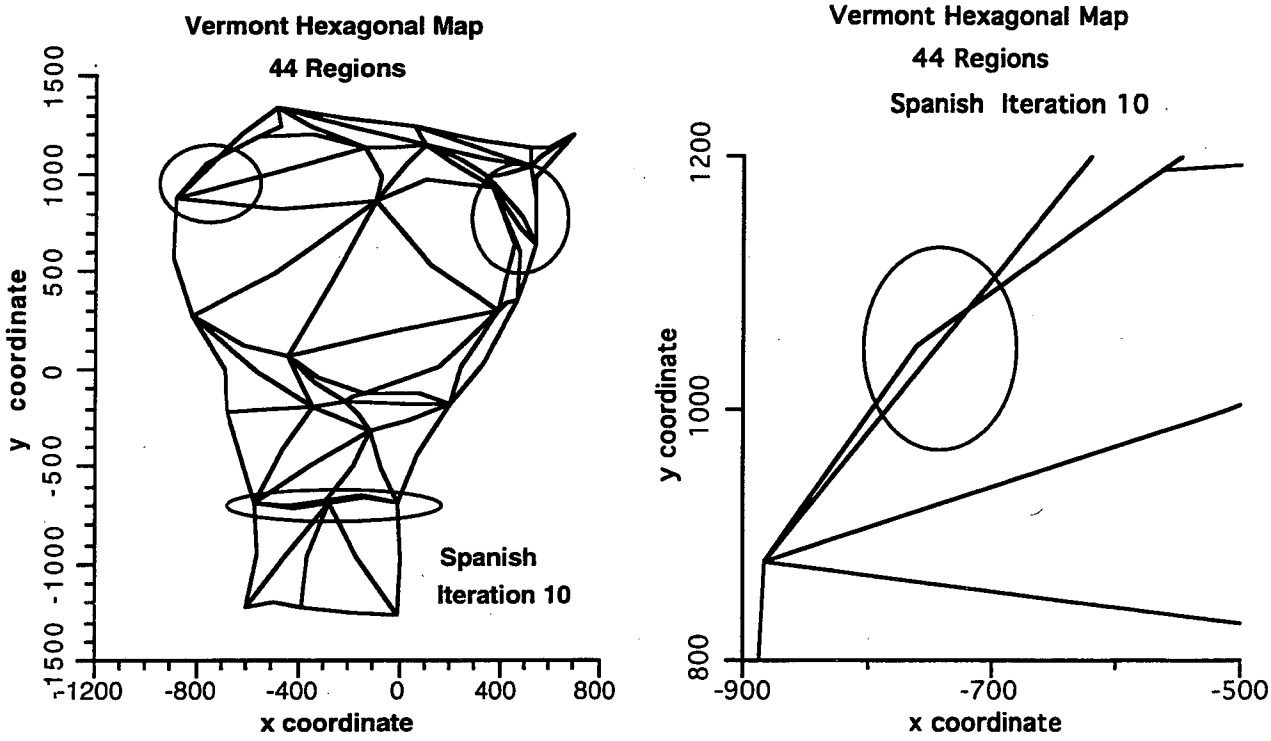
No regions are lost, but there is a region that folds upon itself. For the 104-region map, this region comes close to folding, but does not. The other circled regions do not fold.

Figure 27. Final Mapping 104 Regions Other



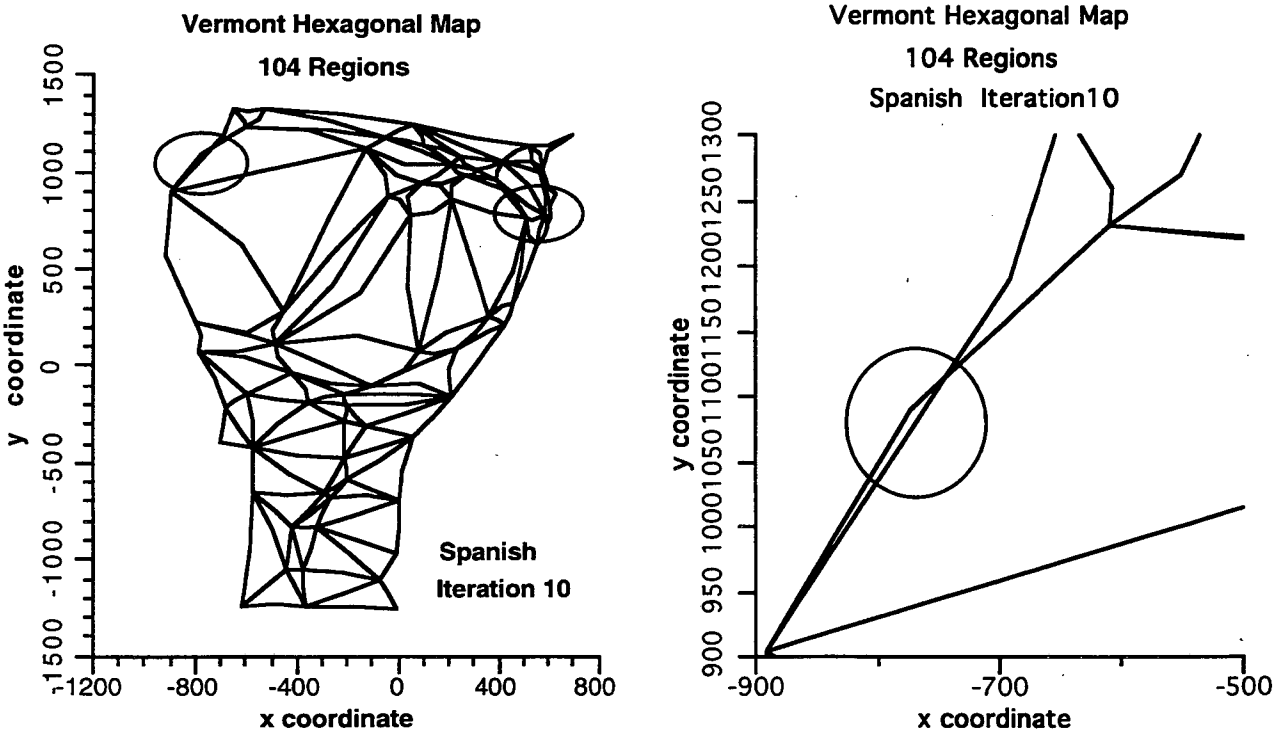
No regions are lost, and there are no boundary crossings. However, there is a region that comes close to folding upon itself.

Figure 28. Final Mapping 44 Regions Spanish



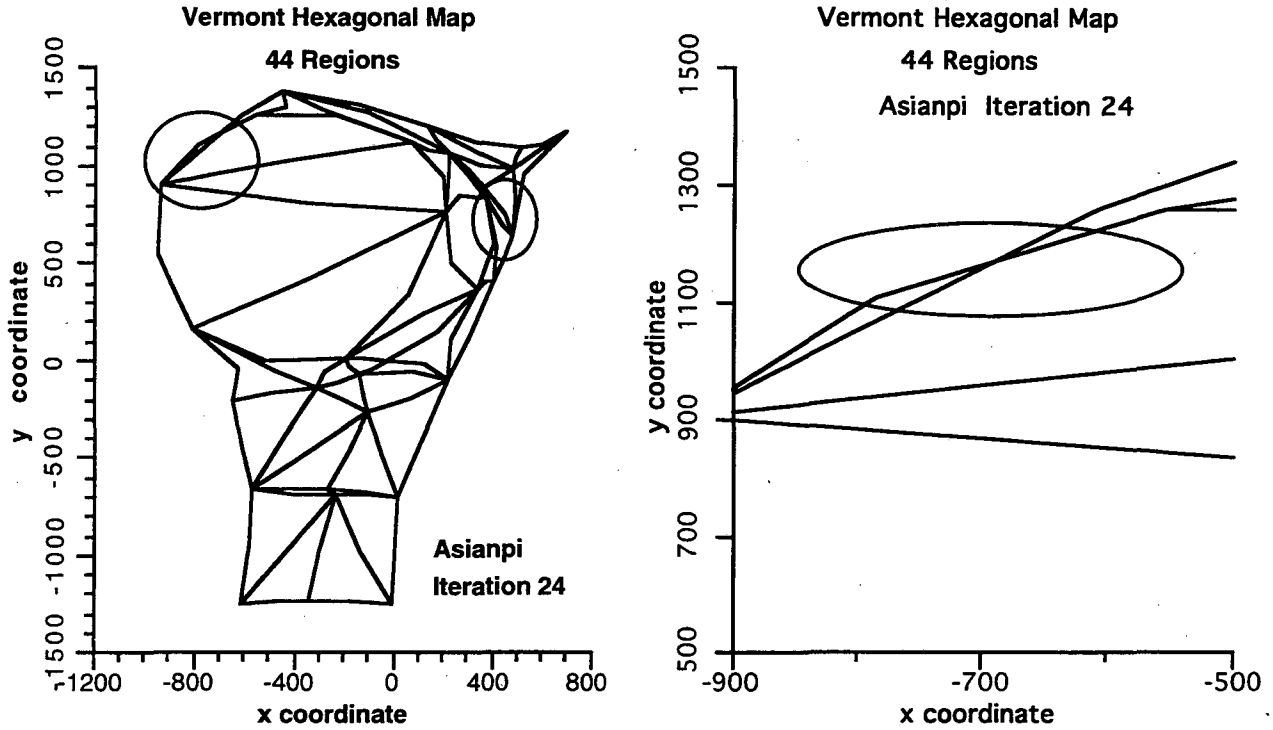
No regions are lost. There is one region with a boundary crossing. The other circled regions do not cross, although on the full scale map, they appear as if they might.

Figure 29. Final Mapping 104 Regions Spanish



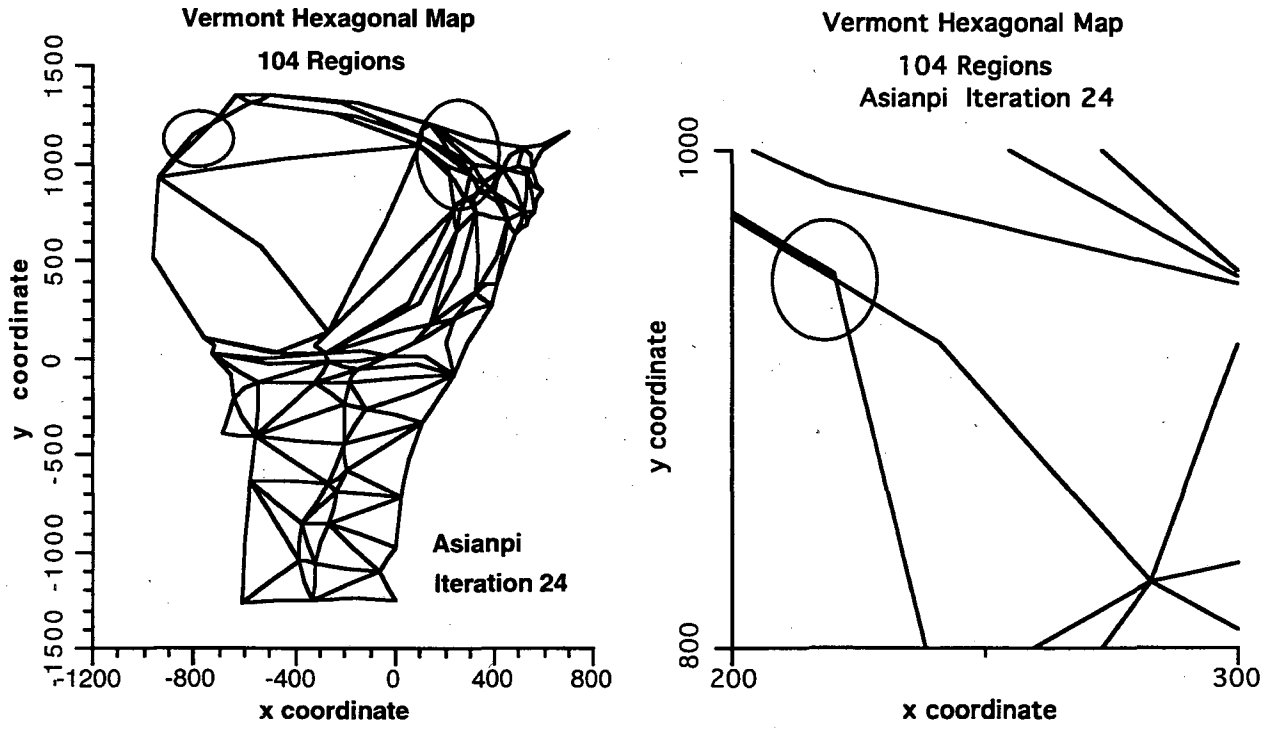
No regions are lost. There is a boundary that crosses itself. The other circled region does not cross, but comes close to doing so.

Figure 30. Final Mapping 44 Regions Asianpi



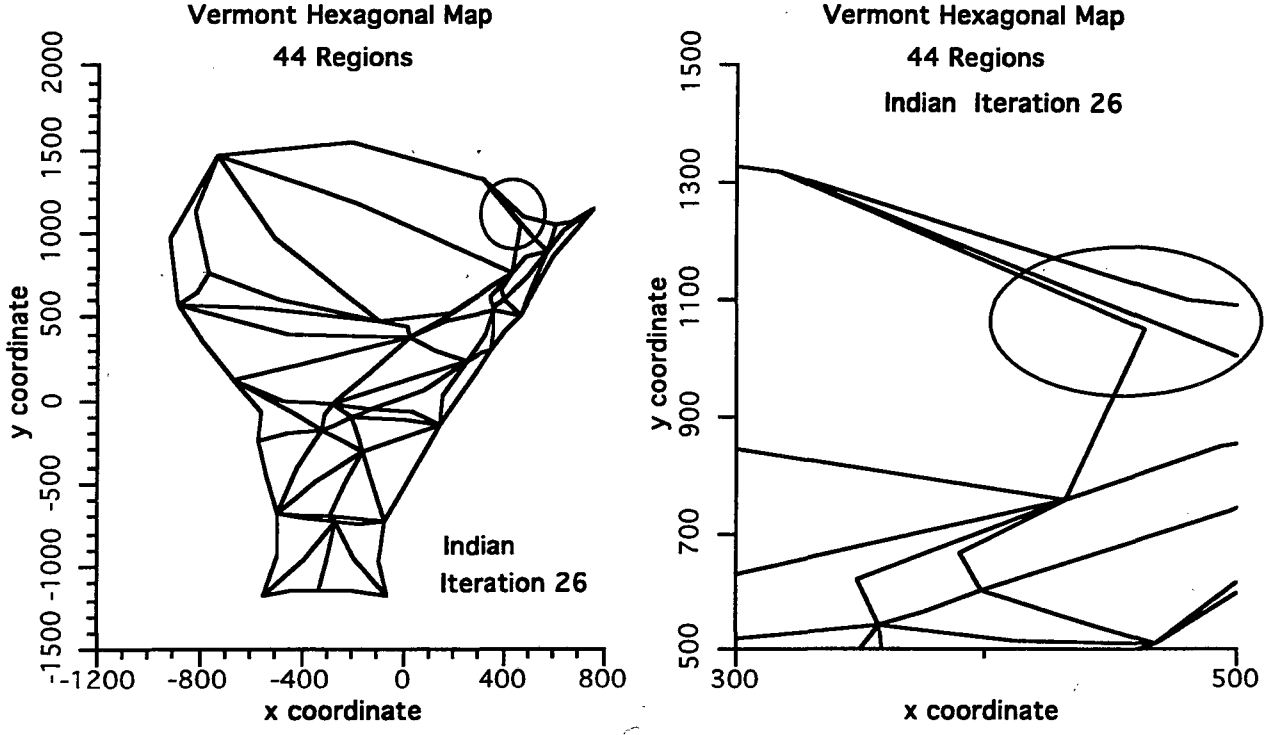
No regions are lost, but there is one region that has a boundary that crosses. The upper left circle is expanded, the upper right circle region does not cross. However, it does cross in the 104-region map.

Figure 31. Final Mapping 104 Regions Asianpi



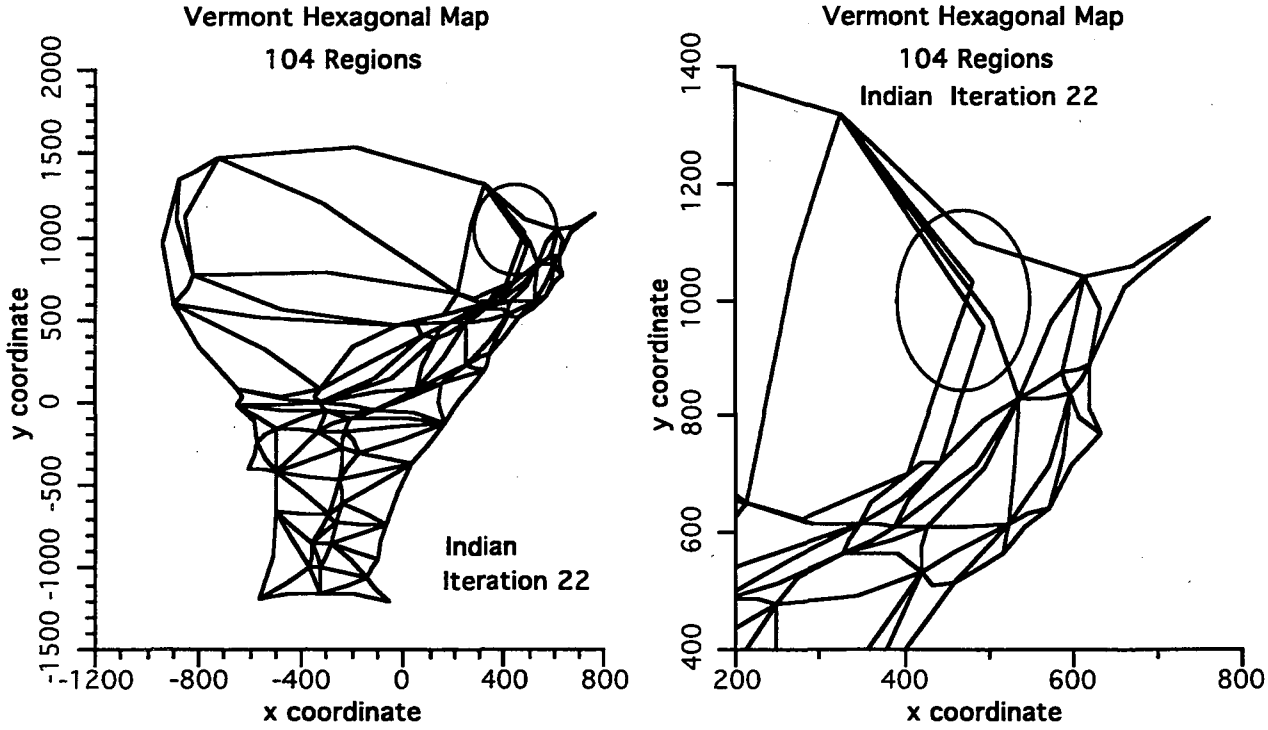
No regions are lost, but there are two regions that have boundaries that cross. The area in the upper right circle is expanded to show the boundaries crossing in the full map.

Figure 32. Final Mapping 44 Regions Indian



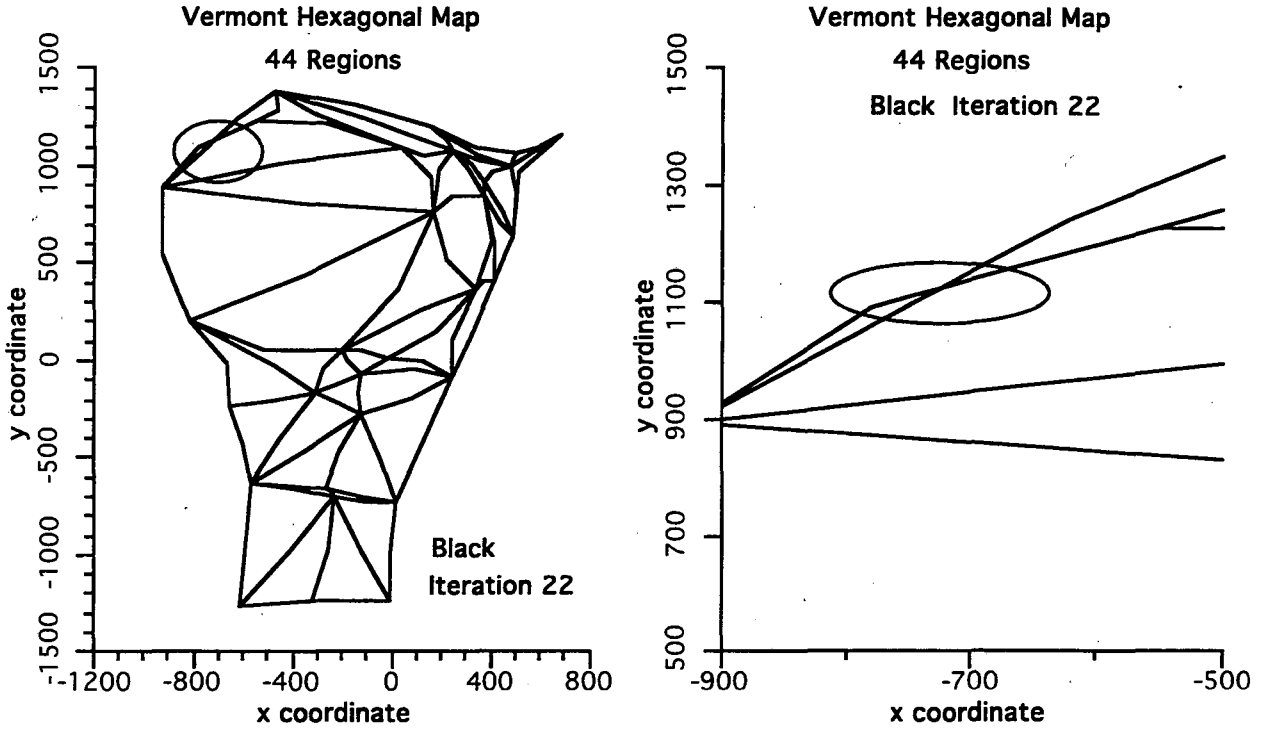
No regions lost. The circled, expanded region does not cross; but it does cross in the 104-region map. There are no boundaries that cross in the 44-region map.

Figure 33. Final Mapping 104 Regions Indian



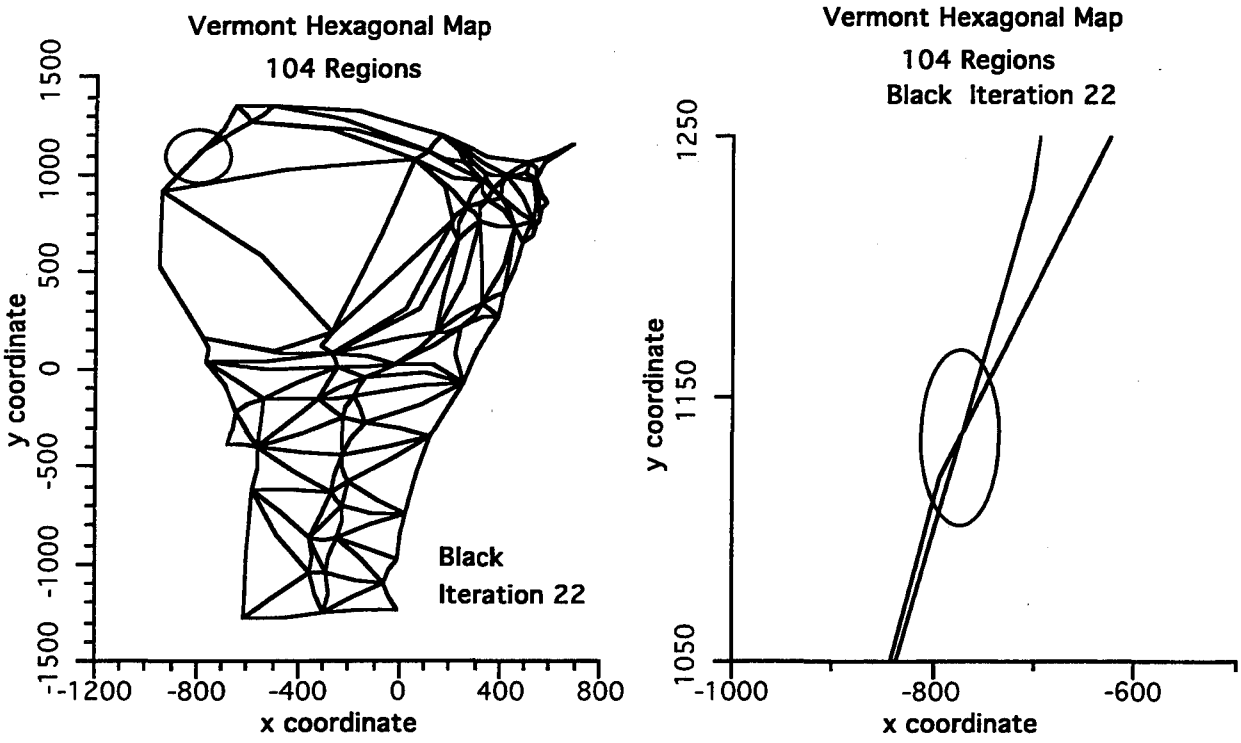
No regions are lost. However, there is an illegal boundary crossing which is circled in the expanded plot. A close examination of the map shows that this is the only such region.

Figure 34. Final Mapping 44 Regions Black



No lost regions. The circled region has a boundary crossing. No other regions have boundaries that cross.

Figure 35. Final Mapping 104 Regions Black



No lost regions. However, the expanded view shows that there is a region with a boundary that crosses. The same region had this problem in the 44-region map.

The final maps that we have shown in detail above indicate that there can be a boundary crossing problem with the mappings. Presently there is no checking done to detect these crossings and hence no strategy for preventing such crossings. Increasing the number of points in each boundary should help prevent the occurrence of these crossings. It is seen from the above maps that increasing the number of regions, while keeping the same six points per boundary does not necessarily prevent crossings. Again, the basic difference in the 104-region maps and the 44-region maps is the number of regions that make up the total map.

We should also note that simply looking at the convergence, or the area scatter plots of the maps is not sufficient. The maps shown above all converged well and the scatter plots have no regions that are far from the target areas. It is necessary to visually inspect the maps to determine whether they are satisfactory. There is sufficient output from the program to do this as the mapping proceeds from iteration to iteration. Alternately an algorithm for detecting boundary crossings could be implemented.

7.4 Timing Comparisons

In this section we examine the time that it takes to perform the mappings. This can be measured in different ways. For example, we can measure the time that it takes to construct the translation vector for one point in the map. Alternatively, we can measure the time that it takes to complete one full mapping iteration to obtain the next map in the sequence of transformed maps. Or we could determine how the time varies as the number of regions in the map increases. And finally, we could measure the time it takes to go from the initial to the final map; *i.e.*, how many iterations it takes to converge a map. In the sections that follow we discuss these different times.

7.4.1 Timing the Line Integral Mapping Algorithm

In line integral algorithm the construction of the translation vector for any point in the map requires that the pushes from all the boundary line segments be calculated and summed. The number of boundary segments in each subregion D_k is equal to the number of points used to represent the boundary.

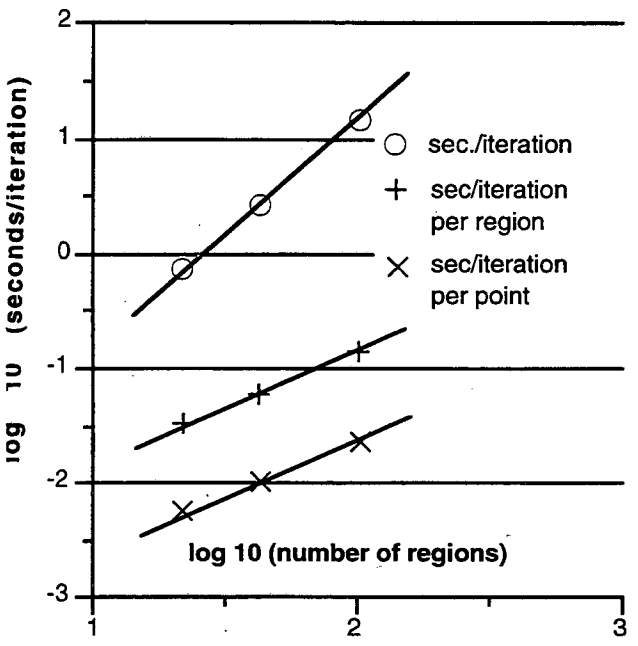
For the hexagonal maps of this report there are six points per boundary, hence six segments per boundary. The current version of RLInt has a full set of boundary points for each region. Thus, a 44-region map contains 264 points, 6x44 points that are mapped. The first region for this map does not enter into generating the transformation vector, so there only 258 boundary points that generate a push from their associated line segments; 43 regions enter into the construction of the mapping.

The 104-region map contains 624 points, 6x104 points that are mapped. Again, the first region of this map does not enter into the generation of the transformation vector; there are in this case 618 boundary line segments.

We expect that the time taken to generate a transformation for each point to be a linear function of the number of boundary segments; those that enter into the construction of the translation vector. Since the number of segments approximates the number of points, we also expect that the time for generating one complete iteration to vary as the square of the number of boundary segments that generate a push. In the simple hexagon cases that we have run the number of points is directly proportional to the number of regions. If we double the number of regions, we double the number of boundary line segments. We have therefore plotted the results as a function of the number of regions in the map. In general, the number of points can vary for each boundary region.

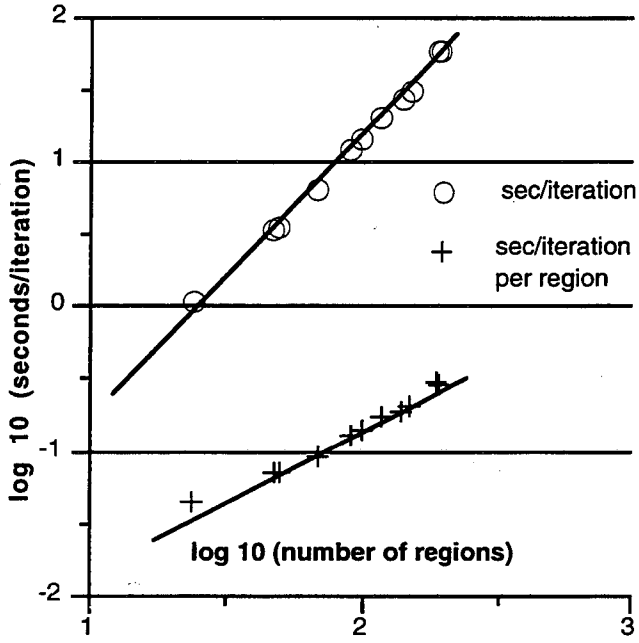
In Figure 36, "Timing for Hexagon Regions," we plot three quantities: the seconds per complete iteration, the seconds per iteration per region, and the seconds per iteration per point. The data points are for 22, 43 and 103 regions. We note that we have correctly taken the number of regions that enter into the generation of the translation vector. For timing purposes region 1 has been omitted. These plotted results do not contain non-pushing regions, or other non-boundary points.

Figure 36. Timing for Hexagon Regions



We have plotted the logarithmic values of the number of regions and the time in seconds. For reference we have plotted on this graph a line for which the slope is 2 and a line for which the slope is 1. From this log-log plot we see that the seconds per iteration varies as the square of the number of regions, as expected. The time per iteration per region and the time per iteration per point vary linearly with the number of regions. This graph simply verifies what we already know about the algorithm. It is, however, useful in that it can be used to predict the time for hexagon maps that contain different numbers of regions.

Figure 37. Timing of 4-County Map



The above results were obtained from uniformly defined hexagon maps. These maps had no anomalous regions and all the regions had 6 points per boundary. We show in the next graph the same type of results for maps in which the regions do not all have the same number of boundary points. We used a 4-county map that has 265 regions. This map has been used here strictly for timing. It has been run for subsets of these regions. During the runs, the mapping was adjusted so that no regions were lost. The only quantity of interest during the runs was the time to complete a fixed number of iterations. These results were combined with previous runs for this map in which regions were lost. This collection of runs yields qualitatively the same quadratic dependence on the number of regions and the same linear dependence on the number of points.

In the next two graphs, Figure 38, "Number of Points in Each Region for the 4 County Map," and Figure 39, "Cumulative Number of Points for the 4 County Map," we show the number of boundary points for each region and the cumulative number of points as the number of regions goes from 1 to 265. We have in the previous graph plotted the results versus the number of regions. We see from the next two graphs that the total number of points in the map is close to linearly dependent on the number of regions. This confirms that the plotting the timing as a function of the number of regions is reasonable for this map even though the number of boundary points for any given region is not constant.

Figure 38. Number of Points in Each Region for the 4-County Map

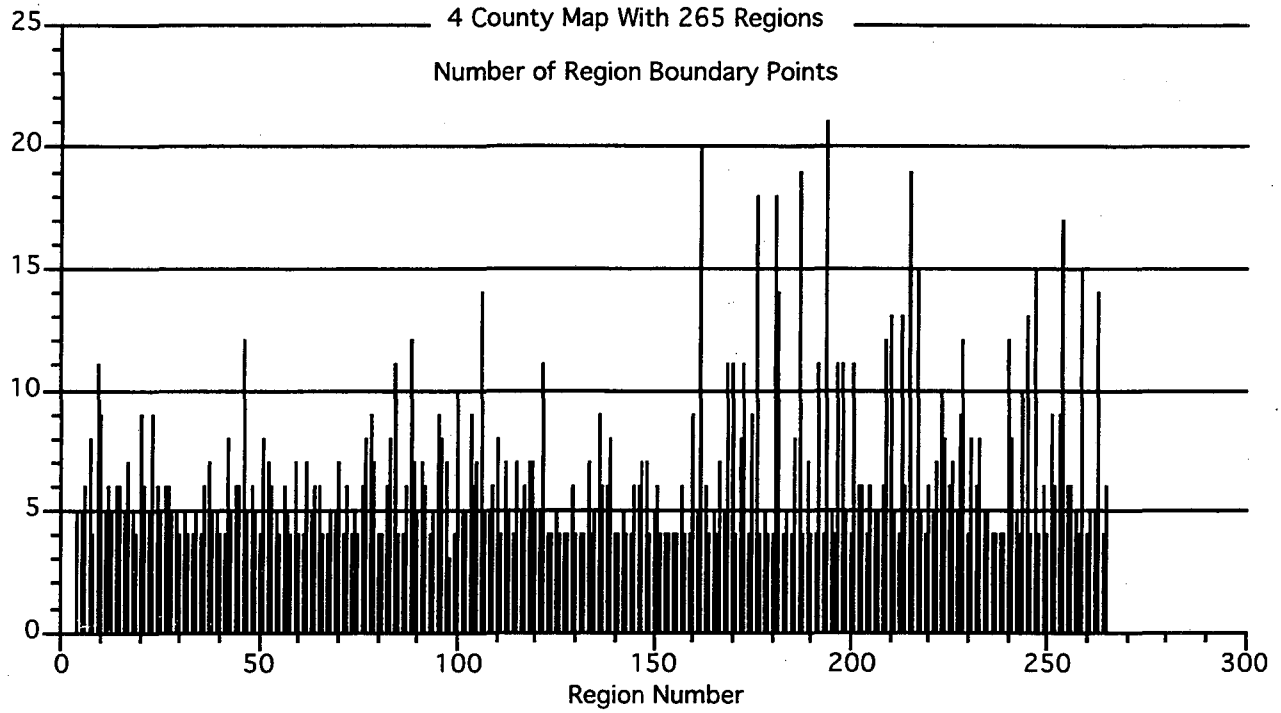
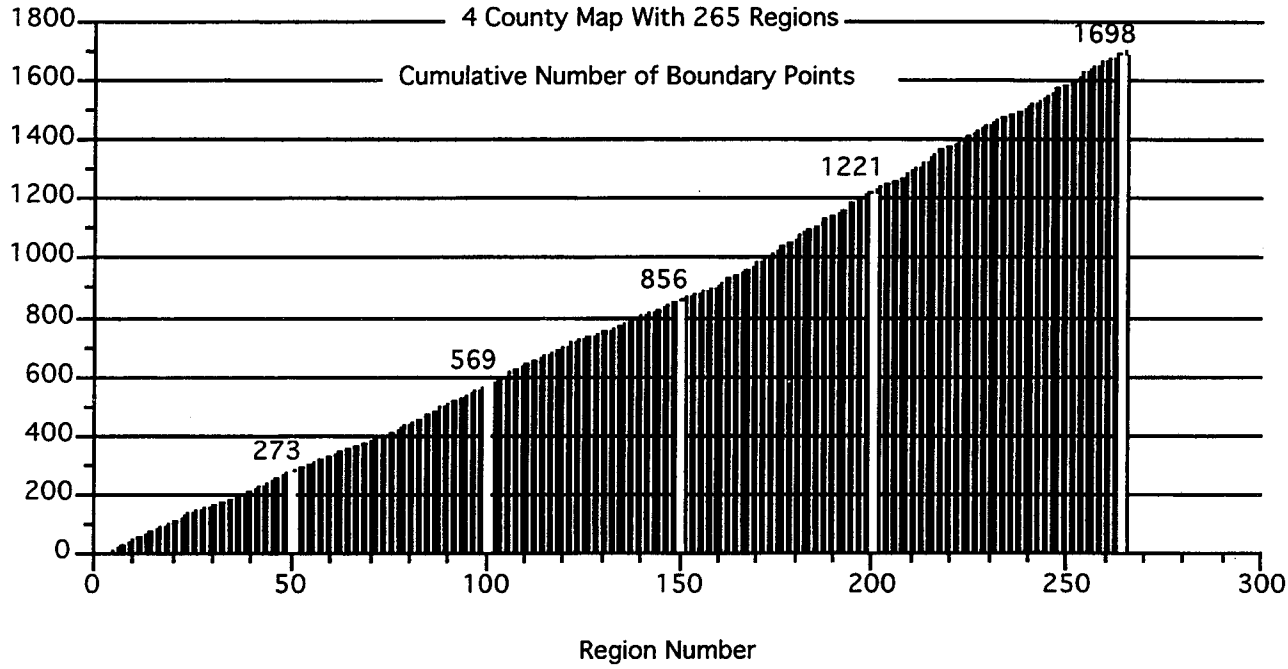


Figure 39. Cumulative Number of Points for the 4 County Map



In reference 12, LBL 32640, times are given for some particular maps. In Figure 20, p. 32 of that report a value of ≈500 seconds per iteration is given for a map labeled SF. This is for a SPARC 2 machine listed at 16 mips, million instructions per second. The characteristics of this map are given on p.3 of the cited report. The map has 270 point, 502 triangles and 771 segments. For the purpose of making a timing estimate, we will assume that the triangles have 6 points, as in our hexagon map. This would be equivalent to running a hexagon map with 502 regions; a total of 3012 points.

We assume that the time per iteration scales quadratically with the number of regions. Going from 103 regions to 502 regions is a factor of 5, so we estimate a factor of 25 increase in time for this map. The time for the 103 region hexagon map is 14.42 seconds/iteration. The time per iteration for the 502 region SF map should be about 360 seconds per iteration on a SPARC 10, instead of a SPARC 2. It is not possible to obtain an absolute ratio of times for these two machines on the original SF map runs without rerunning the examples. However, it seems reasonable to estimate a factor of 2-4 increase in speed on the SPARC 10. If we use 3 as our number for the speed increase, then the 500 seconds per iteration become more like 160-170 seconds per iteration. We have about a factor of 2.2 increase in time per iteration. However, that is not the complete story. The rest is how many iterations are needed to converge. We shall address this in the next section of this report.

The other map referenced in that report is the Alameda map. Details are given on p. 4 of the cited report. This map has 1090 triangles and 1670 segments. For our purposes we consider it to be a hexagonal map of 1090 regions. The ratio 1090/103 is about 10. Thus, for the line integral algorithm the time per iteration for this map should be about 1442 seconds, or about 24 minutes per iteration. On p. 8 of reference 12, the time given is 40 minutes, on a VAX 6610 at 35 mips. A direct comparison of this VAX run with SPARC 10 runs given in this report is not possible without rerunning the original problem However, it would not be unreasonable to take the VAX as being within 25% of the SPARC 10. That is the 40 probably would not reduce to less that 35 minutes for the Alameda map. Again, the estimated number of iterations for the

original run to converge is 100 to 300 iterations. This probably would not be the case with the line integral algorithm. Again this will be addressed in the next section.

7.4.2 Number of Iterations Required for Convergence

We have in the previous section determined the time per iteration as a function of the number of iterations, the number of mapping steps. This, as expected, is quadratic with respect to the number of boundary points. We have also shown that it is close to that with respect to the number of regions; even when the number of boundary points per region is not constant.

In this section, we are interested in how many iterations it takes to obtain convergence. This will determine the actual total time that it takes to generate the final map. Above in Section 7, "**Summary of Results,**" we have shown graphs of the convergence of the maps.

It is necessary to decide what it means for a map to have "converged." It has been noted previously that the single precision program RLInt cannot be expected to get the RMS function much smaller than 10^{-6}, in practice we stop at 5.0×10^{-5}. However, we see from the graphs presented in Section 7.3.2,"Area Scatter Plot Convergence," that 10^{-3} gives an area scatter plot that is visually quite good and that the maps at this stage of convergence look quite the same as those further in the sequence. If the maps look satisfactory, and if the scatter plots do not specify regions that are far from the line of convergence, we can estimate that an RMS function value of 10^{-3}, equivalently an Hsum function of about 10^{-6}, has about 3-4 significant figures in the current area. Whether this is enough accuracy for analyzing the final map quantities will depend on the map and what it represents. Further convergence will lead to better agreement with the target areas.

Referring to Figure 10 and Figure 11 in Section 7.3.1, "**RMS Convergence,**" we see that the number of iterations needed to reach full convergence did not vary significantly with the number of regions, at least not for the 44- and 104-region hexagonal maps. However, this quantity did depend on how far the map was away from a converged value. This measure of difficulty could be seen by looking at the initial Hsum function value. As this value became larger, the number of iterations needed became more. And there could be some difficulties with partially upside down triangles, regions with boundaries that crossed. In fact, for some races, we had to change to the binary 1/2 option to obtain good convergence and not have negative triangles.

Our comment that the total number of iterations is not dependent on the number of regions does not mean that as the number of regions becomes large, there will not be an increase in the required number of iterations. But it does not appear that this parameter in itself is a strong factor for determining the number of required iterations to reach convergence. The mapping is global in nature and appears to converge rather fast.

We refer now to the Vermont Populations graph in Figure 11, Section 7.3.1, "**RMS Convergence,**" in which the seven Vermont maps are plotted. From that graph, we see that the full step (fixed point) iteration RMS value converged by a factor of 100(2 significant figures) every 5 iterations or 0.4 decade per interation. The binary 1/2 option maps converged very smoothly at 0.18 decade per iteration, very close to 2 significant figures per 10 iterations. The first three maps, pop80, White, and Other, converged in 7-10 iterations for an average convergence rate of about 0.5 decades per iteration. The last 4 maps, Spanish, Asianpi, Indian, and Black converged in 22 to 24 iterations for an average convergence rate of about 0.2 decades per iteration.

To summarize, the maps with lower initial RMS gained about 0.4 to 0.5 decade of RMS per interation. Those that had a higher Hsum value gained about 0.25 decade per interation The maximum that we can ask for in this single precision program is an RMS of the order of 10^{-5}, or 5 significant figures.

The hexagon maps are simple and well-behaved. We have run triangle maps, described in Section 7.2.1,"**The 44-Region Vermont Triangle Map,**" and concluded that it is necessary to have the additional degrees of freedom that the addition of the segment midpoint gives. It is also to be noted that this algorithm may need even more boundary points to accurately represent the region boundaries and to prevent boundary crossings. We have not investigated this aspect of the mapping in detail.

It is interesting to estimate the time required to converge the 502 region map and the Alameda map of 1090 regions when mapped with RLInt. For this purpose, as was done in the previous section, we assume that the maps are triangularized with six points per triangle. We assume that each region has all six points, again as was done in the previous section. We make the estimate for converging three and for converging four decades. Where this take us with regard to the number of significant figures in the final, current, areas depends on where we start. If we start with an RMS on the order of 10, then we still arrive in the four decade case with an RMS of 10^{-3}. We use this range simply as an example, other ranges are easily calculated. We take the worst case of 0.2 decades per iteration and we arrive 15-20 iterations. Translated to total time we have for the San Francisco map 360 x (15 or 20). This gives a range of 1.5- 2.0 hours. A similar estimate on the Alameda map would be 1442 x (15 or 20). This gives 6-8 hours for converging 3-4 decades.

In summary: An estimate for running the San Francisco map through 4 decades of convergence is about 2 hours of computing. This assumes 20 iterations. A similar estimate on the Alameda map yields 8 hours. On p. 8 of the cited LBNL report the estimate it "about 6 to 9 days." Even if our estimate is off by a factor of 2-3, this is considerable better that the "6 to 9 days" previously estimated.

As a final comment on convergence, we should refer to Figure 13, "Asianpi Convergence vs. Iteration Count," of Section 7.3.1. This graph shows that convergence is not "guaranteed" with any particular choice of the mapping options. There is a discussion of this graph in that section; we will not repeat it here. Suffice it to say that the convergence of the map must be monitored in general, and if difficulties arise, action must be taken. It seems that it should be possible to restart at some point in the sequence with a choice that causes the map to continue to converge and which maintains the quality of the resultant map. Presently, this monitoring and restarting is done manually.

8. Future Work Required

The LBNL implementation of the Russian line integral algorithm is fairly complete. Test cases have been run to insure the accuracy of the algorithm in that it correctly evaluates the required line integral and that it can take real maps and produce output that is basically correct and useful. However, future work is required to complete this implementation. We discuss in this section some of those tasks.

8.1 Input Maps for the 4-County Area

The preparation of the input maps is described in references 3 and 10. These steps include the following: correction of map errors, removal of excess geographic detail, insertion of connecting segments to eliminate doughnut-shaped polygons, and triangulation of the resulting polygons.

8.2 Handling of Pathological Conditions

As long as the transformation that is applied generates a valid map there is no problem. However, problems with the resultant map can begin to appear. We discuss below in the Section 8.3 "**Program Improvements,**" some problems that should be taken care of with regard to illegal regions. However, it may be that we are unable to prevent the transformation from generating a map that has problem regions. For example, it may be that a water region drifts out of the region that it belongs in, but the rest of the map is satisfactory.

8.2.1 Correction Within DEMP program

The most natural way to correct while inside the DEMP program is to try to adjust the mapping so that each transformation produces a valid map that has no pathological conditions. Some of the additions that can help to prevent this are discussed in Section 8.3, "**Program Improvements.**"

If this is not possible, then it probably is the case that we have found the undesirable condition by examining the resultant map from outside the DEMP program. In that case, it makes sense to try and correct it from outside. This use of the word "outside" however, can refer to an interactive update of the parameters that define the mapping algorithm. The map is examined external to the DEMP program in some way, a problem is found, the DEMP program is interrupted, and then mapping parameters adjusted.

This type of correction requires a map viewing interface and some type of user interaction interface with the DEMP program. It also requires some criteria to be used in making the decisions on what to modify in the mapping algorithm. A restart from a good map can then be done. Again this is in some sense using the DEMP program to prevent the anomalous mapping.

8.2.2 Correction Outside the DEMP Program

The identification of a map problem that is to be corrected outside the DEMP program probably was done outside of the program. The basic item in hand is the present map and possibly saved intermediate maps. In this case, the map can be adjusted and modified. The offending region, or map, problems can be removed. For example, a misplaced water region could be put back where it belonged. If the regions' sizes were modified, or their boundaries moved, it would be necessary to somehow correctly adjust neighboring regions and their associated data. This modification could in all probability be considered a special mapping. The DEMP program is not set

up to do this. To insure that the corrections are correctly done will require a special correction/mapping program. It would be very helpful if this could be done visually and graphically, The simplest way to change a region is to drag its boundary. If this were done, then all the region characteristics would be automatically updated; as too would those of the neighboring regions. Proper, consistent correction of these maps is not trivial.

8.3 Program Improvements

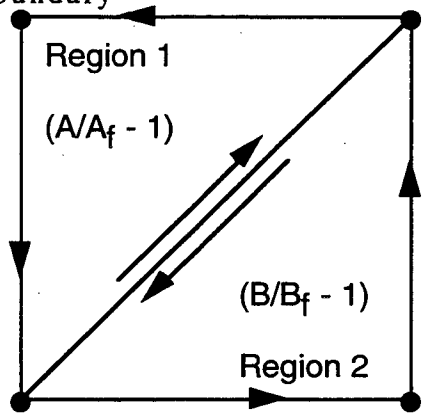
The present version of RLInt is complete in that it can input map data, calculate transformed maps, print reports, and output final maps. It has a number of parameter options that allow the user to select how the mapping will be performed and what output will be generated. However, there are areas in which future work would be beneficial. We discuss some of these areas in this section.

8.3.1 Elimination of Duplicate Boundary Points

The program RLInt uses the line integral algorithm in its simplest, most direct form. To evaluate the line integral around each region the program simply marches around the region's boundary points and accumulates the resultant translation vector contributions from each boundary line segment. Each region boundary is considered to be totally separate with a separate collection of boundary points.

This use of separate regions has the advantage of simplicity and has helped ensure that the line integral around each region is taken in the proper direction. It also makes for easy housekeeping in the program. However, it has the disadvantage of duplicating boundary points for regions with shared boundaries. For large maps with many regions, this is an inefficient way to process the data. In the present program version, the total number of points is directly proportional to the number of boundaries and the number of points on each boundary. In actual practice, the maps may have many less boundary points because of common boundaries.

Figure 40. Shared Region Boundary



Only the line segments that make up the boundaries are required to evaluate the line integral. If we know for each line segment which regions share it as a boundary segment, and if we can determine which directions to assign to the line segment, we can simply accumulate the translation vector contributions as needed. We can illustrate this with Figure 40, "Shared Region Boundary "

We see from this figure that the diagonal segment common to Region 1 and Region 2 is traversed in two directions. Currently these two regions are considered to be totally separate and require 6 points for their definition. This is unnecessary: only 4 points are required. Also, the line integral calculation along the diagonal is done twice, once for each boundary. This is not necessary; the integral can be evaluated once and the result multiplied by the coefficient $\left(A_f/A - B_f/B\right)$. We have assumed that the integral is evaluated from left to right, which is the correct direction for Region 1 but not for Region 2. The current areas are A and B and the target areas are A_f and B_f

This modification would also eliminate some of the checking done to ensure that coincident boundary points, the points on the segment common to both regions in Figure 40, stay coincident.

Implementation of this modification will reduce the calculations needed for evaluating the line integral, it will reduce the storage required for data points, but it will require careful housekeeping to insure that the integrals are properly done.

8.3.2 Illegal Boundary Crossings and Poor Convergence

The assumption is made here that the initial maps are furnished with no illegal boundary crossings, *i.e.*, that the starting map is correctly defined. This assumption has proved to be not necessarily true. Also, the line integral algorithm has the potential of creating transformed regions with boundaries that cross themselves. They have been referred to as regions with folded boundaries, or partially upside down regions. We have seen some mild examples of this problem in the maps shown and discussed in this report. In the worst case, the region is turned over and the area becomes completely negative. We note that a region can have positive, or negative area with only part of the region being upside down; the area calculation gives the net (algebraic) area of the region.

We have also seen that the convergence of the mapping can go from good to bad, and in fact to divergent. This can be associated with the creation of crossed boundaries, or regions lost due to negative boundaries. However, it is possible for the convergence to deteriorate without there being such problems.

The line integral algorithm is such that a negative area is not pushed back into a positive area. Instead, each point is reflected back through the origin - equivalent to a 180^{o} rotation. We need instead a reflection through an arbitrary straight line. These problems need to be addressed and a strategy developed and implemented for dealing with them.

Illegal Boundary Crossings

The region boundaries are polygons with straight line segments. It is possible to implement an algorithm that will detect when a polygonal region has a boundary that crosses itself. The translation vector that takes the map forward one full iteration vector is available in the present version of RLInt. Currently the regions are mapped sequentially by constructing the translation vectors for their boundary points and then applying the translation. It is rather easy to check at that time whether a region has any boundaries that cross. If a region has boundary problems, corrective action can be taken at that step in the process.

There are choices to be made in developing a corrective strategy. The simplest would be to simply not push that region's points so strongly. This, however, causes problems because of shared boundary segments; fixing one region's boundary can cause some very troublesome house keeping problems in updating other regions that have been already mapped.

A simple, and possibly best, solution is to stop the current mapping construction and readjust the fraction of the translation vector that will be taken for all the regions. After doing this, it is not necessary to reconstruct all the translation vectors. It is only necessary to change the coefficient that multiplies the vectors and then again apply them to the boundary points. This process can be repeated until the boundaries do not have crossing problems. At this point, a decision must be made as to whether to continue to use the current value of the coefficient, and if so for how long. This,

while possibly time consuming, is easily done with the current program. In a sense, it is analogous to the automatic step size adjustment used in the integration of ordinary differential equations, particularly systems that are classified as "stiff."

In any case, an algorithm that detects illegal region boundaries should be implemented, and a correction strategy should be decided upon and implemented. This will prevent the construction of maps with illegal boundary crossings and will also insure that furnished maps are in fact correctly constructed.

Non-Boundary Points on Boundary Line Segments

The points that are used to construct the translation vector are the boundary points (segments) of the regions. It is possible to map points that are not boundary points. There is presently in RLInt the capability of doing this. The data set `RLInt.dat` can contain these non-boundary points and they are mapped along with the boundary points. There is, however, no checking to determine whether these points land on a boundary segment. If they happen to coincide with a boundary point, they will stay attached to the boundary points. But if they happen to land on one of the line segments, they will presently not get the right push from that line segment. A point on a line segment should have the push generated from the value of the improper line integral along that segment. Currently RLInt only recognizes when a point lands on one of the line segment end points. This should eventually be corrected.

Convergence Monitoring

The convergence graphs shown in Section 7.3.1, " **RMS Convergence**," of this report show that the convergence rate can vary dramatically depending on what mapping and translation parameters are chosen. There are two monitoring functions, RMS and Hsum, that can be used to determine how well the process is converging. If the map being transformed has no anomalous regions, no zero areas, the behavior of these two functions is basically the same. However, the Hsum function penalizes regions that go negative and it also takes into account the relative areas of the regions. Thus, it probably is the better function to use for automatic monitoring of the mapping convergence.

The problem is two-fold. We have to decide when the process is not converging, or converging poorly, and we have to decide what to do when that is the case. Examples provide some guidance for determining the quality of the convergence. We see from these examples that when all is going well the convergence is monotonic and with a fairly constant slope on a log scale of convergence (Hsum or RMS) versus iterations. The actual rate (slope) of convergence depends to some extent on the number of regions and to a greater extent on the inital value of Hsum or RMS. However, the two properties monotonic convergence and constant slope seem to be a reasonable criterion for recognizing success.

Once it has been determined that the process is not converging well it becomes a task of deciding what to do. There is much information available at each step of the calculation. The RMS and Hsum functions furnish an average of the area errors for the regions. The individual region area errors, magnification errors, are available. The parameter options selected for the mapping (fixed point, binary interval, equal interval, unscaled or HH scaled) are available and can be easily changed to new values. Also, the mapping can be restarted at the current step, or it can be regressed one full step and restarted at that step; all without exiting the subroutine `rtrans` that is constructing the mapping sequence. It is also possible to drop regions that are causing a problem and to pick them up again later.

Summary

It will take some learning with test cases to determine what strategy is appropriate for both the convergence and boundary crossing problems. We should keep in mind that intermediate maps are available. The maps can be viewed as the mapping progresses and decisions can be based upon how these maps look. It is quite possible that the best correction for both the boundary crossing problem and the poor convergence problem can be obtained by user interaction that is based on looking at the maps as they are generated. If this is the case, some work will have to be devoted to establishing an interactive user interface with RLInt that is suitable for this type of corrective action.

References

(For electronically published references, WWW URL's are Uniform Reference Locators in the World Wide Web.)

1. D. Merrill. Cumulative FY1994 PAREP progress report. WWW URL: http://cedr.lbl.gov/mdocs/ftp/status/fy94par.asc.html

2. D. Merrill. Data required for prototype small-area analysis. Task Completion Report due 12/15/93. WWW URL: http://cedr.lbl.gov/mdocs/ftp/status/tr931215.asc.html

3. D. Merrill. Preparation of geographic map files for DEMP transformation. Task Completion Report due 1/15/94. WWW URL: http://cedr.lbl.gov/pdocs/tr940115/all.html

4. Close ER, Merrill DW and Holmes HH. Implementation of a new algorithm for Density Equalizing Map Projections (DEMP). LBL-35738, July 1995. WWW URL: http://cedr.lbl.gov/pdocs/tr940401/all.html

5. Satariano E, Reynolds P, Smith D, Goldman L. The Four County Study of childhood cancer incidence: Interim report I. Environmental Epidemiology and Toxicology Branch, California Department of Health Services. May 1990; to be published.

6. Reynolds P, Satariano E, Smith D. The Four County Study of Childhood cancer incidence: Interim report II. Environmental Epidemiology and Toxicology Program, California Department of Health Services, October 1991; to be published.

7. Schulman J., Selvin S.and Merrill D.W. 1988. Density Equalized Map Projections: A Method for Analyzing Clustering Around a Fixed Point. *Statistics in Medicine* 7:491-505.

8(a). Merrill, D.; Selvin, S.; Mohr, M. S. 1991. Analyzing Geographic Clustered Response. Report LBL-30954, (44 pages), June 1991. Invited paper presented at 1991 Joint Statistical Meetings of the American Statistical Association, Atlanta GA, August 1991. WWW URL: http://cedr.lbl.gov/pdocs/asa91/asa91.txt.html

8(b). Summary version (6 pages) in proceedings, Section on Statistics and the Environment, American Statistical Association, pp. 96-101, published June 1992. Merrill DW, Selvin S and Mohr MS; Analyzing geographic clustered response (44 pages); report LBL-30954, June 1991. WWW URL: http://cedr.lbl.gov/pdocs/asa91/short.txt.html

9. Sabir M. Gusein-Zade and Valdimir S. Tikunov, A new Technique for Constructing Continuous Cartograms, Cartography and Geographic Information Systems, Vol. 20, No. 3, 1993, 167-173

10. Merrill D, Selvin S and Close ER. 1994. Use of density equalizing map projections (DEMP) in the analysis of a reported childhood cancer cluster in McFarland, California, Presented at the Second Conference on Statistics and Computing in Disease Clustering, Vancouver, B.C., Canada, July 21-22, 1994. WWW URL: http://cedr.lbl.gov/pdocs/vancouver/vancouver.html

11. Merrill DW, Selvin S, Close ER and Holmes HH. Use of Density Equalizing Map Projections (DEMP) in the Analysis of Childhood Cancer in Four California Counties. Graphics presented at the 1995 CDC/ATSDR Symposium on Statistical Methods: Small Area Statistics in Public Health; Design, Analysis, Graphic and Spatial Methods; January 25-26, 1995; Atlanta, Georgia. Topic area 5: Detection of temporal and spatial trends in disease patterns (25 pages). WWW URL:
http://cedr.lbl.gov/pdocs/cdc9501/25graphics.html

12. Merrill D, Selvin S and Mohr MS. 1992. Density Equalizing Map Projections: Techniques and Applications. Report LBL-32640, July 1992. Presented at Workshop on Statistics and Computing in Disease Clustering, Stony Brook NY, July 23-24, 1992.

13. Merrill DW, Selvin S, Close ER and Holmes HH. Use of Density Equalizing Map Projections (DEMP) in the Analysis of Childhood Cancer in Four California Counties. Report LBL-36630, January 1995 (74 pages). WWW URL:
http://cedr.lbl.gov/pdocs/cdc9501/lbl36630.html

14. Merrill DW, Selvin S, Close ER and Holmes HH. Use of Density Equalizing Map Projections (DEMP) in the Analysis of Childhood Cancer in Four California Counties. Report LBL-36630 Rev. April 1995 (16 pages). Submitted to *Statistics in Medicine* 4/1/95. For copyright notice and reprint information see WWW URL:
http://cedr.lbl.gov/pdocs/cdc9504/reprint.html

15. Merrill DW. Density Equalizing Map Projections (Cartograms) in Public Health Applications; Dr.P.H. Dissertation, Department of Epidemiology, UC Berkeley School of Public Health; completion expected in December 1995.

Appendix A Integral Transformation and Correction Factors

A.1 Introduction

We derive in this Appendix the basic line integral that defines the mapping transformation. This derivation follows that of reference 9, but is given here in more detail. We also include the explicit evaluation of the line integral used in the program RLInt; this is not in reference 9 which simply gives the result as an indefinite integral. We also evaluate explicitly the area integral for the special case of the circle. This result has been useful in checking the program calculations and also in understanding the mapping. Finally we include in this Appendix a derivation of the local region area calculation. This too is done by means of a line integral.

A.2 Derivation of the Mapping Integrals

We have given a general overview of the mapping derivation in Section 2.2, **"Summary of the Algorithm."**

The original area of an arbitrarily small circular cell is given by

$$\Delta S = \pi R^2$$

and the radius of the cell is

$$R = \left(\sqrt{\Delta S}/\pi\right)$$

The present density within the cell is given by

$$\rho$$

Outside the cell the original density is

$$\rho_f$$

We wish to preserve the present density (ρ_f) outside the cell while expanding the cell to achieve target density ρ_f inside as well.

The target area of the cell is

$$(\Delta S)_f = \rho \Delta S/\rho_f$$

which means that the final (target) radius of the cell is

$$R_f = \sqrt{(\Delta S)_f/\pi}$$

Let $z = (x,y)$ denote the present coordinates of an arbitrary point in the (x,y) plane.

$z' = (x',y')$ is the center of the cell

the present coordinates of an arbitrary point outside the cell are

$$z = (x,y)$$

and the final coordinates of the arbitrary point outside the cell, to be determined, are

$$z_f = (x_f, y_f)$$

The radius vector from the center of the cell (x', y') to the present arbitrary point (x, y) is

$$z - z' = (x - x', y - y')$$

and the radius vector from the center of the cell (x', y') to the final position (x_f, y_f) of the arbitrary point outside the cell is given by

$$z_f - z' = (x_f - x', y_f - y')$$

The vector of translation of z is

$$z_f - z = (x_f - x, y_f - y)$$

The initial distance of z from z' is

$$r = |z - z'| = \sqrt{(x - x')^2 + (y - y')^2}$$

and the final distance of z_f from z' is

$$r_f = |z_f - z'| = \sqrt{(x_f - x')^2 + (y_f - y')^2}$$

We define ϕ, the initial angle of z relative to z', as

$$\phi = \tan^{-1}((y - y')/(x - x'))$$

We note here that it is necessary when obtaining the numerical value of this angle to take into account the signs of $(y - y')$ and $(x - x')$. This is done by using the Fortran intrinsic function atan2.

We also define ϕ_f the final angle of z_f relative to z', as

$$\phi_f = \tan^{-1}((y_f - y')/(x_f - x'))$$

A radial transformation that changes the density from ρ to ρ_f everywhere inside the cell, and maintains constant density ρ_f outside the cell, is:

$$\phi_f = \phi$$

$$r_f = (R_f/R)r \qquad \text{for } r \le R$$

$$r_f = \sqrt{r^2 + R_f^2 - R^2} \qquad \text{for } r \ge R$$

We let R be infinitesimally small so that only the equation for $r \ge R$ applies. Keeping in mind that $\phi_f = \phi$ we can write:

$$z_f - z = (z_f - z') - (z - z')$$

$$= (z_f - z')\left[|z_f - z'|/|z_f - z'| - 1\right]$$

$$= (z - z')\left[(r_f/r) - 1\right]$$

$$= (z - z')\left[\sqrt{r^2 + R_f{}^2 - R^2}/r - 1\right]$$

$$= (z - z')\left[\sqrt{1 + (R_f{}^2 - R^2)/r^2} - 1\right]$$

$$= (z - z')\left[\sqrt{1 + ((\Delta S)_f - \Delta S)/(\pi r^2)} - 1\right]$$

$$= (z - z')\left[\sqrt{1 + (\rho/\rho_f - 1)\Delta S/(\pi r^2)} - 1\right]$$

Ignoring terms of order $(\Delta S)^2$ allows us to write,

$$z_f - z = (z - z')\left[1 + (\rho/\rho_f - 1)\Delta S/(2\pi r^2) - 1\right]$$

$$z_f - z = (z - z')(\rho/\rho_f - 1)\Delta S/(2\pi r^2)$$

Now consider the plane to consist of an infinitely large number of circular cells, each having an infinitesimal area ΔS and centered at a different point z'. The individual cells (which overlap) are arranged so that the total area of the cells in a given region is equal to the area of that region.

For every point z' in the domain D, except z, we perform this transformation. The translation vector $z_f - z$ is obtained as the sum of the translation vectors from all the infinitesimal regions in D, namely over all values of z'. We now let dS be the limit of ΔS as ΔS goes to zero. The sum becomes an integral over the domain D and the resultant translation vector for $z_f - z$ is

$$(z_f - z) = \frac{1}{2\pi}\int_D \left(\frac{\rho - \rho_f}{\rho_f}\right)\frac{(z - z')}{r^2} dS$$

Assume that ρ has a constant value ρ_k over a sub domain D_k. Then the contribution from the subdomain D_k. to the vector of translation is

$$(z_f - z)_k = \frac{1}{2\pi}\left(\frac{\rho_k - \rho_f}{\rho_f}\right)\int_{D_k} \frac{z - z'}{r^2} dS$$

The equation above gives

$$(x_f - x)_k = \frac{1}{2\pi}\left(\frac{\rho_k - \rho_f}{\rho_f}\right)\int_{D_k} \frac{x - x'}{(x - x')^2 + (y - y')^2} dS$$

$$\left(y_f - y\right)_k = \frac{1}{2\pi}\left(\frac{\rho_k - \rho_f}{\rho_f}\right)\int_{D_k}\frac{y - y'}{(x-x')^2 + (y-y')^2}dS$$

We now have two area integrals, one for the x contribution and one for the y contribution. It is important to keep in mind that $dS = dx'dy'$ and that the integration is with respect to the variables x' and y'. The final density ρ_f, referred to elsewhere in this report as the target density, is the density which the infinitesimal regions are to achieve. This is the final density of the region D_k. If the subregions that make up the global region each have different final densities, then the final density in the above formulas should actually be written as $\rho_f^{(k)}$.

We know from Stoke's Theorem, Lass, *Vector and Tensor Analysis,* McGraw-Hill Book Inc., 1950, p.109, that given a vector $\vec{V}$, that we can write

$$\oint_C \vec{V} \bullet d\vec{s} = \int_D curl\vec{V} \bullet \vec{dS}$$

This says that integrating the tangential component of the vector $\vec{V}$ around the boundary C of the region D gives the same results as integrating the normal component of the vector $curl\vec{V}$ throughout the closed region D. This line integral is to be taken in the counter clockwise direction around the boundary C of the region D.

If we define the vectors

$$\vec{V} = \left(V_{x'}, V_{y'}, V_{z'}\right)$$

$$\left(curl\vec{V}\right)_{z'} = \frac{\partial V_{y'}}{\partial x'} - \frac{\partial V_{x'}}{\partial y'}$$

$$d\vec{s} = (dx', dy', dz')$$

$$\vec{dS} = (0, 0, dS)$$

$$(\vec{dS})_{z'} = dS$$

and if we choose the components of an arbitrary $\vec{V}$ to be

$$V_{x'} = \frac{1}{2}\ln\left[(x-x')^2 + (y-y')^2\right]$$

$$V_{y'} = 0$$

$$V_{z'} = 0$$

then from the definition of $curl\vec{V}$ we have

$$\left(curl\vec{V}\right)_{z'} = -\frac{\partial V_{x'}}{\partial y'} = \frac{(y-y')}{(x-x')^2 + (y-y')^2}$$

For this particular vector Stoke's Theorem gives:

$$\oint_C V_{x'} dx' = \oint_D \left(curl\vec{V}\right)_{z'} \left(\vec{dS}\right)_{z'} = -\oint_D \frac{\partial V_{x'}}{\partial y'} dS$$

$$\frac{1}{2}\oint_C \ln\left[(x-x')^2 + (y-y')^2\right] dx' = \int_D \frac{(y-y')}{(x-x')^2 + (y-y')^2} dS$$

In a similar manner if we define the arbitrary vector V to be

$$V_{x'} = 0$$

$$V_{y'} = -\frac{1}{2}\ln\left[(x-x')^2 + (y-y')^2\right]$$

$$V_{z'} = 0$$

then we have that

$$\left(curl\vec{V}\right)_{z'} = \frac{\partial V_{y'}}{\partial x'} = \frac{(x-x')}{(x-x')^2 + (y-y')^2}$$

and Stoke's Theorem gives

$$\oint_C V_{y'} dy' = \oint_D \left(curl\vec{V}\right)_{z'} \left(\vec{dS}\right)_{z'} = \oint_D \frac{\partial V_{y'}}{\partial x'} dS$$

$$-\frac{1}{2}\oint_C \ln\left[(x-x')^2 + (y-y')^2\right] dy' = \int_D \frac{(x-x')}{(x-x')^2 + (y-y')^2} dS$$

This allows us to write for the subdomain D_k

$$\left(x_f - x\right)_k = -\frac{\left(\rho_k - \rho_f\right)}{4\pi\rho_f} \oint_{C_k} \ln\left[(x-x')^2 + (y-y')^2\right] dy' = -\left(\frac{\rho_k - \rho_f}{4\pi\rho_f}\right) \oint_{C_k} \ln\left(r^2\right) dy'$$

and

$$\left(y_f - y\right)_k = +\frac{\left(\rho_k - \rho_f\right)}{4\pi\rho_f} \oint_{C_k} \ln\left[(x-x')^2 + (y-y')^2\right] dx' = +\left(\frac{\rho_k - \rho_f}{4\pi\rho_f}\right) \oint_{C_k} \ln\left(r^2\right) dx'$$

where $r^2 = (x-x')^2 + (y-y')^2$.

The total translation vector is obtained by summing the translation vectors from each of the subdomains D_k:

$$\left(x_f - x\right) = \sum_k \left(x_f - x\right)_k$$

$$\left(y_f - y\right) = \sum_k \left(y_f - y\right)_k$$

We keep in mind that ρ_k is the final, or target, density of the region D_k around which the integral is taken.

A.3 Line Integral Evaluations

The boundaries of the subregions are represented by line segments. For the purpose of obtaining a closed form solution of the line integral we assume in this section that a particular segment has length L. To obtain the contribution from the line segment L to the translation vector, we must evaluate the line integrals

$$\int_0^L \ln\left((x-x')^2+(y-y')^2\right)dx', \quad \int_0^L \ln\left((x-x')^2+(y-y')^2\right)dy'$$

Functionally there is no difference in the two integrals. In reference 9, an indefinite integral formula was given as the value of the integral; however, the indefinite integral formula must be made specific for us to evaluate the line integral numerically.

There is a problem with the above representation when the line along which we are integrating includes the point (x'=x, y'=y). At that point, the integrand $\ln\left((x-x')^2+(y-y')^2\right)$ becomes infinite. Analytically, the line integral remains finite, but it cannot be evaluated numerically.

This problem can be avoided by using the arc length ds as the integration variable. We can write the above line integrals as

$$\int_0^L \ln\left((x-x')^2+(y-y')^2\right)\cos(\alpha)ds, \quad \int_0^L \ln\left((x-x')^2+(y-y')^2\right)\sin(\alpha)ds$$

where α is the angle of the line segment relative to the x axis, measured in a counter clockwise direction. Since the angle α is fixed for each line segment, we shall ignore it in the derivation below. We must, however, remember that the result is multiplied by either $\sin\alpha$ or $\cos\alpha$.

There are, in general, four cases to examine: the point (x,y) is not on the line segment L; the point (x,y) is one of the line segment end points, *i.e.*, (x,y) = (x_1,y_1), or (x,y) = (x_2,y_2); and the point (x,y) is an arbitrary point along the line, but not one of the end points. When the point (x,y) is on the line (three of the four cases) the integral is an improper integral since the integrand becomes negatively infinite. However, the integral does have a finite value. We derive below the formulas used in RLInt for each of these four cases.

A.3.1 Point z not on Line

Start with the basic integral along the line, $z=(x,y)\neq(x',y')$,

$$\int_0^L \ln\left((x-x')^2+(y-y')^2\right)ds$$

where ds is the differential arc length along the line segment.

We have that

$$x' = x_1 + s\cos\alpha$$

$$y' = y_1 + s\sin\alpha$$

where s is the distance from the left end point of the line segment $(x_1, y_1)$to the current point (x', y'). This allow us to write the integral as

$$\int_0^L \ln\left((x - x_1 - s\cos\alpha)^2 + (y - y_1 - s\sin\alpha)^2\right) ds$$

where L is the line segment length.

Complete the square and collect terms to express the integral as

$$\int_0^L \ln\left(a^2 + 2sb + s^2\right) ds$$

where $a^2 = (x - x_1)^2 + (y - y_1)^2$

$$b = -\left[(x - x_1)\cos\alpha + (y - y_1)\sin\alpha\right]$$

The above integral can be written as

$$\int_0^L \ln\left(a^2 - b^2 + (s + b)^2\right) ds$$

Make the change of variables $\xi = s + b$ with $ds = d\xi$ and define $\beta^2 = a^2 - b^2$. This give the formula

$$\int_b^{L+b} \ln\left(\beta^2 + \xi^2\right) ds = \left[\xi \ln\left(\beta^2 + \xi^2\right) - 2\xi + 2\tan^{-1}\left(\frac{\xi}{\beta}\right)\right]_b^{L+b}$$

This later result can be found in tables of integrals. Note that we have assumed that β is constant, which is true for a line segment.

If we perform the indicated algebra we can write

$$\int_0^L \ln\left((x - x')^2 + (y - y')^2\right) ds = \ln\left(\frac{t_2}{t_1}\right) - 2L + 2\beta\tan^{-1}\left(\frac{L\beta}{\beta^2 + \xi_1\xi_2}\right)$$

where

$$t_1 = \left(\beta^2 + \xi_1^{\,2}\right)^{\xi_1} \;;\; \xi_1 = b$$

$$t_2 = \left(\beta^2 + \xi_2^{\,2}\right)^{\xi_2} \;;\; \xi_2 = b + L$$

To arrive at the above expression we have used the trigonometric identity

$$\tan^{-1} u_2 \pm \tan^{-1} u_1 = \tan^{-1}\left[\frac{u_2 \pm u_1}{1 \mp u_2 u_1}\right]$$

Also, we note that numerically we must use the function $\text{atan2}(L\beta, \beta^2 + \xi_1\xi_2)$ to insure that we get the correct quadrant sign. The principal valued inverse tangent function atan does not always give a suitable answer for the above expression.

Numerically there is a problem with the evaluation of t_2/t_1; it is easy to get overflow, *i.e.*, infinite results. A somewhat better result can be obtained. We can write

$$t_2 = \left(\beta^2 + (L+b)^2\right)^{L+b}$$

$$t_1 = \left(\beta^2 + b^2\right)^b$$

and perform the algebra and substitute $\beta^2 + b^2 = a^2$ to arrive at

$$\frac{t_2}{t_1} = \left[\frac{a^2 + L(L+2b)}{a^2}\right]^b \left(a^2 + L(L+2b)\right)^L$$

It is this expression that is used in the program RLInt when evaluating the line integral along a segment. This does not cure the overflow problem. That has been done by scaling the problem so that the largest value of any point coordinate is 1 in absolute value. This keeps the maximum distance less than $2\sqrt{2}$. When this is done there is no overflow problem when run on machines with the standard exponent range for 32 bit numbers.

A.3.2 Point z at the Left End of the Line Segment

In this case $z = (x, y) = (x_1, y_1)$ and the integral becomes

$$\int_0^L \ln\left((-s\cos\alpha)^2 + (-s\sin\alpha)^2\right) ds$$

This integral reduces to the improper integral

$$\int_0^L \ln s^2 ds = 2\int_0^L \ln s ds$$

We can now write

$$2\int_0^L \ln s ds = 2\left(\int_0^1 \ln s ds + \int_1^L \ln s ds\right) \quad 0 \le L \le 1$$

Both integrals are given in tables of integrals as

$$2\int_1^L \ln s ds = 2[s\ln s - s]_1^L = 2[L\ln L - L + 1]$$

$$2\int_0^1 \ln s ds = -2$$

Adding these two quantities gives us the value of the line integral as

$$2[L\ln L - L] \quad \text{for } 0 \le L \le 1$$

We note that

$$\lim_{L\to 0}(L\ln L) = 0$$

A.3.3 Point z at the Right End of the Line Segment

In this case $z=(x,y)=(x_2,y_2)$ and the integral becomes

$$\int_0^L \ln\left((x_2 - x_1 - s\cos\alpha)^2 + (y_2 - y_1 - s\sin\alpha)^2\right) ds$$

We note that

$$x_2 = x_1 + L\cos\alpha,\ y_2 = y_1 + L\sin\alpha$$

Substituting this into the above integrand and doing the algebra gives

$$\int_0^L \ln\left((L-s)^2\right) ds$$

Change the variable to

$$\xi = L - s\ ;\ -\mathrm{ds} = \mathrm{d}\xi$$

and the integral can be written, after a little algebra, as

$$\int_0^L \ln\xi^2 ds = 2\int_0^L \ln\xi d\xi$$

Functionally this is the same as before, so again we get the same answer.

$$2[L\ln L - L] \quad \text{for } 0 \le L \le 1$$

A.3.4 Point z Between the Left and Right End of the Line Segment

Let $L = L_1 + L_2$ with the point in question at the right end of L_1, which is also the left end of L_2. We have that the integral can be considered to be the sum of the integral along L_1, plus the integral along L_2. This means that the value of the integral along the segment can be written as

$$L = 2(L_1\ln L_1 - L_1) + 2(L_2\ln L_2 - L_2) = 2(L_1\ln L_1 + L_2\ln L_2 - L)$$

with

$$L = L_1 + L_2,\quad 0 \le L,\quad 0 \le L_1,\quad 0 \le L_2$$

The current version of RLInt does not check whether the point z lies on a line segment, so this expression is not currently used.

A.4 Special Case: Domain is a Circle of Radius R

A.4.1 Some Examples

Here we derive the result stated in Section 7.1. The original integral representation of the mapping does not map a simple circle in the manner expected. This has been noted above in Section 7.1, "**Circle Expansion With and Without Scaling.**" We derive here (1) the final radius r_f and the translation vector $r_f - r$ of a point initially at distance r from the origin, due to the expansion, or contraction of a circle of radius R centered at the origin. We also find (2) the final radius and translation vector by explicitly evaluating the line integral for a circle. The first of these translation vectors we call the *Expected* translation vector and the second one, from the area integral, we call the *Calculated* translation vector.

In Table 3, "Mapping a Circle," we compare these translation vectors for a circle of radius R = 2 and for a circle of radius R = 1. The areal magnification factor M is defined as

$$M = \frac{\rho}{\rho_f} = \frac{A_f}{A}$$

where ρ is the initial density of the circle and ρ_f is the target density of the circle. The correction factor C is simply the ratio of the *Expected* to the *Calculated* value for the translation vector $(r_f - r)$.

Table 3. Mapping a Circle

R	M	r	r_f Expected	r_f Calculated	$r_f - r$ Expected	$r_f - r$ Calculated	$C = \frac{(r_f - r)_{expt.}}{(r_f - r)_{calc.}}$
2	2	1	1.414	1.5	0.414	0.5	0.828
2	2	2	2.828	3.0	0.828	1.0	0.828
2	2	4	4.472	4.5	0.472	0.5	0.944
1	0	0.5	0.0	0.25	-0.5	-0.25	2.000
1	0	1	0.0	0.5	-1.0	-0.5	2.000
1	0	2	1.732	1.75	-0.268	-0.25	1.072
1	0.5	0.5	0.354	0.375	-0.146	-0.125	1.168
1	0.5	1	0.707	0.75	-0.293	-0.25	1.172
1	0.5	2	1.871	1.875	-0.129	-0.125	1.032
1	2	0.5	0.707	0.75	0.207	0.25	0.828
1	2	1	1.414	1.5	0.414	0.5	0.828
1	2	2	2.236	2.25	0.236	0.25	0.944
1	8	0.5	1.414	2.25	0.914	1.75	0.522
1	8	1	2.828	4.5	1.828	3.5	0.522
1	8	2	3.317	3.75	1.317	1.75	0.753

We see from this table that the values calculated from the area integral are larger near the edge of the circle than those that we get from the simple expansion, or contraction of the circle.

A.4.2 The *Expected* Translation Vector

The *Expected* quantities are arrived at by the simple expansion, or contraction, of a circle. In order to produce the correct areas we need to satisfy the following relations. The original area of the circle is $A = \pi R^2$; the final area, $A_f = M\pi R^2$. For points inside the circle, $r \le R$, we require that $r_f / \left(\sqrt{M} R\right) = r/R$. For points outside the circle, $r \ge R$, we require that $\pi r_f^2 - M\pi R^2 = \pi r^2 - \pi R^2$.

From this we obtain the *Expected* final positions as:

$$r_f = r\sqrt{M} \qquad r \le R$$

$$r_f = \sqrt{r^2 + (M-1)R^2} \qquad r \ge R$$

$$r_f = R\sqrt{M} \qquad r = R$$

The *Expected* translation vectors are:

$$r_f - r = r\left(\sqrt{M} - 1\right) \qquad r \le R$$

$$r_f - r = \sqrt{r^2 + (M-1)R^2} - r \qquad r \ge R$$

$$r_f - r = R\left(\sqrt{M} - 1\right) \qquad r = R$$

A.4.3 The *Calculated* Translation Vector

The line integral can be evaluated explicitly in the special case where the contour of integration is a circle.

Evaluation of the Area Integral for a Circle:

There are two distinct cases that we treat separately. The first is when the point z for which the translation vector is being constructed lies outside the circle and the other is when the point z lies inside the circle.

The Point z is Outside the Circle on the X Axis:

The boundary of integration is a circle. We place the origin at the center of the circle and assume that the point z lies on the positive x-axis (x > R, y = 0). To evaluate the line integral we transform x' and y' to polar coordinates:

$$x' = R\cos(\theta)$$

$$y' = R\sin(\theta)$$

$$dL = Rd\theta$$

$$dx' = dL\,dx'/dL = -R\sin(\theta)d\theta$$

$$dy' = dL\,dy'/dL = \; R\cos(\theta)d\theta$$

$$x - x' = x - R\cos(\theta)$$

$$y - y' = \; -R\sin(\theta)$$

Using the above we can write

$$r^2 = (x - x')^2 + (y - y')^2$$

$$r^2 = (x - R\cos(\theta))^2 + (R\sin(\theta))^2$$

Perform the algebra and simplify to obtain

$$r^2 = x^2(1 - 2a\cos(\theta) + a^2)$$

$$a = R/x \quad (a < 1)$$

This gives us

$$\ln(r^2) = 2\ln(x) + \ln(1 - 2a\cos(\theta) + a^2)$$

The line integrals around the circle R are equal to

$$\oint_C \ln(r^2)dy' = R\int_{-\pi}^{\pi} \ln(r^2)\cos(\theta)d\theta$$

$$\oint_C \ln(r^2)dx' = R\int_{-\pi}^{\pi} \ln(r^2)\sin(\theta)d\theta$$

The X-component of Translation Vector:
We start from the basic relation

$$(x_f - x) = -\frac{1}{4\pi}\frac{\rho - \rho_f}{\rho_f}\oint_C \ln(r^2)dy'$$

$$(x_f - x) = -\frac{R}{4\pi}\frac{\rho - \rho_f}{\rho_f}\int_{-\pi}^{\pi} \ln(r^2)\cos(\theta)d\theta$$

$$(x_f - x) = -\frac{R}{4\pi}\frac{\rho - \rho_f}{\rho_f}\int_{-\pi}^{\pi} \left(2\ln(x) + \ln(1 - 2a\cos(\theta) + a^2)\right)\cos(\theta)d\theta$$

The first term of this integral vanishes because

$$\int_{-\pi}^{\pi} \cos(\theta)d\theta = 0$$

This leaves the integral

$$(x_f - x) = -\frac{R}{4\pi}\frac{\rho - \rho_f}{\rho_f}\int_{-\pi}^{\pi} \left(\ln(1 - 2a\cos(\theta) + a^2)\right)\cos(\theta)d\theta$$

Because $\cos(-\theta)=\cos(\theta)$, the integrand is an even function of θ and we can write

$$\left(x_f - x\right) = -\frac{R}{2\pi}\frac{\rho-\rho_f}{\rho_f}\int_0^{\pi}\left(\ln\left(1-2a\cos(\theta)+a^2\right)\right)\cos(\theta)d\theta$$

From Eq. 865.74 in *Dwight's Tables of Integrals* we obtain

$$\int_0^{\pi}\left(\ln\left(1-2a\cos(\theta)+a^2\right)\right)\cos(\theta)d\theta = -\pi a \quad \text{for } 0<a<1$$

Thus we arrive at

$$\left(x_f - x\right) = -\frac{R}{2\pi}\frac{\rho-\rho_f}{\rho_f}(-\pi a) \quad \text{where } a = R/x$$

which becomes

$$\left(x_f - x\right) = \frac{R^2}{2x}\frac{\rho-\rho_f}{\rho_f} = \frac{R^2}{2x}(M-1)$$

where

$$M = \frac{\rho}{\rho_f} = \frac{\text{original density of } D}{\text{target density of } D} = \frac{\text{target area of } D}{\text{original area of } D} = \frac{A_f}{A}$$

The Y-component of Translation Vector:
We again start from the basic integral

$$\left(y_f - y\right) = +\frac{1}{4\pi}\frac{\rho-\rho_f}{\rho_f}\oint \ln\left(r^2\right)dx'$$

which becomes

$$\left(y_f - y\right) = -\frac{R}{4\pi}\frac{\rho-\rho_f}{\rho_f}\int_{-\pi}^{\pi}\ln\left(r^2\right)\sin(\theta)d\theta$$

Because the integrand is an odd function of θ, the integral from $-\pi \le \theta \le \pi$ is zero.

$$\left(y_f - y\right) = 0$$

The Point z is Inside the Circle on the X Axis:
We have

$$(x < R, y = 0)$$

We start from the expression for r which is

$$r^2 = \left(x - R\cos(\theta)\right)^2 + \left(R\sin(\theta)\right)^2$$

Perform the algebra and simplify to obtain

$$r^2 = R^2\left(1-2b\cos(\theta)+b^2\right)$$

with

$$b = x/R \quad (b<1)$$

Taking the logarithm we get

$$\ln(r^2) = 2\ln(R) + \ln(1 - 2b\cos(\theta) + b^2)$$

The X-component of the Translation Vector:

We again start with the integral that gives the x component of the translation vector.

$$(x_f - x) = -\frac{R}{4\pi}\frac{\rho - \rho_f}{\rho_f}\int_{-\pi}^{\pi}\ln(r^2)\cos(\theta)d\theta$$

Substituting we obtain

$$(x_f - x) = -\frac{R}{4\pi}\frac{\rho - \rho_f}{\rho_f}\int_{-\pi}^{\pi}\left(2\ln(R) + \ln(1 - 2b\cos(\theta) + b^2)\right)\cos(\theta)d\theta$$

This integral is functionally the same as the previous X-component translation vector integral. Instead of $\ln(x)$ we have $\ln(R)$ and instead of $a = R/x \quad (a<1)$, we have $b = x/R \quad (b<1)$. Thus we can write

$$(x_f - x) = -\frac{R}{2\pi}\frac{\rho - \rho_f}{\rho_f}(-\pi b) \text{ where } b = x/R$$

which becomes

$$(x_f - x) = \frac{x}{2}\frac{\rho - \rho_f}{\rho_f} = \frac{x}{2}(M-1)$$

The Y-component of the Translation Vector:

The integral can be written as

$$(y_f - y) = -\frac{R}{4\pi}\frac{\rho - \rho_f}{\rho_f}\int_{-\pi}^{\pi}\ln(r^2)\sin(\theta)d\theta$$

Because the integrand is an odd function of θ, the integral from $-\pi \le \theta \le \pi$ is zero. This gives

$$(y_f - y) = 0$$

The *Calculated* Results for a Circle:

In summary, for a point lying on the positive X-axis,

$$(x_f - x) = \frac{R^2}{2x}(M-1) \qquad \text{if } x \ge R$$

$$(x_f - x) = \frac{x}{2}(M-1) \qquad \text{if } x \le R$$

$$(x_f - x) = \frac{R}{2}(M-1) \qquad \text{if } x = R$$

The symmmetry of a circle allows us to write these results as follows.

The *Calculated* translation vectors:

$$(r_f - r) = \frac{R^2}{2r}(M-1) \qquad \text{if } r \geq R$$

$$(r_f - r) = \frac{r}{2}(M-1) \qquad \text{if } r \leq R$$

$$(r_f - r) = \frac{R}{2}(M-1) \qquad \text{if } r = R$$

We use these results to get the *calculated* radius r_f:

$$r_f = \frac{2r^2 + R^2(M-1)}{2r} \qquad \text{if } r \geq R$$

$$r_f = \frac{r(M+1)}{2} \qquad \text{if } r \leq R$$

$$r_f = \frac{R}{2}(M+1) \qquad \text{if } r = R$$

A.4.4 The Correction factor

In order to compare the *Expected* and the *Calculated* values for the mapping of a circle we define a *Correction Factor* C as the ratio $\frac{(r_f - r)_{\text{expected}}}{(r_f - r)_{\text{calculated}}}$.

The *Correction Factor* becomes:

$$C = \frac{\left(\sqrt{M} - 1\right)}{(M-1)/2} \qquad \text{if } r \leq R$$

$$C = \frac{\sqrt{r^2 + (M-1)R^2} - r}{\frac{R^2}{2r}(M-1)} \qquad \text{if } r \geq R$$

The first expression, which is independent of r and R, is the HH scaling factor described in this report. Where the HH scaling has been applied, the factor $C = \frac{\left(\sqrt{M_k} - 1\right)}{(M_k - 1)/2}$ has been applied to *all* translation vectors $(r_f - r)$ associated with a given subdomain D_k. An improved correction (which has not been implemented) might use the same expression for points inside D_k, and a more complicated form for points outside D_k:

$$C_k = \frac{\sqrt{r_k^{\,2} + (M_k - 1)R_k^{\,2}} - r_k}{\frac{R_k^{\,2}}{2r_k}(M_k - 1)}$$

where πR_k^2 is the area of subdomain D_k, and r_k is the distance of the point z from the centroid of D_k.

A.5. Calculation of Region Areas

The calculation of the translation vector requires that we evaluate the contribution from each region D_k. We also need to know the area of each of these regions. It is convenient to accumulate that area from the boundary line segments. We derive below formulas that express the region area as a line integral.

We again use Stoke's Theorem

$$\int_D curl\vec{V} \bullet d\vec{S} = \oint_c \vec{V} \bullet d\vec{s}$$

and the notation of section A.2.

This time we chose the vector to be

$$\vec{V} = V_{x'}\vec{i} + V_{y'}\vec{j} + V_{z'}\vec{k} = -y'\vec{i} + 0\vec{j} + 0\vec{k}$$

where $\vec{i}$, $\vec{j}$, and $\vec{k}$ are unit vectors.

For this vector we have

$$curl\vec{V} = \vec{k}\left(\frac{\partial V_{y'}}{\partial x'} - \frac{\partial V_{x'}}{\partial y'}\right) = 1\vec{k}$$

This gives

$$\int_D dS = \text{area of D} = -\oint_C y'\cos(\alpha)ds = -\sum_k \int_0^{L_k} y'\cos(\alpha)ds$$

The line integral along any boundary segment $L = L_k$ can be written as

$$-\int_0^L (y_1 + y)\cos(\alpha)ds = -\int_0^L y_1\cos(\alpha)ds - \int_0^L y\cos(\alpha)ds$$

If we define

$$y = s\sin\alpha$$

these two integrals can be evaluated to give

$$-y_1 L\cos\alpha - \frac{L^2}{2}\sin\alpha\cos\alpha = -y_1(x_2 - x_1) - \frac{1}{2}(x_2 - x_1)(y_2 - y_1)$$

Summing over all the n line segments k which define a subdomain boundary yields:

$$area = \frac{1}{2}\sum_{k=1}^{n}(x_{1k} - x_{2k})(y_{1k} + y_{2k})$$

If we choose the vector instead to be

$$\vec{V} = V_{x'}\vec{i} + V_{y'}\vec{j} + V_{z'}\vec{k} = 0\vec{i} + x'\vec{j} + 0\vec{k}$$

the same analysis gives the area as

$$area = \frac{1}{2}\sum_{k=1}^{n}(x_{1k} + x_{2k})(y_{2k} - y_{1k})$$

In RLInt, the area contribution from line segment k is evaluated at the same time as the contribution of that segment to the displacement vector of all the points z.

Because the n segments bounding any polygon are connected, we can define

$$x_k = x_{1k} = x_{2,k-1} \quad (k = 2,3,...n)$$
$$x_1 = x_{11} = x_{2n}$$

and similar expressions for y.

With this substitution, both formulas for area given above reduce to:

$$area = \frac{1}{2}\sum_{k=1}^{n} x_k y_{k+1} - x_{k+1} y_k$$

Appendix B Description of RLInt I/O Files

B.1 Parameter and cl3 Data Files

The basic RLInt data files are:

File Name	Contents
`RLInt.dat`	the map data input file
`RLInt.par`	the parameter input file

The parameters set in file `RLInt.par` were described above in Section 6.3, "**Program Options**." The file `RLInt.dat` is described in the next section below. These data files are read with FORTRAN list directed read statements. This is, essentially, field free and the data that are numeric values simply appear as standard Fortran constants. The header lines and the skipped lines are alphanumeric information. This information is not read by RLInt but is useful to identify the data and organize the data file.

The data on each line are read from left to right; the fields are separated by one or more blanks, or a comma. Numeric data fields can, in principle, be skipped by furnishing an empty field that is terminated with a comma. This, however, is not advisable because no attempt is made to define a default setting for each parameter.

The list is read until completed. This means that when two items are to be read, the two items can appear on one line, or on more than one line. However, it is best to follow the example furnished here by having one set of data on each line.

The data appearing to the right of the required variable values are regarded as a comment. No comments may be placed between data fields.

B.1.1 Map Data File `RLInt.dat` (cl3 format)

The program reads one map data file `RLInt.dat`. The data file shown below in Figure 41,"Input Data File RLInt.dat," is a complete, valid data set. It has been taken from the first 4 regions of the 44-region Vermont case (a complete listing of this 44-region Vermont case is given in Appendix C.2.2, "**File `RLInt.dat`**" below). In Figure 41, we have changed the number of regions to 4. This region number is simply used as an identifier for the region. The program does not care that the last region is 44 and that no intermediate region numbers exist.

The first two lines are header cards that can contain any information. The next line furnishes the requested error limit. For this version of RLInt this is the RMS error which will be tolerated in the magnification M. RLInt will stop when the RMS value of (M - 1) is less than this value. Next comes the number of regions, in this case, four. This is followed by the region data, in this case, four sets of data.

Each region data set begins with a header card containing arbitrary information. It is customary (but not required) to include the geocodes (state, county, *etc.*) identifying the region.

The next two lines of region data are used by RLInt and are important. They contain

```
target area,    population, original area
present area,   present magnification
```

Their meaning is described below. The information after these numerical fields is a comment.

The region data set then has a line that indicates the number of boundary points in this region (excluding the repeated end point). This is followed by the x,y coordinates for each of the boundary points. Their order is important (see discussion below).

After the complete set of boundary data sets (in this case, four), there are two lines of comments which can contain any alphanumeric information. The program skips them.

Next come the data for the non-boundary points. These begin with one comment line, for example, "Non-boundary points." Then comes the number of non-boundary points and after that the x,y coordinates for each of the points (in the same coordinate system). The example shown contains zero non-boundary points.

The final data line for this type of data is a comment card, for example, the comment "End of non-boundary points."

After reading these data, the program continues to process data lines looking for some additional specific information. The program searches for text specified in the source code. For the present version of RLInt it is

```
parameter (info0 = 'tract information follows:')
parameter (info1 = 'longitude')
parameter (info2 = 'latitude' )
parameter (info3 = 'number of map units')
```

The program expects to find this information and copies it to some of the output files. If this information is not found, there could be an error when output files are written.

B.1.2 Comments on the Region Data

There are some flags associated with the region data. The program takes different actions depending on the value of the region data it reads. In the data set there are lines containing `target area` and `present area`. We describe here the possible values of these data fields.

When the region has a negative `target area` (usually -999), it is flagged as a region to be skipped. These regions are pushed, but their boundaries never generate a push. For example, in the Vermont data set given below the first region is so flagged. This region is the total region, the sum of the individual regions, and it would constitute double counting for it to contribute to the transformation vector. The `population` of this area, though not used by RLInt, is also equal to -999.

For this region the `original area` and the `present magnification` are also specified as negative values. A region that originally had a positive magnification can, because of being pushed during the mapping, later have a negative area. Regions for which the magnification becomes negative are always skipped when constructing the transformation vector. Should a region with a magnification that has become negative later become positive, it would normally be re-included in the construction of the mapping vector. The purpose of specifying the original area as negative is to ensure that this area is permanently excluded from contributing to the mapping.

As mentioned earlier, the `population` data field is not used by the program RLInt. However, a value must be present. The `original area` of the region is used in constructing the Hsum function. See Section 6.4, "**Convergence Criteria**," for a definition of that function. The `present area`, in the initial map, agrees with the `original area`. If the program does exactly zero mappings because `itstp` in the

parameter data is zero, then this area on the output files will agree with the `original area`, if the data have been correctly furnished.

`Present area` is included in `RLInt.dat`, but is not used by the program; present areas are recalculated from the region boundaries. The `present magnification`, defined as (`target area / current area`) for each of the regions, becomes 1 when the mapping has perfectly converged. In practice a converged mapping has values that meet the prescribed error criterion. This quantity must be present, but the program uses its own calculated value. For the identity mapping, `itstp` = 0, the program output files have a value that agrees with the input value from this data file.

The region boundary data are furnished as a set of points that are the end points of the line segments that represent the region. The order of these points is important. The algorithm is basically a line integral around the region boundary. The program is blind and simply assumes that the boundary points are furnished in the correct order, from the first line of data input to the last line. If the regions points are furnished in a counter clockwise direction around the boundary, the current area will be positive. If they are furnished in a clockwise direction around the boundary the current area will be negative. If they represent a boundary that crosses itself, the current area will be correctly calculated and could be positive, negative or zero: it is an algebraic sum. Note that in the Vermont example the area furnished for the first region is negative. The region boundary points are furnished in a clockwise direction. The initial point should not be repeated in `RLInt.dat`; the polygon is closed automatically by the program.

Figure 41. Input Data File `RLInt.dat`

```
DEMP input file in CLOSE3 format
vt_500k2_hex.cl3 Mar 10, 1994
 1e-6   Convergence RMS error Single precision convergence
 4        5 # poly, # codes (seq,st,cy,...poly)
Region     1    0    0    0    1
 -9.9900000E+02 -9.9900000E+02 -2.4621430E+06 target area UNSPEC., pop., orig. area
 -2.4621430E+06 -9.9900000E+02 pres. area, magnification UNSPEC.
       30      number of points
       11     -998
       -1    -1273
     -180     -1270
     -358     -1266
     -501     -1265
     -644     -1263
     -638      -942
     -632      -621
     -675      -381
     -718      -140
     -699       149
     -679       439
     -699       605
     -718       771
     -646      1018
     -574      1266
     -322      1266
      -70      1266
      201      1269
      473      1273
      627      1272
      781      1271
      647       895
      513       519
      429       420
      345       322
      277       103
      209      -116
      116      -420
       23     -723
```

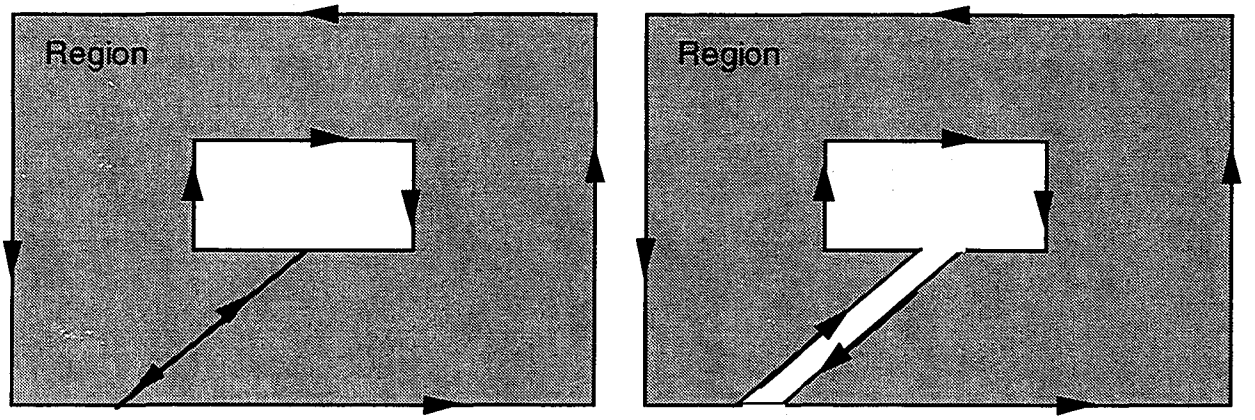
```
Region     2    50     1     0     1
 4.3502570E+04  9.0367021E+03  5.9097500E+04 target area, pop., orig. area
 5.9097500E+04  7.3611528E-01 pres. area, pres. magnification
       6       number of points
    -532        193
    -384        -52
    -383        145
    -381        343
    -530        391
    -679        439
Region     3    50     1     0    -1
 1.0403520E+04  2.1611021E+03  1.4133000E+04 target area, pop., orig. area
 1.4133000E+04  7.3611528E-01 pres. area, pres. magnification
       6       number of points
    -321        202
    -261         62
    -235        119
    -209        176
    -295        259
    -381        343
Region    44    50    27     0    -4
 9.7773094E+04  2.0310211E+04  1.0881800E+05 target area, pop., orig. area
 1.0881800E+05  8.9850098E-01 pres. area, pres. magnification
       6       number of points
      13        -170
    -182        -223
     -80        -473
      23        -723
     116        -420
     209        -116
End of the boundary points, skip this line
skip this line
Non-Boundary points
  0          # of these points
End of non-boundary points
tract information follows:
 -72.47850     longitude in degrees at (x=0)
  43.87635     latitude  in degrees at (y=0)
       10 number of map units in one kilometer
     -718 minimum x (map units)
      781 maximum x (map units)
    -1273 minimum y (map units)
     1273 maximum y (map units)
```

The basic region data thave been described above. There is an attempt to check that the amount of data will not overflow program storage. Also, the runs made have been with array bounds checking, to check further that no storage violations occur. In principle, RLInt checks the number of regions and the number of data points to see whether the allocated storage is sufficient. However, the program does not check for all possible errors in `RLInt.dat` and `RLInt.par`. It is very important that the input data be verified, to the extent that is practical. One easy, quick check is to take exactly one iteration and verify that the output data sets agree with the input data sets with regard to points, areas, etc.

Each region boundary must be completely defined as a separate polygon. RLInt is ineffiecient in that common boundaries are stored twice. Furthermore, the line integrals corresponding to each line segment are evaluated twice, which is unnecessary. In the future, RLInt could be modified to reduce storage requirements and execution time.

A region boundary should be stored counter clockwise for positive area or clockwise for negative area. A hole in a region can be obtained by making a cut to the inner (hole) boundary, traversing the hole boundary clockwise, and then traversing the same cut back out to the outer boundary. The line integral along the cut will cancel out. An example is shown below.

Figure 42. How to Use a Cut for Generating a Doughnut Region



The inner boundary appears to be traversed in a direction that is opposite to the outer boundary. However, it can seen from the figure that the (net) region boundary is traversed in one direction only, counter clockwise. The area calculated for this region will be that of the shaded region in the figure and the line integrals along the cut will cancel. The location of the cut is immaterial as long as it joins the inner and outer boundaries.

A discussion of machine precision, which also pertains to the data input, is given in Section 6.4.2, "**Discussion of Machine Precision.**" We note here that the map is normally scaled by the program to avoid numerical difficulties. If the map is *not* scaled (see the description of the parameter file data), then it must fit in a rectangle whose diagonal is not larger than about 11 or 12. The scaling does not center the origin at the map center; it simply fits the map into a square of side ±1. This means that the maps must be reasonably well centered about the origin of their coordinates. This is a minor restriction that can be handled in a number of ways. Map coordinates can be in any right-handed Cartesian coordinate system. That is, x and y must have the same units, with x increasing to the right and y increasing upward. The units are arbitrary.

B.2 Output Files

B.2.1 Description of Output files

The principal goal of program RLInt is to transform a given initial map into a final map. We have discussed above the basic input file that defines the initial map. We now give descriptions of the output files that can be generated during the run. We note that the generation of output files is controlled by the settings of parameters in the parameter data file `RLInt.par`. A complete example file is included in Appendix C.2.1, "**File `RLInt.par`.**"

For each file that the program can produce, we summarize below its purpose and contents.

B.2.2 Basic Output Files

These files are the basic, minimal output from a mapping run. The files described in Sections B.2.3 and B.2.4 can be optionally obtained by setting the parameters in `RLInt.par`. See Section 6.3, "**Program Options.**"

File Name	Contents
`RLInt.out`	This file is created if `idisk = 0`. Created after the call to `rtrans`, this file is the current state of the boundary points and the non-boundary points. It can be thought of as the image that the mapping produces of the input file `RLInt.dat`. To continue the mapping from the present state, copy this file to the file `RLInt.dat` and run the program. The output file `RLInt.out` has essentially the same format as `RLInt.dat` file listed in Section C.2.2.
`RLInt.plot`	This file is created if `idisk = 0`. This file contains basically the same information as `RLInt.out`, furnished in a comma delimited form that is convenient for input into a plotting program. The regions can be plotted from this file. This file is not suitable as a RLInt data input file; it does not have the required format. The areas and magnifications are omitted from the region data in this file. Only the coordinates of the region boundary points and the non-boundary points are included. In `RLInt.plot`, the boundary points include closure points so that when plotted as line segments there is no gap between the last point and the first point.
`RLInt.out.msg`	This file is always produced. As the iterations are performed, regions may be dropped (negative magnification) or recovered (if magnification returns to positive). This information is logged in this file. Other information pertaining to the progress of the mapping can also be logged here.
`stdout`	This is the standard output file and is always produced. It includes both a summary and a log of the mapping run. The contents of this file can vary, depending on the setting of the parameters in the program parameter file. This file can be redirected to a disk file and saved, if that is desired. When this is done, the program can be run as a batch, or background process, if the parameters are set so that no input from the standard input file `stdin` is requested (`iprmpt = 0`).

B.2.3 Internally Stored Output Files

The next set of files is written when information is stored internally. If requested, the program will save information at each step of the mapping for printing after the mapping subroutine `rtrans` has completed. It is not necessary to store this information. A parameter can be set to skip that step, saving the time and space required to save the results. See Section 6.3.6, "**Output/Reports**," for a description of the parameter settings.

The following tables are created if `icheck` = 1 and `itable` ≥ 1.

File Name	Contents
`RLInt.out.sum`	When information is stored internally during the run (`icheck` = 1), this file can be requested as program output. It contains, in table form, information at each iteration step including the RMS relative error and HSum and, for each region, the magnification, the current area (`arak`), the target area (`ara0k`), and the region's relative error ($M_k - 1$). Only areas with positive magnifications are included in the RMS sum $\left(\sqrt{\frac{1}{n}\sum_{k=1}^{n}(M_k-1)^2}\right)$. The tables are generated for each step in the mapping. If there are ten mappings, there are ten tables, each containing information for all the regions. A map with many regions and many steps requires a lot of disk space. If disk space is insufficient, the program warns the user of that fact. Following these tables, there is an iteration history, giving the x-y coordinates for every boundary point and non-boundary point. The initial coordinates and the values after the last eight iterations are listed. An example is given in Section C.3.3
`RLInt.out.plot`	This file is produced if information has been stored internally during the run (`icheck` = 1). It contains for each iteration step: an iteration history for each region, including the region number, the iteration number, the region magnification, present area and target area, relative error $M_k - 1$, and (for the map as a whole) the RMS relative error, HSum, the target map area and the current total map area. It is comma delimited and can be used to plot a region's behavior as a function of the iteration count. It also contains the region boundaries from iteration 0 and the last few iterations. The difference between `RLInt.out.plot` and `RLInt.out.sum` is that the former gives the behavior of each region as a function of the mapping steps. `RLInt.out.sum` has a table for each step that describes all the regions for that step. `RLInt.out.plot` (Section C.3.4) contains a table for each region, listing the iteration history for that region. `RLInt.out.sum` (Section C.3.3) contains a table for each iteration, listing the values from each region for that step.

B.2.4 Intermediate Mapping Files

The next files are obtained when the program parameters have been set to produce disk file output at intermediate steps in the mapping (`idisk` > 0). They are not created unless requested. When `idisk` > 0, the files `RLInt.out` and `RLInt.plot` files are not created; they would be redundant with `RLInt.*.NNNN`. See Section 6.3.6, "**Output/Reports**," for details on setting the parameters.

File Name	Contents
`RLInt.sum`	This file is created if `isum` = 1. For each region there is a table showing the state of the region for each requested output step of the iteration. The quantities tabulated are the iteration counts, current area (`areak`), target area (`area0k`), current magnification, and relative magnification error (`eps`). There is a summary of the RMS magnification error including the iteration count, the number of dropped regions and the number of regions remaining that contribute to the transformation vector. The function Hsum is also given. The number of dropped regions includes the regions with target area -999, which never enter into the translation vector. This is a comma delimited table; the tables can be plotted to see the convergence of any particular region as the iteration proceeds. `RLInt.sum` contains information for the complete run and is not completed (written) until the current mapping is complete. This is written from the subroutine `rtrans` just before it returns control to the main program. This file collects as a summary the information written in each of the NNNN files described below.
`RLInt.sum.NNNN`	This file is created if `isum` = 1 and `idisk` > 0. The file shows the state of all the regions at step `NNNN` of the mapping. The file is comma delimited and corresponds to the file `RLInt.sum` after step `NNNN`. For each region it includes the area (`areak`), the target area (`area0k`), the magnification (`Magk`) and the region magnification relative error (`Mrelerk`). The `NNNN` files are written after steps which are specified by the parameter `idisk`. See Section 6.3.6, "**Output/Reports**', for a description of the parameters that control this writing.
`RLInt.out.NNNN`	This file is created if `idisk` > 0. This is the `RLInt.out` file at iteration `NNNN`. The file is suitable for use as an input file `RLInt.dat`. It is a little different from `RLInt.out` in that it contains the pushes that produced the current points. If necessary, one could go back one iteration by subtracting these values from the current points. The "push" data follow and are not used by the program. Tract summary information, also not used by RLInt, is written at the end of the file.

RLInt.plot.NNNN	This file is created if idisk > 0. This is the RLInt.plot file at iteration NNNN. The file can be used to make plots of the regions after iteration NNNN. The "pushes" are also included in this file. It is a comma delimited file.
RLInt.*.now	RLInt.out.now and RLInt.plot.now are written if inow > 0; also, RLInt.sum.now is written if inow > 0 and isum >0. They are copies of the lastest version of RLInt.*.NNNN and are updated as the run proceeds. These files are readable during the run and can furnish a dynamic window into the progress of the mapping; for example, they could be used interactively to display the regions as the run progresses.
*.zzz	Files that end in .zzz are temporary binary files, which are deleted at the end of the run. Should they be left over from a run that aborts, they can be deleted, though that is not necessary.

Appendix C. Example of Complete Run

C.1 Description of Example Run

In this appendix the Vermont pop80 44-region map is provided as a complete example. The parameter file `RLInt.par` and the initial map data file `RLInt.dat` are given in their entirety. The program output from this run includes the files `RLInt.std`, `RLInt.msg`, `RLInt.out.sum`, and `RLInt.out.plot`. This run requested intermediate output at each iteration step (`idisk = 1`), which is not presented in this report. The RMS error criterion was specified as 5×10^{-5}, which was achieved in seven iterations. This run took 24.6 seconds on a Sun SPARC 10. The RMS function was equal to 0.703 initially and 0.000019 after seven iterations.

This example can be used to test the program RLInt. It also furnishes an example that can be used with other DEMP algorithms, for comparison with RLInt.

C.2 Data input Files

C.2.1 File `RLInt.par`

The parameter file specifies `iprint = 10` to produce intermediate output; `itable = 10` to create the internally stored tables `RLInt.out.sum` and `RLInt.out.plot`; `maxit0 = 20` to have a maximum of 20 fixed point iterations; `maxit = 0` to have no partial step transformations; `istp = 16` to stop after 16 iterations, (provided convergence is not obtained sooner). The mapping has HH scaling (`iscale = 1`); the data region is scaled into the ± 1 square to avoid overflow (`iara = 1`); `icheck = 1` to save results for files `RLInt.out.sum` and `RLInt.out.plot`. There is no resetting of the mapping done (`ireset` should be left at 0 because not fully implemented). The Russian style of partial transformation mapping is selected (`iciset = 0`); however, these steps are skipped because `maxit = 0`. Intermediate disk output is requested at every iteration step (`idisk = 1`); a final summary file of this output at the end of the run is requested (`isum = 1`); no dynamic (overwritten) output files are requested (`inow = 0`). The program will erase all RLInt files up to NNNN = 16 (`nfdel = 16`). No interactive prompting is requested; the program just reads the data and executes (`iprmpt = 0`).

Data Parameter File `RLInt.par`

```
Parameter File: RLInt.par for program test23.f
 10  iprint   0 no print, > 0 is print out Print Flag
 10  itable   table to RLInt.out.sum, RLInt.out.plot
 20  maxit0   iteration max.  Fixed point
  0  maxit    iteration max.  Transformation
 16  itstp    Stop at exactly iteration itstp
  1  iscale   0 Russian,  1 HHH  Transformation scaling
  1  iara     0 no scale, 1 scale Data Region scaling
  1  icheck   0 no save,  1 save  Result saving flag (in core)
  0  ireset   0 float, no push; 1 reset iteration on Neg. Mag.
  0  iciset   0 Russian, 1 ci = 1/2
  1  idisk    0 no intermediate disk output, <>0  output mod(idisk)
  1  isum     0 no disk summary file, 1 write output (idisk.ne.0)
  0  inow     0 no current iteration files, >0 write output/overwrites
 16  nfdel    <0 no file deletes, >=0 delete RLInt previous run files
  0  iprmpt   0 no interactive prompting,<>0 prompting
End of parameter data
```

C.2.2 File `RLInt.dat`

This file contains the data for the Vermont pop80 44-region map, in the cl3 format described in the body of this report. The file is complete and was used to generate the Vermont pop80 44-region results described in the body of this report.

Data file `RLInt.dat` DEMP input file in CLOSE3 format

```
vt_500k2_hex_pop80.cl3
 5e-5   Convergence RMS error
   44    5 # poly, # codes (seq,st,cy,...poly)
Region  1  0  0  0  1
 -999.0000 -999.0000 -2462143.  targ area, pop, orig area
 -2462143. -999.0000  current area and magnification
   30   number of points
    11  -998
   -1 -1273
  -180 -1270
  -358 -1266
  -501 -1265
  -644 -1263
  -638  -942
  -632  -621
  -675  -381
  -718  -140
  -699   149
  -679   439
  -699   605
  -718   771
  -646  1018
  -574  1266
  -322  1266
  -70   1266
   201  1269
   473  1273
   627  1272
   781  1271
   647   895
   513   519
   429   420
   345   322
   277   103
   209  -116
   116  -420
    23  -723
Region  2  50  1  0  1
 43502.57  9036.702  59097.50  targ area, pop, orig area
 59097.50  0.7361153  current area and magnification
    6   number of points
  -532  193
  -384  -52
  -383  145
  -381  343
  -530  391
  -679  439
Region  3  50  1  0 -1
 10403.52  2161.102  14133.00  targ area, pop, orig area
 14133.00  0.7361153  current area and magnification
    6   number of points
  -321  202
  -261   62
  -235  119
  -209  176
  -295  259
  -381  343
Region  4  50  1  0 -2
 69868.76  14513.70  94915.50  targ area, pop, orig area
 94915.50  0.7361153  current area and magnification
    6   number of points
  -532  193
  -679  439
  -699  149
  -718  -140
  -551 -96
  -384  -52
Region  5  50  1  0 -3
 17785.28  3694.501  24161.00  targ area, pop, orig area
 24161.00  0.7361153  current area and magnification
    6   number of points
  -383  145
  -384  -52
  -323    5
  -261   62
  -321  202
  -381  343
Region  6  50  3  0  1
 64912.10  13484.06  82468.50  targ area, pop, orig area
 82468.50  0.7871139  current area and magnification
    6   number of points
  -501 -1265
  -358 -1266
  -318  -978
  -277  -690
  -461  -977
  -644 -1263
Region  7  50  3  0 -1
 5712.873  1186.724  7258.000  targ area, pop, orig area
 7258.000  0.7871139  current area and magnification
    6   number of points
  -455  -656
  -277  -690
  -296  -666
  -314  -642
  -473  -632
  -632  -621
Region  8  50  3  0 -2
 89897.45  18674.22  114211.5  targ area, pop, orig area
 114211.5  0.7871138  current area and magnification
    6   number of points
  -638  -942
  -644 -1263
  -461  -977
  -277  -690
  -455  -656
  -632  -621
Region  9  50  5  0  1
 13138.90  2729.318  22680.00  targ area, pop, orig area
 22680.00  0.5793167  current area and magnification
    6   number of points
  282  611
  513  519
  300  653
   88  788
   70  746
   52  704
Region 10  50  5  0 -1
 23599.05  4902.183  40736.00  targ area, pop, orig area
 40736.00  0.5793168  current area and magnification
    6   number of points
  198  513
   52  704
   74  536
   96  368
  220  345
  345  322
Region 11  50  5  0 -2
 35294.58  7331.672  60924.50  targ area, pop, orig area
 60924.50  0.5793167  current area and magnification
    6   number of points
  282  611
   52  704
  198  513
  345  322
  429  420
  513  519
Region 12  50  5  0 -3
 52206.87  10844.83  90118.00  targ area, pop, orig area
 90118.00  0.5793167  current area and magnification
    6   number of points
  300  653
  513  519
  469  758
  426  998
  257  893
   88  788
Region 13  50  7  0  1
 108316.1  22500.29  35476.50  targ area, pop, orig area
 35476.50  3.053179  current area and magnification
    6   number of points
  -320  739
  -359  832
  -539  801
  -718  771
  -499  709
  -280  647
Region 14  50  7  0 -1
 153025.3  31787.64  50120.00  targ area, pop, orig area
 50120.00  3.053179  current area and magnification
    6   number of points
  -480  543
  -679  439
  -530  391
  -381  343
  -331  495
  -280  647
```

```
Region 15 50 7 0 -2
 80135.27 16646.34 26246.50 targ area, pop, orig area
 26246.50 3.053179 current area and magnification
   6 number of points
 -539 801
 -359 832
 -481 884
 -602 937
 -660 854
 -718 771
Region 16 50 7 0 -3
 214702.6 44599.73 70321.00 targ area, pop, orig area
 70321.00 3.053179 current area and magnification
   6 number of points
 -480 543
 -280 647
 -499 709
 -718 771
 -699 605
 -679 439
Region 17 50 9 0 1
 21137.01 4390.749 97019.00 targ area, pop, orig area
 97019.00 0.2178647 current area and magnification
   6 number of points
 469 758
 513 519
 647 895
 781 1271
 603 1134
 426 998
Region 18 50 9 0 -1
 9253.693 1922.251 42474.50 targ area, pop, orig area
 42474.50 0.2178647 current area and magnification
   6 number of points
 603 1134
 781 1271
 627 1272
 473 1273
 449 1135
 426 998
Region 19 50 11 0 1
 71803.52 14915.60 79505.50 targ area, pop, orig area
 79505.50 0.9031265 current area and magnification
   6 number of points
 -220 921
 -81 1010
 -328 1138
 -574 1266
 -467 1049
 -359 832
Region 20 50 11 0 -1
 58262.50 12102.75 64512.00 targ area, pop, orig area
 64512.00 0.9031265 current area and magnification
   6 number of points
 -328 1138
 -81 1010
 -76 1138
 -70 1266
 -322 1266
 -574 1266
Region 21 50 11 0 -2
 37402.98 7769.646 41415.00 targ area, pop, orig area
 41415.00 0.9031265 current area and magnification
   6 number of points
 -481 884
 -359 832
 -467 1049
 -574 1266
 -588 1101
 -602 937
Region 22 50 13 0 1
 22206.93 4613.000 16729.00 targ area, pop, orig area
 16729.00 1.327451 current area and magnification
   6 number of points
 -660 854
 -602 937
 -588 1101
 -574 1266
 -646 1018
 -718 771
Region 23 50 15 0 1
 44760.91 9298.095 54634.00 targ area, pop, orig area
 54634.00 0.8192868 current area and magnification
   6 number of points
 -114 675
  52 704
 -15 857
 -81 1010
 -181 828
 -280 647
Region 24 50 15 0 -1
 26848.85 5577.257 32771.00 targ area, pop, orig area
 32771.00 0.8192868 current area and magnification
   6 number of points
 -320 739
 -280 647
 -181 828
 -81 1010
 -220 921
 -359 832
Region 25 50 15 0 -2
 9106.372 1891.648 11115.00 targ area, pop, orig area
 11115.00 0.8192868 current area and magnification
   6 number of points
 -15 857
  52 704
  70 746
  88 788
   3 899
 -81 1010
Region 26 50 17 0 1
 53283.30 11068.43 84585.00 targ area, pop, orig area
 84585.00 0.6299380 current area and magnification
   6 number of points
   0 30
 209 -116
 152 126
  96 368
 -57 272
 -209 176
Region 27 50 17 0 -1
 36390.58 7559.341 57768.50 targ area, pop, orig area
 57768.50 0.6299381 current area and magnification
   6 number of points
 152 126
 209 -116
 277 103
 345 322
 220 345
  96 368
Region 28 50 17 0 -2
 19791.39 4111.226 31418.00 targ area, pop, orig area
 31418.00 0.6299380 current area and magnification
   6 number of points
   0 30
 -209 176
 -235 119
 -261 62
 -26 -27
 209 -116
Region 29 50 19 0 1
 32066.51 6661.111 55315.50 targ area, pop, orig area
 55315.50 0.5797020 current area and magnification
   6 number of points
   3 899
  88 788
 257 893
 426 998
 172 1004
 -81 1010
Region 30 50 19 0 -1
 43075.92 8948.074 74307.00 targ area, pop, orig area
 74307.00 0.5797021 current area and magnification
   6 number of points
 178 1132
 426 998
 449 1135
 473 1273
 201 1269
 -70 1266
Region 31 50 19 0 -2
 37697.45 7830.814 65029.00 targ area, pop, orig area
 65029.00 0.5797021 current area and magnification
   6 number of points
 172 1004
 426 998
 178 1132
 -70 1266
 -76 1138
 -81 1010
Region 32 50 21 0 1
 102431.0 21277.78 84194.50 targ area, pop, orig area
 84194.50 1.216599 current area and magnification
   6 number of points
 -508 -337
 -384 -52
 -551 -96
 -718 -140
 -675 -381
 -632 -621
```

```
Region 33 50 21 0 -1
 82874.13 17215.28 68119.50 targ area, pop, orig area
 68119.50 1.216599 current area and magnification
    6 number of points
 -407 -422
 -632 -621
 -473 -632
 -314 -642
 -248 -433
 -182 -223
Region 34 50 21 0 -2
 95576.66 19853.95 78560.50 targ area, pop, orig area
 78560.50 1.216599 current area and magnification
    6 number of points
 -508 -337
 -632 -621
 -407 -422
 -182 -223
 -283 -138
 -384 -52
Region 35 50 23 0 1
 105689.3 21954.62 71336.50 targ area, pop, orig area
 71336.50 1.481560 current area and magnification
    6 number of points
 -143 355
  96 368
 -92 507
 -280 647
 -331 495
 -381 343
Region 36 50 23 0 -1
 62020.32 12883.36 41861.50 targ area, pop, orig area
 41861.50 1.481560 current area and magnification
    6 number of points
 -143 355
 -381 343
 -295 259
 -209 176
 -57 272
  96 368
Region 37 50 23 0 -2
 84509.66 17555.02 57041.00 targ area, pop, orig area
 57041.00 1.481560 current area and magnification
    6 number of points
 -92 507
  96 368
  74 536
  52 704
 -114 675
 -280 647
Region 38 50 25 0 1
 79167.30 16445.26 82890.00 targ area, pop, orig area
 82890.00 0.9550886 current area and magnification
    6 number of points
  11 -998
  23 -723
 -127 -707
 -277 -690
 -139 -982
 -1 -1273
Region 39 50 25 0 -1
 98627.71 20487.74 103265.5 targ area, pop, orig area
 103265.5 0.9550887 current area and magnification
    6 number of points
 -180 -1270
 -1 -1273
 -139 -982
 -277 -690
 -318 -978
 -358 -1266
Region 40 50 27 0 1
 54005.30 11218.41 60106.00 targ area, pop, orig area
 60106.00 0.8985010 current area and magnification
    6 number of points
  13 -170
  209 -116
 -26 -27
 -261 62
 -222 -81
 -182 -223
Region 41 50 27 0 -1
 5904.949 1226.623 6572.000 targ area, pop, orig area
 6572.000 0.8985010 current area and magnification
    6 number of points
 -146 -683
 -314 -642
 -296 -666
 -277 -690
 -127 -707
  23 -723
Region 42 50 27 0 -2
 19783.64 4109.617 22018.50 targ area, pop, orig area
 22018.50 0.8985010 current area and magnification
    6 number of points
 -283 -138
 -182 -223
 -222 -81
 -261 62
 -323 5
 -384 -52
Region 43 50 27 0 -3
 68190.84 14165.14 75894.00 targ area, pop, orig area
 75894.00 0.8985010 current area and magnification
    6 number of points
 -80 -473
 -182 -223
 -248 -433
 -314 -642
 -146 -683
  23 -723
Region 44 50 27 0 -4
 97773.09 20310.21 108818.0 targ area, pop, orig area
 108818.0 0.8985010 current area and magnification
    6 number of points
  13 -170
 -182 -223
 -80 -473
  23 -723
  116 -420
  209 -116
End of the boundary points, skip this line
skip this line
Non-Boundary points
0   # of these points
End of non-boundary points
geoarea information follows:
 seq no,   3 geocodes, no of polygons,
 pop, target area, present area, mag-1, hterm
 1 50 1 0 4
  29406.00  141560.1  192307.0  -0.2638846  7.3886313E-03
 2 50 3 0 3
  33345.00  160522.4  203938.0  -0.2128861  4.7691595E-03
 3 50 5 0 4
  25808.00  124239.4  214458.5  -0.4206832  2.6608735E-02
 4 50 7 0 4
  115534.0  556179.3  182164.0  2.053179  0.1021529
 5 50 9 0 2
  6313.000  30390.71  139493.5  -0.7821354  0.1590808
 6 50 11 0 3
  34788.00  167469.0  185432.5  -9.6873462E-02  7.8258879E-04
 7 50 13 0 1
  4613.000  22206.93  16729.00  0.3274510  5.4882118E-04
 8 50 15 0 3
  16767.00  80716.14  98520.00  -0.1807132  1.5949785E-03
 9 50 17 0 3
  22739.00  109465.3  173771.5  -0.3700620  1.5343217E-02
 10 50 19 0 3
  23440.00  112839.9  194651.5  -0.4202979  2.4090962E-02
 11 50 21 0 3
  58347.00  280881.8  230874.5  0.2165995  3.6160119E-03
 12 50 23 0 3
  52393.00  252219.3  170239.0  0.4815599  1.0822490E-02
 13 50 25 0 2
  36933.00  177795.0  186155.5  -4.4911265E-02  1.5967226E-04
 14 50 27 0 5
  51030.00  245657.8  273408.5  -0.1014990  1.2732196E-03
 0.2178647  minimum magnification
 3.053179  maximum magnification
 0.00000E+00 hsum (polygons)
 0.00000E+00 log10(hsum) (polygons)
 0.3582322  hsum (geoareas)
 -0.4458354  log10(hsum) (geoareas)
 511456.0  total population
 2462143.  original land area
 2462143.  target land area
 2462143.  present land area
 0.00000E+00 present water area
 2462143.  present total area
-2462143.  external boundary area incl islands
 14 number of (land) geoareas
 43 normal land polygons (normal geocodes, orig area gt 0)
  0 holes in land polygons (normal geocodes, orig area lt 0)
 43 total land polygons
  0 water polygons (zero geocodes, orig area gt 0)
  1 ext boundary incl islands (zero geocodes, orig area lt 0)
 44 total polygons incl ext boundary
 -72.47850  longitude in degrees at (x=0)
 43.87635  latitude  in degrees at (y=0)
 10 number of map units in one kilometer
 -718 minimum x (map units)
 781 maximum x (map units)
 -1273 minimum y (map units)
 1273 maximum y (map units)
```

C.3 Data output Files

C.3.1 File `RLInt.std`

This is output that appears on standard output. In this run, it was redirected to the file `RLInt.std`.

```
Copyright (c) 1995 The Regents of the University of California
BEGIN RLInt
Initialize

OPEN file RLInt.par
  Parameter File: RLInt.par for program test23.f
  Debug Printing Flag,          iprint =    10
  Convergence Table Flag,       itable =    10
  Fixed point iteration max.    maxit0 =    20
  Transformation iteration max. maxit  =     0
  Stop at iteration             itstp  =    16
  Transformation scaling.       iscale =     1
  Data scaling flag             ira    =     1
  Data saving flag              icheck =     1
  Negative magnification flag   ireset =     0
  Coefficient flag              iciset =     0
  Disk files flag               idisk  =     1
  Disk files flag               isum   =     1
  Disk files flag               inow   =     0
  No.of <>.NNNN files to delete nfdel  =    16
  Prompting flag                iprmpt =     0
CLOSE file: RLInt.par

OPEN file RLInt.dat
  DEMP input file in CLOSE3 format
    1          Average density rho0 (not used)
  Convergence relative error =            5.0E-05
  Number of regions,    maxd =                 44
  Number of non-boundary points maxib =         0
CLOSE file: RLInt.dat

DELETE All RLInt output files:
  RLInt.out
  RLInt.out.msg
  RLInt.out.now
  RLInt.out.plot
  RLInt.out.sum
  RLInt.plot
  RLInt.plot.now
  RLInt.sum
  RLInt.sum.now
  The above list of files has been deleted
DELETE files:
  RLInt.sum.0000  thru RLInt.sum.0016
  RLInt.out.0000  thru RLInt.out.0016
  RLInt.plot.0000 thru RLInt.plot.0016
  The above list of files has been deleted

BEGIN rtrans
LOG File: RLInt.out.msg
  Skipping region    1 Target area  -999
Top of Fixed point loop:     it0 =   0
  Top of Transformation loop:it  =   0
    iteration  =  0
    ci         =  1.00000
    Calculate areas, magnifications, errors
     Check for stopping:
     RMS              =   7.034E-01
     DMhsum           =   3.582E-01
     Total Target  Area =   2.462E+06
     Total Present Area =   2.462E+06
     Output File: RLInt.sum.0000
     Output File: RLInt.out.0000
     Output File: RLInt.plot.0000
     Update all Boundary point pushes
     All Non-boundary point pushes
     All points a & b have been updated
  Bottom of transformation loop
Bottom of fixed point loop

Top of Fixed point loop:     it0 =   1
  Top of Transformation loop:it  =   0
    iteration  =  1
    ci         =  1.00000
    Calculate areas, magnifications, errors
     Check for stopping:
     RMS              =   1.231E-01
     DMhsum           =   5.795E-03
     Total Target  Area =   2.462E+06
     Total Present Area =   2.344E+06
     Output File: RLInt.sum.0001
     Output File: RLInt.out.0001
     Output File: RLInt.plot.0001
     Update all Boundary point pushes
     All Non-boundary point pushes
     All points a & b have been updated
  Bottom of transformation loop
Bottom of fixed point loop

Top of Fixed point loop:     it0 =   2
  Top of Transformation loop:it  =   0
    iteration  =  2
    ci         =  1.00000
    Calculate areas, magnifications, errors
     Check for stopping:
     RMS              =   1.101E-02
     DMhsum           =   7.054E-05
     Total Target  Area =   2.462E+06
     Total Present Area =   2.460E+06
     Output File: RLInt.sum.0002
     Output File: RLInt.out.0002
     Output File: RLInt.plot.0002
     Update all Boundary point pushes
     All Non-boundary point pushes
     All points a & b have been updated
  Bottom of transformation loop
Bottom of fixed point loop

Top of Fixed point loop:     it0 =   3
  Top of Transformation loop:it  =   0
    iteration  =  3
    ci         =  1.00000
    Calculate areas, magnifications, errors
     Check for stopping:
     RMS              =   2.402E-03
     DMhsum           =   3.627E-06
     Total Target  Area =   2.462E+06
     Total Present Area =   2.462E+06
     Output File: RLInt.sum.0003
     Output File: RLInt.out.0003
     Output File: RLInt.plot.0003
     Update all Boundary point pushes
     All Non-boundary point pushes
     All points a & b have been updated
  Bottom of transformation loop
Bottom of fixed point loop

Top of Fixed point loop:     it0 =   4
  Top of Transformation loop:it  =   0
    iteration  =  4
    ci         =  1.00000
    Calculate areas, magnifications, errors
     Check for stopping:
     RMS              =   6.802E-04
     DMhsum           =   2.697E-07
     Total Target  Area =   2.462E+06
     Total Present Area =   2.462E+06
     Output File: RLInt.sum.0004
     Output File: RLInt.out.0004
     Output File: RLInt.plot.0004
     Update all Boundary point pushes
     All Non-boundary point pushes
     All points a & b have been updated
  Bottom of transformation loop
Bottom of fixed point loop

Top of Fixed point loop:     it0 =   5
  Top of Transformation loop:it  =   0
    iteration  =  5
    ci         =  1.00000
    Calculate areas, magnifications, errors
     Check for stopping:
     RMS              =   2.035E-04
     DMhsum           =   2.225E-08
     Total Target  Area =   2.462E+06
     Total Present Area =   2.462E+06
     Output File: RLInt.sum.0005
     Output File: RLInt.out.0005
     Output File: RLInt.plot.0005
     Update all Boundary point pushes
     All Non-boundary point pushes
     All points a & b have been updated
  Bottom of transformation loop
Bottom of fixed point loop
```

```
Top of Fixed point loop:     it0 =   6
 Top of Transformation loop:it =   0
  iteration =   6
  ci        =   1.00000
  Calculate areas, magnifications, errors
  Check for stopping:
  RMS               =   6.187E-05
  DMhsum            =   1.905E-09
  Total Target  Area =   2.462E+06
  Total Present Area =   2.462E+06
  Output File: RLInt.sum.0006
  Output File: RLInt.out.0006
  Output File: RLInt.plot.0006
  Update all Boundary point pushes
  All Non-boundary point pushes
  All points a & b have been updated
 Bottom of transformation loop
Bottom of fixed point loop

Top of Fixed point loop:     it0 =   7
 Top of Transformation loop:it =   0
  iteration =   7
  ci        =   1.00000
  Calculate areas, magnifications, errors
  Check for stopping:
  RMS               =   1.897E-05
  DMhsum            =   1.721E-10
  Total Target  Area =   2.462E+06
  Total Present Area =   2.462E+06
  Output File: RLInt.sum.0007
  Output File: RLInt.out.0007
  Output File: RLInt.plot.0007
GOTO DONE


Transformation Converged:
 RMS Magnification error 0th step   =   7.03E-01
 RMS Magnification error all regions =   1.90E-05
 DM h function                      =   1.72E-10
 Requested error                    =   5.00E-05
 Total Target  area                 =   2.46E+06
 Total Current area                 =   2.46E+06
 Number of non-skipped regions      =   43
 Transformation iterations taken    =   0
 Fixed point    iterations taken    =   7
 Mapping coefficient ci             =   1.00E+00
  Output File: RLInt.sum
END rtrans
 Convergence Table on File: RLInt.out.sum
 Plot Tables       on File: RLInt.out.plot
END RLInt
```

C.3.2 File `RLInt.msg`

This file contains messages that indicate which regions were lost due to negative magnifications. If the regions were later recovered, that information would be reported here. For this run, no regions were lost.

```
RMS Magnification error 0th step   =   7.03E-01
RMS Magnification error all regions =   1.90E-05
DM h function                      =   1.72E-10
Requested  error                   =   5.00E-05
Total Target  area                 =   2.46E+06
Total Current area                 =   2.46E+06
Number of non-skipped regions      =   43
Transformation iterations taken    =   0
Fixed point    iterations taken    =   7
Mapping coefficient ci             =   1.00E+00

      END FILE: RLInt.out.msg
```

C.3.3 File `RLInt.out.sum`

This file was written from data stored internally during the run by specifying `icheck` = 1. The first section describes each region after each iteration. The second section provides initial boundary point coordinates, and values after each of the seven steps. In this example there were no non-boundary points.

```
File: RLInt.out.sum
Input  scale flag  iara  =      1
Target RMS error:  relerr = 5.00E-05
Achieved RMS error  reler  = 1.90E-05
DM h function       dmhsum = 1.72E-10
Target total  area  aretot = 2.46E+06
Current total area  arctot = 2.46E+06
Scaling flag        iscale =      1
Convergence Table:
j it0 it   ci     RMS       DMhsum    trgtArea  curntArea
  k  armag      arak          ara0k        reler
-------------------------------------------------------------
0  0  0 1.0000  7.03E-01  3.58E-01  2.46E+06  2.46E+06
  1   0.0004 -2462142.750000    -999.000000  0.00E+00
  2   0.7361    59097.500000   43502.570312 -2.64E-01
  3   0.7361    14132.995117   10403.519531 -2.64E-01
  4   0.7361    94915.492188   69868.757812 -2.64E-01
  5   0.7361    24161.001953   17785.279297 -2.64E-01
  6   0.7871    82468.515625   64912.101562 -2.13E-01
  7   0.7871     7257.984375    5712.873047 -2.13E-01
  8   0.7871   114211.500000   89897.453125 -2.13E-01
  9   0.5793    22679.982422   13138.900391 -4.21E-01
 10   0.5793    40736.007812   23599.050781 -4.21E-01
 11   0.5793    60924.488281   35294.578125 -4.21E-01
 12   0.5793    90118.023438   52206.871094 -4.21E-01
 13   3.0532    35476.492188  108316.101562  2.05E+00
 14   3.0532    50120.003906  153025.296875  2.05E+00
 15   3.0532    26246.525391   80135.273438  2.05E+00
 16   3.0532    70321.007812  214702.578125  2.05E+00
 17   0.2179    97019.007812   21137.009766 -7.82E-01
 18   0.2179    42474.476562    9253.693359 -7.82E-01
 19   0.9031    79505.476562   71803.523438 -9.69E-02
 20   0.9031    64511.976562   58262.500000 -9.69E-02
 21   0.9031    41414.976562   37402.980469 -9.69E-02
 22   1.3275    16728.990234   22206.929688  3.27E-01
 23   0.8193    54633.984375   44760.910156 -1.81E-01
 24   0.8193    32771.000000   26848.849609 -1.81E-01
 25   0.8193    11115.009766    9106.372070 -1.81E-01
 26   0.6299    84585.000000   53283.296875 -3.70E-01
 27   0.6299    57768.496094   36390.578125 -3.70E-01
 28   0.6299    31417.994141   19791.390625 -3.70E-01
 29   0.5797    55315.519531   32066.509766 -4.20E-01
 30   0.5797    74307.007812   43075.921875 -4.20E-01
 31   0.5797    65028.988281   37697.449219 -4.20E-01
 32   1.2166    84194.492188  102431.000000  2.17E-01
 33   1.2166    68119.492188   82874.132812  2.17E-01
 34   1.2166    78560.515625   95576.656250  2.17E-01
 35   1.4816    71336.500000  105689.296875  4.82E-01
 36   1.4816    41861.507812   62020.320312  4.82E-01
 37   1.4816    57041.000000   84509.656250  4.82E-01
 38   0.9551    82889.992188   79167.296875 -4.49E-02
 39   0.9551   103265.500000   98627.710938 -4.49E-02
 40   0.8985    60106.000000   54005.296875 -1.01E-01
 41   0.8985     6572.042969    5904.949219 -1.02E-01
 42   0.8985    22018.498047   19783.640625 -1.01E-01
```

43	0.8985	75893.976562	68190.843750	-1.01E-01
44	0.8985	108818.000000	97773.093750	-1.01E-01
1 1	0 1.0000	1.23E-01	5.80E-03	2.46E+06 2.34E+06
1	0.0004	-2344419.500000	-999.000000	0.00E+00
2	1.1211	38803.429688	43502.570312	1.21E-01
3	1.1863	8769.461914	10403.519531	1.86E-01
4	0.9764	71558.140625	69868.757812	-2.36E-02
5	1.1666	15245.409180	17785.279297	1.67E-01
6	0.9809	66174.273438	64912.101562	-1.91E-02
7	1.0369	5509.504883	5712.873047	3.69E-02
8	0.9647	93183.234375	89897.453125	-3.53E-02
9	1.1585	11340.939453	13138.900391	1.59E-01
10	1.2978	18183.488281	23599.050781	2.98E-01
11	1.0827	32599.457031	35294.578125	8.27E-02
12	1.0611	49199.953125	52206.871094	6.11E-02
13	1.0588	102300.976562	108316.101562	5.88E-02
14	1.0461	146281.968750	153025.296875	4.61E-02
15	1.0601	75592.703125	80135.273438	6.01E-02
16	1.0408	206288.609375	214702.578125	4.08E-02
17	1.0341	20440.066406	21137.009766	3.41E-02
18	0.9347	9900.603516	9253.693359	-6.53E-02
19	1.1695	61394.960938	71803.523438	1.70E-01
20	1.1054	52707.683594	58262.500000	1.05E-01
21	1.0981	34060.644531	37402.980469	9.81E-02
22	1.4521	15293.260742	22206.929688	4.52E-01
23	1.0049	44542.339844	44760.910156	4.91E-03
24	1.0032	26762.626953	26848.849609	3.22E-03
25	1.1188	8139.691406	9106.372070	1.19E-01
26	0.9932	53646.656250	53283.296875	-6.77E-03
27	1.2684	28689.134766	36390.578125	2.68E-01
28	1.1559	17122.027344	19791.390625	1.56E-01
29	0.9822	32648.691406	32066.509766	-1.78E-02
30	1.0793	39911.902344	43075.921875	7.93E-02
31	1.0187	37006.609375	37697.449219	1.87E-02
32	1.0142	101000.304688	102431.000000	1.42E-02
33	1.0263	80747.242188	82874.132812	2.63E-02
34	1.0267	93087.179688	95576.656250	2.67E-02
35	1.0295	102663.054688	105689.296875	2.95E-02
36	1.1576	53575.812500	62020.320312	1.58E-01
37	1.1390	74197.859375	84509.656250	1.39E-01
38	1.0088	78477.210938	79167.296875	8.79E-03
39	1.0146	97207.820312	98627.710938	1.46E-02
40	1.0582	51036.839844	54005.296875	5.82E-02
41	1.0884	5425.308105	5904.949219	8.84E-02
42	1.0056	19673.224609	19783.640625	5.61E-03
43	0.9876	69047.148438	68190.843750	-1.24E-02
44	1.0294	94981.757812	97773.093750	2.94E-02
2 2	0 1.0000	1.10E-02	7.05E-05	2.46E+06 2.46E+06
1	0.0004	-2459751.250000	-999.000000	0.00E+00
2	1.0081	43152.800781	43502.570312	8.11E-03
3	0.9949	10457.264648	10403.519531	-5.14E-03
4	0.9898	70590.953125	69868.757812	-1.02E-02
5	1.0082	17641.191406	17785.279297	8.17E-03
6	0.9998	64925.785156	64912.101562	-2.11E-04
7	1.0040	5690.118164	5712.873047	4.00E-03
8	0.9918	90640.531250	89897.453125	-8.20E-03
9	1.0064	13055.752930	13138.900391	6.37E-03
10	1.0098	23369.603516	23599.050781	9.82E-03
11	0.9963	35427.234375	35294.578125	-3.74E-03
12	0.9999	52212.285156	52206.871094	-1.04E-04
13	1.0118	107056.929688	108316.101562	1.18E-02
14	0.9990	153177.000000	153025.296875	-9.90E-04
15	0.9976	80325.750000	80135.273438	-2.37E-03
16	0.9987	214971.515625	214702.578125	-1.25E-03
17	0.9982	21175.033203	21137.009766	-1.80E-03
18	0.9842	9402.681641	9253.693359	-1.58E-02
19	1.0097	71115.476562	71803.523438	9.68E-03
20	1.0010	58206.585938	58262.500000	9.61E-04
21	0.9947	37603.539062	37402.980469	-5.33E-03
22	1.0406	21340.875000	22206.929688	4.06E-02
23	0.9886	45276.609375	44760.910156	-1.14E-02
24	0.9895	27133.367188	26848.849609	-1.05E-02
25	1.0233	8899.108398	9106.372070	2.33E-02
26	0.9892	53865.964844	53283.296875	-1.08E-02
27	1.0166	35796.269531	36390.578125	1.66E-02
28	1.0096	19603.373047	19791.390625	9.59E-03
29	1.0002	32059.824219	32066.509766	2.08E-04
30	1.0148	42447.636719	43075.921875	1.48E-02
31	0.9863	38221.183594	37697.449219	-1.37E-02
32	1.0009	102336.812500	102431.000000	9.20E-04
33	1.0071	82288.187500	82874.132812	7.12E-03
34	1.0034	95255.328125	95576.656250	3.37E-03
35	0.9899	106765.468750	105689.296875	-1.01E-02
36	1.0135	61195.488281	62020.320312	1.35E-02
37	1.0101	83667.929688	84509.656250	1.01E-02
38	0.9991	79241.101562	79167.296875	-9.31E-04
39	1.0051	98123.078125	98627.710938	5.14E-03
40	1.0031	53840.585938	54005.296875	3.06E-03
41	1.0142	5822.303223	5904.949219	1.42E-02
42	0.9876	20032.273438	19783.640625	-1.24E-02
43	0.9948	68548.289062	68190.843750	-5.21E-03
44	0.9998	97792.359375	97773.093750	-1.97E-04
3 3	0 1.0000	2.40E-03	3.63E-06	2.46E+06 2.46E+06
1	0.0004	-2462122.750000	-999.000000	0.00E+00
2	1.0017	43426.726562	43502.570312	1.75E-03
3	0.9961	10444.639648	10403.519531	-3.94E-03
4	0.9977	70027.601562	69868.757812	-2.27E-03
5	1.0017	17754.642578	17785.279297	1.73E-03
6	1.0004	64884.664062	64912.101562	4.23E-04
7	1.0009	5707.528809	5712.873047	9.36E-04
8	0.9980	90079.250000	89897.453125	-2.02E-03
9	0.9998	13141.357422	13138.900391	-1.87E-04
10	0.9995	23611.896484	23599.050781	-5.44E-04
11	0.9987	35341.726562	35294.578125	-1.33E-03
12	1.0000	52205.269531	52206.871094	3.06E-05
13	1.0030	107987.851562	108316.101562	3.04E-03
14	0.9996	153085.562500	153025.296875	-3.94E-04
15	0.9990	80218.375000	80135.273438	-1.04E-03
16	0.9996	214789.171875	214702.578125	-4.03E-04
17	0.9996	21145.109375	21137.009766	-3.83E-04
18	0.9966	9285.530273	9253.693359	-3.43E-03
19	1.0015	71697.773438	71803.523438	1.47E-03
20	1.0003	58246.238281	58262.500000	2.79E-04
21	0.9967	37525.460938	37402.980469	-3.26E-03
22	1.0035	22129.515625	22206.929688	3.50E-03
23	0.9965	44918.792969	44760.910156	-3.51E-03
24	0.9970	26929.921875	26848.849609	-3.01E-03
25	1.0063	9049.343750	9106.372070	6.30E-03
26	0.9972	53431.656250	53283.296875	-2.78E-03
27	1.0018	36325.925781	36390.578125	1.78E-03
28	1.0025	19741.066406	19791.390625	2.55E-03
29	1.0008	32042.269531	32066.509766	7.57E-04
30	1.0037	42917.019531	43075.921875	3.70E-03
31	0.9961	37844.968750	37697.449219	-3.90E-03
32	0.9999	102438.656250	102431.000000	-7.47E-05
33	1.0017	82736.679688	82874.132812	1.66E-03
34	1.0009	95487.914062	95576.656250	9.29E-04
35	0.9978	105918.109375	105689.296875	-2.16E-03
36	1.0028	61846.066406	62020.320312	2.82E-03
37	1.0028	84270.203125	84509.656250	2.84E-03
38	0.9995	79203.992188	79167.296875	-4.63E-04
39	1.0012	98507.039062	98627.710938	1.22E-03
40	1.0007	53965.714844	54005.296875	7.33E-04
41	1.0041	5880.849121	5904.949219	4.10E-03
42	0.9964	19854.744141	19783.640625	-3.58E-03
43	0.9984	68296.750000	68190.843750	-1.55E-03
44	0.9999	97778.960938	97773.093750	-6.00E-05
4 4	0 1.0000	6.80E-04	2.70E-07	2.46E+06 2.46E+06
1	0.0004	-2462142.500000	-999.000000	0.00E+00
2	1.0005	43482.351562	43502.570312	4.65E-04
3	0.9986	10418.593750	10403.519531	-1.45E-03
4	0.9995	69903.140625	69868.757812	-4.92E-04
5	1.0005	17775.640625	17785.279297	5.42E-04
6	1.0002	64899.996094	64912.101562	1.87E-04
7	1.0001	5712.333984	5712.873047	9.43E-05
8	0.9995	89943.265625	89897.453125	-5.09E-04
9	0.9998	13141.986328	13138.900391	-2.35E-04
10	0.9996	23607.603516	23599.050781	-3.62E-04
11	0.9996	35307.300781	35294.578125	-3.60E-04
12	1.0001	52203.113281	52206.871094	7.19E-05
13	1.0008	108225.562500	108316.101562	8.37E-04
14	0.9998	153050.187500	153025.296875	-1.63E-04
15	0.9997	80161.445312	80135.273438	-3.27E-04
16	0.9999	214719.140625	214702.578125	-7.71E-05
17	1.0000	21136.892578	21137.009766	5.60E-06
18	0.9992	9260.981445	9253.693359	-7.87E-04
19	1.0004	71772.437500	71803.523438	4.33E-04
20	1.0002	58250.316406	58262.500000	2.09E-04
21	0.9990	37439.515625	37402.980469	-9.76E-04
22	1.0003	22200.341797	22206.929688	2.97E-04
23	0.9989	44808.863281	44760.910156	-1.07E-03
24	0.9992	26869.783203	26848.849609	-7.79E-04
25	1.0020	9088.541016	9106.372070	1.96E-03
26	0.9993	53322.648438	53283.296875	-7.38E-04
27	1.0004	36376.675781	36390.578125	3.82E-04
28	1.0009	19773.201172	19791.390625	9.20E-04
29	1.0003	32057.361328	32066.509766	2.85E-04
30	1.0009	43036.839844	43075.921875	9.08E-04
31	0.9989	37739.042969	37697.449219	-1.10E-03
32	0.9999	102440.085938	102431.000000	-8.86E-05
33	1.0004	82840.843750	82874.132812	4.02E-04
34	1.0003	95547.937500	95576.656250	3.01E-04
35	0.9995	105741.093750	105689.296875	-4.90E-04
36	1.0007	61977.816406	62020.320312	6.86E-04
37	1.0008	84438.617188	84509.656250	8.41E-04
38	0.9999	79179.078125	79167.296875	-1.49E-04
39	1.0003	98599.164062	98627.710938	2.90E-04
40	1.0001	53997.222656	54005.296875	1.50E-04
41	1.0012	5897.970215	5904.949219	1.18E-03
42	0.9990	19803.898438	19783.640625	-1.02E-03
43	0.9996	68219.976562	68190.843750	-4.27E-04
44	1.0000	97773.906250	97773.093750	-8.34E-06

5	5	0 1.0000	2.04E-04	2.23E-08	2.46E+06 2.46E+06
	1	0.0004	-2462142.250000		-999.000000 0.00E+00
	2	1.0001	43497.253906	43502.570312	1.22E-04
	3	0.9995	10408.563477	10403.519531	-4.85E-04
	4	0.9999	69876.109375	69868.757812	-1.05E-04
	5	1.0002	17782.152344	17785.279297	1.76E-04
	6	1.0001	64908.085938	64912.101562	6.19E-05
	7	1.0000	5713.083008	5712.873047	-3.67E-05
	8	0.9999	89909.093750	89897.453125	-1.30E-04
	9	0.9999	13140.378906	13138.900391	-1.13E-04
	10	0.9999	23602.046875	23599.050781	-1.27E-04
	11	0.9999	35298.050781	35294.578125	-9.84E-05
	12	1.0000	52204.972656	52206.871094	3.64E-05
	13	1.0002	108290.203125	108316.101562	2.39E-04
	14	0.9999	153035.015625	153025.296875	-6.35E-05
	15	0.9999	80143.546875	80135.273438	-1.03E-04
	16	1.0000	214705.453125	214702.578125	-1.34E-05
	17	1.0000	21136.433594	21137.009766	2.73E-05
	18	0.9998	9255.367188	9253.693359	-1.81E-04
	19	1.0001	71794.046875	71803.523438	1.32E-04
	20	1.0001	58257.683594	58262.500000	8.27E-05
	21	0.9997	37413.242188	37402.980469	-2.74E-04
	22	1.0000	22206.740234	22206.929688	8.46E-06
	23	0.9997	44775.699219	44760.910156	-3.30E-04
	24	0.9998	26854.412109	26848.849609	-2.07E-04
	25	1.0006	9100.700195	9106.372070	6.23E-04
	26	0.9998	53294.386719	53283.296875	-2.08E-04
	27	1.0001	36386.562500	36390.578125	1.10E-04
	28	1.0003	19785.023438	19791.390625	3.22E-04
	29	1.0001	32063.421875	32066.509766	9.63E-05
	30	1.0002	43066.230469	43075.921875	2.25E-04
	31	0.9997	37709.660156	37697.449219	-3.24E-04
	32	1.0000	102435.171875	102431.000000	-4.07E-05
	33	1.0001	82866.101562	82874.132812	9.69E-05
	34	1.0001	95567.078125	95576.656250	1.00E-04
	35	0.9999	105701.421875	105689.296875	-1.15E-04
	36	1.0002	62009.351562	62020.320312	1.77E-04
	37	1.0002	84489.023438	84509.656250	2.44E-04
	38	1.0000	79171.007812	79167.296875	-4.69E-05
	39	1.0001	98620.828125	98627.710938	6.97E-05
	40	1.0000	54004.367188	54005.296875	1.72E-05
	41	1.0004	5902.872070	5904.949219	3.52E-04
	42	0.9997	19789.501953	19783.640625	-2.96E-04
	43	0.9999	68198.890625	68190.843750	-1.18E-04
	44	1.0000	97773.234375	97773.093750	-1.49E-06
6	6	0 1.0000	6.19E-05	1.91E-09	2.46E+06 2.46E+06
	1	0.0004	-2462141.750000		-999.000000 0.00E+00
	2	1.0000	43501.191406	43502.570312	3.17E-05
	3	0.9998	10405.129883	10403.519531	-1.55E-04
	4	1.0000	69870.359375	69868.757812	-2.29E-05
	5	1.0001	17784.199219	17785.279297	6.08E-05
	6	1.0000	64910.777344	64912.101562	2.04E-05
	7	1.0000	5713.010254	5712.873047	-2.40E-05
	8	1.0000	89900.390625	89897.453125	-3.27E-05
	9	1.0000	13139.501953	13138.900391	-4.58E-05
	10	1.0000	23599.982422	23599.050781	-3.95E-05
	11	1.0000	35295.480469	35294.578125	-2.56E-05
	12	1.0000	52206.054688	52206.871094	1.56E-05
	13	1.0001	108308.484375	108316.101562	7.03E-05
	14	1.0000	153028.718750	153025.296875	-2.24E-05
	15	1.0000	80137.742188	80135.273438	-3.08E-05
	16	1.0000	214703.015625	214702.578125	-2.03E-06
	17	1.0000	21136.722656	21137.009766	1.36E-05
	18	1.0000	9254.075195	9253.693359	-4.12E-05
	19	1.0000	71800.710938	71803.523438	3.91E-05
	20	1.0000	58260.824219	58262.500000	2.88E-05
	21	0.9999	37405.714844	37402.980469	-7.31E-05
	22	1.0000	22207.283203	22206.929688	-1.60E-05
	23	0.9999	44765.398438	44760.910156	-1.00E-04
	24	0.9999	26850.367188	26848.849609	-5.66E-05
	25	1.0002	9104.563477	9106.372070	1.99E-04
	26	0.9999	53286.539062	53283.296875	-6.09E-05
	27	1.0000	36389.355469	36390.578125	3.36E-05
	28	1.0001	19789.244141	19791.390625	1.08E-04
	29	1.0000	32065.474609	32066.509766	3.23E-05
	30	1.0001	43073.570312	43075.921875	5.46E-05
	31	0.9999	37700.933594	37697.449219	-9.24E-05
	32	1.0000	102432.703125	102431.000000	-1.66E-05
	33	1.0000	82872.187500	82874.132812	2.35E-05
	34	1.0000	95573.460938	95576.656250	3.34E-05
	35	1.0000	105692.046875	105689.296875	-2.60E-05
	36	1.0000	62017.375000	62020.320312	4.74E-05
	37	1.0001	84503.726562	84509.656250	7.01E-05
	38	1.0000	79168.437500	79167.296875	-1.44E-05
	39	1.0000	98625.992188	98627.710938	1.74E-05
	40	1.0000	54005.527344	54005.296875	-4.23E-06
	41	1.0001	5904.296875	5904.949219	1.11E-04
	42	0.9999	19785.285156	19783.640625	-8.31E-05
	43	1.0000	68193.195312	68190.843750	-3.45E-05
	44	1.0000	97773.148438	97773.093750	-5.36E-07
7	7	0 1.0000	1.90E-05	1.72E-10	2.46E+06 2.46E+06
	1	0.0004	-2462143.500000		-999.000000 0.00E+00
	2	1.0000	43502.355469	43502.570312	4.89E-06
	3	1.0000	10404.003906	10403.519531	-4.66E-05
	4	1.0000	69869.156250	69868.757812	-5.60E-06
	5	1.0000	17784.910156	17785.279297	2.07E-05
	6	1.0000	64911.609375	64912.101562	7.51E-06
	7	1.0000	5712.974121	5712.873047	-1.77E-05
	8	1.0000	89898.351562	89897.453125	-1.00E-05
	9	1.0000	13139.135742	13138.900391	-1.79E-05
	10	1.0000	23599.308594	23599.050781	-1.09E-05
	11	1.0000	35294.792969	35294.578125	-6.08E-06
	12	1.0000	52206.542969	52206.871094	6.20E-06
	13	1.0000	108313.968750	108316.101562	1.98E-05
	14	1.0000	153026.593750	153025.296875	-8.52E-06
	15	1.0000	80136.070312	80135.273438	-9.95E-06
	16	1.0000	214702.781250	214702.578125	-8.94E-07
	17	1.0000	21136.910156	21137.009766	4.77E-06
	18	1.0000	9253.785156	9253.693359	-9.95E-06
	19	1.0000	71802.648438	71803.523438	1.22E-05
	20	1.0000	58262.101562	58262.500000	6.79E-06
	21	1.0000	37403.769531	37402.980469	-2.10E-05
	22	1.0000	22206.994141	22206.929688	-2.92E-06
	23	1.0000	44762.300781	44760.910156	-3.10E-05
	24	1.0000	26849.220703	26848.849609	-1.38E-05
	25	1.0001	9105.806641	9106.372070	6.21E-05
	26	1.0000	53284.222656	53283.296875	-1.73E-05
	27	1.0000	36390.171875	36390.578125	1.12E-05
	28	1.0000	19790.648438	19791.390625	3.76E-05
	29	1.0000	32066.248047	32066.509766	8.23E-06
	30	1.0000	43075.332031	43075.921875	1.37E-05
	31	1.0000	37698.554688	37697.449219	-2.94E-05
	32	1.0000	102431.585938	102431.000000	-5.66E-06
	33	1.0000	82873.523438	82874.132812	7.39E-06
	34	1.0000	95575.601562	95576.656250	1.11E-05
	35	1.0000	105689.828125	105689.296875	-5.01E-06
	36	1.0000	62019.480469	62020.320312	1.35E-05
	37	1.0000	84508.054688	84509.656250	1.90E-05
	38	1.0000	79167.656250	79167.296875	-4.47E-06
	39	1.0000	98627.250000	98627.710938	4.77E-06
	40	1.0000	54005.445312	54005.296875	-2.68E-06
	41	1.0000	5904.767578	5904.949219	3.08E-05
	42	1.0000	19784.117188	19783.640625	-2.40E-05
	43	1.0000	68191.773438	68190.843750	-1.36E-05
	44	1.0000	97773.210938	97773.093750	-1.25E-06

Boundary Point Table
Boundary 1

j	0	1	2	3	4	5	6	7
ax	11.000000	8.010808	9.322116	9.396812	9.402699	9.402502	9.401915	9.401873
ay	-998.000000	-982.087769	-994.324402	-994.410400	-994.368408	-994.356445	-994.353333	-994.352722
ax	-1.000000	-2.835188	-2.105273	-1.928004	-1.883902	-1.873126	-1.870437	-1.869430
ay	-1273.000000	-1252.154663	-1263.010254	-1263.305054	-1263.321655	-1263.323730	-1263.323975	-1263.323975
ax	-180.000000	-179.251083	-180.738571	-180.834335	-180.842010	-180.842087	-180.841232	-180.841125
ay	-1270.000000	-1246.014648	-1257.106323	-1257.591919	-1257.660156	-1257.675415	-1257.678955	-1257.679932
ax	-358.000000	-355.371552	-359.046722	-359.479065	-359.556396	-359.571442	-359.574585	-359.575073
ay	-1266.000000	-1235.484741	-1244.872681	-1245.121094	-1245.151611	-1245.161133	-1245.163940	-1245.164795
ax	-501.000000	-484.857666	-487.862366	-488.168304	-488.235229	-488.251770	-488.255859	-488.256683
ay	-1265.000000	-1232.416260	-1240.452881	-1240.462769	-1240.444092	-1240.442871	-1240.443115	-1240.443481
ax	-644.000000	-615.385132	-617.534912	-617.626587	-617.653259	-617.662537	-617.665100	-617.665894
ay	-1263.000000	-1240.222656	-1248.187378	-1248.118530	-1248.067139	-1248.055176	-1248.052368	-1248.051392
ax	-638.000000	-595.224121	-597.105591	-596.842468	-596.770203	-596.753418	-596.748901	-596.748108
ay	-942.000000	-944.471191	-955.879578	-956.021606	-955.992188	-955.982178	-955.979309	-955.978455
ax	-632.000000	-605.955505	-612.331116	-612.559814	-612.589111	-612.595947	-612.597290	-612.597778

ay	-621.000000	-656.332336	-672.236389	-672.993042	-673.106201	-673.132446	-673.138916	-673.140381
ax	-675.000000	-665.012695	-675.719788	-676.108032	-676.150146	-676.157410	-676.158813	-676.158875
ay	-381.000000	-398.189575	-411.565186	-411.699310	-411.668793	-411.659821	-411.657410	-411.656769
ax	-718.000000	-694.314575	-707.529114	-707.589661	-707.548523	-707.536987	-707.534058	-707.533203
ay	-140.000000	-142.594910	-153.823441	-153.566071	-153.468155	-153.448730	-153.445267	-153.444916
ax	-699.000000	-661.499023	-677.759644	-677.617737	-677.535278	-677.517639	-677.514038	-677.513489
ay	149.000000	91.258286	78.950996	78.657928	78.638428	78.634155	78.632980	78.632324
ax	-679.000000	-746.269043	-766.432068	-766.536682	-766.496643	-766.485962	-766.482666	-766.481873
ay	439.000000	312.402863	303.762451	303.362274	303.319244	303.309692	303.306854	303.306000
ax	-699.000000	-818.514893	-839.909973	-840.150391	-840.129822	-840.126587	-840.126038	-840.126404
ay	605.000000	574.697021	571.751160	571.326599	571.271545	571.259827	571.256592	571.255676
ax	-718.000000	-824.787292	-847.574890	-848.237000	-848.280579	-848.289734	-848.292114	-848.293274
ay	771.000000	831.881836	834.198059	833.756470	833.707520	833.703064	833.703125	833.703247
ax	-646.000000	-627.380737	-654.109497	-655.430725	-655.483276	-655.480408	-655.478149	-655.477478
ay	1018.000000	1115.184082	1131.668945	1132.273315	1132.310303	1132.311890	1132.311035	1132.310669
ax	-574.000000	-519.304077	-531.529175	-531.596191	-531.590698	-531.598022	-531.601318	-531.602844
ay	1266.000000	1304.516968	1331.675415	1332.710815	1332.709839	1332.694580	1332.689209	1332.687378
ax	-322.000000	-236.396744	-235.238220	-234.944885	-234.948181	-234.956970	-234.960083	-234.961182
ay	1266.000000	1278.788574	1306.642090	1307.318848	1307.360352	1307.376587	1307.382446	1307.384766
ax	-70.000000	37.062527	48.237804	48.694942	48.731117	48.745377	48.750622	48.752579
ay	1266.000000	1227.945068	1250.848755	1250.985718	1250.935425	1250.926147	1250.924316	1250.923950
ax	201.000000	272.615875	285.144287	285.231659	285.169434	285.151978	285.146881	285.146027
ay	1269.000000	1180.678711	1199.302734	1199.764526	1199.826050	1199.840942	1199.844238	1199.844849
ax	473.000000	504.891846	522.742371	523.724731	523.892517	523.931702	523.940796	523.943420
ay	1273.000000	1145.384155	1158.782715	1158.965698	1158.981079	1158.988159	1158.990967	1158.991821
ax	627.000000	594.566895	609.874207	610.296631	610.340332	610.351257	610.354004	610.355286
ay	1272.000000	1142.418945	1153.179077	1153.089600	1153.040161	1153.031982	1153.030884	1153.031128
ax	781.000000	697.926147	711.752563	711.991638	711.995361	711.997925	711.999023	712.000000
ay	1271.000000	1194.409424	1204.217407	1204.290405	1204.277100	1204.275879	1204.276245	1204.276611
ax	647.000000	538.885315	559.278198	559.668579	559.694397	559.704468	559.707886	559.709473
ay	895.000000	935.270874	944.798462	944.925354	944.909546	944.904663	944.903503	944.903198
ax	513.000000	480.972992	508.574066	509.022217	509.002502	508.995605	508.993439	508.993103
ay	519.000000	601.749329	602.360168	602.376770	602.354858	602.344727	602.341125	602.339722
ax	429.000000	419.741852	448.109650	448.653473	448.646698	448.642273	448.640747	448.640564
ay	420.000000	495.052032	491.122223	491.227386	491.247803	491.249878	491.249939	491.249725
ax	345.000000	352.492615	382.502197	383.400391	383.446960	383.454224	383.456055	383.456665
ay	322.000000	382.852142	373.316437	373.312592	373.346436	373.354889	373.357056	373.357269
ax	277.000000	276.894684	298.921265	299.384338	299.378784	299.374817	299.373871	299.373657
ay	103.000000	145.887970	127.842056	127.208672	127.158447	127.146225	127.142326	127.141075
ax	209.000000	216.748550	230.470337	230.565735	230.557724	230.556213	230.555466	230.555084
ay	-116.000000	-70.008408	-89.856010	-90.265198	-90.251999	-90.248253	-90.247810	-90.248100
ax	116.000000	126.679756	133.631180	133.651718	133.631989	133.626266	133.624680	133.624588
ay	-420.000000	-396.886841	-414.914215	-415.272400	-415.289917	-415.295074	-415.296661	-415.297424
ax	23.000000	26.617027	28.680613	28.622435	28.587276	28.576437	28.572962	28.572025
ay	-723.000000	-707.588257	-722.452820	-722.578003	-722.544495	-722.536255	-722.534424	-722.534119

Boundary 2

j	0	1	2	3	4	5	6	7
ax	-532.000000	-493.963654	-514.313293	-515.330750	-515.506409	-515.547546	-515.557068	-515.559448
ay	193.000000	106.341980	89.337395	88.554451	88.431084	88.404106	88.398293	88.396652
ax	-384.000000	-330.976868	-343.208038	-343.614410	-343.660553	-343.669586	-343.671082	-343.671356
ay	-52.000000	-41.876698	-63.767120	-64.073174	-64.068443	-64.068176	-64.068214	-64.068535
ax	-383.000000	-324.417572	-338.139374	-338.598236	-338.644867	-338.654236	-338.656219	-338.656769
ay	145.000000	75.417419	59.459557	59.132450	59.116245	59.115150	59.115925	59.116077
ax	-381.000000	-301.078796	-313.058929	-313.365845	-313.384338	-313.386597	-313.386993	-313.386993
ay	343.000000	180.171432	174.121124	174.408585	174.489548	174.511353	174.517822	174.519348
ax	-530.000000	-526.178528	-544.099182	-544.634521	-544.703064	-544.717346	-544.719971	-544.720215
ay	391.000000	203.591858	193.791321	193.580688	193.585678	193.593948	193.597672	193.599030
ax	-679.000000	-746.269043	-766.432068	-766.536682	-766.496643	-766.485962	-766.482666	-766.481873
ay	439.000000	312.402863	303.762451	303.362274	303.319244	303.309692	303.306854	303.306000

Boundary 3

j	0	1	2	3	4	5	6	7
ax	-321.000000	-252.596344	-261.648773	-261.794373	-261.748474	-261.725800	-261.717224	-261.714020
ay	202.000000	109.072029	95.420326	95.313454	95.354935	95.371994	95.378235	95.379990
ax	-261.000000	-187.053116	-191.146149	-191.519791	-191.607437	-191.631821	-191.638718	-191.640198
ay	62.000000	36.286488	13.835632	13.257878	13.234482	13.234234	13.235188	13.235323
ax	-235.000000	-151.728043	-153.417877	-153.889969	-154.048370	-154.101562	-154.119064	-154.124207
ay	119.000000	65.497154	46.723122	46.413864	46.449669	46.468517	46.475586	46.477711
ax	-209.000000	-116.457123	-114.855957	-115.036400	-115.125717	-115.159996	-115.172058	-115.175537
ay	176.000000	89.588737	73.369965	72.946106	72.916824	72.915314	72.916138	72.916588
ax	-295.000000	-223.457565	-230.657379	-231.156418	-231.259094	-231.286713	-231.294540	-231.296783
ay	259.000000	143.990662	133.934052	133.633774	133.548218	133.519073	133.509979	133.506653
ax	-381.000000	-301.078796	-313.058929	-313.365845	-313.384338	-313.386597	-313.386993	-313.386993
ay	343.000000	180.171432	174.121124	174.408585	174.489548	174.511353	174.517822	174.519348

Boundary 4

j	0	1	2	3	4	5	6	7
ax	-532.000000	-493.963654	-514.313293	-515.330750	-515.506409	-515.547546	-515.557068	-515.559448
ay	193.000000	106.341980	89.337395	88.554451	88.431084	88.404106	88.398293	88.396652
ax	-679.000000	-746.269043	-766.432068	-766.536682	-766.496643	-766.485962	-766.482666	-766.481873
ay	439.000000	312.402863	303.762451	303.362274	303.319244	303.309692	303.306854	303.306000
ax	-699.000000	-661.499023	-677.759644	-677.617737	-677.535278	-677.517639	-677.514038	-677.513489
ay	149.000000	91.258286	78.950996	78.657928	78.638428	78.634155	78.632980	78.632324
ax	-718.000000	-694.314575	-707.529114	-707.589661	-707.548523	-707.536987	-707.534058	-707.533203
ay	-140.000000	-142.594910	-153.823441	-153.566071	-153.468155	-153.448730	-153.445267	-153.444916
ax	-551.000000	-519.233948	-533.930603	-534.523071	-534.612732	-534.634155	-534.639526	-534.641113
ay	-96.000000	-82.599579	-96.065147	-95.622612	-95.483765	-95.456947	-95.452316	-95.451965
ax	-384.000000	-330.976868	-343.208038	-343.614410	-343.660553	-343.669586	-343.671082	-343.671356
ay	-52.000000	-41.876698	-63.767120	-64.073174	-64.068443	-64.068176	-64.068214	-64.068535

Boundary 5

j	0	1	2	3	4	5	6	7
ax	-383.000000	-324.417572	-338.139374	-338.598236	-338.644867	-338.654236	-338.656219	-338.656769
ay	145.000000	75.417419	59.459557	59.132450	59.116245	59.115150	59.115925	59.116077
ax	-384.000000	-330.976868	-343.208038	-343.614410	-343.660553	-343.669586	-343.671082	-343.671356
ay	-52.000000	-41.876698	-63.767120	-64.073174	-64.068443	-64.068176	-64.068214	-64.068535
ax	-323.000000	-263.424225	-270.307770	-270.352295	-270.319489	-270.305634	-270.300446	-270.298584
ay	5.000000	-4.846634	-28.491846	-29.274696	-29.382883	-29.412550	-29.420986	-29.423809
ax	-261.000000	-187.053116	-191.146149	-191.519791	-191.607437	-191.631821	-191.638718	-191.640198
ay	62.000000	36.286488	13.835632	13.257878	13.234482	13.234234	13.235188	13.235323
ax	-321.000000	-252.596344	-261.648773	-261.794373	-261.748474	-261.725800	-261.717224	-261.714020
ay	202.000000	109.072029	95.420326	95.313454	95.354935	95.371994	95.378235	95.379990
ax	-381.000000	-301.078796	-313.058929	-313.365845	-313.384338	-313.386597	-313.386993	-313.386993
ay	343.000000	180.171432	174.121124	174.408585	174.489548	174.511353	174.517822	174.519348

Boundary 6

j	0	1	2	3	4	5	6	7
ax	-501.000000	-484.857666	-487.862366	-488.168304	-488.235229	-488.251770	-488.255859	-488.256683
ay	-1265.000000	-1232.416260	-1240.452881	-1240.462769	-1240.444092	-1240.442871	-1240.443115	-1240.443481
ax	-358.000000	-355.371552	-359.046722	-359.479065	-359.556396	-359.571442	-359.574585	-359.575073
ay	-1266.000000	-1235.484741	-1244.872681	-1245.121094	-1245.151611	-1245.161133	-1245.163940	-1245.164795
ax	-318.000000	-319.687927	-325.392242	-326.105316	-326.237579	-326.264709	-326.270142	-326.271240
ay	-978.000000	-968.853210	-980.549438	-980.512024	-980.437012	-980.416077	-980.410645	-980.409363
ax	-277.000000	-258.348419	-262.448639	-262.662354	-262.692749	-262.700958	-262.703369	-262.703949
ay	-690.000000	-703.876953	-721.055725	-721.638123	-721.707703	-721.724182	-721.728577	-721.730042
ax	-461.000000	-442.790619	-447.647461	-448.337250	-448.505035	-448.548615	-448.559479	-448.562103
ay	-977.000000	-972.842651	-984.031372	-983.901978	-983.793091	-983.761230	-983.752563	-983.750305
ax	-644.000000	-615.385132	-617.534912	-617.626587	-617.653259	-617.662537	-617.665100	-617.665894
ay	-1263.000000	-1240.222656	-1248.187378	-1248.118530	-1248.067139	-1248.055176	-1248.052368	-1248.051392

Boundary 7

j	0	1	2	3	4	5	6	7
ax	-455.000000	-427.610260	-433.152283	-433.524078	-433.594696	-433.611389	-433.614655	-433.615295
ay	-656.000000	-695.596497	-714.379883	-715.573242	-715.786987	-715.837097	-715.849060	-715.851990
ax	-277.000000	-258.348419	-262.448639	-262.662354	-262.692749	-262.700958	-262.703369	-262.703949
ay	-690.000000	-703.876953	-721.055725	-721.638123	-721.707703	-721.724182	-721.728577	-721.730042
ax	-296.000000	-271.916290	-275.895630	-276.059174	-276.082336	-276.090210	-276.093079	-276.094086
ay	-666.000000	-686.895996	-703.841553	-704.435913	-704.503052	-704.518677	-704.522766	-704.523926

ax	-314.000000	-283.109833	-286.604218	-286.615021	-286.598267	-286.595001	-286.594696	-286.594849
ay	-642.000000	-670.633667	-687.278076	-687.868408	-687.929932	-687.944031	-687.947876	-687.949036
ax	-473.000000	-445.257751	-450.960632	-451.328308	-451.397278	-451.413361	-451.416626	-451.417267
ay	-632.000000	-677.382812	-695.708557	-696.872070	-697.079712	-697.130676	-697.143982	-697.147583
ax	-632.000000	-605.955505	-612.331116	-612.559814	-612.589111	-612.595947	-612.597290	-612.597778
ay	-621.000000	-656.332336	-672.236389	-672.993042	-673.106201	-673.132446	-673.138916	-673.140381

Boundary 8

j	0	1	2	3	4	5	6	7
ax	-638.000000	-595.224121	-597.105591	-596.842468	-596.770203	-596.753418	-596.748901	-596.748108
ay	-942.000000	-944.471191	-955.879578	-956.021606	-955.992188	-955.982178	-955.979309	-955.978455
ax	-644.000000	-615.385132	-617.534912	-617.626587	-617.653259	-617.662537	-617.665100	-617.665894
ay	-1263.000000	-1240.222656	-1248.187378	-1248.118530	-1248.067139	-1248.055176	-1248.052368	-1248.051392
ax	-461.000000	-442.790619	-447.647461	-448.337250	-448.505035	-448.548615	-448.559479	-448.562103
ay	-977.000000	-972.842651	-984.031372	-983.901978	-983.793091	-983.761230	-983.752563	-983.750305
ax	-277.000000	-258.348419	-262.448639	-262.662354	-262.692749	-262.700958	-262.703369	-262.703949
ay	-690.000000	-703.876953	-721.055725	-721.638123	-721.707703	-721.724182	-721.728577	-721.730042
ax	-455.000000	-427.610260	-433.152283	-433.524078	-433.594696	-433.611389	-433.614655	-433.615295
ay	-656.000000	-695.596497	-714.379883	-715.573242	-715.786987	-715.837097	-715.849060	-715.851990
ax	-632.000000	-605.955505	-612.331116	-612.559814	-612.589111	-612.595947	-612.597290	-612.597778
ay	-621.000000	-656.332336	-672.236389	-672.993042	-673.106201	-673.132446	-673.138916	-673.140381

Boundary 9

j	0	1	2	3	4	5	6	7
ax	282.000000	364.352539	386.216949	386.720703	386.761749	386.772736	386.775787	386.776764
ay	611.000000	661.217834	666.384705	666.353394	666.309448	666.295166	666.291260	666.290039
ax	513.000000	480.972992	508.574066	509.022217	509.002502	508.995605	508.993439	508.993103
ay	519.000000	601.749329	602.360168	602.376770	602.354858	602.344727	602.341125	602.339722
ax	300.000000	377.333191	400.255310	400.855957	400.886414	400.889801	400.889526	400.889313
ay	653.000000	702.729980	711.330688	711.522583	711.489014	711.472168	711.466064	711.463745
ax	88.000000	247.369965	260.446747	260.861420	260.902069	260.914001	260.917694	260.919250
ay	788.000000	837.492249	854.016907	854.479919	854.503601	854.504944	854.504395	854.503601
ax	70.000000	240.595016	251.832062	252.026932	252.044250	252.051895	252.054703	252.056168
ay	746.000000	800.373230	814.939819	815.232056	815.231140	815.225037	815.221985	815.220642
ax	52.000000	236.662689	247.534363	247.612045	247.614655	247.617966	247.619202	247.620010
ay	704.000000	769.339233	783.263855	783.671753	783.728882	783.742004	783.745422	783.745972

Boundary 10

j	0	1	2	3	4	5	6	7
ax	198.000000	314.447754	338.113434	339.023102	339.152130	339.185608	339.194885	339.197510
ay	513.000000	553.838501	556.264221	556.446289	556.436035	556.427185	556.424133	556.423035
ax	52.000000	236.662689	247.534363	247.612045	247.614655	247.617966	247.619202	247.620010
ay	704.000000	769.339233	783.263855	783.671753	783.728882	783.742004	783.745422	783.745972
ax	74.000000	268.459412	282.375854	282.891083	283.077942	283.139130	283.157684	283.163422
ay	536.000000	572.173950	574.971558	575.066589	575.050232	575.041565	575.038818	575.037903
ax	96.000000	247.986237	256.860596	256.331604	256.277588	256.272797	256.272522	256.272644
ay	368.000000	361.859772	352.452942	352.307251	352.267914	352.252960	352.248505	352.246674
ax	220.000000	293.925903	312.704651	312.935486	312.953217	312.960815	312.963196	312.964081
ay	345.000000	379.053986	369.769897	369.769867	369.796112	369.802826	369.805054	369.805237
ax	345.000000	352.492615	382.502197	383.400391	383.446960	383.454224	383.456055	383.456665
ay	322.000000	382.852142	373.316437	373.312592	373.346436	373.354889	373.357056	373.357269

Boundary 11

j	0	1	2	3	4	5	6	7
ax	282.000000	364.352539	386.216949	386.720703	386.761749	386.772736	386.775787	386.776764
ay	611.000000	661.217834	666.384705	666.353394	666.309448	666.295166	666.291260	666.290039
ax	52.000000	236.662689	247.534363	247.612045	247.614655	247.617966	247.619202	247.620010
ay	704.000000	769.339233	783.263855	783.671753	783.728882	783.742004	783.745422	783.745972
ax	198.000000	314.447754	338.113434	339.023102	339.152130	339.185608	339.194885	339.197510
ay	513.000000	553.838501	556.264221	556.446289	556.436035	556.427185	556.424133	556.423035
ax	345.000000	352.492615	382.502197	383.400391	383.446960	383.454224	383.456055	383.456665
ay	322.000000	382.852142	373.316437	373.312592	373.346436	373.354889	373.357056	373.357269
ax	429.000000	419.741852	448.109650	448.653473	448.646698	448.642273	448.640747	448.640564
ay	420.000000	495.052032	491.122223	491.227386	491.247803	491.249878	491.249939	491.249725
ax	513.000000	480.972992	508.574066	509.022217	509.002502	508.995605	508.993439	508.993103
ay	519.000000	601.749329	602.360168	602.376770	602.354858	602.344727	602.341125	602.339722

Boundary 12

j	0	1	2	3	4	5	6	7
ax	300.000000	377.333191	400.255310	400.855957	400.886414	400.889801	400.889526	400.889313
ay	653.000000	702.729980	711.330688	711.522583	711.489014	711.472168	711.466064	711.463745
ax	513.000000	480.972992	508.574066	509.022217	509.002502	508.995605	508.993439	508.993103
ay	519.000000	601.749329	602.360168	602.376770	602.354858	602.344727	602.341125	602.339722
ax	469.000000	495.471710	518.452698	518.893250	518.915833	518.923584	518.926147	518.927246
ay	758.000000	813.079773	821.219788	821.362732	821.345398	821.337708	821.335205	821.334167
ax	426.000000	489.057281	507.577911	508.145294	508.213715	508.228607	508.231567	508.232666
ay	998.000000	985.695801	997.128418	997.058044	996.993469	996.980713	996.978455	996.978210
ax	257.000000	351.737762	366.271149	366.544617	366.542938	366.541748	366.540924	366.541321
ay	893.000000	910.187378	924.946106	925.184265	925.174927	925.173889	925.174255	925.174377
ax	88.000000	247.369965	260.446747	260.861420	260.902069	260.914001	260.917694	260.919250
ay	788.000000	837.492249	854.016907	854.479919	854.503601	854.504944	854.504395	854.503601

Boundary 13

j	0	1	2	3	4	5	6	7
ax	-320.000000	-86.510666	-82.387367	-81.020905	-80.700935	-80.616379	-80.592461	-80.585312
ay	739.000000	842.220642	848.998413	849.139954	849.220276	849.242493	849.248596	849.250244
ax	-359.000000	-160.521164	-158.331680	-157.522232	-157.376160	-157.345047	-157.337189	-157.334549
ay	832.000000	972.798096	976.053284	975.960266	975.996338	976.005676	976.007629	976.008301
ax	-539.000000	-489.832855	-502.004272	-502.365448	-502.450256	-502.480438	-502.489838	-502.493042
ay	801.000000	905.760193	908.616882	908.854858	909.037292	909.098938	909.117981	909.123901
ax	-718.000000	-824.787292	-847.574890	-848.237000	-848.280579	-848.289734	-848.292114	-848.293274
ay	771.000000	831.881836	834.198059	833.756470	833.707520	833.703064	833.703125	833.703247
ax	-499.000000	-429.760468	-440.697021	-441.020538	-441.092682	-441.116333	-441.123199	-441.125549
ay	709.000000	761.937500	761.755981	760.562073	760.327209	760.270142	760.254761	760.250427
ax	-280.000000	-51.421230	-53.034142	-52.822388	-52.798317	-52.795628	-52.795059	-52.794598
ay	647.000000	705.234070	710.897400	710.542847	710.507568	710.502686	710.502075	710.502075

Boundary 14

j	0	1	2	3	4	5	6	7
ax	-480.000000	-417.648254	-429.760162	-429.937653	-429.954437	-429.957855	-429.957855	-429.957733
ay	543.000000	486.815887	483.768188	483.267822	483.184052	483.156921	483.147308	483.144104
ax	-679.000000	-746.269043	-766.432068	-766.536682	-766.496643	-766.485962	-766.482666	-766.481873
ay	439.000000	312.402863	303.762451	303.362274	303.319244	303.309692	303.306854	303.306000
ax	-530.000000	-526.178528	-544.099182	-544.634521	-544.703064	-544.717346	-544.719971	-544.720215
ay	391.000000	203.591858	193.791321	193.580688	193.585678	193.593948	193.597672	193.599030
ax	-381.000000	-301.078796	-313.058929	-313.365845	-313.384338	-313.386597	-313.386993	-313.386993
ay	343.000000	180.171432	174.121124	174.408585	174.489548	174.511353	174.517822	174.519348
ax	-331.000000	-155.424988	-159.522842	-159.285324	-159.247864	-159.248703	-159.251755	-159.253159
ay	495.000000	428.974091	426.057251	425.707977	425.647705	425.631317	425.626862	425.625275
ax	-280.000000	-51.421230	-53.034142	-52.822388	-52.798317	-52.795628	-52.795059	-52.794598
ay	647.000000	705.234070	710.897400	710.542847	710.507568	710.502686	710.502075	710.502075

Boundary 15

j	0	1	2	3	4	5	6	7
ax	-539.000000	-489.832855	-502.004272	-502.365448	-502.450256	-502.480438	-502.489838	-502.493042
ay	801.000000	905.760193	908.616882	908.854858	909.037292	909.098938	909.117981	909.123901
ax	-359.000000	-160.521164	-158.331680	-157.522232	-157.376160	-157.345047	-157.337189	-157.334549
ay	832.000000	972.798096	976.053284	975.960266	975.996338	976.005676	976.007629	976.008301
ax	-481.000000	-364.960236	-372.666931	-372.766815	-372.840302	-372.873688	-372.885376	-372.888977
ay	884.000000	1056.562988	1064.116089	1064.233765	1064.330933	1064.359497	1064.366699	1064.368774
ax	-602.000000	-576.135132	-587.777527	-587.084900	-586.932983	-586.907166	-586.901794	-586.901367
ay	937.000000	1086.204956	1091.641968	1090.930664	1090.884766	1090.886719	1090.888306	1090.889038
ax	-660.000000	-717.865356	-740.518982	-741.161743	-741.176941	-741.175659	-741.174561	-741.174744
ay	854.000000	971.532104	976.049744	975.404175	975.346069	975.343323	975.343933	975.344482
ax	-718.000000	-824.787292	-847.574890	-848.237000	-848.280579	-848.289734	-848.292114	-848.293274
ay	771.000000	831.881836	834.198059	833.756470	833.707520	833.703064	833.703125	833.703247

Boundary 16

j	0	1	2	3	4	5	6	7
ax	-480.000000	-417.648254	-429.760162	-429.937653	-429.954437	-429.957855	-429.957855	-429.957733
ay	543.000000	486.815887	483.768188	483.267822	483.184052	483.156921	483.147308	483.144104
ax	-280.000000	-51.421230	-53.034142	-52.822388	-52.798317	-52.795628	-52.795059	-52.794598
ay	647.000000	705.234070	710.897400	710.542847	710.507568	710.502686	710.502075	710.502075
ax	-499.000000	-429.760468	-440.697021	-441.020538	-441.092682	-441.116333	-441.123199	-441.125549
ay	709.000000	761.937500	761.755981	760.562073	760.327209	760.270142	760.254761	760.250427

ax	-718.000000	-824.787292	-847.574890	-848.237000	-848.280579	-848.289734	-848.292114	-848.293274
ay	771.000000	831.881836	834.198059	833.756470	833.707520	833.703064	833.703125	833.703247
ax	-699.000000	-818.514893	-839.909973	-840.150391	-840.129822	-840.126587	-840.126038	-840.126404
ay	605.000000	574.697021	571.751160	571.326599	571.271545	571.259827	571.256592	571.255676
ax	-679.000000	-746.269043	-766.432068	-766.536682	-766.496643	-766.485962	-766.482666	-766.481873
ay	439.000000	312.402863	303.762451	303.362274	303.319244	303.309692	303.306854	303.306000

Boundary 17

j	0	1	2	3	4	5	6	7
ax	469.000000	495.471710	518.452698	518.893250	518.915833	518.923584	518.926147	518.927246
ay	758.000000	813.079773	821.219788	821.362732	821.345398	821.337708	821.335205	821.334167
ax	513.000000	480.972992	508.574066	509.022217	509.002502	508.995605	508.993439	508.993103
ay	519.000000	601.749329	602.360168	602.376770	602.354858	602.344727	602.341125	602.339722
ax	647.000000	538.885315	559.278198	559.668579	559.694397	559.704468	559.707886	559.709473
ay	895.000000	935.270874	944.798462	944.925354	944.909546	944.904663	944.903503	944.903198
ax	781.000000	697.926147	711.752563	711.991638	711.995361	711.997925	711.999023	712.000000
ay	1271.000000	1194.409424	1204.217407	1204.290405	1204.277100	1204.275879	1204.276245	1204.276611
ax	603.000000	562.532288	578.690918	579.121521	579.168518	579.179749	579.182312	579.183655
ay	1134.000000	1079.547852	1092.370850	1092.683350	1092.715942	1092.726807	1092.730225	1092.731567
ax	426.000000	489.057281	507.577911	508.145294	508.213715	508.228607	508.231567	508.232666
ay	998.000000	985.695801	997.128418	997.058044	996.993469	996.980713	996.978455	996.978210

Boundary 18

j	0	1	2	3	4	5	6	7
ax	603.000000	562.532288	578.690918	579.121521	579.168518	579.179749	579.182312	579.183655
ay	1134.000000	1079.547852	1092.370850	1092.683350	1092.715942	1092.726807	1092.730225	1092.731567
ax	781.000000	697.926147	711.752563	711.991638	711.995361	711.997925	711.999023	712.000000
ay	1271.000000	1194.409424	1204.217407	1204.290405	1204.277100	1204.275879	1204.276245	1204.276611
ax	627.000000	594.566895	609.874207	610.296631	610.340332	610.351257	610.354004	610.355286
ay	1272.000000	1142.418945	1153.179077	1153.089600	1153.040161	1153.031982	1153.030884	1153.031128
ax	473.000000	504.891846	522.742371	523.724731	523.892517	523.931702	523.940796	523.943420
ay	1273.000000	1145.384155	1158.782715	1158.965698	1158.981079	1158.988159	1158.990967	1158.991821
ax	449.000000	503.623749	523.640015	524.828918	525.043457	525.093445	525.104919	525.108154
ay	1135.000000	1068.713135	1081.130981	1081.205322	1081.182739	1081.180054	1081.180176	1081.180664
ax	426.000000	489.057281	507.577911	508.145294	508.213715	508.228607	508.231567	508.232666
ay	998.000000	985.695801	997.128418	997.058044	996.993469	996.980713	996.978455	996.978210

Boundary 19

j	0	1	2	3	4	5	6	7
ax	-220.000000	-56.609509	-48.226143	-47.217823	-47.044521	-46.999908	-46.987282	-46.983383
ay	921.000000	989.040649	993.239502	992.568909	992.410889	992.365662	992.352295	992.348511
ax	-81.000000	76.996605	90.802467	91.484230	91.543045	91.560333	91.566025	91.568008
ay	1010.000000	1040.268311	1051.823120	1051.885986	1051.861206	1051.853394	1051.851074	1051.850464
ax	-328.000000	-224.025162	-222.562744	-222.149506	-222.147766	-222.158691	-222.162857	-222.164230
ay	1138.000000	1180.804810	1202.841187	1203.555908	1203.607666	1203.618286	1203.620972	1203.621704
ax	-574.000000	-519.304077	-531.529175	-531.596191	-531.590698	-531.598022	-531.601318	-531.602844
ay	1266.000000	1304.516968	1331.675415	1332.710815	1332.709839	1332.694580	1332.689209	1332.687378
ax	-467.000000	-377.475159	-385.389954	-385.652802	-385.811279	-385.871948	-385.891724	-385.897858
ay	1049.000000	1138.872314	1152.204102	1152.104370	1152.009521	1151.979248	1151.969849	1151.967529
ax	-359.000000	-160.521164	-158.331680	-157.522232	-157.376160	-157.345047	-157.337189	-157.334549
ay	832.000000	972.798096	976.053284	975.960266	975.996338	976.005676	976.007629	976.008301

Boundary 20

j	0	1	2	3	4	5	6	7
ax	-328.000000	-224.025162	-222.562744	-222.149506	-222.147766	-222.158691	-222.162857	-222.164230
ay	1138.000000	1180.804810	1202.841187	1203.555908	1203.607666	1203.618286	1203.620972	1203.621704
ax	-81.000000	76.996605	90.802467	91.484230	91.543045	91.560333	91.566025	91.568008
ay	1010.000000	1040.268311	1051.823120	1051.885986	1051.861206	1051.853394	1051.851074	1051.850464
ax	-76.000000	62.123814	76.051254	76.896088	77.025352	77.067436	77.081314	77.085602
ay	1138.000000	1134.334351	1151.754028	1151.805054	1151.749268	1151.735840	1151.732422	1151.731689
ax	-70.000000	37.062527	48.237804	48.694942	48.731117	48.745377	48.750622	48.752579
ay	1266.000000	1227.945068	1250.848755	1250.985718	1250.935425	1250.926147	1250.924316	1250.923950
ax	-322.000000	-236.396744	-235.238220	-234.944885	-234.948181	-234.956970	-234.960083	-234.961182
ay	1266.000000	1278.788574	1306.642090	1307.318848	1307.360352	1307.376587	1307.382446	1307.384766
ax	-574.000000	-519.304077	-531.529175	-531.596191	-531.590698	-531.598022	-531.601318	-531.602844
ay	1266.000000	1304.516968	1331.675415	1332.710815	1332.709839	1332.694580	1332.689209	1332.687378

Boundary 21

j	0	1	2	3	4	5	6	7
ax	-481.000000	-364.960236	-372.666931	-372.766815	-372.840302	-372.873688	-372.885376	-372.888977
ay	884.000000	1056.562988	1064.116089	1064.233765	1064.330933	1064.359497	1064.366699	1064.368774
ax	-359.000000	-160.521164	-158.331680	-157.522232	-157.376160	-157.345047	-157.337189	-157.334549
ay	832.000000	972.798096	976.053284	975.960266	975.996338	976.005676	976.007629	976.008301
ax	-467.000000	-377.475159	-385.389954	-385.652802	-385.811279	-385.871948	-385.891724	-385.897858
ay	1049.000000	1138.872314	1152.204102	1152.104370	1152.009521	1151.979248	1151.969849	1151.967529
ax	-574.000000	-519.304077	-531.529175	-531.596191	-531.590698	-531.598022	-531.601318	-531.602844
ay	1266.000000	1304.516968	1331.675415	1332.710815	1332.709839	1332.694580	1332.689209	1332.687378
ax	-588.000000	-529.107849	-537.356689	-536.338257	-536.153870	-536.126099	-536.121094	-536.120972
ay	1101.000000	1178.084351	1194.367188	1194.521606	1194.467285	1194.448975	1194.443237	1194.441772
ax	-602.000000	-576.135132	-587.777527	-587.084900	-586.932983	-586.907166	-586.901794	-586.901367
ay	937.000000	1086.204956	1091.641968	1090.930664	1090.884766	1090.886719	1090.888306	1090.889038

Boundary 22

j	0	1	2	3	4	5	6	7
ax	-660.000000	-717.865356	-740.518982	-741.161743	-741.176941	-741.175659	-741.174561	-741.174744
ay	854.000000	971.532104	976.049744	975.404175	975.346069	975.343323	975.343933	975.344482
ax	-602.000000	-576.135132	-587.777527	-587.084900	-586.932983	-586.907166	-586.901794	-586.901367
ay	937.000000	1086.204956	1091.641968	1090.930664	1090.884766	1090.886719	1090.888306	1090.889038
ax	-588.000000	-529.107849	-537.356689	-536.338257	-536.153870	-536.126099	-536.121094	-536.120972
ay	1101.000000	1178.084351	1194.367188	1194.521606	1194.467285	1194.448975	1194.443237	1194.441772
ax	-574.000000	-519.304077	-531.529175	-531.596191	-531.590698	-531.598022	-531.601318	-531.602844
ay	1266.000000	1304.516968	1331.675415	1332.710815	1332.709839	1332.694580	1332.689209	1332.687378
ax	-646.000000	-627.380737	-654.109497	-655.430725	-655.483276	-655.480408	-655.478149	-655.477478
ay	1018.000000	1115.184082	1131.668945	1132.273315	1132.310303	1132.311890	1132.311035	1132.310669
ax	-718.000000	-824.787292	-847.574890	-848.237000	-848.280579	-848.289734	-848.292114	-848.293274
ay	771.000000	831.881836	834.198059	833.756470	833.707520	833.703064	833.703125	833.703247

Boundary 23

j	0	1	2	3	4	5	6	7
ax	-114.000000	66.918976	69.995499	69.814041	69.724442	69.697433	69.689407	69.687378
ay	675.000000	755.483582	768.849915	769.781677	770.064392	770.149475	770.175110	770.182617
ax	52.000000	236.662689	247.534363	247.612045	247.614655	247.617966	247.619202	247.620010
ay	704.000000	769.339233	783.263855	783.671753	783.728882	783.742004	783.745422	783.745972
ax	-15.000000	150.017181	158.981827	158.383591	158.112457	158.028366	158.002289	157.994598
ay	857.000000	910.172180	922.585999	922.574585	922.522034	922.503784	922.497681	922.495483
ax	-81.000000	76.996605	90.802467	91.484230	91.543045	91.560333	91.566025	91.568008
ay	1010.000000	1040.268311	1051.823120	1051.885986	1051.861206	1051.853394	1051.851074	1051.850464
ax	-181.000000	-1.815540	4.934396	5.587512	5.694028	5.725240	5.735265	5.738637
ay	828.000000	902.223816	910.529907	910.478760	910.472534	910.468140	910.466003	910.465332
ax	-280.000000	-51.421230	-53.034142	-52.822388	-52.798317	-52.795628	-52.795059	-52.794598
ay	647.000000	705.234070	710.897400	710.542847	710.507568	710.502686	710.502075	710.502075

Boundary 24

j	0	1	2	3	4	5	6	7
ax	-320.000000	-86.510666	-82.387367	-81.020905	-80.700935	-80.616379	-80.592461	-80.585312
ay	739.000000	842.220642	848.998413	849.139954	849.220276	849.242493	849.248596	849.250244
ax	-280.000000	-51.421230	-53.034142	-52.822388	-52.798317	-52.795628	-52.795059	-52.794598
ay	647.000000	705.234070	710.897400	710.542847	710.507568	710.502686	710.502075	710.502075
ax	-181.000000	-1.815540	4.934396	5.587512	5.694028	5.725240	5.735265	5.738637
ay	828.000000	902.223816	910.529907	910.478760	910.472534	910.468140	910.466003	910.465332
ax	-81.000000	76.996605	90.802467	91.484230	91.543045	91.560333	91.566025	91.568008
ay	1010.000000	1040.268311	1051.823120	1051.885986	1051.861206	1051.853394	1051.851074	1051.850464
ax	-220.000000	-56.609509	-48.226143	-47.217823	-47.044521	-46.999908	-46.987282	-46.983383
ay	921.000000	989.040649	993.239502	992.568909	992.410889	992.365662	992.352295	992.348511
ax	-359.000000	-160.521164	-158.331680	-157.522232	-157.376160	-157.345047	-157.337189	-157.334549
ay	832.000000	972.798096	976.053284	975.960266	975.996338	976.005676	976.007629	976.008301

Boundary 25

j	0	1	2	3	4	5	6	7
ax	-15.000000	150.017181	158.981827	158.383591	158.112457	158.028366	158.002289	157.994598
ay	857.000000	910.172180	922.585999	922.574585	922.522034	922.503784	922.497681	922.495483
ax	52.000000	236.662689	247.534363	247.612045	247.614655	247.617966	247.619202	247.620010
ay	704.000000	769.339233	783.263855	783.671753	783.728882	783.742004	783.745422	783.745972
ax	70.000000	240.595016	251.832062	252.026932	252.044250	252.051895	252.054703	252.056168
ay	746.000000	800.373230	814.939819	815.232056	815.231140	815.225037	815.221985	815.220642

ax	88.000000	247.369965	260.446747	260.861420	260.902069	260.914001	260.917694	260.919250
ay	788.000000	837.492249	854.016907	854.479919	854.503601	854.504944	854.504395	854.503601
ax	3.000000	164.313965	176.773376	176.884491	176.799545	176.773239	176.765610	176.763718
ay	899.000000	945.573975	959.837708	960.321960	960.402649	960.426575	960.434021	960.436401
ax	-81.000000	76.996605	90.802467	91.484230	91.543045	91.560333	91.566025	91.568008
ay	1010.000000	1040.268311	1051.823120	1051.885986	1051.861206	1051.853394	1051.851074	1051.850464

Boundary 26

j	0	1	2	3	4	5	6	7
ax	0.000000	67.846100	77.167938	77.483231	77.566154	77.590698	77.597794	77.599876
ay	30.000000	42.757633	28.721792	28.934359	29.076349	29.118225	29.130497	29.133595
ax	209.000000	216.748550	230.470337	230.565735	230.557724	230.556213	230.555466	230.555084
ay	-116.000000	-70.008408	-89.856010	-90.265198	-90.251999	-90.248253	-90.247810	-90.248100
ax	152.000000	199.684860	205.299484	204.400558	204.228729	204.187042	204.175003	204.171356
ay	126.000000	146.655777	130.652390	130.210464	130.176285	130.167282	130.164413	130.163223
ax	96.000000	247.986237	256.860596	256.331604	256.277588	256.272797	256.272522	256.272644
ay	368.000000	361.859772	352.452942	352.307251	352.267914	352.252960	352.248505	352.246674
ax	-57.000000	83.383804	92.590576	92.900581	92.992111	93.016434	93.022697	93.024643
ay	272.000000	201.444473	184.895584	184.062637	183.905655	183.863281	183.851074	183.847168
ax	-209.000000	-116.457123	-114.855957	-115.036400	-115.125717	-115.159996	-115.172058	-115.175537
ay	176.000000	89.588737	73.369965	72.946106	72.916824	72.915314	72.916138	72.916588

Boundary 27

j	0	1	2	3	4	5	6	7
ax	152.000000	199.684860	205.299484	204.400558	204.228729	204.187042	204.175003	204.171356
ay	126.000000	146.655777	130.652390	130.210464	130.176285	130.167282	130.164413	130.163223
ax	209.000000	216.748550	230.470337	230.565735	230.557724	230.556213	230.555466	230.555084
ay	-116.000000	-70.008408	-89.856010	-90.265198	-90.251999	-90.248253	-90.247810	-90.248100
ax	277.000000	276.894684	298.921265	299.384338	299.378784	299.374817	299.373871	299.373657
ay	103.000000	145.887970	127.842056	127.208672	127.158447	127.146225	127.142326	127.141075
ax	345.000000	352.492615	382.502197	383.400391	383.446960	383.454224	383.456055	383.456665
ay	322.000000	382.852142	373.316437	373.312592	373.346436	373.354889	373.357056	373.357269
ax	220.000000	293.925903	312.704651	312.935486	312.953217	312.960815	312.963196	312.964081
ay	345.000000	379.053986	369.769897	369.769867	369.796112	369.802826	369.805054	369.805237
ax	96.000000	247.986237	256.860596	256.331604	256.277588	256.272797	256.272522	256.272644
ay	368.000000	361.859772	352.452942	352.307251	352.267914	352.252960	352.248505	352.246674

Boundary 28

j	0	1	2	3	4	5	6	7
ax	0.000000	67.846100	77.167938	77.483231	77.566154	77.590698	77.597794	77.599876
ay	30.000000	42.757633	28.721792	28.934359	29.076349	29.118225	29.130497	29.133595
ax	-209.000000	-116.457123	-114.855957	-115.036400	-115.125717	-115.159996	-115.172058	-115.175537
ay	176.000000	89.588737	73.369965	72.946106	72.916824	72.915314	72.916138	72.916588
ax	-235.000000	-151.728043	-153.417877	-153.889969	-154.048370	-154.101562	-154.119064	-154.124207
ay	119.000000	65.497154	46.723122	46.413864	46.449669	46.468517	46.475586	46.477711
ax	-261.000000	-187.053116	-191.146149	-191.519791	-191.607437	-191.631821	-191.638718	-191.640198
ay	62.000000	36.286488	13.835632	13.257878	13.234482	13.234234	13.235188	13.235323
ax	-26.000000	40.397327	47.077961	47.209564	47.236458	47.241711	47.242321	47.242126
ay	-27.000000	-1.861507	-21.052298	-21.294720	-21.272797	-21.274035	-21.277149	-21.279043
ax	209.000000	216.748550	230.470337	230.565735	230.557724	230.556213	230.555466	230.555084
ay	-116.000000	-70.008408	-89.856010	-90.265198	-90.251999	-90.248253	-90.247810	-90.248100

Boundary 29

j	0	1	2	3	4	5	6	7
ax	3.000000	164.313965	176.773376	176.884491	176.799545	176.773239	176.765610	176.763718
ay	899.000000	945.573975	959.837708	960.321960	960.402649	960.426575	960.434021	960.436401
ax	88.000000	247.369965	260.446747	260.861420	260.902069	260.914001	260.917694	260.919250
ay	788.000000	837.492249	854.016907	854.479919	854.503601	854.504944	854.504395	854.503601
ax	257.000000	351.737762	366.271149	366.544617	366.542938	366.541748	366.540924	366.541321
ay	893.000000	910.187378	924.946106	925.184265	925.174927	925.173889	925.174255	925.174377
ax	426.000000	489.057281	507.577911	508.145294	508.213715	508.228607	508.231567	508.232666
ay	998.000000	985.695801	997.128418	997.058044	996.993469	996.980713	996.978455	996.978210
ax	172.000000	278.796997	291.772949	291.829376	291.759155	291.739960	291.734467	291.733368
ay	1004.000000	1000.712402	1012.834106	1013.247620	1013.342468	1013.376221	1013.387878	1013.391602
ax	-81.000000	76.996605	90.802467	91.484230	91.543045	91.560333	91.566025	91.568008
ay	1010.000000	1040.268311	1051.823120	1051.885986	1051.861206	1051.853394	1051.851074	1051.850464

Boundary 30

j	0	1	2	3	4	5	6	7
ax	178.000000	269.573975	281.550995	281.151215	280.956024	280.903778	280.889160	280.885834
ay	1132.000000	1091.242310	1104.302124	1103.653442	1103.441650	1103.389282	1103.375610	1103.372192
ax	426.000000	489.057281	507.577911	508.145294	508.213715	508.228607	508.231567	508.232666
ay	998.000000	985.695801	997.128418	997.058044	996.993469	996.980713	996.978455	996.978210
ax	449.000000	503.623749	523.640015	524.828918	525.043457	525.093445	525.104919	525.108154
ay	1135.000000	1068.713135	1081.130981	1081.205322	1081.182739	1081.180054	1081.180176	1081.180664
ax	473.000000	504.891846	522.742371	523.724731	523.892517	523.931702	523.940796	523.943420
ay	1273.000000	1145.384155	1158.782715	1158.965698	1158.981079	1158.988159	1158.990967	1158.991821
ax	201.000000	272.615875	285.144287	285.231659	285.169434	285.151978	285.146881	285.146027
ay	1269.000000	1180.678711	1199.302734	1199.764526	1199.826050	1199.840942	1199.844238	1199.844849
ax	-70.000000	37.062527	48.237804	48.694942	48.731117	48.745377	48.750622	48.752579
ay	1266.000000	1227.945068	1250.848755	1250.985718	1250.935425	1250.926147	1250.924316	1250.923950

Boundary 31

j	0	1	2	3	4	5	6	7
ax	172.000000	278.796997	291.772949	291.829376	291.759155	291.739960	291.734467	291.733368
ay	1004.000000	1000.712402	1012.834106	1013.247620	1013.342468	1013.376221	1013.387878	1013.391602
ax	426.000000	489.057281	507.577911	508.145294	508.213715	508.228607	508.231567	508.232666
ay	998.000000	985.695801	997.128418	997.058044	996.993469	996.980713	996.978455	996.978210
ax	178.000000	269.573975	281.550995	281.151215	280.956024	280.903778	280.889160	280.885834
ay	1132.000000	1091.242310	1104.302124	1103.653442	1103.441650	1103.389282	1103.375610	1103.372192
ax	-70.000000	37.062527	48.237804	48.694942	48.731117	48.745377	48.750622	48.752579
ay	1266.000000	1227.945068	1250.848755	1250.985718	1250.935425	1250.926147	1250.924316	1250.923950
ax	-76.000000	62.123814	76.051254	76.896088	77.025352	77.067436	77.081314	77.085602
ay	1138.000000	1134.334351	1151.754028	1151.805054	1151.749268	1151.735840	1151.732422	1151.731689
ax	-81.000000	76.996605	90.802467	91.484230	91.543045	91.560333	91.566025	91.568008
ay	1010.000000	1040.268311	1051.823120	1051.885986	1051.861206	1051.853394	1051.851074	1051.850464

Boundary 32

j	0	1	2	3	4	5	6	7
ax	-508.000000	-475.961029	-485.354523	-485.831726	-485.937073	-485.970551	-485.981049	-485.984375
ay	-337.000000	-346.744110	-362.026337	-362.027832	-361.955627	-361.935486	-361.929993	-361.928497
ax	-384.000000	-330.976868	-343.208038	-343.614410	-343.660553	-343.669586	-343.671082	-343.671356
ay	-52.000000	-41.876698	-63.767120	-64.073174	-64.068443	-64.068176	-64.068214	-64.068535
ax	-551.000000	-519.233948	-533.930603	-534.523071	-534.612732	-534.634155	-534.639526	-534.641113
ay	-96.000000	-82.599579	-96.065147	-95.622612	-95.483765	-95.456947	-95.452316	-95.451965
ax	-718.000000	-694.314575	-707.529114	-707.589661	-707.548523	-707.536987	-707.534058	-707.533203
ay	-140.000000	-142.594910	-153.823441	-153.566071	-153.468155	-153.448730	-153.445267	-153.444916
ax	-675.000000	-665.012695	-675.719788	-676.108032	-676.150146	-676.157410	-676.158813	-676.158875
ay	-381.000000	-398.189575	-411.565186	-411.699310	-411.668793	-411.659821	-411.657410	-411.656769
ax	-632.000000	-605.955505	-612.331116	-612.559814	-612.589111	-612.595947	-612.597290	-612.597778
ay	-621.000000	-656.332336	-672.236389	-672.993042	-673.106201	-673.132446	-673.138916	-673.140381

Boundary 33

j	0	1	2	3	4	5	6	7
ax	-407.000000	-366.252258	-371.775269	-372.012299	-372.045685	-372.052826	-372.053528	-372.053497
ay	-422.000000	-436.782623	-453.769836	-453.923340	-453.897522	-453.894470	-453.895142	-453.895966
ax	-632.000000	-605.955505	-612.331116	-612.559814	-612.589111	-612.595947	-612.597290	-612.597778
ay	-621.000000	-656.332336	-672.236389	-672.993042	-673.106201	-673.132446	-673.138916	-673.140381
ax	-473.000000	-445.257751	-450.960632	-451.328308	-451.397278	-451.413361	-451.416626	-451.417267
ay	-632.000000	-677.382812	-695.708557	-696.872070	-697.079712	-697.130676	-697.143982	-697.147583
ax	-314.000000	-283.109833	-286.604218	-286.615021	-286.598267	-286.595001	-286.594696	-286.594849
ay	-642.000000	-670.633667	-687.278076	-687.868408	-687.929932	-687.944031	-687.947876	-687.949036
ax	-248.000000	-188.887939	-189.167221	-188.532028	-188.359543	-188.314621	-188.302673	-188.299805
ay	-433.000000	-442.524170	-461.321655	-461.857910	-461.925507	-461.944611	-461.950134	-461.952118
ax	-182.000000	-113.201660	-114.481316	-114.470642	-114.467087	-114.465126	-114.463890	-114.463287
ay	-223.000000	-209.930298	-231.222107	-231.471878	-231.453506	-231.447540	-231.445206	-231.444458

Boundary 34

j	0	1	2	3	4	5	6	7
ax	-508.000000	-475.961029	-485.354523	-485.831726	-485.937073	-485.970551	-485.981049	-485.984375
ay	-337.000000	-346.744110	-362.026337	-362.027832	-361.955627	-361.935486	-361.929993	-361.928497
ax	-632.000000	-605.955505	-612.331116	-612.559814	-612.589111	-612.595947	-612.597290	-612.597778
ay	-621.000000	-656.332336	-672.236389	-672.993042	-673.106201	-673.132446	-673.138916	-673.140381
ax	-407.000000	-366.252258	-371.775269	-372.012299	-372.045685	-372.052826	-372.053528	-372.053497
ay	-422.000000	-436.782623	-453.769836	-453.923340	-453.897522	-453.894470	-453.895142	-453.895966

ax	-182.000000	-113.201660	-114.481316	-114.470642	-114.467087	-114.465126	-114.463890	-114.463287
ay	-223.000000	-209.930298	-231.222107	-231.471878	-231.453506	-231.447540	-231.445206	-231.444458
ax	-283.000000	-214.357971	-219.322754	-219.299347	-219.268433	-219.256500	-219.251923	-219.250107
ay	-138.000000	-122.338287	-142.560150	-142.371857	-142.224930	-142.182190	-142.169495	-142.166077
ax	-384.000000	-330.976868	-343.208038	-343.614410	-343.660553	-343.669586	-343.671082	-343.671356
ay	-52.000000	-41.876698	-63.767120	-64.073174	-64.068443	-64.068176	-64.068214	-64.068535

Boundary 35

j	0	1	2	3	4	5	6	7
ax	-143.000000	-4.114292	-3.765493	-4.286761	-4.392023	-4.419407	-4.427302	-4.429573
ay	355.000000	297.703583	294.629120	295.562988	295.768433	295.812347	295.822144	295.824005
ax	96.000000	247.986237	256.860596	256.331604	256.277588	256.272797	256.272522	256.272644
ay	368.000000	361.859772	352.452942	352.307251	352.267914	352.252960	352.248505	352.246674
ax	-92.000000	79.352852	78.182632	77.094780	76.841232	76.774734	76.756378	76.751442
ay	507.000000	512.946716	508.816864	507.994446	507.790741	507.733887	507.718323	507.713623
ax	-280.000000	-51.421230	-53.034142	-52.822388	-52.798317	-52.795628	-52.795059	-52.794598
ay	647.000000	705.234070	710.897400	710.542847	710.507568	710.502686	710.502075	710.502075
ax	-331.000000	-155.424988	-159.522842	-159.285324	-159.247864	-159.248703	-159.251755	-159.253159
ay	495.000000	428.974091	426.057251	425.707977	425.647705	425.631317	425.626862	425.625275
ax	-381.000000	-301.078796	-313.058929	-313.365845	-313.384338	-313.386597	-313.386993	-313.386993
ay	343.000000	180.171432	174.121124	174.408585	174.489548	174.511353	174.517822	174.519348

Boundary 36

j	0	1	2	3	4	5	6	7
ax	-143.000000	-4.114292	-3.765493	-4.286761	-4.392023	-4.419407	-4.427302	-4.429573
ay	355.000000	297.703583	294.629120	295.562988	295.768433	295.812347	295.822144	295.824005
ax	-381.000000	-301.078796	-313.058929	-313.365845	-313.384338	-313.386597	-313.386993	-313.386993
ay	343.000000	180.171432	174.121124	174.408585	174.489548	174.511353	174.517822	174.519348
ax	-295.000000	-223.457565	-230.657379	-231.156418	-231.259094	-231.286713	-231.294540	-231.296783
ay	259.000000	143.990662	133.934052	133.633774	133.548218	133.519073	133.509979	133.506653
ax	-209.000000	-116.457123	-114.855957	-115.036400	-115.125717	-115.159996	-115.172058	-115.175537
ay	176.000000	89.588737	73.369965	72.946106	72.916824	72.915314	72.916138	72.916588
ax	-57.000000	83.383804	92.590576	92.900581	92.992111	93.016434	93.022697	93.024643
ay	272.000000	201.444473	184.895584	184.062637	183.905655	183.863281	183.851074	183.847168
ax	96.000000	247.986237	256.860596	256.331604	256.277588	256.272797	256.272522	256.272644
ay	368.000000	361.859772	352.452942	352.307251	352.267914	352.252960	352.248505	352.246674

Boundary 37

j	0	1	2	3	4	5	6	7
ax	-92.000000	79.352852	78.182632	77.094780	76.841232	76.774734	76.756378	76.751442
ay	507.000000	512.946716	508.816864	507.994446	507.790741	507.733887	507.718323	507.713623
ax	96.000000	247.986237	256.860596	256.331604	256.277588	256.272797	256.272522	256.272644
ay	368.000000	361.859772	352.452942	352.307251	352.267914	352.252960	352.248505	352.246674
ax	74.000000	268.459412	282.375854	282.891083	283.077942	283.139130	283.157684	283.163422
ay	536.000000	572.173950	574.971558	575.066589	575.050232	575.041565	575.038818	575.037903
ax	52.000000	236.662689	247.534363	247.612045	247.614655	247.617966	247.619202	247.620010
ay	704.000000	769.339233	783.263855	783.671753	783.728882	783.742004	783.745422	783.745972
ax	-114.000000	66.918976	69.995499	69.814041	69.724442	69.697433	69.689407	69.687378
ay	675.000000	755.483582	768.849915	769.781677	770.064392	770.149475	770.175110	770.182617
ax	-280.000000	-51.421230	-53.034142	-52.822388	-52.798317	-52.795628	-52.795059	-52.794598
ay	647.000000	705.234070	710.897400	710.542847	710.507568	710.502686	710.502075	710.502075

Boundary 38

j	0	1	2	3	4	5	6	7
ax	11.000000	8.010808	9.322116	9.396812	9.402699	9.402502	9.401915	9.401873
ay	-998.000000	-982.087769	-994.324402	-994.410400	-994.368408	-994.356445	-994.353333	-994.352722
ax	23.000000	26.617027	28.680613	28.622435	28.587276	28.576437	28.572962	28.572025
ay	-723.000000	-707.588257	-722.452820	-722.578003	-722.544495	-722.536255	-722.534424	-722.534119
ax	-127.000000	-112.740776	-112.962563	-112.839760	-112.801254	-112.790840	-112.787933	-112.786957
ay	-707.000000	-700.784973	-716.116394	-716.326111	-716.314819	-716.312866	-716.313049	-716.313477
ax	-277.000000	-258.348419	-262.448639	-262.662354	-262.692749	-262.700958	-262.703369	-262.703949
ay	-690.000000	-703.876953	-721.055725	-721.638123	-721.707703	-721.724182	-721.728577	-721.730042
ax	-139.000000	-138.894211	-140.163589	-139.990509	-139.923492	-139.904510	-139.898956	-139.897232
ay	-982.000000	-969.157898	-981.434143	-981.381958	-981.300537	-981.278320	-981.272644	-981.271362
ax	-1.000000	-2.835188	-2.105273	-1.928004	-1.883902	-1.873126	-1.870437	-1.869430
ay	-1273.000000	-1252.154663	-1263.010254	-1263.305054	-1263.321655	-1263.323730	-1263.323975	-1263.323975

Boundary 39

j	0	1	2	3	4	5	6	7
ax	-180.000000	-179.251083	-180.738571	-180.834335	-180.842010	-180.842087	-180.841232	-180.841125
ay	-1270.000000	-1246.014648	-1257.106323	-1257.591919	-1257.660156	-1257.675415	-1257.678955	-1257.679932
ax	-1.000000	-2.835188	-2.105273	-1.928004	-1.883902	-1.873126	-1.870437	-1.869430
ay	-1273.000000	-1252.154663	-1263.010254	-1263.305054	-1263.321655	-1263.323730	-1263.323975	-1263.323975
ax	-139.000000	-138.894211	-140.163589	-139.990509	-139.923492	-139.904510	-139.898956	-139.897232
ay	-982.000000	-969.157898	-981.434143	-981.381958	-981.300537	-981.278320	-981.272644	-981.271362
ax	-277.000000	-258.348419	-262.448639	-262.662354	-262.692749	-262.700958	-262.703369	-262.703949
ay	-690.000000	-703.876953	-721.055725	-721.638123	-721.707703	-721.724182	-721.728577	-721.730042
ax	-318.000000	-319.687927	-325.392242	-326.105316	-326.237579	-326.264709	-326.270142	-326.271240
ay	-978.000000	-968.853210	-980.549438	-980.512024	-980.437012	-980.416077	-980.410645	-980.409363
ax	-358.000000	-355.371552	-359.046722	-359.479065	-359.556396	-359.571442	-359.574585	-359.575073
ay	-1266.000000	-1235.484741	-1244.872681	-1245.121094	-1245.151611	-1245.161133	-1245.163940	-1245.164795

Boundary 40

j	0	1	2	3	4	5	6	7
ax	13.000000	56.299335	62.890556	62.987972	62.997253	62.997810	62.997349	62.996777
ay	-170.000000	-143.784943	-164.867798	-165.372009	-165.411942	-165.422195	-165.424576	-165.425293
ax	209.000000	216.748550	230.470337	230.565735	230.557724	230.556213	230.555466	230.555084
ay	-116.000000	-70.008408	-89.856010	-90.265198	-90.251999	-90.248253	-90.247810	-90.248100
ax	-26.000000	40.397327	47.077961	47.209564	47.236458	47.241711	47.242321	47.242126
ay	-27.000000	-1.861507	-21.052298	-21.294720	-21.272797	-21.274035	-21.277149	-21.279043
ax	-261.000000	-187.053116	-191.146149	-191.519791	-191.607437	-191.631821	-191.638718	-191.640198
ay	62.000000	36.286488	13.835632	13.257878	13.234482	13.234234	13.235188	13.235323
ax	-222.000000	-153.503540	-157.642044	-158.251389	-158.408966	-158.451126	-158.462402	-158.464844
ay	-81.000000	-73.347336	-94.987000	-95.391586	-95.403290	-95.405991	-95.406517	-95.406982
ax	-182.000000	-113.201660	-114.481316	-114.470642	-114.467087	-114.465126	-114.463890	-114.463287
ay	-223.000000	-209.930298	-231.222107	-231.471878	-231.453506	-231.447540	-231.445206	-231.444458

Boundary 41

j	0	1	2	3	4	5	6	7
ax	-146.000000	-128.640656	-128.992996	-128.832031	-128.781937	-128.768738	-128.764771	-128.763474
ay	-683.000000	-679.829102	-693.909485	-693.836060	-693.737000	-693.709351	-693.701782	-693.699829
ax	-314.000000	-283.109833	-286.604218	-286.615021	-286.598267	-286.595001	-286.594696	-286.594849
ay	-642.000000	-670.633667	-687.278076	-687.868408	-687.929932	-687.944031	-687.947876	-687.949036
ax	-296.000000	-271.916290	-275.895630	-276.059174	-276.082336	-276.090210	-276.093079	-276.094086
ay	-666.000000	-686.895996	-703.841553	-704.435913	-704.503052	-704.518677	-704.522766	-704.523926
ax	-277.000000	-258.348419	-262.448639	-262.662354	-262.692749	-262.700958	-262.703369	-262.703949
ay	-690.000000	-703.876953	-721.055725	-721.638123	-721.707703	-721.724182	-721.728577	-721.730042
ax	-127.000000	-112.740776	-112.962563	-112.839760	-112.801254	-112.790840	-112.787933	-112.786957
ay	-707.000000	-700.784973	-716.116394	-716.326111	-716.314819	-716.312866	-716.313049	-716.313477
ax	23.000000	26.617027	28.680613	28.622435	28.587276	28.576437	28.572962	28.572025
ay	-723.000000	-707.588257	-722.452820	-722.578003	-722.544495	-722.536255	-722.534424	-722.534119

Boundary 42

j	0	1	2	3	4	5	6	7
ax	-283.000000	-214.357971	-219.322754	-219.299347	-219.268433	-219.256500	-219.251923	-219.250107
ay	-138.000000	-122.338287	-142.560150	-142.371857	-142.224930	-142.182190	-142.169495	-142.166077
ax	-182.000000	-113.201660	-114.481316	-114.470642	-114.467087	-114.465126	-114.463890	-114.463287
ay	-223.000000	-209.930298	-231.222107	-231.471878	-231.453506	-231.447540	-231.445206	-231.444458
ax	-222.000000	-153.503540	-157.642044	-158.251389	-158.408966	-158.451126	-158.462402	-158.464844
ay	-81.000000	-73.347336	-94.987000	-95.391586	-95.403290	-95.405991	-95.406517	-95.406982
ax	-261.000000	-187.053116	-191.146149	-191.519791	-191.607437	-191.631821	-191.638718	-191.640198
ay	62.000000	36.286488	13.835632	13.257878	13.234482	13.234234	13.235188	13.235323
ax	-323.000000	-263.424225	-270.307770	-270.352295	-270.319489	-270.305634	-270.300446	-270.298584
ay	5.000000	-4.846634	-28.491846	-29.274696	-29.382883	-29.412550	-29.420986	-29.423809
ax	-384.000000	-330.976868	-343.208038	-343.614410	-343.660553	-343.669586	-343.671082	-343.671356
ay	-52.000000	-41.876698	-63.767120	-64.073174	-64.068443	-64.068176	-64.068214	-64.068535

Boundary 43

j	0	1	2	3	4	5	6	7
ax	-80.000000	-48.133102	-48.384205	-48.444397	-48.477474	-48.488392	-48.491611	-48.492336
ay	-473.000000	-465.602417	-484.198853	-484.643280	-484.687897	-484.700714	-484.704437	-484.705750
ax	-182.000000	-113.201660	-114.481316	-114.470642	-114.467087	-114.465126	-114.463890	-114.463287
ay	-223.000000	-209.930298	-231.222107	-231.471878	-231.453506	-231.447540	-231.445206	-231.444458
ax	-248.000000	-188.887939	-189.167221	-188.532028	-188.359543	-188.314621	-188.302673	-188.299805
ay	-433.000000	-442.524170	-461.321655	-461.857910	-461.925507	-461.944611	-461.950134	-461.952118

```
ax  -314.000000  -283.109833  -286.604218  -286.615021  -286.598267  -286.595001  -286.594696  -286.594849
ay  -642.000000  -670.633667  -687.278076  -687.868408  -687.929932  -687.944031  -687.947876  -687.949036

ax  -146.000000  -128.640656  -128.992996  -128.832031  -128.781937  -128.768738  -128.764771  -128.763474
ay  -683.000000  -679.829102  -693.909485  -693.836060  -693.737000  -693.709351  -693.701782  -693.699829

ax   23.000000   26.617027   28.680613   28.622435   28.587276   28.576437   28.572962   28.572025
ay  -723.000000  -707.588257  -722.452820  -722.578003  -722.544495  -722.536255  -722.534424  -722.534119

Boundary  44

j    0        1        2        3        4        5        6        7
ax   13.000000   56.299335   62.890556   62.987972   62.997253   62.997810   62.997349   62.996777
ay  -170.000000  -143.784943  -164.867798  -165.372009  -165.411942  -165.422195  -165.424576  -165.425293

ax  -182.000000  -113.201660  -114.481316  -114.470642  -114.467087  -114.465126  -114.463890  -114.463287
ay  -223.000000  -209.930298  -231.222107  -231.471878  -231.453506  -231.447540  -231.445206  -231.444458

ax   -80.000000   -48.133102   -48.384205   -48.444397   -48.477474   -48.488392   -48.491611   -48.492336
ay  -473.000000  -465.602417  -484.198853  -484.643280  -484.687897  -484.700714  -484.704437  -484.705750

ax   23.000000   26.617027   28.680613   28.622435   28.587276   28.576437   28.572962   28.572025
ay  -723.000000  -707.588257  -722.452820  -722.578003  -722.544495  -722.536255  -722.534424  -722.534119

ax   116.000000   126.679756   133.631180   133.651718   133.631989   133.626266   133.624680   133.624588
ay  -420.000000  -396.886841  -414.914215  -415.272400  -415.289917  -415.295074  -415.296661  -415.297424

ax   209.000000   216.748550   230.470337   230.565735   230.557724   230.556213   230.555466   230.555084
ay  -116.000000   -70.008408   -89.856010   -90.265198   -90.251999   -90.248253   -90.247810   -90.248100


Non-Boundary Point Table
j   0    1    2    3    4    5    6    7

End of Convergence Tables
```

C.3.4 File `RLInt.out.plot`

This is a comma delimited table that is useful for making plots. It contains region information and the regions' boundaries at the initial and final iteration. It was created by specifying `icheck = 1`.

```
File: RLInt.out.plot
Input scale flag  iara  =      1
Target RMS error:  relerr = 5.00E-05
Achieved RMS error  reler  = 1.90E-05
DM h function       dmhsum = 1.72E-10
Target total  area  aretot = 2.46E+06
Current total area  arctot = 2.46E+06
Scaling flag        iscale =      1
/ Delimited Convergence Table

k,it0 ,it    ,ci  ,j    ,armag       ,arak       ,ara0k   ,reler    ,RMS     ,DMhsum   ,trgtArea  ,curtArea
 1,  0,  0,  1.0000,  0, 4.0574E-04,-2.462143E+06,-9.990000E+02, 0.00E+00, 7.03E-01, 3.58E-01, 2.4621E+06, 2.4621E+06,
 1,  1,  0,  1.0000,  1, 4.2612E-04,-2.344420E+06,-9.990000E+02, 0.00E+00, 1.23E-01, 5.80E-03, 2.4621E+06, 2.3444E+06,
 1,  2,  0,  1.0000,  2, 4.0614E-04,-2.459751E+06,-9.990000E+02, 0.00E+00, 1.10E-02, 7.05E-05, 2.4621E+06, 2.4598E+06,
 1,  3,  0,  1.0000,  3, 4.0575E-04,-2.462123E+06,-9.990000E+02, 0.00E+00, 2.40E-03, 3.63E-06, 2.4621E+06, 2.4621E+06,
 1,  4,  0,  1.0000,  4, 4.0574E-04,-2.462142E+06,-9.990000E+02, 0.00E+00, 6.80E-04, 2.70E-07, 2.4621E+06, 2.4621E+06,
 1,  5,  0,  1.0000,  5, 4.0574E-04,-2.462142E+06,-9.990000E+02, 0.00E+00, 2.04E-04, 2.23E-08, 2.4621E+06, 2.4621E+06,
 1,  6,  0,  1.0000,  6, 4.0574E-04,-2.462142E+06,-9.990000E+02, 0.00E+00, 6.19E-05, 1.91E-09, 2.4621E+06, 2.4621E+06,
 1,  7,  0,  1.0000,  7, 4.0574E-04,-2.462144E+06,-9.990000E+02, 0.00E+00, 1.90E-05, 1.72E-10, 2.4621E+06, 2.4621E+06,

k,it0 ,it    ,ci  ,j    ,armag       ,arak       ,ara0k   ,reler    ,RMS     ,DMhsum   ,trgtArea  ,curtArea
 2,  0,  0,  1.0000,  0, 7.3612E-01, 5.909750E+04, 4.350257E+04,-2.64E-01, 7.03E-01, 3.58E-01, 2.4621E+06, 2.4621E+06,
 2,  1,  0,  1.0000,  1, 1.1211E+00, 3.880343E+04, 4.350257E+04, 1.21E-01, 1.23E-01, 5.80E-03, 2.4621E+06, 2.3444E+06,
 2,  2,  0,  1.0000,  2, 1.0081E+00, 4.315280E+04, 4.350257E+04, 8.11E-03, 1.10E-02, 7.05E-05, 2.4621E+06, 2.4598E+06,
 2,  3,  0,  1.0000,  3, 1.0017E+00, 4.342673E+04, 4.350257E+04, 1.75E-03, 2.40E-03, 3.63E-06, 2.4621E+06, 2.4621E+06,
 2,  4,  0,  1.0000,  4, 1.0005E+00, 4.348235E+04, 4.350257E+04, 4.65E-04, 6.80E-04, 2.70E-07, 2.4621E+06, 2.4621E+06,
 2,  5,  0,  1.0000,  5, 1.0001E+00, 4.349725E+04, 4.350257E+04, 1.22E-04, 2.04E-04, 2.23E-08, 2.4621E+06, 2.4621E+06,
 2,  6,  0,  1.0000,  6, 1.0000E+00, 4.350119E+04, 4.350257E+04, 3.17E-05, 6.19E-05, 1.91E-09, 2.4621E+06, 2.4621E+06,
 2,  7,  0,  1.0000,  7, 1.0000E+00, 4.350236E+04, 4.350257E+04, 4.89E-06, 1.90E-05, 1.72E-10, 2.4621E+06, 2.4621E+06,

k,it0 ,it    ,ci  ,j    ,armag       ,arak       ,ara0k   ,reler    ,RMS     ,DMhsum   ,trgtArea  ,curtArea
 3,  0,  0,  1.0000,  0, 7.3612E-01, 1.413300E+04, 1.040352E+04,-2.64E-01, 7.03E-01, 3.58E-01, 2.4621E+06, 2.4621E+06,
 3,  1,  0,  1.0000,  1, 1.1863E+00, 8.769462E+03, 1.040352E+04, 1.86E-01, 1.23E-01, 5.80E-03, 2.4621E+06, 2.3444E+06,
 3,  2,  0,  1.0000,  2, 9.9486E-01, 1.045726E+04, 1.040352E+04,-5.14E-03, 1.10E-02, 7.05E-05, 2.4621E+06, 2.4598E+06,
 3,  3,  0,  1.0000,  3, 9.9606E-01, 1.044464E+04, 1.040352E+04,-3.94E-03, 2.40E-03, 3.63E-06, 2.4621E+06, 2.4621E+06,
 3,  4,  0,  1.0000,  4, 9.9855E-01, 1.041859E+04, 1.040352E+04,-1.45E-03, 6.80E-04, 2.70E-07, 2.4621E+06, 2.4621E+06,
 3,  5,  0,  1.0000,  5, 9.9952E-01, 1.040856E+04, 1.040352E+04,-4.85E-04, 2.04E-04, 2.23E-08, 2.4621E+06, 2.4621E+06,
 3,  6,  0,  1.0000,  6, 9.9985E-01, 1.040513E+04, 1.040352E+04,-1.55E-04, 6.19E-05, 1.91E-09, 2.4621E+06, 2.4621E+06,
 3,  7,  0,  1.0000,  7, 9.9995E-01, 1.040400E+04, 1.040352E+04,-4.66E-05, 1.90E-05, 1.72E-10, 2.4621E+06, 2.4621E+06,

k,it0 ,it    ,ci  ,j    ,armag       ,arak       ,ara0k   ,reler    ,RMS     ,DMhsum   ,trgtArea  ,curtArea
 4,  0,  0,  1.0000,  0, 7.3612E-01, 9.491549E+04, 6.986876E+04,-2.64E-01, 7.03E-01, 3.58E-01, 2.4621E+06, 2.4621E+06,
 4,  1,  0,  1.0000,  1, 9.7639E-01, 7.155814E+04, 6.986876E+04,-2.36E-02, 1.23E-01, 5.80E-03, 2.4621E+06, 2.3444E+06,
 4,  2,  0,  1.0000,  2, 9.8977E-01, 7.059095E+04, 6.986876E+04,-1.02E-02, 1.10E-02, 7.05E-05, 2.4621E+06, 2.4598E+06,
 4,  3,  0,  1.0000,  3, 9.9773E-01, 7.002760E+04, 6.986876E+04,-2.27E-03, 2.40E-03, 3.63E-06, 2.4621E+06, 2.4621E+06,
 4,  4,  0,  1.0000,  4, 9.9951E-01, 6.990314E+04, 6.986876E+04,-4.92E-04, 6.80E-04, 2.70E-07, 2.4621E+06, 2.4621E+06,
 4,  5,  0,  1.0000,  5, 9.9989E-01, 6.987611E+04, 6.986876E+04,-1.05E-04, 2.04E-04, 2.23E-08, 2.4621E+06, 2.4621E+06,
 4,  6,  0,  1.0000,  6, 9.9998E-01, 6.987036E+04, 6.986876E+04,-2.29E-05, 6.19E-05, 1.91E-09, 2.4621E+06, 2.4621E+06,
```

4, 7, 0, 1.0000, 7, 9.9999E-01, 6.986916E+04, 6.986876E+04,-5.60E-06, 1.90E-05, 1.72E-10, 2.4621E+06, 2.4621E+06,

k,it0 ,it ,ci ,j ,armag ,arak ,ara0k ,reler ,RMS ,DMhsum ,trgtArea ,curtArea
5, 0, 0, 1.0000, 0, 7.3612E-01, 2.416100E+04, 1.778528E+04,-2.64E-01, 7.03E-01, 3.58E-01, 2.4621E+06, 2.4621E+06,
5, 1, 0, 1.0000, 1, 1.1666E+00, 1.524541E+04, 1.778528E+04, 1.67E-01, 1.23E-01, 5.80E-03, 2.4621E+06, 2.3444E+06,
5, 2, 0, 1.0000, 2, 1.0082E+00, 1.764119E+04, 1.778528E+04, 8.17E-03, 1.10E-02, 7.05E-05, 2.4621E+06, 2.4598E+06,
5, 3, 0, 1.0000, 3, 1.0017E+00, 1.775464E+04, 1.778528E+04, 1.73E-03, 2.40E-03, 3.63E-06, 2.4621E+06, 2.4621E+06,
5, 4, 0, 1.0000, 4, 1.0005E+00, 1.777564E+04, 1.778528E+04, 5.42E-04, 6.80E-04, 2.70E-07, 2.4621E+06, 2.4621E+06,
5, 5, 0, 1.0000, 5, 1.0002E+00, 1.778215E+04, 1.778528E+04, 1.76E-04, 2.04E-04, 2.23E-08, 2.4621E+06, 2.4621E+06,
5, 6, 0, 1.0000, 6, 1.0001E+00, 1.778420E+04, 1.778528E+04, 6.08E-05, 6.19E-05, 1.91E-09, 2.4621E+06, 2.4621E+06,
5, 7, 0, 1.0000, 7, 1.0000E+00, 1.778491E+04, 1.778528E+04, 2.07E-05, 1.90E-05, 1.72E-10, 2.4621E+06, 2.4621E+06,

k,it0 ,it ,ci ,j ,armag ,arak ,ara0k ,reler ,RMS ,DMhsum ,trgtArea ,curtArea
6, 0, 0, 1.0000, 0, 7.8711E-01, 8.246852E+04, 6.491210E+04,-2.13E-01, 7.03E-01, 3.58E-01, 2.4621E+06, 2.4621E+06,
6, 1, 0, 1.0000, 1, 9.8093E-01, 6.617427E+04, 6.491210E+04,-1.91E-02, 1.23E-01, 5.80E-03, 2.4621E+06, 2.3444E+06,
6, 2, 0, 1.0000, 2, 9.9979E-01, 6.492579E+04, 6.491210E+04,-2.11E-04, 1.10E-02, 7.05E-05, 2.4621E+06, 2.4598E+06,
6, 3, 0, 1.0000, 3, 1.0004E+00, 6.488466E+04, 6.491210E+04, 4.23E-04, 2.40E-03, 3.63E-06, 2.4621E+06, 2.4621E+06,
6, 4, 0, 1.0000, 4, 1.0002E+00, 6.490000E+04, 6.491210E+04, 1.87E-04, 6.80E-04, 2.70E-07, 2.4621E+06, 2.4621E+06,
6, 5, 0, 1.0000, 5, 1.0001E+00, 6.490809E+04, 6.491210E+04, 6.19E-05, 2.04E-04, 2.23E-08, 2.4621E+06, 2.4621E+06,
6, 6, 0, 1.0000, 6, 1.0000E+00, 6.491078E+04, 6.491210E+04, 2.04E-05, 6.19E-05, 1.91E-09, 2.4621E+06, 2.4621E+06,
6, 7, 0, 1.0000, 7, 1.0000E+00, 6.491161E+04, 6.491210E+04, 7.51E-06, 1.90E-05, 1.72E-10, 2.4621E+06, 2.4621E+06,

k,it0 ,it ,ci ,j ,armag ,arak ,ara0k ,reler ,RMS ,DMhsum ,trgtArea ,curtArea
7, 0, 0, 1.0000, 0, 7.8712E-01, 7.257984E+03, 5.712873E+03,-2.13E-01, 7.03E-01, 3.58E-01, 2.4621E+06, 2.4621E+06,
7, 1, 0, 1.0000, 1, 1.0369E+00, 5.509505E+03, 5.712873E+03, 3.69E-02, 1.23E-01, 5.80E-03, 2.4621E+06, 2.3444E+06,
7, 2, 0, 1.0000, 2, 1.0040E+00, 5.690118E+03, 5.712873E+03, 4.00E-03, 1.10E-02, 7.05E-05, 2.4621E+06, 2.4598E+06,
7, 3, 0, 1.0000, 3, 1.0009E+00, 5.707529E+03, 5.712873E+03, 9.36E-04, 2.40E-03, 3.63E-06, 2.4621E+06, 2.4621E+06,
7, 4, 0, 1.0000, 4, 1.0001E+00, 5.712334E+03, 5.712873E+03, 9.43E-05, 6.80E-04, 2.70E-07, 2.4621E+06, 2.4621E+06,
7, 5, 0, 1.0000, 5, 9.9996E-01, 5.713083E+03, 5.712873E+03,-3.67E-05, 2.04E-04, 2.23E-08, 2.4621E+06, 2.4621E+06,
7, 6, 0, 1.0000, 6, 9.9998E-01, 5.713010E+03, 5.712873E+03,-2.40E-05, 6.19E-05, 1.91E-09, 2.4621E+06, 2.4621E+06,
7, 7, 0, 1.0000, 7, 9.9998E-01, 5.712974E+03, 5.712873E+03,-1.77E-05, 1.90E-05, 1.72E-10, 2.4621E+06, 2.4621E+06,

k,it0 ,it ,ci ,j ,armag ,arak ,ara0k ,reler ,RMS ,DMhsum ,trgtArea ,curtArea
8, 0, 0, 1.0000, 0, 7.8711E-01, 1.142115E+05, 8.989745E+04,-2.13E-01, 7.03E-01, 3.58E-01, 2.4621E+06, 2.4621E+06,
8, 1, 0, 1.0000, 1, 9.6474E-01, 9.318323E+04, 8.989745E+04,-3.53E-02, 1.23E-01, 5.80E-03, 2.4621E+06, 2.3444E+06,
8, 2, 0, 1.0000, 2, 9.9180E-01, 9.064053E+04, 8.989745E+04,-8.20E-03, 1.10E-02, 7.05E-05, 2.4621E+06, 2.4598E+06,
8, 3, 0, 1.0000, 3, 9.9798E-01, 9.007925E+04, 8.989745E+04,-2.02E-03, 2.40E-03, 3.63E-06, 2.4621E+06, 2.4621E+06,
8, 4, 0, 1.0000, 4, 9.9949E-01, 8.994327E+04, 8.989745E+04,-5.09E-04, 6.80E-04, 2.70E-07, 2.4621E+06, 2.4621E+06,
8, 5, 0, 1.0000, 5, 9.9987E-01, 8.990909E+04, 8.989745E+04,-1.30E-04, 2.04E-04, 2.23E-08, 2.4621E+06, 2.4621E+06,
8, 6, 0, 1.0000, 6, 9.9997E-01, 8.990039E+04, 8.989745E+04,-3.27E-05, 6.19E-05, 1.91E-09, 2.4621E+06, 2.4621E+06,
8, 7, 0, 1.0000, 7, 9.9999E-01, 8.989835E+04, 8.989745E+04,-1.00E-05, 1.90E-05, 1.72E-10, 2.4621E+06, 2.4621E+06,

k,it0 ,it ,ci ,j ,armag ,arak ,ara0k ,reler ,RMS ,DMhsum ,trgtArea ,curtArea
9, 0, 0, 1.0000, 0, 5.7932E-01, 2.267998E+04, 1.313890E+04,-4.21E-01, 7.03E-01, 3.58E-01, 2.4621E+06, 2.4621E+06,
9, 1, 0, 1.0000, 1, 1.1585E+00, 1.134094E+04, 1.313890E+04, 1.59E-01, 1.23E-01, 5.80E-03, 2.4621E+06, 2.3444E+06,
9, 2, 0, 1.0000, 2, 1.0064E+00, 1.305575E+04, 1.313890E+04, 6.37E-03, 1.10E-02, 7.05E-05, 2.4621E+06, 2.4598E+06,
9, 3, 0, 1.0000, 3, 9.9981E-01, 1.314136E+04, 1.313890E+04,-1.87E-04, 2.40E-03, 3.63E-06, 2.4621E+06, 2.4621E+06,
9, 4, 0, 1.0000, 4, 9.9977E-01, 1.314199E+04, 1.313890E+04,-2.35E-04, 6.80E-04, 2.70E-07, 2.4621E+06, 2.4621E+06,
9, 5, 0, 1.0000, 5, 9.9989E-01, 1.314038E+04, 1.313890E+04,-1.13E-04, 2.04E-04, 2.23E-08, 2.4621E+06, 2.4621E+06,
9, 6, 0, 1.0000, 6, 9.9995E-01, 1.313950E+04, 1.313890E+04,-4.58E-05, 6.19E-05, 1.91E-09, 2.4621E+06, 2.4621E+06,
9, 7, 0, 1.0000, 7, 9.9998E-01, 1.313914E+04, 1.313890E+04,-1.79E-05, 1.90E-05, 1.72E-10, 2.4621E+06, 2.4621E+06,

k,it0 ,it ,ci ,j ,armag ,arak ,ara0k ,reler ,RMS ,DMhsum ,trgtArea ,curtArea
10, 0, 0, 1.0000, 0, 5.7932E-01, 4.073601E+04, 2.359905E+04,-4.21E-01, 7.03E-01, 3.58E-01, 2.4621E+06, 2.4621E+06,
10, 1, 0, 1.0000, 1, 1.2978E+00, 1.818349E+04, 2.359905E+04, 2.98E-01, 1.23E-01, 5.80E-03, 2.4621E+06, 2.3444E+06,
10, 2, 0, 1.0000, 2, 1.0098E+00, 2.336960E+04, 2.359905E+04, 9.82E-03, 1.10E-02, 7.05E-05, 2.4621E+06, 2.4598E+06,
10, 3, 0, 1.0000, 3, 9.9946E-01, 2.361190E+04, 2.359905E+04,-5.44E-04, 2.40E-03, 3.63E-06, 2.4621E+06, 2.4621E+06,
10, 4, 0, 1.0000, 4, 9.9964E-01, 2.360760E+04, 2.359905E+04,-3.62E-04, 6.80E-04, 2.70E-07, 2.4621E+06, 2.4621E+06,
10, 5, 0, 1.0000, 5, 9.9987E-01, 2.360205E+04, 2.359905E+04,-1.27E-04, 2.04E-04, 2.23E-08, 2.4621E+06, 2.4621E+06,
10, 6, 0, 1.0000, 6, 9.9996E-01, 2.359998E+04, 2.359905E+04,-3.95E-05, 6.19E-05, 1.91E-09, 2.4621E+06, 2.4621E+06,
10, 7, 0, 1.0000, 7, 9.9999E-01, 2.359931E+04, 2.359905E+04,-1.09E-05, 1.90E-05, 1.72E-10, 2.4621E+06, 2.4621E+06,

k,it0 ,it ,ci ,j ,armag ,arak ,ara0k ,reler ,RMS ,DMhsum ,trgtArea ,curtArea
11, 0, 0, 1.0000, 0, 5.7932E-01, 6.092449E+04, 3.529458E+04,-4.21E-01, 7.03E-01, 3.58E-01, 2.4621E+06, 2.4621E+06,
11, 1, 0, 1.0000, 1, 1.0827E+00, 3.259946E+04, 3.529458E+04, 8.27E-02, 1.23E-01, 5.80E-03, 2.4621E+06, 2.3444E+06,
11, 2, 0, 1.0000, 2, 9.9626E-01, 3.542723E+04, 3.529458E+04,-3.74E-03, 1.10E-02, 7.05E-05, 2.4621E+06, 2.4598E+06,
11, 3, 0, 1.0000, 3, 9.9867E-01, 3.534173E+04, 3.529458E+04,-1.33E-03, 2.40E-03, 3.63E-06, 2.4621E+06, 2.4621E+06,
11, 4, 0, 1.0000, 4, 9.9964E-01, 3.530730E+04, 3.529458E+04,-3.60E-04, 6.80E-04, 2.70E-07, 2.4621E+06, 2.4621E+06,
11, 5, 0, 1.0000, 5, 9.9990E-01, 3.529805E+04, 3.529458E+04,-9.84E-05, 2.04E-04, 2.23E-08, 2.4621E+06, 2.4621E+06,
11, 6, 0, 1.0000, 6, 9.9997E-01, 3.529548E+04, 3.529458E+04,-2.56E-05, 6.19E-05, 1.91E-09, 2.4621E+06, 2.4621E+06,
11, 7, 0, 1.0000, 7, 9.9999E-01, 3.529479E+04, 3.529458E+04,-6.08E-06, 1.90E-05, 1.72E-10, 2.4621E+06, 2.4621E+06,

k,it0 ,it ,ci ,j ,armag ,arak ,ara0k ,reler ,RMS ,DMhsum ,trgtArea ,curtArea
12, 0, 0, 1.0000, 0, 5.7932E-01, 9.011802E+04, 5.220687E+04,-4.21E-01, 7.03E-01, 3.58E-01, 2.4621E+06, 2.4621E+06,
12, 1, 0, 1.0000, 1, 1.0611E+00, 4.919995E+04, 5.220687E+04, 6.11E-02, 1.23E-01, 5.80E-03, 2.4621E+06, 2.3444E+06,
12, 2, 0, 1.0000, 2, 9.9990E-01, 5.221229E+04, 5.220687E+04,-1.04E-04, 1.10E-02, 7.05E-05, 2.4621E+06, 2.4598E+06,
12, 3, 0, 1.0000, 3, 1.0000E+00, 5.220527E+04, 5.220687E+04, 3.06E-05, 2.40E-03, 3.63E-06, 2.4621E+06, 2.4621E+06,
12, 4, 0, 1.0000, 4, 1.0001E+00, 5.220311E+04, 5.220687E+04, 7.19E-05, 6.80E-04, 2.70E-07, 2.4621E+06, 2.4621E+06,
12, 5, 0, 1.0000, 5, 1.0000E+00, 5.220497E+04, 5.220687E+04, 3.64E-05, 2.04E-04, 2.23E-08, 2.4621E+06, 2.4621E+06,
12, 6, 0, 1.0000, 6, 1.0000E+00, 5.220605E+04, 5.220687E+04, 1.56E-05, 6.19E-05, 1.91E-09, 2.4621E+06, 2.4621E+06,
12, 7, 0, 1.0000, 7, 1.0000E+00, 5.220654E+04, 5.220687E+04, 6.20E-06, 1.90E-05, 1.72E-10, 2.4621E+06, 2.4621E+06,

k,it0 ,it ,ci ,j ,armag ,arak ,ara0k ,reler ,RMS ,DMhsum ,trgtArea ,curtArea
13, 0, 0, 1.0000, 0, 3.0532E+00, 3.547649E+04, 1.083161E+05, 2.05E+00, 7.03E-01, 3.58E-01, 2.4621E+06, 2.4621E+06,
13, 1, 0, 1.0000, 1, 1.0588E+00, 1.023010E+05, 1.083161E+05, 5.88E-02, 1.23E-01, 5.80E-03, 2.4621E+06, 2.3444E+06,
13, 2, 0, 1.0000, 2, 1.0118E+00, 1.070569E+05, 1.083161E+05, 1.18E-02, 1.10E-02, 7.05E-05, 2.4621E+06, 2.4598E+06,
13, 3, 0, 1.0000, 3, 1.0030E+00, 1.079879E+05, 1.083161E+05, 3.04E-03, 2.40E-03, 3.63E-06, 2.4621E+06, 2.4621E+06,
13, 4, 0, 1.0000, 4, 1.0008E+00, 1.082256E+05, 1.083161E+05, 8.37E-04, 6.80E-04, 2.70E-07, 2.4621E+06, 2.4621E+06,
13, 5, 0, 1.0000, 5, 1.0002E+00, 1.082902E+05, 1.083161E+05, 2.39E-04, 2.04E-04, 2.23E-08, 2.4621E+06, 2.4621E+06,
13, 6, 0, 1.0000, 6, 1.0001E+00, 1.083085E+05, 1.083161E+05, 7.03E-05, 6.19E-05, 1.91E-09, 2.4621E+06, 2.4621E+06,
13, 7, 0, 1.0000, 7, 1.0000E+00, 1.083140E+05, 1.083161E+05, 1.98E-05, 1.90E-05, 1.72E-10, 2.4621E+06, 2.4621E+06,

k,it0 ,it ,ci ,j ,armag ,arak ,ara0k ,reler ,RMS ,DMhsum ,trgtArea ,curtArea
14, 0, 0, 1.0000, 0, 3.0532E+00, 5.012000E+04, 1.530253E+05, 2.05E+00, 7.03E-01, 3.58E-01, 2.4621E+06, 2.4621E+06,
14, 1, 0, 1.0000, 1, 1.0461E+00, 1.462820E+05, 1.530253E+05, 4.61E-02, 1.23E-01, 5.80E-03, 2.4621E+06, 2.3444E+06,
14, 2, 0, 1.0000, 2, 9.9901E-01, 1.531770E+05, 1.530253E+05,-9.90E-04, 1.10E-02, 7.05E-05, 2.4621E+06, 2.4598E+06,
14, 3, 0, 1.0000, 3, 9.9961E-01, 1.530856E+05, 1.530253E+05,-3.94E-04, 2.40E-03, 3.63E-06, 2.4621E+06, 2.4621E+06,
14, 4, 0, 1.0000, 4, 9.9984E-01, 1.530502E+05, 1.530253E+05,-1.63E-04, 6.80E-04, 2.70E-07, 2.4621E+06, 2.4621E+06,
14, 5, 0, 1.0000, 5, 9.9994E-01, 1.530350E+05, 1.530253E+05,-6.35E-05, 2.04E-04, 2.23E-08, 2.4621E+06, 2.4621E+06,
14, 6, 0, 1.0000, 6, 9.9998E-01, 1.530287E+05, 1.530253E+05,-2.24E-05, 6.19E-05, 1.91E-09, 2.4621E+06, 2.4621E+06,
14, 7, 0, 1.0000, 7, 9.9999E-01, 1.530266E+05, 1.530253E+05,-8.52E-06, 1.90E-05, 1.72E-10, 2.4621E+06, 2.4621E+06,

k,it0 ,it ,ci ,j ,armag ,arak ,ara0k ,reler ,RMS ,DMhsum ,trgtArea ,curtArea
15, 0, 0, 1.0000, 0, 3.0532E+00, 2.624653E+04, 8.013527E+04, 2.05E+00, 7.03E-01, 3.58E-01, 2.4621E+06, 2.4621E+06,
15, 1, 0, 1.0000, 1, 1.0601E+00, 7.559270E+04, 8.013527E+04, 6.01E-02, 1.23E-01, 5.80E-03, 2.4621E+06, 2.3444E+06,
15, 2, 0, 1.0000, 2, 9.9763E-01, 8.032575E+04, 8.013527E+04,-2.37E-03, 1.10E-02, 7.05E-05, 2.4621E+06, 2.4598E+06,
15, 3, 0, 1.0000, 3, 9.9896E-01, 8.021838E+04, 8.013527E+04,-1.04E-03, 2.40E-03, 3.63E-06, 2.4621E+06, 2.4621E+06,
15, 4, 0, 1.0000, 4, 9.9967E-01, 8.016145E+04, 8.013527E+04,-3.27E-04, 6.80E-04, 2.70E-07, 2.4621E+06, 2.4621E+06,
15, 5, 0, 1.0000, 5, 9.9990E-01, 8.014355E+04, 8.013527E+04,-1.03E-04, 2.04E-04, 2.23E-08, 2.4621E+06, 2.4621E+06,
15, 6, 0, 1.0000, 6, 9.9997E-01, 8.013774E+04, 8.013527E+04,-3.08E-05, 6.19E-05, 1.91E-09, 2.4621E+06, 2.4621E+06,
15, 7, 0, 1.0000, 7, 9.9999E-01, 8.013607E+04, 8.013527E+04,-9.95E-06, 1.90E-05, 1.72E-10, 2.4621E+06, 2.4621E+06,

k,it0 ,it ,ci ,j ,armag ,arak ,ara0k ,reler ,RMS ,DMhsum ,trgtArea ,curtArea
16, 0, 0, 1.0000, 0, 3.0532E+00, 7.032101E+04, 2.147026E+05, 2.05E+00, 7.03E-01, 3.58E-01, 2.4621E+06, 2.4621E+06,
16, 1, 0, 1.0000, 1, 1.0408E+00, 2.062886E+05, 2.147026E+05, 4.08E-02, 1.23E-01, 5.80E-03, 2.4621E+06, 2.3444E+06,
16, 2, 0, 1.0000, 2, 9.9875E-01, 2.149715E+05, 2.147026E+05,-1.25E-03, 1.10E-02, 7.05E-05, 2.4621E+06, 2.4598E+06,
16, 3, 0, 1.0000, 3, 9.9960E-01, 2.147892E+05, 2.147026E+05,-4.03E-04, 2.40E-03, 3.63E-06, 2.4621E+06, 2.4621E+06,
16, 4, 0, 1.0000, 4, 9.9992E-01, 2.147191E+05, 2.147026E+05,-7.71E-05, 6.80E-04, 2.70E-07, 2.4621E+06, 2.4621E+06,
16, 5, 0, 1.0000, 5, 9.9999E-01, 2.147055E+05, 2.147026E+05,-1.34E-05, 2.04E-04, 2.23E-08, 2.4621E+06, 2.4621E+06,
16, 6, 0, 1.0000, 6, 1.0000E+00, 2.147030E+05, 2.147026E+05,-2.03E-06, 6.19E-05, 1.91E-09, 2.4621E+06, 2.4621E+06,
16, 7, 0, 1.0000, 7, 1.0000E+00, 2.147028E+05, 2.147026E+05,-8.94E-07, 1.90E-05, 1.72E-10, 2.4621E+06, 2.4621E+06,

k,it0 ,it ,ci ,j ,armag ,arak ,ara0k ,reler ,RMS ,DMhsum ,trgtArea ,curtArea
17, 0, 0, 1.0000, 0, 2.1786E-01, 9.701901E+04, 2.113701E+04,-7.82E-01, 7.03E-01, 3.58E-01, 2.4621E+06, 2.4621E+06,
17, 1, 0, 1.0000, 1, 1.0341E+00, 2.044007E+04, 2.113701E+04, 3.41E-02, 1.23E-01, 5.80E-03, 2.4621E+06, 2.3444E+06,
17, 2, 0, 1.0000, 2, 9.9820E-01, 2.117503E+04, 2.113701E+04,-1.80E-03, 1.10E-02, 7.05E-05, 2.4621E+06, 2.4598E+06,
17, 3, 0, 1.0000, 3, 9.9962E-01, 2.114511E+04, 2.113701E+04,-3.83E-04, 2.40E-03, 3.63E-06, 2.4621E+06, 2.4621E+06,
17, 4, 0, 1.0000, 4, 1.0000E+00, 2.113689E+04, 2.113701E+04, 5.60E-06, 6.80E-04, 2.70E-07, 2.4621E+06, 2.4621E+06,
17, 5, 0, 1.0000, 5, 1.0000E+00, 2.113643E+04, 2.113701E+04, 2.73E-05, 2.04E-04, 2.23E-08, 2.4621E+06, 2.4621E+06,
17, 6, 0, 1.0000, 6, 1.0000E+00, 2.113672E+04, 2.113701E+04, 1.36E-05, 6.19E-05, 1.91E-09, 2.4621E+06, 2.4621E+06,
17, 7, 0, 1.0000, 7, 1.0000E+00, 2.113691E+04, 2.113701E+04, 4.77E-06, 1.90E-05, 1.72E-10, 2.4621E+06, 2.4621E+06,

k,it0 ,it ,ci ,j ,armag ,arak ,ara0k ,reler ,RMS ,DMhsum ,trgtArea ,curtArea
18, 0, 0, 1.0000, 0, 2.1786E-01, 4.247448E+04, 9.253693E+03,-7.82E-01, 7.03E-01, 3.58E-01, 2.4621E+06, 2.4621E+06,
18, 1, 0, 1.0000, 1, 9.3466E-01, 9.900604E+03, 9.253693E+03,-6.53E-02, 1.23E-01, 5.80E-03, 2.4621E+06, 2.3444E+06,
18, 2, 0, 1.0000, 2, 9.8415E-01, 9.402682E+03, 9.253693E+03,-1.58E-02, 1.10E-02, 7.05E-05, 2.4621E+06, 2.4598E+06,
18, 3, 0, 1.0000, 3, 9.9657E-01, 9.285530E+03, 9.253693E+03,-3.43E-03, 2.40E-03, 3.63E-06, 2.4621E+06, 2.4621E+06,
18, 4, 0, 1.0000, 4, 9.9921E-01, 9.260981E+03, 9.253693E+03,-7.87E-04, 6.80E-04, 2.70E-07, 2.4621E+06, 2.4621E+06,
18, 5, 0, 1.0000, 5, 9.9982E-01, 9.255367E+03, 9.253693E+03,-1.81E-04, 2.04E-04, 2.23E-08, 2.4621E+06, 2.4621E+06,
18, 6, 0, 1.0000, 6, 9.9996E-01, 9.254075E+03, 9.253693E+03,-4.12E-05, 6.19E-05, 1.91E-09, 2.4621E+06, 2.4621E+06,
18, 7, 0, 1.0000, 7, 9.9999E-01, 9.253785E+03, 9.253693E+03,-9.95E-06, 1.90E-05, 1.72E-10, 2.4621E+06, 2.4621E+06,

k,it0 ,it ,ci ,j ,armag ,arak ,ara0k ,reler ,RMS ,DMhsum ,trgtArea ,curtArea
19, 0, 0, 1.0000, 0, 9.0313E-01, 7.950548E+04, 7.180352E+04,-9.69E-02, 7.03E-01, 3.58E-01, 2.4621E+06, 2.4621E+06,
19, 1, 0, 1.0000, 1, 1.1695E+00, 6.139496E+04, 7.180352E+04, 1.70E-01, 1.23E-01, 5.80E-03, 2.4621E+06, 2.3444E+06,
19, 2, 0, 1.0000, 2, 1.0097E+00, 7.111548E+04, 7.180352E+04, 9.68E-03, 1.10E-02, 7.05E-05, 2.4621E+06, 2.4598E+06,
19, 3, 0, 1.0000, 3, 1.0015E+00, 7.169777E+04, 7.180352E+04, 1.47E-03, 2.40E-03, 3.63E-06, 2.4621E+06, 2.4621E+06,
19, 4, 0, 1.0000, 4, 1.0004E+00, 7.177244E+04, 7.180352E+04, 4.33E-04, 6.80E-04, 2.70E-07, 2.4621E+06, 2.4621E+06,
19, 5, 0, 1.0000, 5, 1.0001E+00, 7.179405E+04, 7.180352E+04, 1.32E-04, 2.04E-04, 2.23E-08, 2.4621E+06, 2.4621E+06,
19, 6, 0, 1.0000, 6, 1.0000E+00, 7.180071E+04, 7.180352E+04, 3.91E-05, 6.19E-05, 1.91E-09, 2.4621E+06, 2.4621E+06,
19, 7, 0, 1.0000, 7, 1.0000E+00, 7.180265E+04, 7.180352E+04, 1.22E-05, 1.90E-05, 1.72E-10, 2.4621E+06, 2.4621E+06,

k,it0 ,it ,ci ,j ,armag ,arak ,ara0k ,reler ,RMS ,DMhsum ,trgtArea ,curtArea
20, 0, 0, 1.0000, 0, 9.0313E-01, 6.451198E+04, 5.826250E+04,-9.69E-02, 7.03E-01, 3.58E-01, 2.4621E+06, 2.4621E+06,
20, 1, 0, 1.0000, 1, 1.1054E+00, 5.270768E+04, 5.826250E+04, 1.05E-01, 1.23E-01, 5.80E-03, 2.4621E+06, 2.3444E+06,
20, 2, 0, 1.0000, 2, 1.0010E+00, 5.820659E+04, 5.826250E+04, 9.61E-04, 1.10E-02, 7.05E-05, 2.4621E+06, 2.4598E+06,
20, 3, 0, 1.0000, 3, 1.0003E+00, 5.824624E+04, 5.826250E+04, 2.79E-04, 2.40E-03, 3.63E-06, 2.4621E+06, 2.4621E+06,
20, 4, 0, 1.0000, 4, 1.0002E+00, 5.825032E+04, 5.826250E+04, 2.09E-04, 6.80E-04, 2.70E-07, 2.4621E+06, 2.4621E+06,
20, 5, 0, 1.0000, 5, 1.0001E+00, 5.825768E+04, 5.826250E+04, 8.27E-05, 2.04E-04, 2.23E-08, 2.4621E+06, 2.4621E+06,
20, 6, 0, 1.0000, 6, 1.0000E+00, 5.826082E+04, 5.826250E+04, 2.88E-05, 6.19E-05, 1.91E-09, 2.4621E+06, 2.4621E+06,
20, 7, 0, 1.0000, 7, 1.0000E+00, 5.826210E+04, 5.826250E+04, 6.79E-06, 1.90E-05, 1.72E-10, 2.4621E+06, 2.4621E+06,

k,it0 ,it ,ci ,j ,armag ,arak ,ara0k ,reler ,RMS ,DMhsum ,trgtArea ,curtArea
21, 0, 0, 1.0000, 0, 9.0313E-01, 4.141498E+04, 3.740298E+04,-9.69E-02, 7.03E-01, 3.58E-01, 2.4621E+06, 2.4621E+06,
21, 1, 0, 1.0000, 1, 1.0981E+00, 3.406064E+04, 3.740298E+04, 9.81E-02, 1.23E-01, 5.80E-03, 2.4621E+06, 2.3444E+06,
21, 2, 0, 1.0000, 2, 9.9467E-01, 3.760354E+04, 3.740298E+04,-5.33E-03, 1.10E-02, 7.05E-05, 2.4621E+06, 2.4598E+06,
21, 3, 0, 1.0000, 3, 9.9674E-01, 3.752546E+04, 3.740298E+04,-3.26E-03, 2.40E-03, 3.63E-06, 2.4621E+06, 2.4621E+06,
21, 4, 0, 1.0000, 4, 9.9902E-01, 3.743952E+04, 3.740298E+04,-9.76E-04, 6.80E-04, 2.70E-07, 2.4621E+06, 2.4621E+06,
21, 5, 0, 1.0000, 5, 9.9973E-01, 3.741324E+04, 3.740298E+04,-2.74E-04, 2.04E-04, 2.23E-08, 2.4621E+06, 2.4621E+06,
21, 6, 0, 1.0000, 6, 9.9993E-01, 3.740571E+04, 3.740298E+04,-7.31E-05, 6.19E-05, 1.91E-09, 2.4621E+06, 2.4621E+06,
21, 7, 0, 1.0000, 7, 9.9998E-01, 3.740377E+04, 3.740298E+04,-2.10E-05, 1.90E-05, 1.72E-10, 2.4621E+06, 2.4621E+06,

k,it0 ,it ,ci ,j ,armag ,arak ,ara0k ,reler ,RMS ,DMhsum ,trgtArea ,curtArea
22, 0, 0, 1.0000, 0, 1.3275E+00, 1.672899E+04, 2.220693E+04, 3.27E-01, 7.03E-01, 3.58E-01, 2.4621E+06, 2.4621E+06,
22, 1, 0, 1.0000, 1, 1.4521E+00, 1.529326E+04, 2.220693E+04, 4.52E-01, 1.23E-01, 5.80E-03, 2.4621E+06, 2.3444E+06,
22, 2, 0, 1.0000, 2, 1.0406E+00, 2.134088E+04, 2.220693E+04, 4.06E-02, 1.10E-02, 7.05E-05, 2.4621E+06, 2.4598E+06,
22, 3, 0, 1.0000, 3, 1.0035E+00, 2.212952E+04, 2.220693E+04, 3.50E-03, 2.40E-03, 3.63E-06, 2.4621E+06, 2.4621E+06,
22, 4, 0, 1.0000, 4, 1.0003E+00, 2.220034E+04, 2.220693E+04, 2.97E-04, 6.80E-04, 2.70E-07, 2.4621E+06, 2.4621E+06,
22, 5, 0, 1.0000, 5, 1.0000E+00, 2.220674E+04, 2.220693E+04, 8.46E-06, 2.04E-04, 2.23E-08, 2.4621E+06, 2.4621E+06,
22, 6, 0, 1.0000, 6, 9.9998E-01, 2.220728E+04, 2.220693E+04,-1.60E-05, 6.19E-05, 1.91E-09, 2.4621E+06, 2.4621E+06,
22, 7, 0, 1.0000, 7, 1.0000E+00, 2.220699E+04, 2.220693E+04,-2.92E-06, 1.90E-05, 1.72E-10, 2.4621E+06, 2.4621E+06,

k,it0 ,it ,ci ,j ,armag ,arak ,ara0k ,reler ,RMS ,DMhsum ,trgtArea ,curtArea
23, 0, 0, 1.0000, 0, 8.1929E-01, 5.463398E+04, 4.476091E+04,-1.81E-01, 7.03E-01, 3.58E-01, 2.4621E+06, 2.4621E+06,
23, 1, 0, 1.0000, 1, 1.0049E+00, 4.454234E+04, 4.476091E+04, 4.91E-03, 1.23E-01, 5.80E-03, 2.4621E+06, 2.3444E+06,
23, 2, 0, 1.0000, 2, 9.8861E-01, 4.527661E+04, 4.476091E+04,-1.14E-02, 1.10E-02, 7.05E-05, 2.4621E+06, 2.4598E+06,
23, 3, 0, 1.0000, 3, 9.9649E-01, 4.491879E+04, 4.476091E+04,-3.51E-03, 2.40E-03, 3.63E-06, 2.4621E+06, 2.4621E+06,
23, 4, 0, 1.0000, 4, 9.9893E-01, 4.480886E+04, 4.476091E+04,-1.07E-03, 6.80E-04, 2.70E-07, 2.4621E+06, 2.4621E+06,
23, 5, 0, 1.0000, 5, 9.9967E-01, 4.477570E+04, 4.476091E+04,-3.30E-04, 2.04E-04, 2.23E-08, 2.4621E+06, 2.4621E+06,
23, 6, 0, 1.0000, 6, 9.9990E-01, 4.476540E+04, 4.476091E+04,-1.00E-04, 6.19E-05, 1.91E-09, 2.4621E+06, 2.4621E+06,
23, 7, 0, 1.0000, 7, 9.9997E-01, 4.476230E+04, 4.476091E+04,-3.10E-05, 1.90E-05, 1.72E-10, 2.4621E+06, 2.4621E+06,

k,it0 ,it ,ci ,j ,armag ,arak ,ara0k ,reler ,RMS ,DMhsum ,trgtArea ,curtArea
24, 0, 0, 1.0000, 0, 8.1929E-01, 3.277100E+04, 2.684885E+04,-1.81E-01, 7.03E-01, 3.58E-01, 2.4621E+06, 2.4621E+06,
24, 1, 0, 1.0000, 1, 1.0032E+00, 2.676263E+04, 2.684885E+04, 3.22E-03, 1.23E-01, 5.80E-03, 2.4621E+06, 2.3444E+06,
24, 2, 0, 1.0000, 2, 9.8951E-01, 2.713337E+04, 2.684885E+04,-1.05E-02, 1.10E-02, 7.05E-05, 2.4621E+06, 2.4598E+06,
24, 3, 0, 1.0000, 3, 9.9699E-01, 2.692992E+04, 2.684885E+04,-3.01E-03, 2.40E-03, 3.63E-06, 2.4621E+06, 2.4621E+06,
24, 4, 0, 1.0000, 4, 9.9922E-01, 2.686978E+04, 2.684885E+04,-7.79E-04, 6.80E-04, 2.70E-07, 2.4621E+06, 2.4621E+06,
24, 5, 0, 1.0000, 5, 9.9979E-01, 2.685441E+04, 2.684885E+04,-2.07E-04, 2.04E-04, 2.23E-08, 2.4621E+06, 2.4621E+06,
24, 6, 0, 1.0000, 6, 9.9994E-01, 2.685037E+04, 2.684885E+04,-5.66E-05, 6.19E-05, 1.91E-09, 2.4621E+06, 2.4621E+06,
24, 7, 0, 1.0000, 7, 9.9999E-01, 2.684922E+04, 2.684885E+04,-1.38E-05, 1.90E-05, 1.72E-10, 2.4621E+06, 2.4621E+06,

k,it0 ,it ,ci ,j ,armag ,arak ,ara0k ,reler ,RMS ,DMhsum ,trgtArea ,curtArea
25, 0, 0, 1.0000, 0, 8.1929E-01, 1.111501E+04, 9.106372E+03,-1.81E-01, 7.03E-01, 3.58E-01, 2.4621E+06, 2.4621E+06,
25, 1, 0, 1.0000, 1, 1.1188E+00, 8.139691E+03, 9.106372E+03, 1.19E-01, 1.23E-01, 5.80E-03, 2.4621E+06, 2.3444E+06,
25, 2, 0, 1.0000, 2, 1.0233E+00, 8.899108E+03, 9.106372E+03, 2.33E-02, 1.10E-02, 7.05E-05, 2.4621E+06, 2.4598E+06,
25, 3, 0, 1.0000, 3, 1.0063E+00, 9.049344E+03, 9.106372E+03, 6.30E-03, 2.40E-03, 3.63E-06, 2.4621E+06, 2.4621E+06,
25, 4, 0, 1.0000, 4, 1.0020E+00, 9.088541E+03, 9.106372E+03, 1.96E-03, 6.80E-04, 2.70E-07, 2.4621E+06, 2.4621E+06,
25, 5, 0, 1.0000, 5, 1.0006E+00, 9.100700E+03, 9.106372E+03, 6.23E-04, 2.04E-04, 2.23E-08, 2.4621E+06, 2.4621E+06,
25, 6, 0, 1.0000, 6, 1.0002E+00, 9.104563E+03, 9.106372E+03, 1.99E-04, 6.19E-05, 1.91E-09, 2.4621E+06, 2.4621E+06,
25, 7, 0, 1.0000, 7, 1.0001E+00, 9.105807E+03, 9.106372E+03, 6.21E-05, 1.90E-05, 1.72E-10, 2.4621E+06, 2.4621E+06,

k,it0 ,it ,ci ,j ,armag ,arak ,ara0k ,reler ,RMS ,DMhsum ,trgtArea ,curtArea
26, 0, 0, 1.0000, 0, 6.2994E-01, 8.458500E+04, 5.328330E+04,-3.70E-01, 7.03E-01, 3.58E-01, 2.4621E+06, 2.4621E+06,
26, 1, 0, 1.0000, 1, 9.9323E-01, 5.364666E+04, 5.328330E+04,-6.77E-03, 1.23E-01, 5.80E-03, 2.4621E+06, 2.3444E+06,
26, 2, 0, 1.0000, 2, 9.8918E-01, 5.386596E+04, 5.328330E+04,-1.08E-02, 1.10E-02, 7.05E-05, 2.4621E+06, 2.4598E+06,
26, 3, 0, 1.0000, 3, 9.9722E-01, 5.343166E+04, 5.328330E+04,-2.78E-03, 2.40E-03, 3.63E-06, 2.4621E+06, 2.4621E+06,
26, 4, 0, 1.0000, 4, 9.9926E-01, 5.332265E+04, 5.328330E+04,-7.38E-04, 6.80E-04, 2.70E-07, 2.4621E+06, 2.4621E+06,
26, 5, 0, 1.0000, 5, 9.9979E-01, 5.329439E+04, 5.328330E+04,-2.08E-04, 2.04E-04, 2.23E-08, 2.4621E+06, 2.4621E+06,
26, 6, 0, 1.0000, 6, 9.9994E-01, 5.328654E+04, 5.328330E+04,-6.09E-05, 6.19E-05, 1.91E-09, 2.4621E+06, 2.4621E+06,
26, 7, 0, 1.0000, 7, 9.9998E-01, 5.328422E+04, 5.328330E+04,-1.73E-05, 1.90E-05, 1.72E-10, 2.4621E+06, 2.4621E+06,

k,it0 ,it ,ci ,j ,armag ,arak ,ara0k ,reler ,RMS ,DMhsum ,trgtArea ,curtArea
27, 0, 0, 1.0000, 0, 6.2994E-01, 5.776850E+04, 3.639058E+04,-3.70E-01, 7.03E-01, 3.58E-01, 2.4621E+06, 2.4621E+06,
27, 1, 0, 1.0000, 1, 1.2684E+00, 2.868913E+04, 3.639058E+04, 2.68E-01, 1.23E-01, 5.80E-03, 2.4621E+06, 2.3444E+06,
27, 2, 0, 1.0000, 2, 1.0166E+00, 3.579627E+04, 3.639058E+04, 1.66E-02, 1.10E-02, 7.05E-05, 2.4621E+06, 2.4598E+06,
27, 3, 0, 1.0000, 3, 1.0018E+00, 3.632593E+04, 3.639058E+04, 1.78E-03, 2.40E-03, 3.63E-06, 2.4621E+06, 2.4621E+06,
27, 4, 0, 1.0000, 4, 1.0004E+00, 3.637668E+04, 3.639058E+04, 3.82E-04, 6.80E-04, 2.70E-07, 2.4621E+06, 2.4621E+06,
27, 5, 0, 1.0000, 5, 1.0001E+00, 3.638656E+04, 3.639058E+04, 1.10E-04, 2.04E-04, 2.23E-08, 2.4621E+06, 2.4621E+06,
27, 6, 0, 1.0000, 6, 1.0000E+00, 3.638936E+04, 3.639058E+04, 3.36E-05, 6.19E-05, 1.91E-09, 2.4621E+06, 2.4621E+06,
27, 7, 0, 1.0000, 7, 1.0000E+00, 3.639017E+04, 3.639058E+04, 1.12E-05, 1.90E-05, 1.72E-10, 2.4621E+06, 2.4621E+06,

k,it0 ,it ,ci ,j ,armag ,arak ,ara0k ,reler ,RMS ,DMhsum ,trgtArea ,curtArea
28, 0, 0, 1.0000, 0, 6.2994E-01, 3.141799E+04, 1.979139E+04,-3.70E-01, 7.03E-01, 3.58E-01, 2.4621E+06, 2.4621E+06,
28, 1, 0, 1.0000, 1, 1.1559E+00, 1.712203E+04, 1.979139E+04, 1.56E-01, 1.23E-01, 5.80E-03, 2.4621E+06, 2.3444E+06,
28, 2, 0, 1.0000, 2, 1.0096E+00, 1.960337E+04, 1.979139E+04, 9.59E-03, 1.10E-02, 7.05E-05, 2.4621E+06, 2.4598E+06,
28, 3, 0, 1.0000, 3, 1.0025E+00, 1.974107E+04, 1.979139E+04, 2.55E-03, 2.40E-03, 3.63E-06, 2.4621E+06, 2.4621E+06,
28, 4, 0, 1.0000, 4, 1.0009E+00, 1.977320E+04, 1.979139E+04, 9.20E-04, 6.80E-04, 2.70E-07, 2.4621E+06, 2.4621E+06,
28, 5, 0, 1.0000, 5, 1.0003E+00, 1.978502E+04, 1.979139E+04, 3.22E-04, 2.04E-04, 2.23E-08, 2.4621E+06, 2.4621E+06,
28, 6, 0, 1.0000, 6, 1.0001E+00, 1.978924E+04, 1.979139E+04, 1.08E-04, 6.19E-05, 1.91E-09, 2.4621E+06, 2.4621E+06,
28, 7, 0, 1.0000, 7, 1.0000E+00, 1.979065E+04, 1.979139E+04, 3.76E-05, 1.90E-05, 1.72E-10, 2.4621E+06, 2.4621E+06,

k,it0 ,it ,ci ,j ,armag ,arak ,ara0k ,reler ,RMS ,DMhsum ,trgtArea ,curtArea
29, 0, 0, 1.0000, 0, 5.7970E-01, 5.531552E+04, 3.206651E+04,-4.20E-01, 7.03E-01, 3.58E-01, 2.4621E+06, 2.4621E+06,
29, 1, 0, 1.0000, 1, 9.8217E-01, 3.264869E+04, 3.206651E+04,-1.78E-02, 1.23E-01, 5.80E-03, 2.4621E+06, 2.3444E+06,
29, 2, 0, 1.0000, 2, 1.0002E+00, 3.205982E+04, 3.206651E+04, 2.08E-04, 1.10E-02, 7.05E-05, 2.4621E+06, 2.4598E+06,
29, 3, 0, 1.0000, 3, 1.0008E+00, 3.204227E+04, 3.206651E+04, 7.57E-04, 2.40E-03, 3.63E-06, 2.4621E+06, 2.4621E+06,
29, 4, 0, 1.0000, 4, 1.0003E+00, 3.205736E+04, 3.206651E+04, 2.85E-04, 6.80E-04, 2.70E-07, 2.4621E+06, 2.4621E+06,
29, 5, 0, 1.0000, 5, 1.0001E+00, 3.206342E+04, 3.206651E+04, 9.63E-05, 2.04E-04, 2.23E-08, 2.4621E+06, 2.4621E+06,
29, 6, 0, 1.0000, 6, 1.0000E+00, 3.206547E+04, 3.206651E+04, 3.23E-05, 6.19E-05, 1.91E-09, 2.4621E+06, 2.4621E+06,
29, 7, 0, 1.0000, 7, 1.0000E+00, 3.206625E+04, 3.206651E+04, 8.23E-06, 1.90E-05, 1.72E-10, 2.4621E+06, 2.4621E+06,

k,it0 ,it ,ci ,j ,armag ,arak ,ara0k ,reler ,RMS ,DMhsum ,trgtArea ,curtArea
30, 0, 0, 1.0000, 0, 5.7970E-01, 7.430701E+04, 4.307592E+04,-4.20E-01, 7.03E-01, 3.58E-01, 2.4621E+06, 2.4621E+06,
30, 1, 0, 1.0000, 1, 1.0793E+00, 3.991190E+04, 4.307592E+04, 7.93E-02, 1.23E-01, 5.80E-03, 2.4621E+06, 2.3444E+06,
30, 2, 0, 1.0000, 2, 1.0148E+00, 4.244764E+04, 4.307592E+04, 1.48E-02, 1.10E-02, 7.05E-05, 2.4621E+06, 2.4598E+06,
30, 3, 0, 1.0000, 3, 1.0037E+00, 4.291702E+04, 4.307592E+04, 3.70E-03, 2.40E-03, 3.63E-06, 2.4621E+06, 2.4621E+06,
30, 4, 0, 1.0000, 4, 1.0009E+00, 4.303684E+04, 4.307592E+04, 9.08E-04, 6.80E-04, 2.70E-07, 2.4621E+06, 2.4621E+06,
30, 5, 0, 1.0000, 5, 1.0002E+00, 4.306623E+04, 4.307592E+04, 2.25E-04, 2.04E-04, 2.23E-08, 2.4621E+06, 2.4621E+06,
30, 6, 0, 1.0000, 6, 1.0001E+00, 4.307357E+04, 4.307592E+04, 5.46E-05, 6.19E-05, 1.91E-09, 2.4621E+06, 2.4621E+06,
30, 7, 0, 1.0000, 7, 1.0000E+00, 4.307533E+04, 4.307592E+04, 1.37E-05, 1.90E-05, 1.72E-10, 2.4621E+06, 2.4621E+06,

k,it0 ,it ,ci ,j ,armag ,arak ,ara0k ,reler ,RMS ,DMhsum ,trgtArea ,curtArea
31, 0, 0, 1.0000, 0, 5.7970E-01, 6.502899E+04, 3.769745E+04,-4.20E-01, 7.03E-01, 3.58E-01, 2.4621E+06, 2.4621E+06,
31, 1, 0, 1.0000, 1, 1.0187E+00, 3.700661E+04, 3.769745E+04, 1.87E-02, 1.23E-01, 5.80E-03, 2.4621E+06, 2.3444E+06,
31, 2, 0, 1.0000, 2, 9.8630E-01, 3.822118E+04, 3.769745E+04,-1.37E-02, 1.10E-02, 7.05E-05, 2.4621E+06, 2.4598E+06,
31, 3, 0, 1.0000, 3, 9.9610E-01, 3.784497E+04, 3.769745E+04,-3.90E-03, 2.40E-03, 3.63E-06, 2.4621E+06, 2.4621E+06,
31, 4, 0, 1.0000, 4, 9.9890E-01, 3.773904E+04, 3.769745E+04,-1.10E-03, 6.80E-04, 2.70E-07, 2.4621E+06, 2.4621E+06,
31, 5, 0, 1.0000, 5, 9.9968E-01, 3.770966E+04, 3.769745E+04,-3.24E-04, 2.04E-04, 2.23E-08, 2.4621E+06, 2.4621E+06,
31, 6, 0, 1.0000, 6, 9.9991E-01, 3.770093E+04, 3.769745E+04,-9.24E-05, 6.19E-05, 1.91E-09, 2.4621E+06, 2.4621E+06,
31, 7, 0, 1.0000, 7, 9.9997E-01, 3.769855E+04, 3.769745E+04,-2.94E-05, 1.90E-05, 1.72E-10, 2.4621E+06, 2.4621E+06,

k,it0	,it	,ci	,j	,armag	,arak	,ara0k	,reler	,RMS	,DMhsum	,trgtArea	,curtArea
32,	0,	0,	1.0000,	0,	1.2166E+00,	8.419449E+04,	1.024310E+05,	2.17E-01,	7.03E-01,	3.58E-01,	2.4621E+06, 2.4621E+06,
32,	1,	0,	1.0000,	1,	1.0142E+00,	1.010003E+05,	1.024310E+05,	1.42E-02,	1.23E-01,	5.80E-03,	2.4621E+06, 2.3444E+06,
32,	2,	0,	1.0000,	2,	1.0009E+00,	1.023368E+05,	1.024310E+05,	9.20E-04,	1.10E-02,	7.05E-05,	2.4621E+06, 2.4598E+06,
32,	3,	0,	1.0000,	3,	9.9993E-01,	1.024387E+05,	1.024310E+05,-7.47E-05,	2.40E-03,	3.63E-06,	2.4621E+06,	2.4621E+06,
32,	4,	0,	1.0000,	4,	9.9991E-01,	1.024401E+05,	1.024310E+05,-8.86E-05,	6.80E-04,	2.70E-07,	2.4621E+06,	2.4621E+06,
32,	5,	0.	1.0000,	5,	9.9996E-01,	1.024352E+05,	1.024310E+05,-4.07E-05,	2.04E-04,	2.23E-08,	2.4621E+06,	2.4621E+06,
32,	6,	0,	1.0000,	6,	9.9998E-01,	1.024327E+05,	1.024310E+05,-1.66E-05,	6.19E-05,	1.91E-09,	2.4621E+06,	2.4621E+06,
32,	7,	0,	1.0000,	7,	9.9999E-01,	1.024316E+05,	1.024310E+05,-5.66E-06,	1.90E-05,	1.72E-10,	2.4621E+06,	2.4621E+06,

k,it0	,it	,ci	,j	,armag	,arak	,ara0k	,reler	,RMS	,DMhsum	,trgtArea	,curtArea
33,	0,	0,	1.0000,	0,	1.2166E+00,	6.811949E+04,	8.287413E+04,	2.17E-01,	7.03E-01,	3.58E-01,	2.4621E+06, 2.4621E+06,
33,	1,	0,	1.0000,	1,	1.0263E+00,	8.074724E+04,	8.287413E+04,	2.63E-02,	1.23E-01,	5.80E-03,	2.4621E+06, 2.3444E+06,
33,	2,	0,	1.0000,	2,	1.0071E+00,	8.228819E+04,	8.287413E+04,	7.12E-03,	1.10E-02,	7.05E-05,	2.4621E+06, 2.4598E+06,
33,	3,	0,	1.0000,	3,	1.0017E+00,	8.273668E+04,	8.287413E+04,	1.66E-03,	2.40E-03,	3.63E-06,	2.4621E+06, 2.4621E+06,
33,	4,	0,	1.0000,	4,	1.0004E+00,	8.284084E+04,	8.287413E+04,	4.02E-04,	6.80E-04,	2.70E-07,	2.4621E+06, 2.4621E+06,
33,	5,	0,	1.0000,	5,	1.0001E+00,	8.286610E+04,	8.287413E+04,	9.69E-05,	2.04E-04,	2.23E-08,	2.4621E+06, 2.4621E+06,
33,	6,	0,	1.0000,	6,	1.0000E+00,	8.287219E+04,	8.287413E+04,	2.35E-05,	6.19E-05,	1.91E-09,	2.4621E+06, 2.4621E+06,
33,	7,	0,	1.0000,	7,	1.0000E+00,	8.287352E+04,	8.287413E+04,	7.39E-06,	1.90E-05,	1.72E-10,	2.4621E+06, 2.4621E+06,

k,it0	,it	,ci	,j	,armag	,arak	,ara0k	,reler	,RMS	,DMhsum	,trgtArea	,curtArea
34,	0,	0,	1.0000,	0,	1.2166E+00,	7.856052E+04,	9.557666E+04,	2.17E-01,	7.03E-01,	3.58E-01,	2.4621E+06, 2.4621E+06,
34,	1,	0,	1.0000,	1,	1.0267E+00,	9.308718E+04,	9.557666E+04,	2.67E-02,	1.23E-01,	5.80E-03,	2.4621E+06, 2.3444E+06,
34,	2,	0,	1.0000,	2,	1.0034E+00,	9.525533E+04,	9.557666E+04,	3.37E-03,	1.10E-02,	7.05E-05,	2.4621E+06, 2.4598E+06,
34,	3,	0,	1.0000,	3,	1.0009E+00,	9.548791E+04,	9.557666E+04,	9.29E-04,	2.40E-03,	3.63E-06,	2.4621E+06, 2.4621E+06,
34,	4,	0,	1.0000,	4,	1.0003E+00,	9.554794E+04,	9.557666E+04,	3.01E-04,	6.80E-04,	2.70E-07,	2.4621E+06, 2.4621E+06,
34,	5,	0,	1.0000,	5,	1.0001E+00,	9.556708E+04,	9.557666E+04,	1.00E-04,	2.04E-04,	2.23E-08,	2.4621E+06, 2.4621E+06,
34,	6,	0,	1.0000,	6,	1.0000E+00,	9.557346E+04,	9.557666E+04,	3.34E-05,	6.19E-05,	1.91E-09,	2.4621E+06, 2.4621E+06,
34,	7,	0,	1.0000,	7,	1.0000E+00,	9.557560E+04,	9.557666E+04,	1.11E-05,	1.90E-05,	1.72E-10,	2.4621E+06, 2.4621E+06,

k,it0	,it	,ci	,j	,armag	,arak	,ara0k	,reler	,RMS	,DMhsum	,trgtArea	,curtArea
35,	0,	0,	1.0000,	0,	1.4816E+00,	7.133650E+04,	1.056893E+05,	4.82E-01,	7.03E-01,	3.58E-01,	2.4621E+06, 2.4621E+06,
35,	1,	0,	1.0000,	1,	1.0295E+00,	1.026631E+05,	1.056893E+05,	2.95E-02,	1.23E-01,	5.80E-03,	2.4621E+06, 2.3444E+06,
35,	2,	0,	1.0000,	2,	9.8992E-01,	1.067655E+05,	1.056893E+05,-1.01E-02,	1.10E-02,	7.05E-05,	2.4621E+06,	2.4598E+06,
35,	3,	0,	1.0000,	3,	9.9784E-01,	1.059181E+05,	1.056893E+05,-2.16E-03,	2.40E-03,	3.63E-06,	2.4621E+06,	2.4621E+06,
35,	4,	0,	1.0000,	4,	9.9951E-01,	1.057411E+05,	1.056893E+05,-4.90E-04,	6.80E-04,	2.70E-07,	2.4621E+06,	2.4621E+06,
35,	5,	0,	1.0000,	5,	9.9989E-01,	1.057014E+05,	1.056893E+05,-1.15E-04,	2.04E-04,	2.23E-08,	2.4621E+06,	2.4621E+06,
35,	6,	0,	1.0000,	6,	9.9997E-01,	1.056920E+05,	1.056893E+05,-2.60E-05,	6.19E-05,	1.91E-09,	2.4621E+06,	2.4621E+06,
35,	7,	0,	1.0000,	7,	9.9999E-01,	1.056898E+05,	1.056893E+05,-5.01E-06,	1.90E-05,	1.72E-10,	2.4621E+06,	2.4621E+06,

k,it0	,it	,ci	,j	,armag	,arak	,ara0k	,reler	,RMS	,DMhsum	,trgtArea	,curtArea
36,	0,	0,	1.0000,	0,	1.4816E+00,	4.186151E+04,	6.202032E+04,	4.82E-01,	7.03E-01,	3.58E-01,	2.4621E+06, 2.4621E+06,
36,	1,	0,	1.0000,	1,	1.1576E+00,	5.357581E+04,	6.202032E+04,	1.58E-01,	1.23E-01,	5.80E-03,	2.4621E+06, 2.3444E+06,
36,	2,	0,	1.0000,	2,	1.0135E+00,	6.119549E+04,	6.202032E+04,	1.35E-02,	1.10E-02,	7.05E-05,	2.4621E+06, 2.4598E+06,
36,	3,	0,	1.0000,	3,	1.0028E+00,	6.184607E+04,	6.202032E+04,	2.82E-03,	2.40E-03,	3.63E-06,	2.4621E+06, 2.4621E+06,
36,	4,	0,	1.0000,	4,	1.0007E+00,	6.197782E+04,	6.202032E+04,	6.86E-04,	6.80E-04,	2.70E-07,	2.4621E+06, 2.4621E+06,
36,	5,	0,	1.0000,	5,	1.0002E+00,	6.200935E+04,	6.202032E+04,	1.77E-04,	2.04E-04,	2.23E-08,	2.4621E+06, 2.4621E+06,
36,	6,	0,	1.0000,	6,	1.0000E+00,	6.201738E+04,	6.202032E+04,	4.74E-05,	6.19E-05,	1.91E-09,	2.4621E+06, 2.4621E+06,
36,	7,	0,	1.0000,	7,	1.0000E+00,	6.201948E+04,	6.202032E+04,	1.35E-05,	1.90E-05,	1.72E-10,	2.4621E+06, 2.4621E+06,

k,it0	,it	,ci	,j	,armag	,arak	,ara0k	,reler	,RMS	,DMhsum	,trgtArea	,curtArea
37,	0,	0,	1.0000,	0,	1.4816E+00,	5.704100E+04,	8.450966E+04,	4.82E-01,	7.03E-01,	3.58E-01,	2.4621E+06, 2.4621E+06,
37,	1,	0,	1.0000,	1,	1.1390E+00,	7.419786E+04,	8.450966E+04,	1.39E-01,	1.23E-01,	5.80E-03,	2.4621E+06, 2.3444E+06,
37,	2,	0,	1.0000,	2,	1.0101E+00,	8.366793E+04,	8.450966E+04,	1.01E-02,	1.10E-02,	7.05E-05,	2.4621E+06, 2.4598E+06,
37,	3,	0,	1.0000,	3,	1.0028E+00,	8.427020E+04,	8.450966E+04,	2.84E-03,	2.40E-03,	3.63E-06,	2.4621E+06, 2.4621E+06,
37,	4,	0,	1.0000,	4,	1.0008E+00,	8.443862E+04,	8.450966E+04,	8.41E-04,	6.80E-04,	2.70E-07,	2.4621E+06, 2.4621E+06,
37,	5,	0,	1.0000,	5,	1.0002E+00,	8.448902E+04,	8.450966E+04,	2.44E-04,	2.04E-04,	2.23E-08,	2.4621E+06, 2.4621E+06,
37,	6,	0,	1.0000,	6,	1.0001E+00,	8.450373E+04,	8.450966E+04,	7.01E-05,	6.19E-05,	1.91E-09,	2.4621E+06, 2.4621E+06,
37,	7,	0,	1.0000,	7,	1.0000E+00,	8.450805E+04,	8.450966E+04,	1.90E-05,	1.90E-05,	1.72E-10,	2.4621E+06, 2.4621E+06,

k,it0	,it	,ci	,j	,armag	,arak	,ara0k	,reler	,RMS	,DMhsum	,trgtArea	,curtArea
38,	0,	0,	1.0000,	0,	9.5509E-01,	8.288999E+04,	7.916730E+04,-4.49E-02,	7.03E-01,	3.58E-01,	2.4621E+06,	2.4621E+06,
38,	1,	0,	1.0000,	1,	1.0088E+00,	7.847721E+04,	7.916730E+04,	8.79E-03,	1.23E-01,	5.80E-03,	2.4621E+06, 2.3444E+06,
38,	2,	0,	1.0000,	2,	9.9907E-01,	7.924110E+04,	7.916730E+04,-9.31E-04,	1.10E-02,	7.05E-05,	2.4621E+06,	2.4598E+06,
38,	3,	0,	1.0000,	3,	9.9954E-01,	7.920399E+04,	7.916730E+04,-4.63E-04,	2.40E-03,	3.63E-06,	2.4621E+06,	2.4621E+06,
38,	4,	0,	1.0000,	4,	9.9985E-01,	7.917908E+04,	7.916730E+04,-1.49E-04,	6.80E-04,	2.70E-07,	2.4621E+06,	2.4621E+06,
38,	5,	0,	1.0000,	5,	9.9995E-01,	7.917101E+04,	7.916730E+04,-4.69E-05,	2.04E-04,	2.23E-08,	2.4621E+06,	2.4621E+06,
38,	6,	0,	1.0000,	6,	9.9999E-01,	7.916844E+04,	7.916730E+04,-1.44E-05,	6.19E-05,	1.91E-09,	2.4621E+06,	2.4621E+06,
38,	7,	0,	1.0000,	7,	1.0000E+00,	7.916766E+04,	7.916730E+04,-4.47E-06,	1.90E-05,	1.72E-10,	2.4621E+06,	2.4621E+06,

k,it0	,it	,ci	,j	,armag	,arak	,ara0k	,reler	,RMS	,DMhsum	,trgtArea	,curtArea
39,	0,	0,	1.0000,	0,	9.5509E-01,	1.032655E+05,	9.862771E+04,-4.49E-02,	7.03E-01,	3.58E-01,	2.4621E+06,	2.4621E+06,
39,	1,	0,	1.0000,	1,	1.0146E+00,	9.720782E+04,	9.862771E+04,	1.46E-02,	1.23E-01,	5.80E-03,	2.4621E+06, 2.3444E+06,
39,	2,	0,	1.0000,	2,	1.0051E+00,	9.812308E+04,	9.862771E+04,	5.14E-03,	1.10E-02,	7.05E-05,	2.4621E+06, 2.4598E+06,
39,	3,	0,	1.0000,	3,	1.0012E+00,	9.850704E+04,	9.862771E+04,	1.22E-03,	2.40E-03,	3.63E-06,	2.4621E+06, 2.4621E+06,
39,	4,	0,	1.0000,	4,	1.0003E+00,	9.859916E+04,	9.862771E+04,	2.90E-04,	6.80E-04,	2.70E-07,	2.4621E+06, 2.4621E+06,
39,	5,	0,	1.0000,	5,	1.0001E+00,	9.862083E+04,	9.862771E+04,	6.97E-05,	2.04E-04,	2.23E-08,	2.4621E+06, 2.4621E+06,
39,	6,	0,	1.0000,	6,	1.0000E+00,	9.862599E+04,	9.862771E+04,	1.74E-05,	6.19E-05,	1.91E-09,	2.4621E+06, 2.4621E+06,
39,	7,	0,	1.0000,	7,	1.0000E+00,	9.862725E+04,	9.862771E+04,	4.77E-06,	1.90E-05,	1.72E-10,	2.4621E+06, 2.4621E+06,

k,it0	,it	,ci	,j	,armag	,arak	,ara0k	,reler	,RMS	,DMhsum	,trgtArea	,curtArea
40,	0,	0,	1.0000,	0,	8.9850E-01,	6.010600E+04,	5.400530E+04,-1.01E-01,	7.03E-01,	3.58E-01,	2.4621E+06,	2.4621E+06,
40,	1,	0,	1.0000,	1,	1.0582E+00,	5.103684E+04,	5.400530E+04,	5.82E-02,	1.23E-01,	5.80E-03,	2.4621E+06, 2.3444E+06,
40,	2,	0.	1.0000,	2,	1.0031E+00,	5.384059E+04,	5.400530E+04,	3.06E-03,	1.10E-02,	7.05E-05,	2.4621E+06, 2.4598E+06,
40,	3,	0,	1.0000,	3,	1.0007E+00,	5.396571E+04,	5.400530E+04,	7.33E-04,	2.40E-03,	3.63E-06,	2.4621E+06, 2.4621E+06,
40,	4,	0,	1.0000,	4,	1.0001E+00,	5.399722E+04,	5.400530E+04,	1.50E-04,	6.80E-04,	2.70E-07,	2.4621E+06, 2.4621E+06,
40,	5,	0,	1.0000,	5,	1.0000E+00,	5.400437E+04,	5.400530E+04,	1.72E-05,	2.04E-04,	2.23E-08,	2.4621E+06, 2.4621E+06,
40,	6,	0,	1.0000,	6,	1.0000E+00,	5.400553E+04,	5.400530E+04,-4.23E-06,	6.19E-05,	1.91E-09,	2.4621E+06,	2.4621E+06,
40,	7,	0,	1.0000,	7,	1.0000E+00,	5.400545E+04,	5.400530E+04,-2.68E-06,	1.90E-05,	1.72E-10,	2.4621E+06,	2.4621E+06,

k,it0	,it	,ci	,j	,armag	,arak	,ara0k	,reler	,RMS	,DMhsum	,trgtArea	,curtArea
41,	0,	0,	1.0000,	0,	8.9850E-01,	6.572043E+03,	5.904949E+03,	-1.02E-01,	7.03E-01,	3.58E-01,	2.4621E+06, 2.4621E+06,
41,	1,	0,	1.0000,	1,	1.0884E+00,	5.425308E+03,	5.904949E+03,	8.84E-02,	1.23E-01,	5.80E-03,	2.4621E+06, 2.3444E+06,
41,	2,	0,	1.0000,	2,	1.0142E+00,	5.822303E+03,	5.904949E+03,	1.42E-02,	1.10E-02,	7.05E-05,	2.4621E+06, 2.4598E+06,
41,	3,	0,	1.0000,	3,	1.0041E+00,	5.880849E+03,	5.904949E+03,	4.10E-03,	2.40E-03,	3.63E-06,	2.4621E+06, 2.4621E+06,
41,	4,	0,	1.0000,	4,	1.0012E+00,	5.897970E+03,	5.904949E+03,	1.18E-03,	6.80E-04,	2.70E-07,	2.4621E+06, 2.4621E+06,
41,	5,	0,	1.0000,	5,	1.0004E+00,	5.902872E+03,	5.904949E+03,	3.52E-04,	2.04E-04,	2.23E-08,	2.4621E+06, 2.4621E+06,
41,	6,	0,	1.0000,	6,	1.0001E+00,	5.904297E+03,	5.904949E+03,	1.11E-04,	6.19E-05,	1.91E-09,	2.4621E+06, 2.4621E+06,
41,	7,	0,	1.0000,	7,	1.0000E+00,	5.904768E+03,	5.904949E+03,	3.08E-05,	1.90E-05,	1.72E-10,	2.4621E+06, 2.4621E+06,

k,it0	,it	,ci	,j	,armag	,arak	,ara0k	,reler	,RMS	,DMhsum	,trgtArea	,curtArea
42,	0,	0,	1.0000,	0,	8.9850E-01,	2.201850E+04,	1.978364E+04,	-1.01E-01,	7.03E-01,	3.58E-01,	2.4621E+06, 2.4621E+06,
42,	1,	0,	1.0000,	1,	1.0056E+00,	1.967322E+04,	1.978364E+04,	5.61E-03,	1.23E-01,	5.80E-03,	2.4621E+06, 2.3444E+06,
42,	2,	0,	1.0000,	2,	9.8759E-01,	2.003227E+04,	1.978364E+04,	-1.24E-02,	1.10E-02,	7.05E-05,	2.4621E+06, 2.4598E+06,
42,	3,	0,	1.0000,	3,	9.9642E-01,	1.985474E+04,	1.978364E+04,	-3.58E-03,	2.40E-03,	3.63E-06,	2.4621E+06, 2.4621E+06,
42,	4,	0,	1.0000,	4,	9.9898E-01,	1.980390E+04,	1.978364E+04,	-1.02E-03,	6.80E-04,	2.70E-07,	2.4621E+06, 2.4621E+06,
42,	5,	0,	1.0000,	5,	9.9970E-01,	1.978950E+04,	1.978364E+04,	-2.96E-04,	2.04E-04,	2.23E-08,	2.4621E+06, 2.4621E+06,
42,	6,	0,	1.0000,	6,	9.9992E-01,	1.978529E+04,	1.978364E+04,	-8.31E-05,	6.19E-05,	1.91E-09,	2.4621E+06, 2.4621E+06,
42,	7,	0,	1.0000,	7,	9.9998E-01,	1.978412E+04,	1.978364E+04,	-2.40E-05,	1.90E-05,	1.72E-10,	2.4621E+06, 2.4621E+06,

k,it0	,it	,ci	,j	,armag	,arak	,ara0k	,reler	,RMS	,DMhsum	,trgtArea	,curtArea
43,	0,	0,	1.0000,	0,	8.9850E-01,	7.589398E+04,	6.819084E+04,	-1.01E-01,	7.03E-01,	3.58E-01,	2.4621E+06, 2.4621E+06,
43,	1,	0,	1.0000,	1,	9.8760E-01,	6.904715E+04,	6.819084E+04,	-1.24E-02,	1.23E-01,	5.80E-03,	2.4621E+06, 2.3444E+06,
43,	2,	0,	1.0000,	2,	9.9479E-01,	6.854829E+04,	6.819084E+04,	-5.21E-03,	1.10E-02,	7.05E-05,	2.4621E+06, 2.4598E+06,
43,	3,	0,	1.0000,	3,	9.9845E-01,	6.829675E+04,	6.819084E+04,	-1.55E-03,	2.40E-03,	3.63E-06,	2.4621E+06, 2.4621E+06,
43,	4,	0,	1.0000,	4,	9.9957E-01,	6.821998E+04,	6.819084E+04,	-4.27E-04,	6.80E-04,	2.70E-07,	2.4621E+06, 2.4621E+06,
43,	5,	0,	1.0000,	5,	9.9988E-01,	6.819889E+04,	6.819084E+04,	-1.18E-04,	2.04E-04,	2.23E-08,	2.4621E+06, 2.4621E+06,
43,	6,	0,	1.0000,	6,	9.9997E-01,	6.819320E+04,	6.819084E+04,	-3.45E-05,	6.19E-05,	1.91E-09,	2.4621E+06, 2.4621E+06,
43,	7,	0,	1.0000,	7,	9.9999E-01,	6.819177E+04,	6.819084E+04,	-1.36E-05,	1.90E-05,	1.72E-10,	2.4621E+06, 2.4621E+06,

k,it0	,it	,ci	,j	,armag	,arak	,ara0k	,reler	,RMS	,DMhsum	,trgtArea	,curtArea
44,	0,	0,	1.0000,	0,	8.9850E-01,	1.088180E+05,	9.777309E+04,	-1.01E-01,	7.03E-01,	3.58E-01,	2.4621E+06, 2.4621E+06,
44,	1,	0,	1.0000,	1,	1.0294E+00,	9.498176E+04,	9.777309E+04,	2.94E-02,	1.23E-01,	5.80E-03,	2.4621E+06, 2.3444E+06,
44,	2,	0,	1.0000,	2,	9.9980E-01,	9.779236E+04,	9.777309E+04,	-1.97E-04,	1.10E-02,	7.05E-05,	2.4621E+06, 2.4598E+06,
44,	3,	0,	1.0000,	3,	9.9994E-01,	9.777896E+04,	9.777309E+04,	-6.00E-05,	2.40E-03,	3.63E-06,	2.4621E+06, 2.4621E+06,
44,	4,	0,	1.0000,	4,	9.9999E-01,	9.777391E+04,	9.777309E+04,	-8.34E-06,	6.80E-04,	2.70E-07,	2.4621E+06, 2.4621E+06,
44,	5,	0,	1.0000,	5,	1.0000E+00,	9.777323E+04,	9.777309E+04,	-1.49E-06,	2.04E-04,	2.23E-08,	2.4621E+06, 2.4621E+06,
44,	6,	0,	1.0000,	6,	1.0000E+00,	9.777315E+04,	9.777309E+04,	-5.36E-07,	6.19E-05,	1.91E-09,	2.4621E+06, 2.4621E+06,
44,	7,	0,	1.0000,	7,	1.0000E+00,	9.777321E+04,	9.777309E+04,	-1.25E-06,	1.90E-05,	1.72E-10,	2.4621E+06, 2.4621E+06,

Boundary Point Table

Boundary 1

j.	0x ,	0y ,	7x ,	7y
axy,	11.000000,	-998.000000,	9.401873,	-994.352722
axy,	-1.000000,	-1273.000000,	-1.869430,	-1263.323975
axy,	-180.000000,	-1270.000000,	-180.841125,	-1257.679932
axy,	-358.000000,	-1266.000000,	-359.575073,	-1245.164795
axy,	-501.000000,	-1265.000000,	-488.256683,	-1240.443481
axy,	-644.000000,	-1263.000000,	-617.665894,	-1248.051392
axy,	-638.000000,	-942.000000,	-596.748108,	-955.978455
axy,	-632.000000,	-621.000000,	-612.597778,	-673.140381
axy,	-675.000000,	-381.000000,	-676.158875,	-411.656769
axy,	-718.000000,	-140.000000,	-707.533203,	-153.444916
axy,	-699.000000,	149.000000,	-677.513489,	78.632324
axy,	-679.000000,	439.000000,	-766.481873,	303.306000
axy,	-699.000000,	605.000000,	-840.126404,	571.255676
axy,	-718.000000,	771.000000,	-848.293274,	833.703247
axy,	-646.000000,	1018.000000,	-655.477478,	1132.310669
axy,	-574.000000,	1266.000000,	-531.602844,	1332.687378
axy,	-322.000000,	1266.000000,	-234.961182,	1307.384766
axy,	-70.000000,	1266.000000,	48.752579,	1250.923950
axy,	201.000000,	1269.000000,	285.146027,	1199.844849
axy,	473.000000,	1273.000000,	523.943420,	1158.991821
axy,	627.000000,	1272.000000,	610.355286,	1153.031128
axy,	781.000000,	1271.000000,	712.000000,	1204.276611
axy,	647.000000,	895.000000,	559.709473,	944.903198
axy,	513.000000,	519.000000,	508.993103,	602.339722
axy,	429.000000,	420.000000,	448.640564,	491.249725
axy,	345.000000,	322.000000,	383.456665,	373.357269
axy,	277.000000,	103.000000,	299.373657,	127.141075
axy,	209.000000,	-116.000000,	230.555084,	-90.248100
axy,	116.000000,	-420.000000,	133.624588,	-415.297424
axy,	23.000000,	-723.000000,	28.572025,	-722.534119

Boundary 2

j,	0x ,	0y ,	7x ,	7y
axy,	-532.000000,	193.000000,	-515.559448,	88.396652
axy,	-384.000000,	-52.000000,	-343.671356,	-64.068535
axy,	-383.000000,	145.000000,	-338.656769,	59.116077
axy,	-381.000000,	343.000000,	-313.386993,	174.519348
axy,	-530.000000,	391.000000,	-544.720215,	193.599030
axy,	-679.000000,	439.000000,	-766.481873,	303.306000

Boundary 3

j,	0x ,	0y ,	7x ,	7y
axy,	-321.000000,	202.000000,	-261.714020,	95.379990
axy,	-261.000000,	62.000000,	-191.640198,	13.235323
axy,	-235.000000,	119.000000,	-154.124207,	46.477711
axy,	-209.000000,	176.000000,	-115.175537,	72.916588
axy,	-295.000000,	259.000000,	-231.296783,	133.506653
axy,	-381.000000,	343.000000,	-313.386993,	174.519348

Boundary 4

j,	0x ,	0y ,	7x ,	7y
axy,	-532.000000,	193.000000,	-515.559448,	88.396652
axy,	-679.000000,	439.000000,	-766.481873,	303.306000
axy,	-699.000000,	149.000000,	-677.513489,	78.632324
axy,	-718.000000,	-140.000000,	-707.533203,	-153.444916
axy,	-551.000000,	-96.000000,	-534.641113,	-95.451965
axy,	-384.000000,	-52.000000,	-343.671356,	-64.068535

Boundary 5

j,	0x ,	0y ,	7x ,	7y
axy,	-383.000000,	145.000000,	-338.656769,	59.116077
axy,	-384.000000,	-52.000000,	-343.671356,	-64.068535
axy,	-323.000000,	5.000000,	-270.298584,	-29.423809
axy,	-261.000000,	62.000000,	-191.640198,	13.235323
axy,	-321.000000,	202.000000,	-261.714020,	95.379990
axy,	-381.000000,	343.000000,	-313.386993,	174.519348

Boundary 6

j,	0x ,	0y ,	7x ,	7y
axy,	-501.000000,	-1265.000000,	-488.256683,	-1240.443481
axy,	-358.000000,	-1266.000000,	-359.575073,	-1245.164795
axy,	-318.000000,	-978.000000,	-326.271240,	-980.409363
axy,	-277.000000,	-690.000000,	-262.703949,	-721.730042
axy,	-461.000000,	-977.000000,	-448.562103,	-983.750305
axy,	-644.000000,	-1263.000000,	-617.665894,	-1248.051392

Boundary 7

j,	0x ,	0y ,	7x ,	7y
axy,	-455.000000,	-656.000000,	-433.615295,	-715.851990
axy,	-277.000000,	-690.000000,	-262.703949,	-721.730042
axy,	-296.000000,	-666.000000,	-276.094086,	-704.523926
axy,	-314.000000,	-642.000000,	-286.594849,	-687.949036
axy,	-473.000000,	-632.000000,	-451.417267,	-697.147583
axy,	-632.000000,	-621.000000,	-612.597778,	-673.140381

Boundary 8

j,	0x ,	0y ,	7x ,	7y
axy,	-638.000000,	-942.000000,	-596.748108,	-955.978455
axy,	-644.000000,	-1263.000000,	-617.665894,	-1248.051392
axy,	-461.000000,	-977.000000,	-448.562103,	-983.750305
axy,	-277.000000,	-690.000000,	-262.703949,	-721.730042
axy,	-455.000000,	-656.000000,	-433.615295,	-715.851990
axy,	-632.000000,	-621.000000,	-612.597778,	-673.140381

Boundary 9

j,	0x ,	0y ,	7x ,	7y
axy,	282.000000,	611.000000,	386.776764,	666.290039
axy,	513.000000,	519.000000,	508.993103,	602.339722
axy,	300.000000,	653.000000,	400.889313,	711.463745
axy,	88.000000,	788.000000,	260.919250,	854.503601
axy,	70.000000,	746.000000,	252.056168,	815.220642
axy,	52.000000,	704.000000,	247.620010,	783.745972

Boundary 10

j,	0x ,	0y ,	7x ,	7y
axy,	198.000000,	513.000000,	339.197510,	556.423035
axy,	52.000000,	704.000000,	247.620010,	783.745972
axy,	74.000000,	536.000000,	283.163422,	575.037903
axy,	96.000000,	368.000000,	256.272644,	352.246674
axy,	220.000000,	345.000000,	312.964081,	369.805237
axy,	345.000000,	322.000000,	383.456665,	373.357269

Boundary 11

j,	0x ,	0y ,	7x ,	7y
axy,	282.000000,	611.000000,	386.776764,	666.290039
axy,	52.000000,	704.000000,	247.620010,	783.745972
axy,	198.000000,	513.000000,	339.197510,	556.423035
axy,	345.000000,	322.000000,	383.456665,	373.357269
axy,	429.000000,	420.000000,	448.640564,	491.249725
axy,	513.000000,	519.000000,	508.993103,	602.339722

Boundary 12

j,	0x ,	0y ,	7x ,	7y
axy,	300.000000,	653.000000,	400.889313,	711.463745
axy,	513.000000,	519.000000,	508.993103,	602.339722
axy,	469.000000,	758.000000,	518.927246,	821.334167
axy,	426.000000,	998.000000,	508.232666,	996.978210
axy,	257.000000,	893.000000,	366.541321,	925.174377
axy,	88.000000,	788.000000,	260.919250,	854.503601

Boundary 13

j,	0x ,	0y ,	7x ,	7y
axy,	-320.000000,	739.000000,	-80.585312,	849.250244
axy,	-359.000000,	832.000000,	-157.334549,	976.008301
axy,	-539.000000,	801.000000,	-502.493042,	909.123901
axy,	-718.000000,	771.000000,	-848.293274,	833.703247
axy,	-499.000000,	709.000000,	-441.125549,	760.250427
axy,	-280.000000,	647.000000,	-52.794598,	710.502075

Boundary 14

j,	0x ,	0y ,	7x ,	7y
axy,	-480.000000,	543.000000,	-429.957733,	483.144104
axy,	-679.000000,	439.000000,	-766.481873,	303.306000
axy,	-530.000000,	391.000000,	-544.720215,	193.599030
axy,	-381.000000,	343.000000,	-313.386993,	174.519348
axy,	-331.000000,	495.000000,	-159.253159,	425.625275
axy,	-280.000000,	647.000000,	-52.794598,	710.502075

Boundary 15

j,	0x ,	0y ,	7x ,	7y
axy,	-539.000000,	801.000000,	-502.493042,	909.123901
axy,	-359.000000,	832.000000,	-157.334549,	976.008301
axy,	-481.000000,	884.000000,	-372.888977,	1064.368774
axy,	-602.000000,	937.000000,	-586.901367,	1090.889038
axy,	-660.000000,	854.000000,	-741.174744,	975.344482
axy,	-718.000000,	771.000000,	-848.293274,	833.703247

Boundary 16

j,	0x ,	0y ,	7x ,	7y
axy,	-480.000000,	543.000000,	-429.957733,	483.144104
axy,	-280.000000,	647.000000,	-52.794598,	710.502075
axy,	-499.000000,	709.000000,	-441.125549,	760.250427
axy,	-718.000000,	771.000000,	-848.293274,	833.703247
axy,	-699.000000,	605.000000,	-840.126404,	571.255676
axy,	-679.000000,	439.000000,	-766.481873,	303.306000

Boundary 17

j,	0x ,	0y ,	7x ,	7y
axy,	469.000000,	758.000000,	518.927246,	821.334167
axy,	513.000000,	519.000000,	508.993103,	602.339722
axy,	647.000000,	895.000000,	559.709473,	944.903198
axy,	781.000000,	1271.000000,	712.000000,	1204.276611
axy,	603.000000,	1134.000000,	579.183655,	1092.731567
axy,	426.000000,	998.000000,	508.232666,	996.978210

Boundary 18

j,	0x ,	0y ,	7x ,	7y
axy,	603.000000,	1134.000000,	579.183655,	1092.731567
axy,	781.000000,	1271.000000,	712.000000,	1204.276611
axy,	627.000000,	1272.000000,	610.355286,	1153.031128
axy,	473.000000,	1273.000000,	523.943420,	1158.991821
axy,	449.000000,	1135.000000,	525.108154,	1081.180664
axy,	426.000000,	998.000000,	508.232666,	996.978210

Boundary 19

j,	0x ,	0y ,	7x ,	7y
axy,	-220.000000,	921.000000,	-46.983383,	992.348511
axy,	-81.000000,	1010.000000,	91.568008,	1051.850464
axy,	-328.000000,	1138.000000,	-222.164230,	1203.621704
axy,	-574.000000,	1266.000000,	-531.602844,	1332.687378
axy,	-467.000000,	1049.000000,	-385.897858,	1151.967529
axy,	-359.000000,	832.000000,	-157.334549,	976.008301

Boundary 20

j,	0x ,	0y ,	7x ,	7y
axy,	-328.000000,	1138.000000,	-222.164230,	1203.621704
axy,	-81.000000,	1010.000000,	91.568008,	1051.850464
axy,	-76.000000,	1138.000000,	77.085602,	1151.731689
axy,	-70.000000,	1266.000000,	48.752579,	1250.923950
axy,	-322.000000,	1266.000000,	-234.961182,	1307.384766
axy,	-574.000000,	1266.000000,	-531.602844,	1332.687378

Boundary 21

j,	0x ,	0y ,	7x ,	7y
axy,	-481.000000,	884.000000,	-372.888977,	1064.368774
axy,	-359.000000,	832.000000,	-157.334549,	976.008301
axy,	-467.000000,	1049.000000,	-385.897858,	1151.967529
axy,	-574.000000,	1266.000000,	-531.602844,	1332.687378
axy,	-588.000000,	1101.000000,	-536.120972,	1194.441772
axy,	-602.000000,	937.000000,	-586.901367,	1090.889038

Boundary 22

j,	0x ,	0y ,	7x ,	7y
axy,	-660.000000,	854.000000,	-741.174744,	975.344482
axy,	-602.000000,	937.000000,	-586.901367,	1090.889038
axy,	-588.000000,	1101.000000,	-536.120972,	1194.441772
axy,	-574.000000,	1266.000000,	-531.602844,	1332.687378
axy,	-646.000000,	1018.000000,	-655.477478,	1132.310669
axy,	-718.000000,	771.000000,	-848.293274,	833.703247

Boundary 23

j,	0x ,	0y ,	7x ,	7y
axy,	-114.000000,	675.000000,	69.687378,	770.182617
axy,	52.000000,	704.000000,	247.620010,	783.745972
axy,	-15.000000,	857.000000,	157.994598,	922.495483
axy,	-81.000000,	1010.000000,	91.568008,	1051.850464
axy,	-181.000000,	828.000000,	5.738637,	910.465332
axy,	-280.000000,	647.000000,	-52.794598,	710.502075

Boundary 24

j,	0x ,	0y ,	7x ,	7y
axy,	-320.000000,	739.000000,	-80.585312,	849.250244
axy,	-280.000000,	647.000000,	-52.794598,	710.502075
axy,	-181.000000,	828.000000,	5.738637,	910.465332
axy,	-81.000000,	1010.000000,	91.568008,	1051.850464
axy,	-220.000000,	921.000000,	-46.983383,	992.348511
axy,	-359.000000,	832.000000,	-157.334549,	976.008301

Boundary 25

j,	0x ,	0y ,	7x ,	7y
axy,	-15.000000,	857.000000,	157.994598,	922.495483
axy,	52.000000,	704.000000,	247.620010,	783.745972
axy,	70.000000,	746.000000,	252.056168,	815.220642
axy,	88.000000,	788.000000,	260.919250,	854.503601
axy,	3.000000,	899.000000,	176.763718,	960.436401
axy,	-81.000000,	1010.000000,	91.568008,	1051.850464

Boundary 26

j,	0x ,	0y ,	7x ,	7y
axy,	0.000000,	30.000000,	77.599876,	29.133595
axy,	209.000000,	-116.000000,	230.555084,	-90.248100
axy,	152.000000,	126.000000,	204.171356,	130.163223
axy,	96.000000,	368.000000,	256.272644,	352.246674
axy,	-57.000000,	272.000000,	93.024643,	183.847168
axy,	-209.000000,	176.000000,	-115.175537,	72.916588

Boundary 27

j,	0x ,	0y ,	7x ,	7y
axy,	152.000000,	126.000000,	204.171356,	130.163223
axy,	209.000000,	-116.000000,	230.555084,	-90.248100
axy,	277.000000,	103.000000,	299.373657,	127.141075
axy,	345.000000,	322.000000,	383.456665,	373.357269
axy,	220.000000,	345.000000,	312.964081,	369.805237
axy,	96.000000,	368.000000,	256.272644,	352.246674

Boundary 28

j,	0x ,	0y ,	7x ,	7y
axy,	0.000000,	30.000000,	77.599876,	29.133595
axy,	-209.000000,	176.000000,	-115.175537,	72.916588
axy,	-235.000000,	119.000000,	-154.124207,	46.477711
axy,	-261.000000,	62.000000,	-191.640198,	13.235323
axy,	-26.000000,	-27.000000,	47.242126,	-21.279043
axy,	209.000000,	-116.000000,	230.555084,	-90.248100

Boundary 29

j,	0x ,	0y ,	7x ,	7y
axy,	3.000000,	899.000000,	176.763718,	960.436401
axy,	88.000000,	788.000000,	260.919250,	854.503601
axy,	257.000000,	893.000000,	366.541321,	925.174377
axy,	426.000000,	998.000000,	508.232666,	996.978210
axy,	172.000000,	1004.000000,	291.733368,	1013.391602
axy,	-81.000000,	1010.000000,	91.568008,	1051.850464

Boundary 30

j,	0x ,	0y ,	7x ,	7y
axy,	178.000000,	1132.000000,	280.885834,	1103.372192
axy,	426.000000,	998.000000,	508.232666,	996.978210
axy,	449.000000,	1135.000000,	525.108154,	1081.180664
axy,	473.000000,	1273.000000,	523.943420,	1158.991821
axy,	201.000000,	1269.000000,	285.146027,	1199.844849
axy,	-70.000000,	1266.000000,	48.752579,	1250.923950

Boundary 31

j,	0x ,	0y ,	7x ,	7y
axy,	172.000000,	1004.000000,	291.733368,	1013.391602
axy,	426.000000,	998.000000,	508.232666,	996.978210
axy,	178.000000,	1132.000000,	280.885834,	1103.372192
axy,	-70.000000,	1266.000000,	48.752579,	1250.923950
axy,	-76.000000,	1138.000000,	77.085602,	1151.731689
axy,	-81.000000,	1010.000000,	91.568008,	1051.850464

Boundary 32

j,	0x ,	0y ,	7x ,	7y
axy,	-508.000000,	-337.000000,	-485.984375,	-361.928497
axy,	-384.000000,	-52.000000,	-343.671356,	-64.068535
axy,	-551.000000,	-96.000000,	-534.641113,	-95.451965
axy,	-718.000000,	-140.000000,	-707.533203,	-153.444916
axy,	-675.000000,	-381.000000,	-676.158875,	-411.656769
axy,	-632.000000,	-621.000000,	-612.597778,	-673.140381

Boundary 33

j,	0x ,	0y ,	7x ,	7y
axy,	-407.000000,	-422.000000,	-372.053497,	-453.895966
axy,	-632.000000,	-621.000000,	-612.597778,	-673.140381
axy,	-473.000000,	-632.000000,	-451.417267,	-697.147583
axy,	-314.000000,	-642.000000,	-286.594849,	-687.949036
axy,	-248.000000,	-433.000000,	-188.299805,	-461.952118
axy,	-182.000000,	-223.000000,	-114.463287,	-231.444458

Boundary 34

j,	0x ,	0y ,	7x ,	7y
axy,	-508.000000,	-337.000000,	-485.984375,	-361.928497
axy,	-632.000000,	-621.000000,	-612.597778,	-673.140381
axy,	-407.000000,	-422.000000,	-372.053497,	-453.895966
axy,	-182.000000,	-223.000000,	-114.463287,	-231.444458
axy,	-283.000000,	-138.000000,	-219.250107,	-142.166077
axy,	-384.000000,	-52.000000,	-343.671356,	-64.068535

Boundary 35

j,	0x ,	0y ,	7x ,	7y
axy,	-143.000000,	355.000000,	-4.429573,	295.824005
axy,	96.000000,	368.000000,	256.272644,	352.246674
axy,	-92.000000,	507.000000,	76.751442,	507.713623
axy,	-280.000000,	647.000000,	-52.794598,	710.502075
axy,	-331.000000,	495.000000,	-159.253159,	425.625275
axy,	-381.000000,	343.000000,	-313.386993,	174.519348

Boundary 36

j,	0x ,	0y ,	7x ,	7y
axy,	-143.000000,	355.000000,	-4.429573,	295.824005
axy,	-381.000000,	343.000000,	-313.386993,	174.519348
axy,	-295.000000,	259.000000,	-231.296783,	133.506653
axy,	-209.000000,	176.000000,	-115.175537,	72.916588
axy,	-57.000000,	272.000000,	93.024643,	183.847168
axy,	96.000000,	368.000000,	256.272644,	352.246674

Boundary 37

j,	0x ,	0y ,	7x ,	7y
axy,	-92.000000,	507.000000,	76.751442,	507.713623
axy,	96.000000,	368.000000,	256.272644,	352.246674
axy,	74.000000,	536.000000,	283.163422,	575.037903
axy,	52.000000,	704.000000,	247.620010,	783.745972
axy,	-114.000000,	675.000000,	69.687378,	770.182617
axy,	-280.000000,	647.000000,	-52.794598,	710.502075

Boundary 38

j,	0x ,	0y ,	7x ,	7y
axy,	11.000000,	-998.000000,	9.401873,	-994.352722
axy,	23.000000,	-723.000000,	28.572025,	-722.534119
axy,	-127.000000,	-707.000000,	-112.786957,	-716.313477
axy,	-277.000000,	-690.000000,	-262.703949,	-721.730042
axy,	-139.000000,	-982.000000,	-139.897232,	-981.271362
axy,	-1.000000,	-1273.000000,	-1.869430,	-1263.323975

Boundary 39

j,	0x ,	0y ,	7x ,	7y
axy,	-180.000000,	-1270.000000,	-180.841125,	-1257.679932
axy,	-1.000000,	-1273.000000,	-1.869430,	-1263.323975
axy,	-139.000000,	-982.000000,	-139.897232,	-981.271362
axy,	-277.000000,	-690.000000,	-262.703949,	-721.730042
axy,	-318.000000,	-978.000000,	-326.271240,	-980.409363
axy,	-358.000000,	-1266.000000,	-359.575073,	-1245.164795

Boundary 40

j,	0x ,	0y ,	7x ,	7y
axy,	13.000000,	-170.000000,	62.996777,	-165.425293
axy,	209.000000,	-116.000000,	230.555084,	-90.248100
axy,	-26.000000,	-27.000000,	47.242126,	-21.279043
axy,	-261.000000,	62.000000,	-191.640198,	13.235323
axy,	-222.000000,	-81.000000,	-158.464844,	-95.406982
axy,	-182.000000,	-223.000000,	-114.463287,	-231.444458

Boundary 41

j,	0x ,	0y ,	7x ,	7y
axy,	-146.000000,	-683.000000,	-128.763474,	-693.699829
axy,	-314.000000,	-642.000000,	-286.594849,	-687.949036
axy,	-296.000000,	-666.000000,	-276.094086,	-704.523926
axy,	-277.000000,	-690.000000,	-262.703949,	-721.730042
axy,	-127.000000,	-707.000000,	-112.786957,	-716.313477
axy,	23.000000,	-723.000000,	28.572025,	-722.534119

Boundary 42

j,	0x ,	0y ,	7x ,	7y
axy,	-283.000000,	-138.000000,	-219.250107,	-142.166077
axy,	-182.000000,	-223.000000,	-114.463287,	-231.444458
axy,	-222.000000,	-81.000000,	-158.464844,	-95.406982
axy,	-261.000000,	62.000000,	-191.640198,	13.235323
axy,	-323.000000,	5.000000,	-270.298584,	-29.423809
axy,	-384.000000,	-52.000000,	-343.671356,	-64.068535

Boundary 43

j,	0x ,	0y ,	7x ,	7y
axy,	-80.000000,	-473.000000,	-48.492336,	-484.705750
axy,	-182.000000,	-223.000000,	-114.463287,	-231.444458
axy,	-248.000000,	-433.000000,	-188.299805,	-461.952118
axy,	-314.000000,	-642.000000,	-286.594849,	-687.949036
axy,	-146.000000,	-683.000000,	-128.763474,	-693.699829
axy,	23.000000,	-723.000000,	28.572025,	-722.534119

Boundary 44

j,	0x ,	0y ,	7x ,	7y
axy,	13.000000,	-170.000000,	62.996777,	-165.425293
axy,	-182.000000,	-223.000000,	-114.463287,	-231.444458
axy,	-80.000000,	-473.000000,	-48.492336,	-484.705750
axy,	23.000000,	-723.000000,	28.572025,	-722.534119
axy,	116.000000,	-420.000000,	133.624588,	-415.297424
axy,	209.000000,	-116.000000,	230.555084,	-90.248100

Non-Boundary Point Table

j,	0x ,	0y ,	7x ,	7y

End file:RLInt.out.plot

C.3.5 File RLInt.sum.0007

After seven iterations, the map had fully converged to single precision accuracy, about five significant figures. This file shows for each region the current area, target area, the current region magnification M_k, and the current relative error $M_k - 1$. If fully converged, the magnification for each region should be 1.

DEMP input file in CLOSE3 format
vt_500k2_hex_pop80.cl3

it0, it, ci, magRMS, DMHsum, nDrop, nleft
7, 0, 1.000E+00, 1.897E-05, 1.721E-10, 1, 43
Region k, areak, area0k, Magk, Mrelerk

1,-2.46214E+06,-9.99000E+02, 4.05744E-04, 0.00E+00
2, 4.35024E+04, 4.35026E+04, 1.00000E+00, 4.89E-06
3, 1.04040E+04, 1.04035E+04, 9.99953E-01,-4.66E-05
4, 6.98692E+04, 6.98688E+04, 9.99994E-01,-5.60E-06
5, 1.77849E+04, 1.77853E+04, 1.00002E+00, 2.07E-05
6, 6.49116E+04, 6.49121E+04, 1.00001E+00, 7.51E-06
7, 5.71297E+03, 5.71287E+03, 9.99982E-01,-1.77E-05
8, 8.98984E+04, 8.98975E+04, 9.99990E-01,-1.00E-05
9, 1.31391E+04, 1.31389E+04, 9.99982E-01,-1.79E-05
10, 2.35993E+04, 2.35991E+04, 9.99989E-01,-1.09E-05
11, 3.52948E+04, 3.52946E+04, 9.99994E-01,-6.08E-06
12, 5.22065E+04, 5.22069E+04, 1.00001E+00, 6.20E-06
13, 1.08314E+05, 1.08316E+05, 1.00002E+00, 1.98E-05
14, 1.53027E+05, 1.53025E+05, 9.99991E-01,-8.52E-06
15, 8.01361E+04, 8.01353E+04, 9.99990E-01,-9.95E-06
16, 2.14703E+05, 2.14703E+05, 9.99999E-01,-8.94E-07
17, 2.11369E+04, 2.11370E+04, 1.00000E+00, 4.77E-06
18, 9.25379E+03, 9.25369E+03, 9.99990E-01,-9.95E-06
19, 7.18026E+04, 7.18035E+04, 1.00001E+00, 1.22E-05
20, 5.82621E+04, 5.82625E+04, 1.00001E+00, 6.79E-06
21, 3.74038E+04, 3.74030E+04, 9.99979E-01,-2.10E-05
22, 2.22070E+04, 2.22069E+04, 9.99997E-01,-2.92E-06
23, 4.47623E+04, 4.47609E+04, 9.99969E-01,-3.10E-05
24, 2.68492E+04, 2.68488E+04, 9.99986E-01,-1.38E-05
25, 9.10581E+03, 9.10637E+03, 1.00006E+00, 6.21E-05
26, 5.32842E+04, 5.32833E+04, 9.99983E-01,-1.73E-05
27, 3.63902E+04, 3.63906E+04, 1.00001E+00, 1.12E-05
28, 1.97906E+04, 1.97914E+04, 1.00004E+00, 3.76E-05
29, 3.20662E+04, 3.20665E+04, 1.00001E+00, 8.23E-06
30, 4.30753E+04, 4.30759E+04, 1.00001E+00, 1.37E-05
31, 3.76986E+04, 3.76974E+04, 9.99971E-01,-2.94E-05
32, 1.02432E+05, 1.02431E+05, 9.99994E-01,-5.66E-06
33, 8.28735E+04, 8.28741E+04, 1.00001E+00, 7.39E-06
34, 9.55756E+04, 9.55767E+04, 1.00001E+00, 1.11E-05
35, 1.05690E+05, 1.05689E+05, 9.99995E-01,-5.01E-06
36, 6.20195E+04, 6.20203E+04, 1.00001E+00, 1.35E-05
37, 8.45081E+04, 8.45097E+04, 1.00002E+00, 1.90E-05
38, 7.91677E+04, 7.91673E+04, 9.99996E-01,-4.47E-06
39, 9.86272E+04, 9.86277E+04, 1.00000E+00, 4.77E-06
40, 5.40054E+04, 5.40053E+04, 9.99997E-01,-2.68E-06
41, 5.90477E+03, 5.90495E+03, 1.00003E+00, 3.08E-05
42, 1.97841E+04, 1.97836E+04, 9.99976E-01,-2.40E-05
43, 6.81918E+04, 6.81908E+04, 9.99986E-01,-1.36E-05
44, 9.77732E+04, 9.77731E+04, 9.99999E-01,-1.25E-06

C.3.6 Output Files Not Included in this Appendix

The parameter option `idisk = 1` specified map output after every iteration. The files `RLInt.out.NNNN` and `RLInt.plot.NNNN` contain that information. Any of the files `RLInt.out.NNNN` can be copied to the file `RLInt.dat` and used as input to the program RLInt. When that is done, the mapping starts at the map that exists after iteration `NNNN`. These files also contain the translation vector that was used to map from the previous iteration to the iteration `NNNN`.

The `RLInt.plot.NNNN` files are useful for plotting the region maps after iteration step `NNNN`. They are comma delimited and contain the closure points that are needed to complete the region boundaries.

The files `RLInt.sum.NNNN` were requested by specifying `isum = 1` and `idisk = 1`. They contain information about all 44 regions after iteration `NNNN`. One such file, `RLInt.sum.0007`, is included in the previous section. The file `RLInt.sum` includes the same information for all iterations. Written after the mapping is complete, it shows the history of each region as the run proceeds. The file sizes shown below are for the 44-region Vermont pop80 map that was used here as an example. Other maps would produce files of different sizes.

Size (bytes)	File Name
39282	RLInt.out.0000
39282	RLInt.out.0001
39282	RLInt.out.0002
39282	RLInt.out.0003
39282	RLInt.out.0004
39282	RLInt.out.0005
39282	RLInt.out.0006
39282	RLInt.out.0007
39600	RLInt.plot.0000
39600	RLInt.plot.0001
39600	RLInt.plot.0002
39600	RLInt.plot.0003
39600	RLInt.plot.0004

Size (bytes)	File Name
39600	RLInt.plot.0005
39600	RLInt.plot.0006
39600	RLInt.plot.0007
3040	RLInt.sum.0000
3040	RLInt.sum.0001
3040	RLInt.sum.0002
3040	RLInt.sum.0003
3040	RLInt.sum.0004
3040	RLInt.sum.0005
3040	RLInt.sum.0006
3040	RLInt.sum.0007 *
28884	RLInt.sum

* This file is shown in Section C.3.5.

LAWRENCE BERKELEY LABORATORY
UNIVERSITY OF CALIFORNIA
TECHNICAL INFORMATION DEPARTMENT
BERKELEY, CALIFORNIA 94720

AAF349
LBL Libraries