Pergamon

0098-3004(95)00041-0

Computers & Geosciences Vol. 21, No. 9, pp. 1053–1064, 1995

0098-3004/95 $9.50 + 0.00

DEVELOPING A VARIABLE-SCALE MAP PROJECTION FOR URBAN AREAS

D. FAIRBAIRN and G. TAYLOR

Department of Surveying, University of Newcastle upon Tyne, Newcastle upon Tyne NE1 7RU, U.K.
(*e-mail*: dave.fairbairn@uk.ac.newcastle)

(Received 21 May 1994; accepted 23 October 1994)

Abstract—An algorithm for the development of a variable-scale map projection is presented and the code given in C. Such constructs have applications in urban areas where the central zones, which are detailed densely are to be displayed at a larger scale than the surrounding suburbs which are more extensive, but with sparser mappable features.

Key Words: Map projections, Urban mapping, Map scale, C.

INTRODUCTION

The cartographic representation of urban areas on many types of mapping—topographic, navigational, touristic—has been undertaken at larger scales than their surrounding landscapes. This is inevitable given the greater amount of mapped detail which exists in these areas.

However, this has meant that map overviews which cover differing landscapes (urban and rural, city center, and suburbs) on a single sheet have to compromise in terms of the scale of mapping appropriate to them. Thus, to include a city and its surrounding countryside on one view may well lead to unacceptably small-scale coverage of the central zones (resulting in an overly complex, difficult to interpret cartographic representation) or undesirable large-scale coverage in the rural surroundings (where the level of detail does not justify the large scale selected for the map). The traditional solution to such scale problems, and resultant map size and format difficulties, is to use one or more insets, which are displayed physically on the same sheet or view, but are perceived as being distinctly separate and of a different scale to the main map (American Cartographic Association, 1991).

VARIABLE-SCALE SOLUTIONS

Another solution is to construct a map with a variable scale, such that certain areas (for example, city centers) are represented at large scale, and the scale decreases away from these areas to other zones (for example, rural localities) where the scale is smaller. Such plans can be created using computer-assisted techniques, and the purpose of this paper is to demonstrate the construction of such a variable-scale map, applying the results to urban mapping in the city of Newcastle upon Tyne, in northeastern England and to borehole mapping in an area of Glasgow, in Scotland.

It should be noted that all flat two-dimensional maps of the curved three-dimensional Earth have a variable scale, the only representation showing true distance to scale being a globe. However, every map possesses a nominal scale, which should be quoted to the user, which holds for a given line or point and which, in the situation of large-scale maps and plans, can be regarded as adequate for dimensioning purposes over the whole sheet.

The first attempts to produce variable-scale map projections of the type described here were undertaken by the Swedish geographer Hagerstrand, whose well-known logarithmic azimuthal projection was applied to the study of migration patterns from the small Swedish village of Asby (Hagerstrand, 1957).

Snyder (1987) developed a series of "Magnifying Glass" projections which represent a central area at one particular scale, with the surrounding area at a smaller scale (Fig. 1). The change in scale here is abrupt (further exemplified by Fig. 2), but an effective enlargement of a specific area of interest is achieved. The scale change, of course, would become gradual as the number of bands of differing scales indicated in Figure 2 approach infinity.

The following algorithm outlines a technique for producing a variable-scale transformation of square-grid coordinates. The aim is to produce a variable-scale map, where scale changes in a linear fashion from a given scale at one point to a given scale at another point or series of points. The scale at any

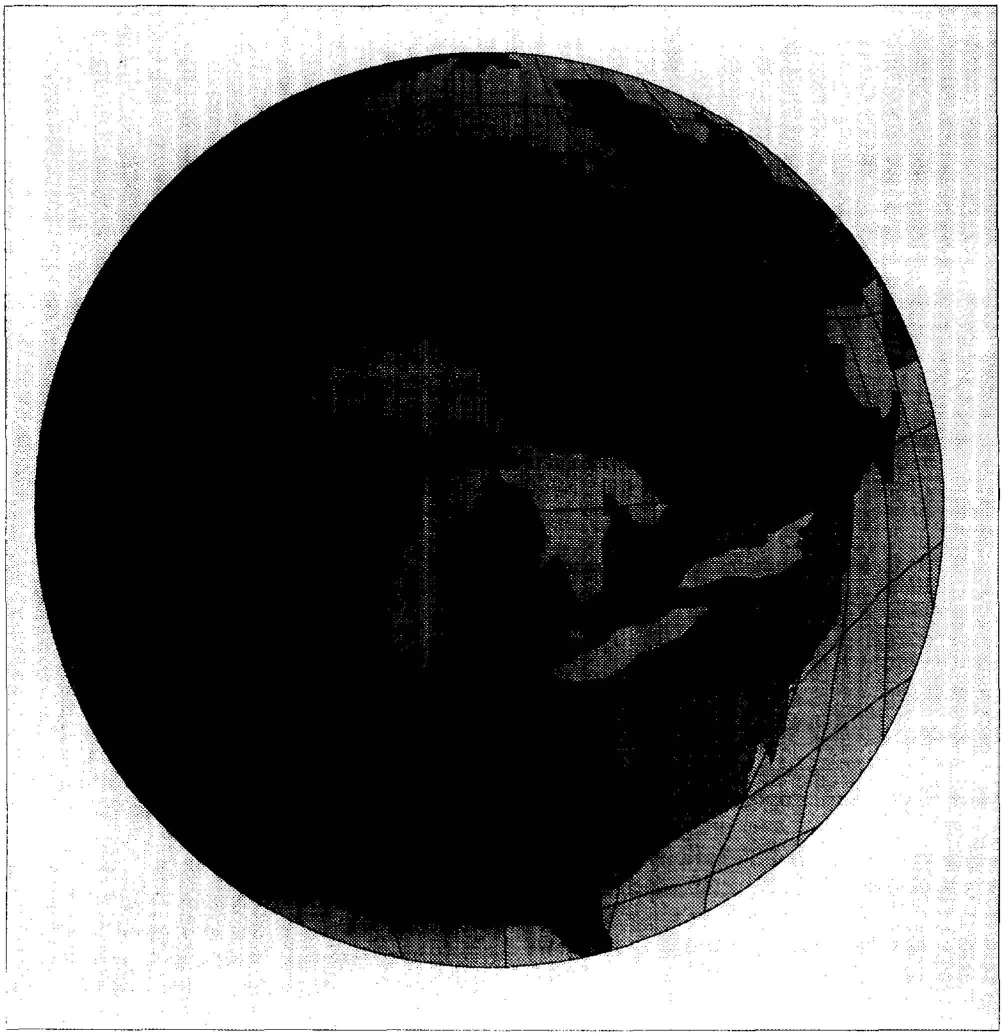

Figure 1. Example of "magnifying glass" projection (from American Cartographic Association, 1991).

point is proportional to the distance from the first given point. The following notation is used:

$SCALE1$ = scale at first point;
X_1, Y_1 = square-grid coordinates for first point;
$SCALE2$ = scale at second point;
X_2, Y_2 = square-grid coordinates for second point;
X, Y = square-grid coordinates for any point;
x, y = transformed grid coordinates.

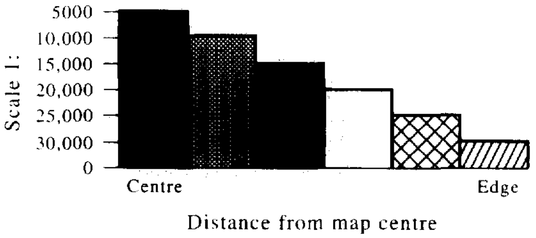

Figure 2. Histogram of scale against distance.

The scale applied to any grid coordinate is given by SC, where:

$$SC = SCALE1 + \frac{SCALE2 - SCALE1}{\sqrt{(X_2 - X_1)^2 + (Y_2 - Y_1)^2}} \cdot D \quad (1)$$

and

$$D = \sqrt{(X - X_1)^2 + (Y - Y_1)^2}. \quad (2)$$

Thus, the transformation to variable scale of any square-grid X coordinate is given by:

$$x = \frac{X}{SC} \quad (3)$$

and similarly,

$$y = \frac{Y}{SC}. \quad (4)$$

The result obtained by considering the limit situation of an infinite number of bands is identical to that obtained by employing the analytical technique outlined by Kadmon (1975). Unfortunately both the original article by Kadmon and a later description of the procedure by Maling (1992) suffer from shortcomings in explanation or errors of transcription.

This analytical approach to deriving an equation for the change of scale with distance from the center of the projection as suggested by Kadmon seems to be impossible, although numerical integration

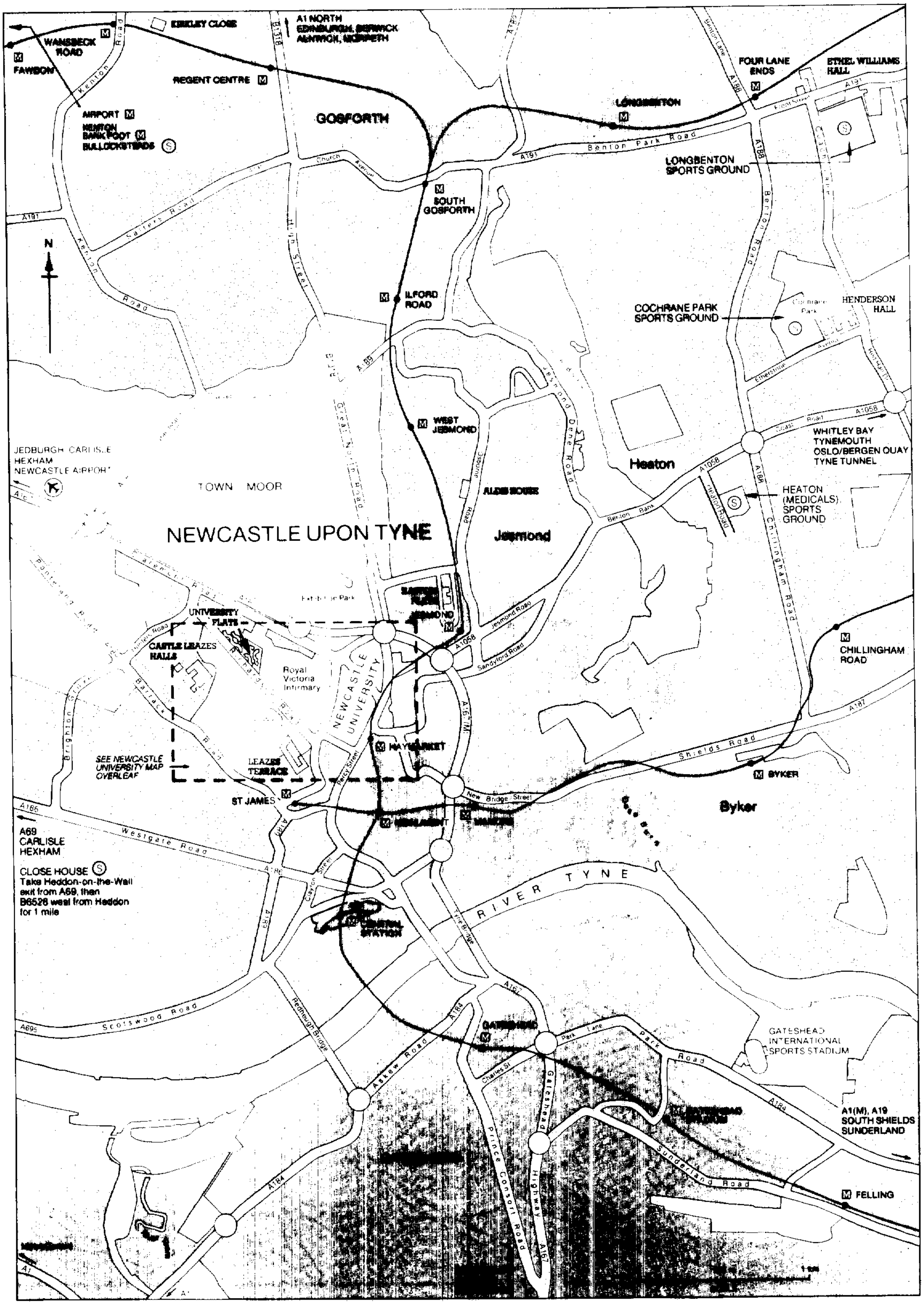

Figure 3. Newcastle and its environs map: before manipulation (scale approx 1:27,600).

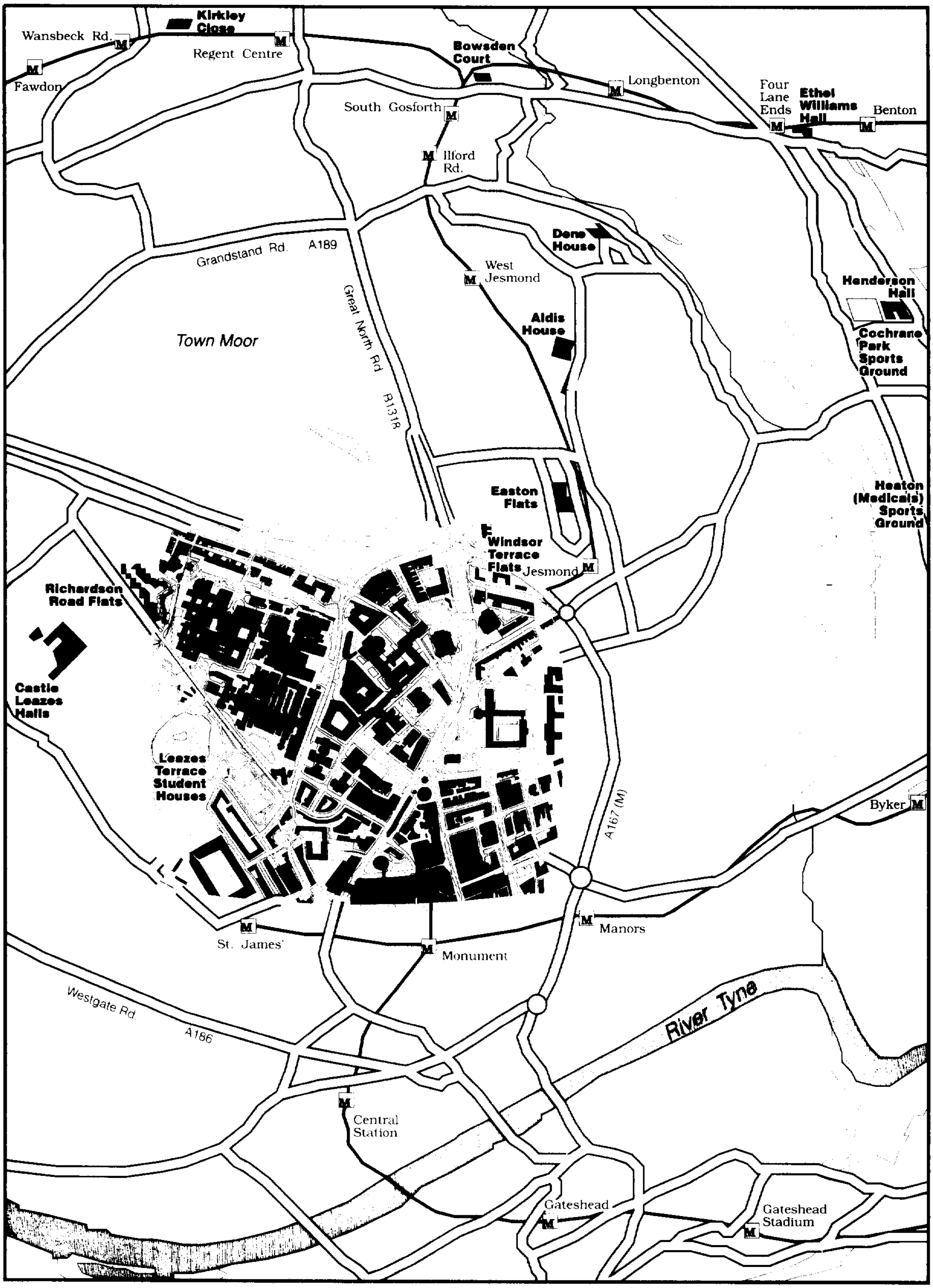

Figure 4. Newcastle and its environs map: after manipulation with scale variation and campus map included (scale 1:10,000 at center of projection, approximately 1:40,000 at corners).

techniques can provide a solution. Our approach gives the same visual result using a simple and elegant solution. This paper demonstrates the full procedures necessary and presents the encoded algorithm in the C language.

As Maling has indicated, this type of variable-scale mapping, termed the hyperbolic projection, was applied to Falk city plans in Europe in the late 1950s. Kadmon developed this approach in Israel, and went further (Kadmon and Shlomi, 1978) to construct polyfocal hyperbolic projections which have a number of enlarged foci, particularly useful for representing adjacent urban areas, or enlarging particular points of interest spread across the map face.

GENERAL URBAN MAPPING

The use of a hyperbolic projection for the city of Newcastle upon Tyne was prompted by the need for a map showing detail in the city center (including the central campus area of the University of Newcastle upon Tyne), but also extending to the outlying suburbs where student halls of residence and sports fields are located.

The original map (Fig. 3) at a scale of approximately 1:27,600 shows that, in order to extend to university facilities shown towards the border, the scale needs to be relatively small, resulting in a generalized representation of the central area. Figure 4 shows the same region with a variable scale.

A

Figure 5A. Caption overleaf.

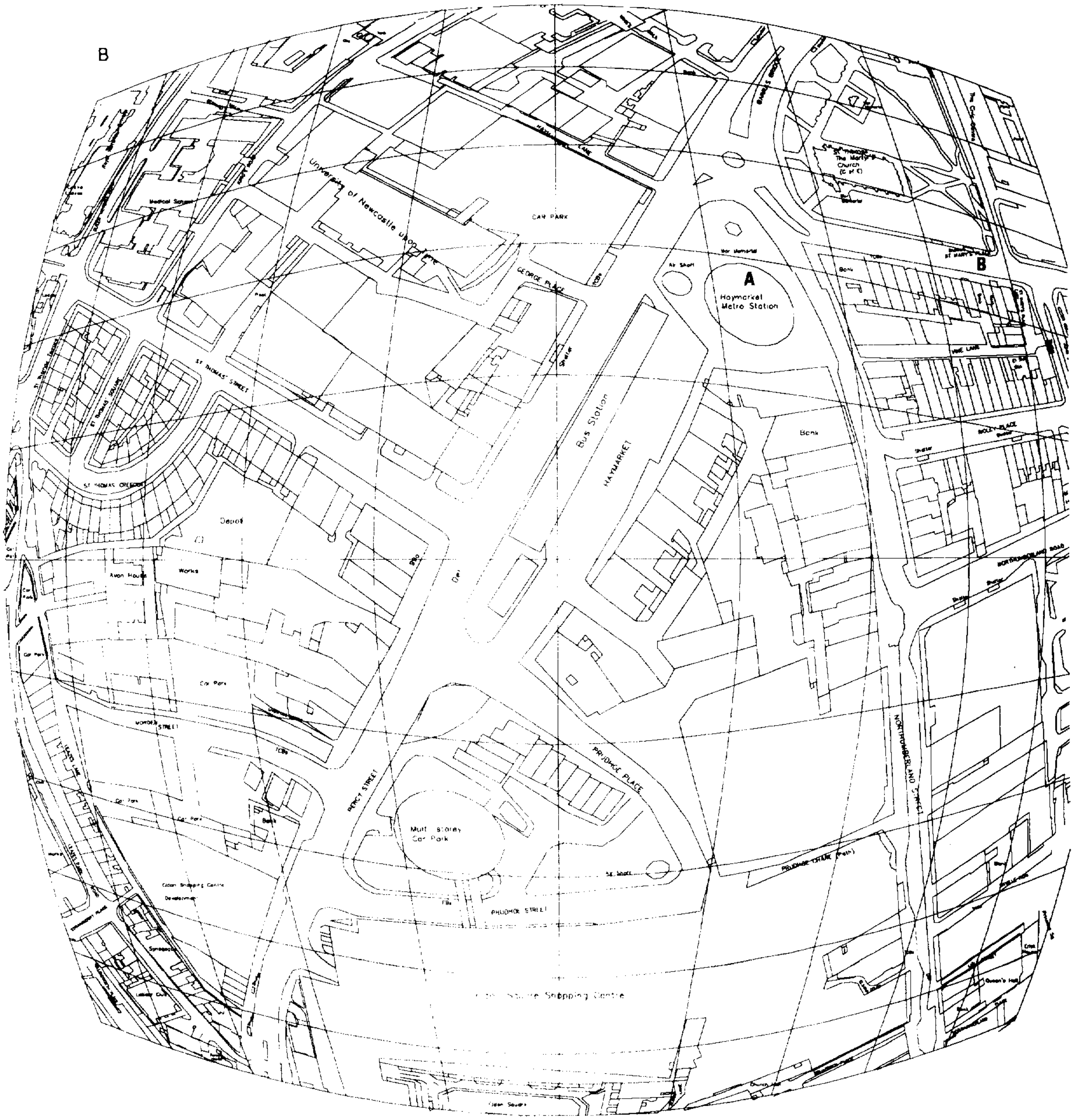

Fig. 5B

Figure 5. A, OS data NZ2464NE (grid interval 50 m, scale 1:3 000). B, OS data NZ2464NE with text problems and shape distortion, as shown by grid pattern after transformations (grid interval 50 m, scale 1:1 250 at center, 1:3 000 at corners).

The proposed solution, coded in an Appendix, allows for an enlarged scale to be selected at any point, followed by a prompt for the intended scale at the edge of the map. The resultant image then can be "squared-off" to give a pleasing layout for the new map, with or without the original grid superimposed. The optimal inclusion of the transformed grid indicates to the map user that the map scale had been distorted.

The distortions which are the result of the variable scale may lead to unusual or undesirable results. For example, consider the effects shown in Figure 5. Point A (Fig. 5B) shows the effect of the new projection on a circular building, made to appear elliptical. This morphological stretching is apparent for features with other regular geometrical properties, for example rectilinearity, parallelism, etc., where angles are not preserved and irregular shapes are produced.

This irregularity is noticeable particularly in digital vector-structured data-sets which reflect the urban landscape. With a predominance of straightline detail, the encoding of the man-made features in such data sets may comprise a skeletal framework of coordinated points. Thus, the four corners of a building or the two endpoints of a long straight road segment are sufficient to define their geometry. However, if variable scaling is applied to the data, there is a danger that long lines may cross, or that irregularities in shape become distracting. Thus, it may be necessary to add intermediate points to some features, or to increase the average density of data points to allow for pleasing renditions of varying scale. Such enhancement of detail can be done in a variety of ways, ranging from fractal enhancement to regular automatic densification of object coordinate pairs.

An example in Figure 5B (point B) indicates the effect of the projection on any encoded text strings. Text in this particular dataset [Ordnance Survey (OS) 1:1250 data] has an attributed angle of display (usually, but not always 90 degrees, that is eastwards on a north-oriented map) which is subject to change. Thus, a street name may start by following the direction of the street, but, by the point where the name has finished, the orientation of the street may have altered. Unless each letter is encoded separately, the text string cannot take this into account with the result shown at point **B**.

An additional effect of changing the scale is to change the density of the total portrayed detail on the map face. There is an obvious requirement for generalization of map detail as the scale of representation decreases and the "space available" to portray real-world features on the graphic diminishes. Because the scale of our mapping differs within the graphic, an imbalance of detail may result. A data set which originally had a higher density of detail in the center,

A

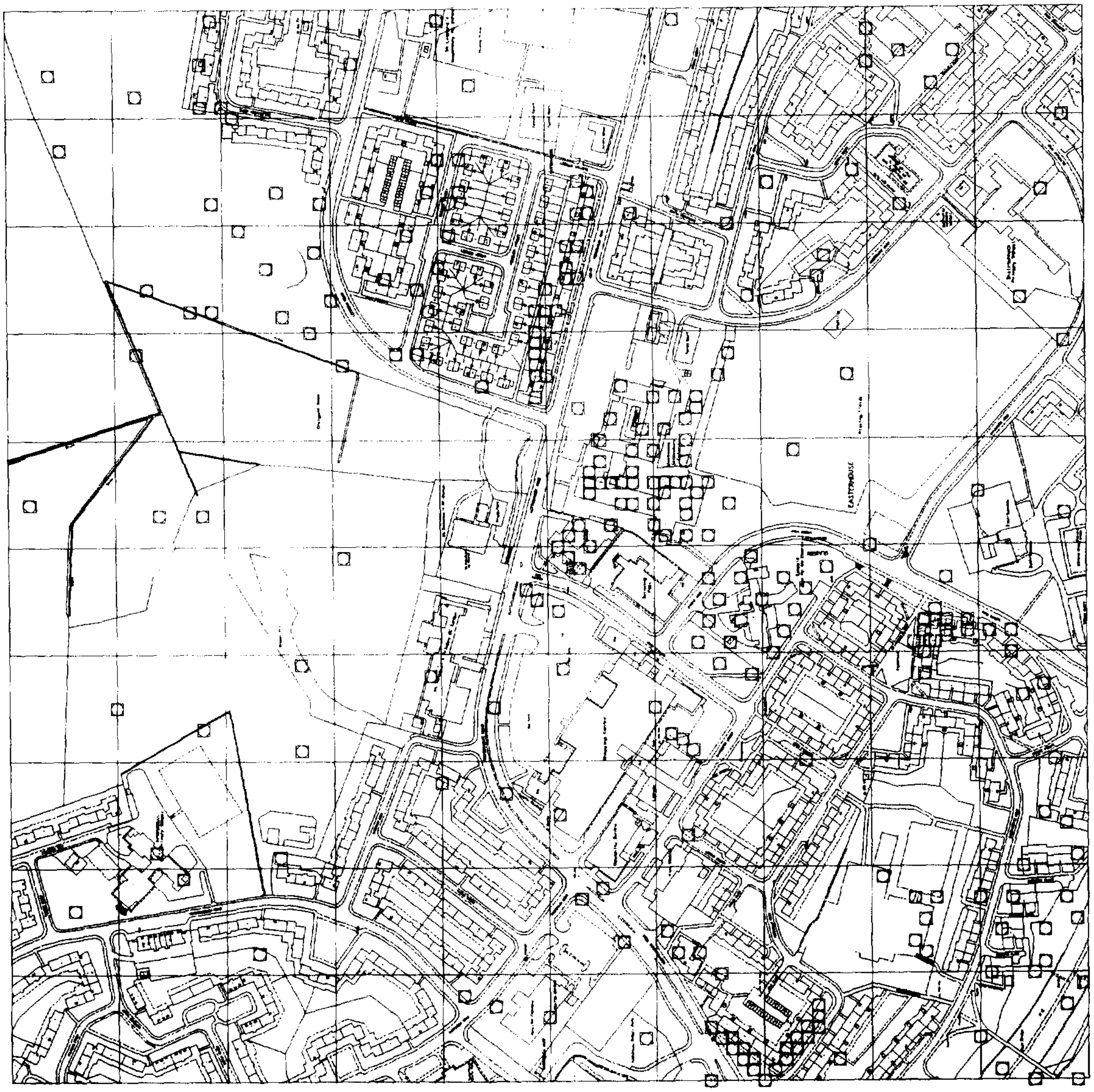

Figure 6A. Caption overleaf.

B

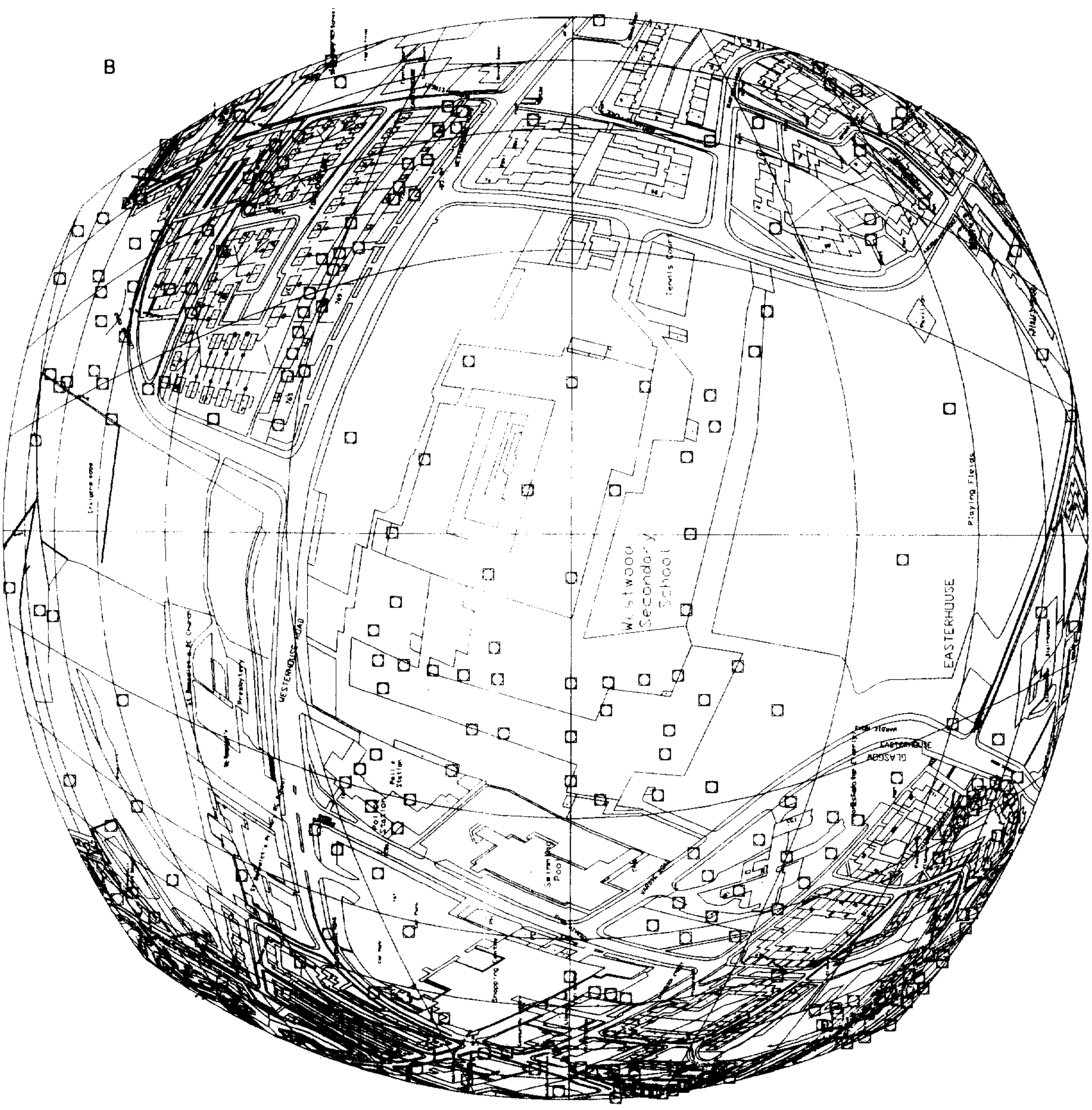

Fig. 6B

Figure 6. A, OS data NS6766SW, NS6766SE, NS6765NW, NS6765NE with BGS borehole data (grid interval 100 m, scale 1:6000). B, Transformed map of OS data from Figure 6A with BGS borehole data (grid interval 100 m, scale 1:1250 at the center of projection, 1:6000 at corners).

declining to sparser detail at the edges, may well reverse such characteristics after the application of a variable-scale projection. The enlarged space available in the center allows city center detail to "spread out", yet compresses the detail towards the edges. In certain circumstances this effect may be welcome, as it will imply to the map reader that the scale is decreasing away from the center.

To avoid this problem one solution is to use a more detailed data set for the central part of the map, and a less detailed representation at the edges. Thus, the distinction between the central zone and the suburbs is maintained. This solution has been adopted in Figure 4 where the central university campus was mapped and digitized at 1:6000, but the detail for the rest of the city center and the suburbs was taken from a map at 1:25,000. Unfortunately another complication is that, without further generalization, the point at which the data sets change is obvious and has led to discontinuities.

POINT MAPPING OF GEOLOGICAL DATA

The massive number of borehole records maintained by the British Geological Survey (BGS), and their unevenness in spatial distribution can lead to interpretation problems when displaying them, either on a video display or on a paper map. The example selected here is for borehole data in an urban area in the northeastern suburbs of Glasgow.

The conventional projection of the OS (Ordinance Survey) digital 1:1250 Landline data can be superimposed onto the borehole records extracted from the Oracle database maintained by BGS (Fig. 6). There are a number of concentrations of data points, where difficulties arise in attaching annotation to points, or where interpretation of the relationship between the ground features and the position of the boreholes is unclear.

The hyperbolic projection has been applied to the OS data and to the borehole records (Fig. 6B), giving a clearer indication of the distribution and density of borehole data within the grounds of Westwood School. The center of the new projection is slightly offset from the center of the original map.

USING THE NEW MAPS

The original application of such hyperbolic projections by Hagerstrand was in the context of mapping migration patterns in rural Sweden. Geographers have noted further diffusion patterns which can be mapped using these techniques. The spread of ideas, the progress of epidemics, the determination of "spheres of influence," and even the price of telephone calls exhibit such distance–decay properties.

Acknowledgments—Much of the preliminary investigative work and the digitizing of the data was undertaken by Nick Butler. The borehole data was supplied by Dr Jeremy Giles of the British Geological Survey, Keyworth. The digital data provided by the Ordnance Survey is Crown Copyright.

REFERENCES

American Cartographic Association, 1991, Matching the map projection to the need: ACSM, Bethesda, Maryland, 30 p.

Hagerstrand, T., 1957, Migration and area: migration in Sweden: Lund Studies in Geography, Series B, Human Geography, No. 13, p. 27–158.

Kadmon, N., 1975, Data-bank derived hyperbolic scale equitemporal town maps: Internationales Jahrbuch für Kartographie, v. 15, p. 47–54.

Kadmon, N., and Shlomi, E., 1978, A polyfocal projection for statistical surfaces: Cartographic Jour., v. 15, no. 1, p. 36–41.

Maling, D., 1992, Coordinate systems and map projections (2nd ed.): Pergamon Press, Oxford, 476 p.

Snyder, J. P., 1987, Magnifying glass azimuthal map projections: Am. Cartographer, v. 14, no. 1, p. 61–68.

APPENDIX

```
/* project.c (MICROSOFT C) */

/*
Read in a dxf file, change x,y and size values to variable scale projection
values. Write out new dxf file.

Scan for DXF GROUP CODES:
10-18  X COORD
20-28  Y COORD
40-49  SIZE VALUE (used for text size, symbol size)

Program expects 7 command line arguments:

dxfin     = full dxf file name e.g. name.dxf - string
scale1    = scale at centre of projection - real
XO        = x coordinate for scale1 (OS National Grid) - real
YO        = y coordinate for scale1 (OS National Grid) - real
scale2    = scale other point - real
Xscale2   = x coordinate for scale2 (OS National Grid) - real
Yscale2   = y coordinate for scale2 (OS National Grid) - real
*/

#include <stdio.h>
#include <stdlib.h>     /* for atoi, exit */
#include <math.h>
#include <string.h>

#define MAXINLINE 80  /* max input/output line length for DXF file*/
```

```
/*transformations */
double project_var(double *sc, double *xng, double *yng);

     double scale1, X0, Y0, scale2, Xscale2, Yscale2;
     double c, m;

_cdecl main(int argc, char *argv[]) {

     FILE *in_fp;                          /* input stream */
     FILE *out_fp;                         /* output stream */
     double xprojected, yprojected;        /* projected coords */
     double xng, yng;                      /* x,y relative to X0 and Y0 */
     double v_inline;                      /* integer value of input line */
     double sp, sng;                       /* transformed value, real size value */
     char dxfin[FILENAME_MAX];                       /* input filename */
     char dxfout[FILENAME_MAX]="project.dxf";        /* output filename */
     char inline[MAXINLINE], savex[MAXINLINE];       /* input line */

/* check no. of args */
     if (argc != 8) {
          printf ("\nProject - , read in dxf file, change to hyperbolic projection");
          printf ("\nand write out dxf file.  Requires 7 arguments");
          printf ("\nE.g. Project dxfin scale1 X0 Y0 scale2 Xscale2 Yscale2");
          printf ("\ndxfin  =  full dxf file name e.g. name.dxf - string");
          printf ("\nscale1  =  scale at centre of projection - real");
          printf ("\nX0  =  x coord for scale1 (OSNG) - real");
          printf ("\nY0  =  y coord for scale1 (OSNG) - real");
          printf ("\nscale2  =  scale other point - real");
          printf ("\nXscale2  =  x coord for scale2 (OSNG) - real");
          printf ("\nYscale2  =  y coord for scale2 (OSNG) - real");
     return 1;
     }

/* get args */
     strncpy(dxfin, argv[1], FILENAME_MAX);
     scale1      = atof(argv[2]);
     X0          = atof(argv[3]);
     Y0          = atof(argv[4]);
     scale2      = atof(argv[5]);
     Xscale2     = atof(argv[6]);
     Yscale2     = atof(argv[7]);

/* scale at any point X (or Y) grid coords in the projection is sc, where
sc  =  ( scale1  +  (scale2 - scale1) / m )  *  d
where m is the grid distance of X from X0, i.e.
m=sqrt((xscale2-X0)*(xscale2-X0)+(yscale2-Y0)*(yscale2-Y0))
and
d=sqrt((X-X0)*(X-X0)+(Y-Y0)*(Y-Y0))
*/

/* calculate constants */
     m  =  sqrt(((X0-Xscale2)*(X0-Xscale2))+((Y0-Yscale2)*(Y0-Yscale2)));
     c  =  (scale2-scale1)/m;
/* open files and set buffers */
```

```
    if ((in_fp = fopen(dxfin, "r")) = = NULL) {
         return 1;
         }
         setvbuf(in_fp, NULL, _IOFBF, 8192);

    if ((out_fp = fopen(dxfout, "w")) = = NULL) {
         printf("\nERROR opening output file %s", dxfout);
         fclose(in_fp);
         return 1;
         }
         setvbuf(out_fp, NULL, _IOFBF, 8192);

/* read input file line by line and check for coords/scale to transform */
    while (fgets(inline, MAXINLINE, in_fp) != NULL) {
         v_inline = atoi(inline);
         /* check for x coord */
         if ((v_inline > = 10) && (v_inline < = 18)) {
              fputs(inline,out_fp);          /* write " 10" etc. */
              fgets(inline, MAXINLINE, in_fp);
              strcpy(savex, inline);         /* save original */
              xng = atof(inline)-X0;
              fgets(inline, MAXINLINE, in_fp);
              v_inline = atoi(inline);
              /* check for y coord */
              if ((v_inline > = 20) && (v_inline < = 28)) {
                   strcpy(savex, inline);
                   fgets(inline, MAXINLINE, in_fp);
                   yng = atof(inline)-Y0;
                   xprojected = project_var(&xng, &xng, &yng);
                   yprojected = project_var(&yng, &xng, &yng);
                   fprintf(out_fp, "%.8f\n",xprojected);     /* write x coord */
                   fputs(savex, out_fp); /* write " 20" etc.  */
                   fprintf(out_fp, "%.8f\n",yprojected);     /* write y coord */
                   } else
                   /* no y to write out: SHOULD NEVER APPLY! */
    {
    fputs(savex, out_fp);     /* write original x coord? */
    fputs(inline,out_fp);     /* write last read line */
    }
    /* else check for a text/symbol size */
    } else if ((v_inline > = 40) && (v_inline < = 49)) {
         fputs(inline,out_fp);                  /* write " 40" etc. */
         fgets(inline, MAXINLINE, in_fp);   /* get size value */
         sng = atof(inline);
         /* use last x,y which must be insert coords */
         sp = project_var(&sng, &xng, &yng);
         fprintf(out_fp, "%.8f\n",sp);          /* write size */
         } else                                 /* write out */
         fputs(inline,out_fp);
    }
    fclose(in_fp);
    fclose(out_fp);
    return 0;
    }
```

```
/****************************************************/
/* process transformation of x,y, or size values return transformed value
sc = value to transform
xng = x relative to X0
yng = y relative to Y0
*/

double project_var(double *sc, double *xng, double *yng) {
double s;
      s = *sc/(scale1 + (c*sqrt((*xng**xng) + (*yng**yng))));
      return s;
      }
/* end project_var*/
/***************************************************/
```