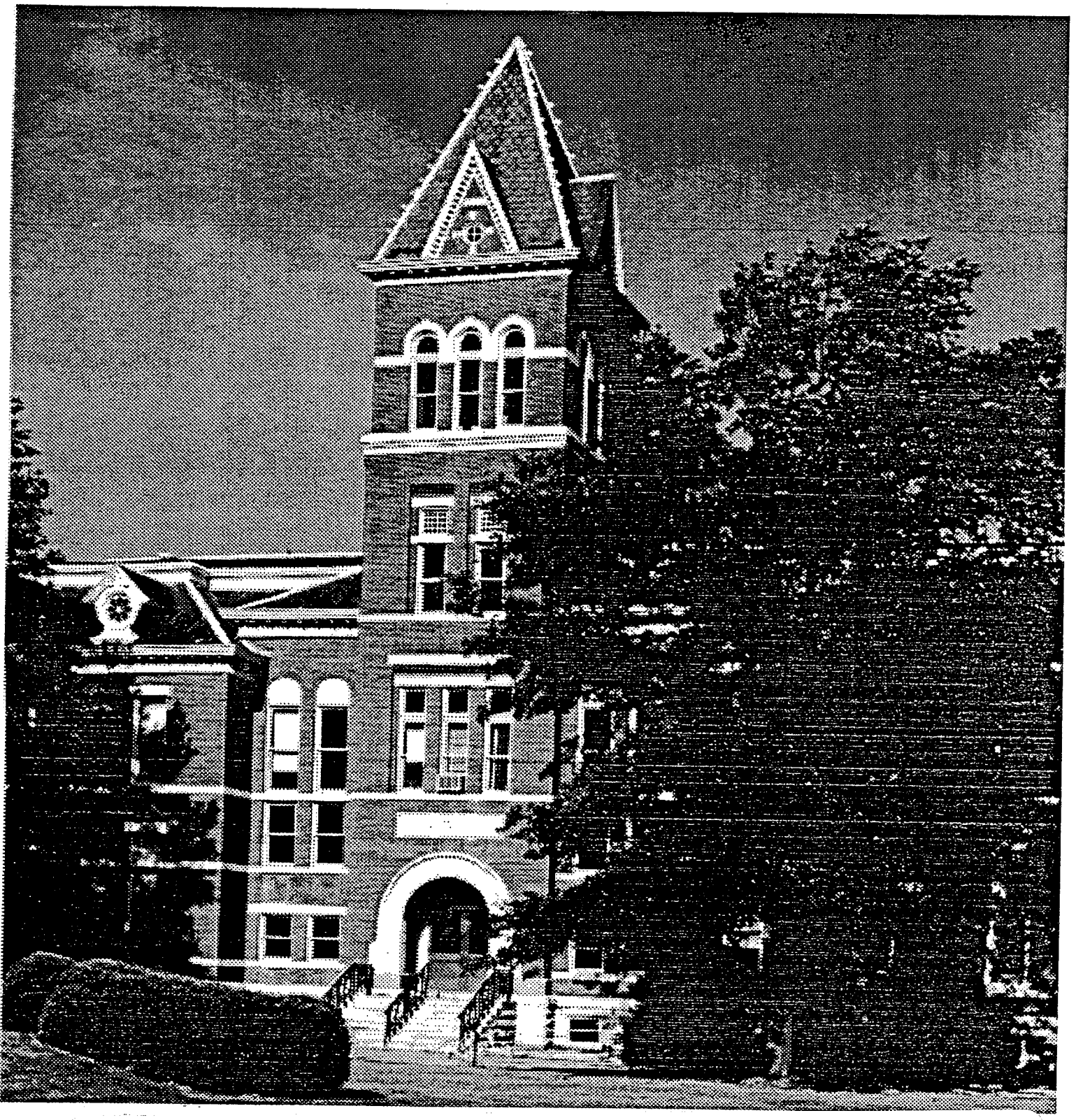
NAG1-875

University of Missouri–Columbia

College of Engineering



ENGINEERING SPECIFICATION AND SYSTEM DESIGN FOR CAD/CAM OF CUSTOM SHOES : UMC PROJECT EFFORT

NASA PROJECT NAG-1-875

FINAL REPORT COVERING PERIOD 6/1/1988 - 12/31/1990

SUBMITTED TO

NATIONAL AERONAUTICS & SPACE ADMINISTRATION

LANGLEY RESEARCH CENTER

HAMPTON, VIRGINIA

BY

HAN P. BAO, PH.D., P.E.

UNIVERSITY OF MISSOURI-COLUMBIA

COLUMBIA, MISSOURI 65211

Technical Report UMC-IE-6-0391

March 1991

SUMMARY

This is the final report for project NASA NAG-1-875 covering the period from June 1988 to December 1990. The goal of this project is to supplement the footwear design system of North Carolina State University (NASA project NAG-1-696) with a software module to design and manufacture a combination sole.

Befitting its nature, the report contains summaries of the work performed in the previous phases, as well as the latest work done between January 1990 and December 1990. The detailed descriptions of the early phases can be found in the following publications :

Technical Report UMC-IE-3-0189 for period 6/88 - 12/88

Technical Report UMC-IE-4-0889 for period 1/89 - 6/89

Technical Report UMC-IE-5-0590 for period 8/89 - 12/89

Work in period 6/88 - 12/88 : Customization of NASCAD to the footwear project; Use of the Cencit data ; Computer-aided manufacturing activities; and Beginning work for bottom elements of shoes.

Work in period 1/89 - 6/89 : Machining of shoe lasts - point-to-point configuration, and patch configuration; and Design and Production of integrated sole.

Work in period 8/89 - 12/89 : Further experimentations to improve the design and fabrication techniques of the integrated sole.

In the period from January 1990 to December 1990, the work done was primarily to improve the machining process of the shoe last using a spiral tool path approach. It was shown that this method not only could cut down the machining time but also could improve the surface quality of the shoe last. Furthermore experimentations with different stock materials such as soft foam indicated remarkable improvement in productivity and quality.

In conclusion, the work assigned to the University of Missouri was accomplished satisfactorily. The software written in C language and concerning the design and manufacturing of the integrated sole was delivered to North Carolina State University some time ago, but it remains to be integrated with the LASTMOD software at NCSU before becoming fully operational. Although not responsible for the machining of the shoe last, we at UMC have continued to work to improve the machining of this item, and have proved conclusively that the novel approach based on a spiral tool path could indeed reduce machining time as well as improving quality.

2- REVIEW OF WORK IN PERIOD 6/88 - 12/88

Full documentation of the work completed in this period is contained in Technical Report UMC-IE-3-0189. The four areas of concentration were :

- Customization of NASCAD to the footwear project
- Use of the CENCIT data
- Computer-Aided Manufacturing Activities
- Beginning Work for Bottom Elements of Shoes

Customization of NASCAD

NASCAD (NASA Computer Aided Design) is an interactive graphics program which supports the construction and display of two or three dimensional geometric structures and text.

In this project phase, methods were sought to enter a foot geometric data file, say from CENCIT or from CyBERWARE, into NASCAD then perform simple shape changes for converting the foot model into a shoe last model. Note that this was not an attempt to duplicate the work of Dr. McAllister at NCSU, but simply an interest in finding out how NASCAD would handle such tasks.

The report includes 2 parts, respectively to review all available NASCAD commands and publish a simplified version of the NASCAD user's manual, and to develop macros (all together 46 of them) to aid in the graphics manipulation of the foot image.

Use of CENCIT data

A foot data file was obtained from CENCIT Corp., St Louis, and distributed to all PI's of the project. The original data was in 16-bit integer format in reverse order bytes. Hence the first conversion work was to swap the order of the bytes (most significant byte with least significant byte in a 2-byte integer). Dr. Rasdorf's group has produced a Pascal program to do just this (see Technical Report " Foot Data Preparation Summary ", Department of Civil Engineering and Computer Science, NCSU, July 1988). The UMC group also did come up with their own conversion program but using a different approach.

Once the data was in the appropriate format, statistics were collected to find out how many good data points exist for each of the three views provided by CENCIT, and their distribution patterns in space. This work was needed as a pre-requisite for using the data for machining purposes.

The report provided detail of the conversion program developed at UMC, and the various coordinate transformations of these data points as required by the machining process.

Computer Aided Manufacturing Activities

These activities included : the making of molds for casting cylindrical plaster blocks and rectangular plaster blocks, the fixture arrangement, the data processing, the selection of machine tool parameters, the selection of cutting tools, and actual machining experimentations.

Data processing is a critical step before machining. The CENCIT data is essentially cartesian data. In a first series of machining tests, it was used directly to machine a shoe last. Of course this was done after discarding the poorly defined or missing points rom the CENCIT data. As expected, the resulting surface was excellent in the middle of each portion, but detoriorated markedly close to the edges.

In a second series of machining tests, the cartesian data was first converted to cylindrical data and the block was machined while held in a rotary device. Surface quality improved considerably because the cuttinf condition was now a purely radial cut. The problem of tool interference was dealt with efficiently by the incorporation of a tool offset as used by Saunders (ref 1) into the cutter location data file. The report explained in detail any software development, jig and fixture construction, choice of cutting parameters, choice of tool geometry, and the results form the machining experiments.

Beginning work for bottom elements of shoes

Previously it was decided that the prime contribution of UMC to this footwear project will be in the design and production of the bottom

elements of the shoes. These elements include insole, outsole, or combination insole/outsole, heels, foot aids, orthotics and many other devices.

To gain a good understanding of this aspect of shoe making process, a series of industry visits, professional society meetings and personal contacts were made in this project phase. While significant work in this area is expected in the coming phase, phase V, already the UMC has identified means of extracting the goemetry of the plantar surface from the foot model then generate NC codes for machining molds for casting the soles using liquid last or neoprene latex.

In summary, the work for this phase was completed on schedule. Good progress was made in the fourth objective which was related to the bottom elements of the shoes. This paved the way to the proposed work in phase V as submitted in our proposal to NASA.

3- REVIEW OF WORK IN PERIOD 1/89 - 6/89

A detailed explanation of this work is provided in Technical Report UMC-IE-4-0889 submitted to NASA in August 1989. The work involved in the previous phase, Phase IV from June 1988 to December 1988, was primarily of a theoretical nature paving the way to actual machining and casting experiments reported in the document quoted above.

For this period, the following areas were involved :

- Machining of shoe lasts - point-to-point configuration
- Machining of shoe lasts - patch configuration
- Design and production of integrated sole

Machining of shoe lasts - point-to-point configuration

The solid object for machining is represented by a wireframe model with its nodes or vertices specified systematically in a grid pattern covering its entire length.

Two sets of data, respectively from CENCIT and CYBERWARE, were used for machining purpose. The machining process itself has been experimented with, using a variety of approaches as suggested in the theoretical work carried out in the previous phase.

It has been found that the indexing technique, that is turning the stock by a small angle then move the tool on a longitudinal path along the foot, yields the best result in terms of ease of programming, saving in wear and tear of machine and cutting tools, and resolution of fine surface details.

Machining of shoe lasts - patch configuration

The work of Dr. McAllister and his group at NCSU through the LASTMOD last design system results in a shoe last specified by a number of congruent surface patches of different sizes. This data format must therefore be adopted to carry out the downstream operation of last machining.

In this report, a means of converting this data into a form amenable to the machine tool is provided. Essentially it involves a series of sorting algorithms and interpolation algorithms to provide the grid pattern that the machine tool needs as was the case in a point-to-point configuration discussed previously.

The resulting machined foot agrees quite well with the one obtained by Dr. Sanii of NCSU although both of us were surprised with the actual length of the object, it being about seven inches long compared to nine inches of expected length.

Design and Production of integrated sole

Although the design and manufacture of a shoe last is the single most important element in shoe making, many other activities also play an important role in the entire spectrum of footwear production. Examples of these activities include the prescription of different types of wedges, the molding of the inner, middle, and outer soles, the use of inserts for arch, heel, or metatarsal support, the use of rigid to soft orthotics and, in general, the use of extra-depth shoes.

This report contains an in-depth treatment of the design and production technique of an integrated sole to complement the task of design and manufacture of the shoe last. Clinical data and essential production parameters are also discussed. Examples of soles made through this process are given as illustrations.

Because the grant was given to UMC very late, only a month before the official end of the phase period, work on orthotic devices such as pads, wedges, and inserts could not be made in time for reporting in this document. Nevertheless the work done in this phase V of the project reported here provides valuable practical solutions to the theoretical propositions laid down in the previous phase. They facilitate the proposed next phase work for the rest of this year.

4- REVIEW OF WORK IN PERIOD 6/89 - 12/89

Documentation for this work can be found in Technical Report UMC-IE-5-0590 submittedd to NASA in MAy 1990.

In this reporting period, further experimentations were made to improve the design and fabrication techniques of the integrated sole. The sole design was shown to be related to the foot position requirements and the actual shape of the foot including presence of neurotropic ulcers or other infections. Factors for consideration were : heel pitch, balance line, and rigidity conditions of the foot. Machining considerations were also part of the design problem. Among these considerations, widths of each contour, tool motion, tool feed rate, depths of cuts, and slopes of cut at the boundary were the key elements.

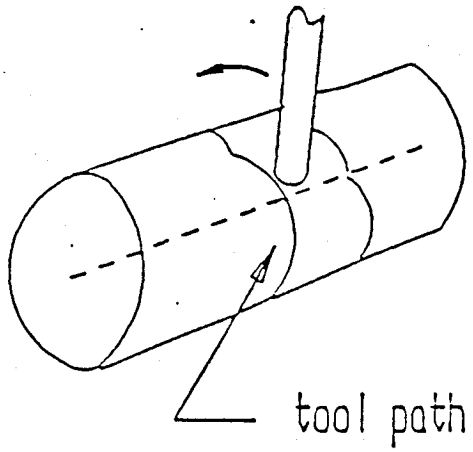
The essential fabrication techniques evolved around the ides of machining a mold then, using quick-form latex material, casting the sole through the mold. Two main mold materials were experimented with : plaster and wood. Plaster was very easy to machine and shape but could barely support the pressure in the hydraulic press required by the casting process. Wood was found to be quite effective in terms of relative cost, strength, and surface smoothness except for the problem of cutting against the fibers which could generate ragged surfaces.

This report also discussed the programming efforts to convert our original Dbase programs into C programs so that they could be executed on the SUN computer at North Carolina State University.

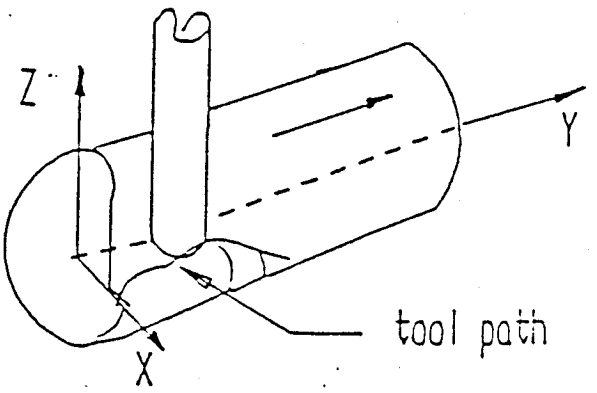
5- SPIRAL MACHINING

5.1 CURRENT TECHNOLOGIES

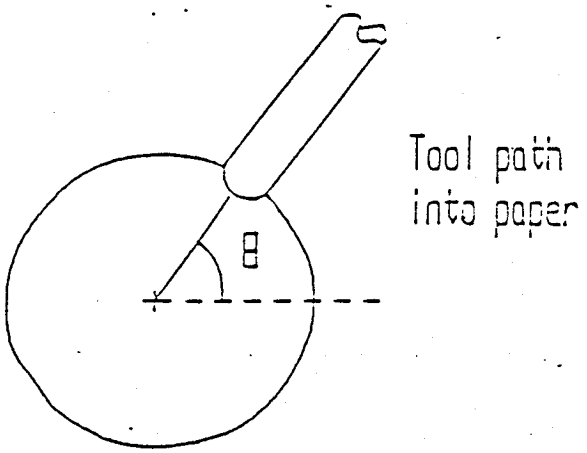
The current techques of machining the shoe lasts have been extensively reviewed in (ref 2). Basically there are four such techniques known respectively as contour machining, cartesian machining, constant theta per pass machining, and variable theta per pass machining. Figure 1 shows the features of these four techniques.



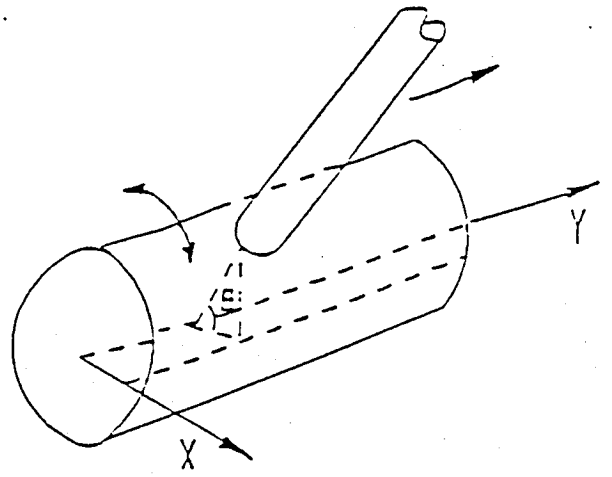
1 - CONTOUR MACHINING



2 - CARTESIAN MACHINING



3 - CONSTANT THETA PER PASS



4 - VARIABLE THETA PER PASS

Figure 1

The following results have been obtained :

1- Contour machining and Constant theta per pass are the easiest approaches to program but, in the case of contour machining, due to rotational limitation of the rotary table, the tool must be programmed to go clockwise in one contour then counter-clockwise in the next contour. Due to constant changes if direction, the rotary table may be subjected to excessive wear and tear. For this reason, a decision was made not to pursue further this method of machining.

2- For the same reason for the contour machining approach, the variable theta per pass approach was also discarded not only because of excessive wear and tear of the rotary table but also because of the complexity in calculating the proper delta angle in both directions for the rotary table.

3- The cartesian machining approach is potentially the least cost technique since it does not even require the use of the rotary table. The major problem for this approach is the increase in depth of the ridges as the tool moves further from its central position atop of the block.

4- Of all the investigated machining techniques, the constant theta per pass, otherwise known as index machining, offers the best solution in terms of ease of data preparation, practicality, and lower wear and tear of the machine tool. Therefore it was recommended as the preferred technique for making shoe lasts.

Initially it was thought that the rotary table could not be programmed to revolve continuously in one direction. This limitation led to the series of machining experiments which resulted in the conclusions made above. One noted drawback of the constant theta per pass approach is the required reversal of the tool movement at the end of each pass. Furthermore the workpiece must also be indexed at or during the time the tool reverses its direction of motion. A smoother machine cut would result if the work can rotate constinuously while the tool moves from one end of the work to the other end. This motion is essentially what happens to a turned part in a lathe. Another way of describing this motion is by imagining a tool always perpendicular to the work and tracing its contour in a spiral path. The technique for obtaining the coordinates of the points along this spiral path is described in the section below.

5.2 Spiral machining

The rectangular grid shown in figure 2 below can be thought of as being the surface of the cylindrical part after being cut open and laid out flat. The path for the tool machining along A_1-H_1 followed by H_2-A_2, etc..., is what the third current machining approach is all about. Machining along A_1-A_5 followed by B_1-B_5 etc... would be the contour machining. Spiral machining involves moving the tool along A_1-B_5 after a single revolution of the work, then repeat the same motion for B_1-C_5, and so on. Here an interpolation process is needed to determine the

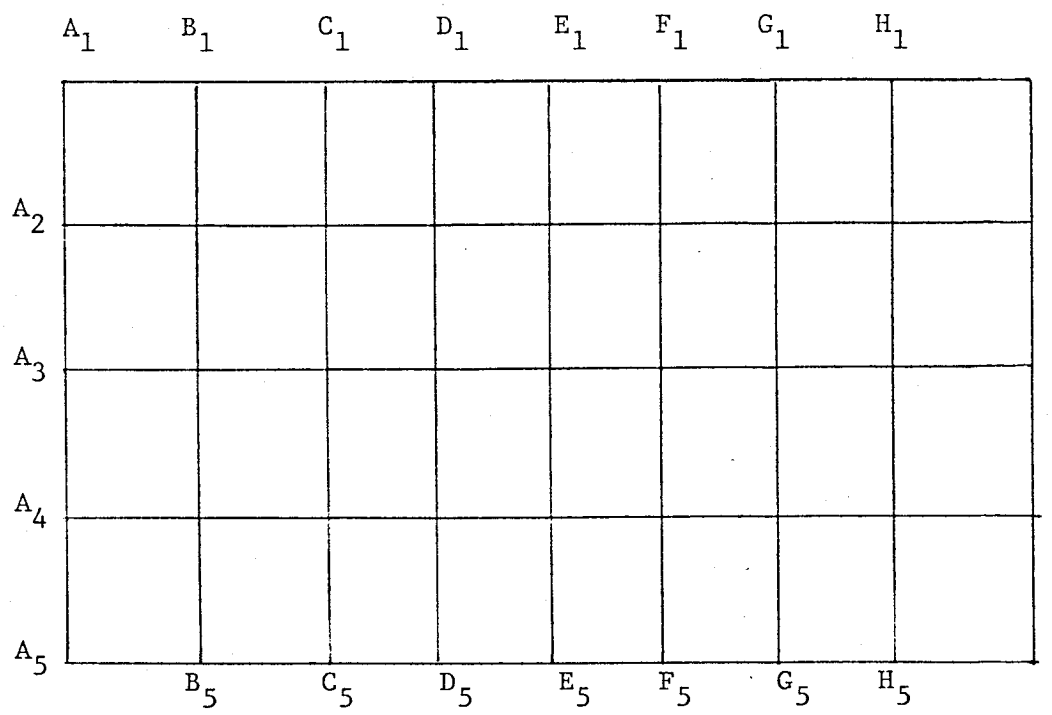


Figure 2

height locations of the tool at the intermediate points between the extreme points. There are two technical issues to be resolved in conjunction with the determination of the tool path. First, the type of interpolation technique must be selected, and second, the tool interference problem must be addressed. These two issues are discussed in the section below.

Interpolation technique

A variety of interpolation techniques exist, such as linear interpolation, quadratic interpolation, Lagrange's method, Hermite's method, and Newton's approach to name a few. After consideration of relative accuracy, stability, and ease of programming, the Newton's

approach was adopted. The particular form of the Newton's approach applied to our problem was known as divided differences which possesses the following recursive property :

$$f[x_1,x_2\cdots x_n] = \frac{f[x_2,x_3\cdots x_n]_n - f[x_1,x_1,\cdots x_{n-1}]_{n-1}}{x_n - x_1}$$

where $f[x_1,x_2,\cdots x_n]$ are known as the divided differences of f.

When a divided difference diagram is formed then the entries at the top of each column become the respective coefficients in the Newton's polynomial expression. Reference to the Newton's approach can be found in many textbooks, for example (ref 3).

Tool interference

Tool interference occurs when the profile of the tool overlaps with the intended surface as illustrated in figure 3 . In order to avoid interference, the tool could be raised by an offset value "x" as shown in figure 3.b . The problem of determining the correct location of the tool tip is actually resolved by starting at point A (a known point on the surface), constructing a vector R n perpendicular to the surface at A, then adding another vector R u along the axis of the tool. This results in a new tool tip position vector given by :

$$r = r_A + R\,n - R\,u$$

where R is the radius of the tool and n and u are respectively the unit

vector perpendicular to the surface at A, and the unit vector along the tool axis.

In the case of radius cut (tool pointing toward center of rotation), the tool offset distance x can be obtained from Saunders's paper (1) and reproduced below :

$$x = \frac{R\left[(r_i)^2 + (r_{i+1})^2 - 2r_i \cdot r_{i+1} \cos(\quad) \right]^{1/2}}{r_{i+1} \cdot (\sin \quad)} - R$$

where r_{i-1}, r_i, r_{i+1} are radial distances of consecutive points on the contour, and is the sustained angle between two consecutive points. Appendix 2 contains the programs for determining the NC codes for driving the milling machine to machine the shoe last. Note that two passes were used in our experiments : the first one was a rough cut involving radiuses arbitrarily set at corresponding true radiuses plus a constant offset of 1/2 inch and performed at a high tool feed rate, and the final pass at the correct radiuses but at a much lower feed rate for creating a smooth surface.

Conclusion

The task of generating a software module for producing a sole given to the University of Missouri has been completed with demonstrated product realization. The software written in C has been delivered to North Carolina State University for inclusion in their design system for

custom footwear known as the LASTMOD system.

Our experience with this project indicates that the proposed technique for casting the sole is practical and precise enough to be seriously considered in the next phase of mass production. The strength of the sole has been shown to depend on the allowable volume of expansion of the liquid foam, that is the lower the volume of expansion, the denser the foam. Thus volume of expansion should be a design parameter which the user could specify to select the required strength or hardness of the sole.

Our machining experiments with the shoe lasts have shown that using a foam material and some strategic approaches such as spiral machining and 2-pass machining (rough cut followed by finished cut) could reduce the time of cut considerably and may be the only practical way of making the shoe lasts in a cost efficient manner.

Our final remark is on the need for having a design system similar to LASTMOD but on a microcomputer platform rather than the SUN computer and its expensive TAAC graphic board. Using an IBM PS/2 microcomputer and the HOOPS software we have demonstrated that this was possible although much work remains to be done to make the system run faster and also have the necessary design features that a practitioner in custom footwear would want to see in the system. It is our intention to pursue this line of work in the future as an extension of the work done in this project.

APPENDIX 1 : REFERENCES

1- Saunders C.G. and Vickers C.G. " A General Approach to Replication of Cylindrical Bodies of Compound Curvatures." Transactions of ASME, Journal of Mechanisms, Transmissions and Automation in Design, Vol. 106, pp.70-76, 1984.

2- Bao H.P., " Engineering Specification and System Design for CAD/CAM of Custom Shoes." , NASA Project NAG-1-875, Technical Report UMC-IE-4-0889,August 1989.

3- Cheney W. and Kincaid D., "Numerical Mathematics and Computing.", Brooks/Cole Publication, Second Edition, 1985, pp.108-118.

APPENDIX II

```
/*****************************************************************/
/*                                                               */
/*                                                               */
/*                                                               */
/*                          LASTMAKE.C                           */
/*                                                               */
/*****************************************************************/


#include <stdio.h>
#include <math.h>
    float   x[40][40], r[40][40], eval[40][40], t, c[31];
    float   z[40][40], z1[40][40], z2[40][40], y[40][40];
    float   yax[40][40], zax[40][40];


main ()
{
    int     i,d,t1,t2,u1,u2,m,k,n,l,s,b;
    FILE    *file1,*file2;
    float   junk;

    junk = 2.0;
    file1 = fopen("cylfoot2.dat","r");
    file2 = fopen("last_1.nc","w");

    for (i=1;i<20;i++)
        {
        for (d=1;d<37;d++)
            {
            fscanf (file1,"%f %f   %f", &x[i][d], &yax[i][d],
            &zax[i][d]);
            r[i][d] = sqrt(pow(yax[i][d],2)+pow(zax[i][d],2));
            }
        }
    fclose (file1);
/*****************************************************************/

/*  Interpolation to determine points at 1/4th inch              */

/*****************************************************************/
for (d=1;d<37;d++)
    {
    s = 0;
    for (t=1.25;t<10.5;t=t+0.5)
```

```
{
s=s+2;
if (t==1.25)
     {
     i=1;
     for (k=1;k<=3;k++)
          {
          c[k] = r[i][d];
          i = i+2;
          }
     for (l=1;l<=2;l++)
          {
          for (m=3;m>=l+1;m--)
               {
               c[m] = (c[m]-c[m-1])/(x[2*m-1][d]-
               x[2*(m-l)-1][d]);
               }
          }
     eval[2][d] = c[3];
     for (m=2;m>=1;m--)
          {
          eval[2][d] = eval[2][d]*(1.25-x[2*m-1][d])
          + c[m];
          }
     r[2][d] = eval[2][d];
     x[2][d] = t;
     }
else if (t==10.25)
     {
     for (k=1;k<=3;k++)
          {
          c[k] = r[35+2*(m-1)][d];
          }
     for (l=1;l<=2;l++)
          {
          for (m=3;m>=l+1;m--)
               {
               c[m] = (c[m]-c[m-1])/(x[35+2*(m-1)][d]-
               x[35+2*(m-l-1)][d]);
               }
          }
     eval[38][d] = c[3];
     for (m=2;m>=1;m--)
          {
          eval[38][d] = eval[38][d]*(10.25-x[41-2*m-1][d])
          + c[m];
          }
     r[38][d] = eval[38][d];
     x[38][d] = t;
     }
else if (t!=1.25 && t!= 10.25)
     {
     for (k=1;k<=4;k++)
```

```
                {
                c[k] = r[s-5+2*k][d];
                }
            for (l=1;l<=3;l++)
                {
                for (m=4;m>=1;m--)
                    {
                    c[m]=(c[m]-c[m-1])/(x[s-5+2*m][d]-
                    x[s-5+2*(m-1)][d]);
                    }
                }
                eval[s][d]=c[4];
                for (m=3;m>=1;m--)
                    {
                    eval[s][d]=eval[s][d]*(t-x[s-5+2*m][d])+c[m];
                    }
                r[s][d]=eval[s][d];
                x[s][d]=t;
            }
        }
    }
/*************************************************************
*                                                            *
*                        Determining the tool path           *
*                                                            *
*************************************************************/
for (i=1;i<=30;i++)
    {
    c[i]=0;
    }
for (d=2;d<=36;d++)
    {
    for (s=1;s<=38;s++)
        {
        if (s==1)
            {
            y[1][d]=x[1][d]+(0.25/36)*(d-1);
            i=1;
            for (k=1;k<=3;k++)
                {
                c[k]=r[i][d];
                i++;
                }
            for (l=1;l<=2;l++)
                {
                for (m=3;m>=l+1;m--)
                    {
                    c[m]=(c[m]-c[m-1])/(x[m][d]-x[m-l][d]);
                    }
                }
            eval[1][d]=c[3];
            for (m=2;m>=1;m--)
                {
```

```
                eval[1][d]=eval[1][d]*(y[1][d]-x[m][d])+c[m];
                }
            z[1][d]=eval[1][d];
            }
        else if (s==38)
            {
            y[38][d]=x[38][d]+(0.25/36)*(d-1);
            for (k=1;k<=3;k++)
                {
                c[k]=r[36+k][d];
                }
            for (l=1;l<=2;l++)
                {
                for (m=3;m>=l+1;m--)
                    {
                    c[m]=(c[m]-c[m-1])/(x[36+m][d]-x[36+m-1][d]);
                    }
                }
            eval[38][d]=c[3];
            for (m=2;m>=1;m--)
                {
                eval[38][d]=eval[38][d]*(y[38][d]-x[36+m][d])
               +c[m];
                }
            z[38][d]=eval[38][d];
            }
        else if ((s!=1) || (s!=38))
            {
            y[s][d]=x[s][d]+(0.25/36)*(d-1);
            for (k=1;k<=4;k++)
                {
                c[k]=r[s+k-2][d];
                }
            for (l=1;l<=3;l++)
                {
                for (m=4;m>=l+1;m--)
                    {
                    c[m]=(c[m]-c[m-1])/(x[s+m-2][d]-
                    x[s+m-2-1][d]);
                    }
                }
            eval[s][d]=c[4];
            for (m=3;m>=1;m--)
                {
                eval[s][d]=eval[s][d]*(y[s][d]-x[s+m-2][d])+c[m];
                }
            z[s][d]=eval[s][d];
            }
        }
    }
/*************************************************************
*                                                            *
*                      ANTI INTERFERENCE FEATURES            *
```

```
*                                                                    *
**********************************************************************/
for (s=1;s<=39;s++)
    {
    z[s][1]=r[s][1];
    }
for (s=1;s<=39;s++)
    {
    for (d=1;d<=36;d++)
        {
        u1=d-1;
        u2=d+1;
        t1=s;
        t2=s;
        z1[s][d]=z[s][d];
        z2[s][d]=z[s][d];
        if (d==1)
            {
            t1=s-1;
            u1=36;
            }
        if ((d!=1) || (s!=1))
            {
            if (d==36)
                {
                t2=s+1;
                u2=1;
                }
            if (z[t1][u1] > z[s][d]/cos(3.414/18.0))
                {
                z1[s][d]=z[s][d]+0.5*(sqrt(pow(z[s][d],2)
                +pow(z[t1][u1],2)-2.0*z[s][d]*z[t1][u1]
                *cos(3.414/18.0))/(z[t1][u1]*sin(3.414/18.0)) -
                1.0);
                }
            if (z[t2][u2] > z[s][d]/cos(3.414/18.0))
                {
                z2[s][d]=z[s][d]+0.5*(sqrt(pow(z[s][d],2)+
                pow(z[t2][u2],2)-2.0*z[s][d]*
                z[t2][u2]*cos(3.414/18.0))/(z[t2][u2]*
                sin(3.414/18.0))-1.0);
                }
            if (z1[s][d]>z2[s][d])
                {
                z2[s][d]=z1[s][d];
                }
            }
        }
    }
for (s=1;s<=38;s++)
    {
    for (d=2;d<=36;d++)
        {
```

```
        x[s][d]=y[s][d];
        z[s][d]=z2[s][d];
        }
    }
/*************************************************************
*                                                            *
*                  GENERATING THE NC CODES FOR MACHINING     *
*                                                            *
*************************************************************/
fprintf (file2,"(XYZ1 )\n");
fprintf (file2,"N1000(9)M06T01$\n");
fprintf (file2,"N1001(9)M03S1000\n");
fprintf (file2,"N1002(E)G90$\n");
fprintf (file2,"N1003(E)G00X0.5Y0B0Z3.75$\n");
fprintf (file2,"N1004(E)G01X1.0Z3.1F60$\n");
l=1005;
for (s=1;s<=38;s++)
    {
    for (d=1;d<=36;d++)
        {
        fprintf(file2,"N%4d(E)",l);
        l++;
        fprintf (file2,"X%5.4fZ%5.4fB%1d$\n",x[s][d],
                z[s][d]+1.0,(d-1)*10);
        }
    }
fprintf(file2,"N%4d(E)X%5.4fZ%5.4fB0$\n",l,x[39][1],z[38][1]
              +1.0);
l++;
fprintf(file2,"N%4d(E)M05$\n",l);
l++;
fprintf(file2,"N%4d(E)M02$\n",l);
fprintf(file2,"END\n");
fclose (file2);
}
```

```
program research (input,output);
(* This program develops the cutter path for machining a shoe
last using a turning operation. The input data is from
cylfoot2.dat. The input data is in the form x,r at each cross
section Anti inteference features of Saunders are also included*)


label
        100, 200;


type
        string = packed array [1..15] of char;
        matrix = array[1..50,1..40] of real;
        number = array[1..30] of real;


var
   c:number;
   a,e,yax,zax:real;
   d,m,k,n,l,s,i,b,t1,t2,u1,u2:integer;
   infile,outfile:string;
   in1,out1:text;
   x,r,eval,y,z,z1,z2:matrix;


begin
        write ('The name of the input data file:');
        readln (infile);
        writeln;
        writeln;
        write('The name of the output data file:');
        readln (outfile);
        open (in1,infile,old);
        open (out1,outfile,new);
        reset (in1);
        rewrite(out1);
        while not eof(in1) do
        begin
             for i := 1 to 38 do
             begin
                for d:= 1 to 36 do
                begin
                readln(in1,x[i,d],yax,zax);
                r[i,d]:=sqrt(sqr(yax)+sqr(zax));
                x[i,d] := x[i,d] + 1
                end
```

```
            end
        end;
(*Interpolation so as to have points at every 1/4 of an inch*)
        for d:= 2 to 36 do
        begin
            for s := 1 to 37 do
            begin
               if s = 1 then
               begin
                  y[1,d]:=s+ (0.25/36)*(d-1);
                  i := 1;
                  for k := 1 to 3 do
                  begin
                       c[k] := r[i,d];
                       i := i + 1
                  end;
                  for l:= 1 to 2 do
                  begin
                       for m := 3 downto l + 1 do
                       c[m] := (c[m] - c[m-1])/(x[m,d] -
                                  x[m-1,d]);
                  end;
                  eval[1,d] := c[3];
                  for m := 2 downto 1 do
                  eval[1,d] := eval[1,d]*(y[1,d] - x[m,d])
                                  + c[m];
                  z[1,d]:= eval[1,d];
                  goto 100
               end;
               if s = 37 then
               begin
                  y[37,d]:= x[s,d] + (0.25/36)*(d-1);
                  for m:= 1 to 3 do
                  c[m] := r[35+m,d];
                  for l:= 1 to 2 do
                  begin
                      for m:= 3 downto l + 1 do
                      c[m] := (c[m]-c[m-1])/(x[35+m,d]-
                                  x[35+m-1,d]);
                  end;
                  eval[37,d] := c[3];
                  for m:=2 downto 1 do
                  eval[37,d] := eval[37,d]*(y[37,d] - x[35+m,d])
                                              + c[m];
                  z[37,d] := eval[37,d];
                  goto 100
               end;
               y[s,d]:= x[s,d]+(0.25/36)*(d-1);
               for k:= 1 to 4 do
               c[k] := r[s+k-2,d];
               for l := 1 to 3 do
               begin
                  for m:= 4 downto l + 1 do
```

```
                    c[m] := (c[m] - c[m-1])/(x[s+m-2,d]
                                - x[s+m-2-1,d]);
               end;
               eval[s,d] := c[4];
               for m:= 3 downto 1 do
               eval[s,d] := eval[s,d]*(y[s,d]- x[s+m-2,d])
                                + c[m];
               z[s,d] := eval[s,d];
100:       end
        end;
        d :=1;
        for s := 1 to 38 do
        z[s,d] := r[s,d];
        for s := 1 to 37 do
        begin
        for d := 1 to 36 do
        begin
           u1 := d-1;
           u2 := d+1;
           t1 := s;
           t2 := s;
           z1[s,d] := z[s,d];
           z2[s,d] := z[s,d];
           if d = 1 then
           begin
                t1 := s-1;
                u1 := 36
           end;
           if (d = 1) and (s = 1) then
           goto 200;
           if d = 36 then
           begin
                t2 := s+ 1;
                u2 := 1
           end;
           if z[t1,u1] > z[s,d]/cos(3.414/18.0) then
           z1[s,d]:=z[s,d]+0.5*(sqrt(sqr(z[s,d])+sqr(z[t1,u1])-
                               2.0*z[s,d]*z[t1,u1]*
                               cos(3.414/18.0))/(z[t1,u1]*
                                    sin(3.414/18.0))-1.0);


           if z[t2,u2] > z[s,d]/cos(3.414/18.0) then
           z2[s,d]:=z[s,d]+0.5*(sqrt(sqr(z[s,d])+sqr(z[t2,u2])-
                               2.0*z[s,d]*z[t2,u2]*
                               cos(3.414/18.0))/(z[t2,u2]*
                               sin(3.414/18.0))-1.0);

           if z1[s,d] > z2[s,d] then
                z2[s,d] := z1[s,d];
200:    end
        end;
        for s := 1 to 37 do
```

```
        begin
           for d:= 2 to 36 do
           begin
                x[s,d]:=y[s,d];
                z[s,d]:=z2[s,d]
           end
        end;
        writeln(out1,'(XYZ1 )');
        writeln(out1,'N1000(9)M06T01$');
        writeln(out1,'N1001(9)M03S1000$');
        writeln(out1,'N1002(E)G90$');
        writeln(out1,'N1003(E)G00X0.5Y0B0Z3.75$');
        writeln(out1,'N1004(E)G01X1.0Z3.1F60$');
        l:=1005;
        for s:= 1 to 37 do
        begin
           for d:= 1 to 36 do
           begin
           write(out1,'N',l:4,'(E)');
           l := l+1;
           writeln (out1,'X',x[s,d]:5:4,'Z',z[s,d]+1.0:5:4,'B',
                    (d-1)*10:1,'$')
           end
        end;
        writeln(out1,'N',l:4,'(E)X',x[38,1]:5:4,'Z',z[38,1]+
                                 1.0:5:4,'B0$');
        l:=l+1;
        writeln(out1,'N',l:4,'(E)M05$');
        l:=l+1;
        writeln(out1,'N',l:4,'(E)M02$');
        writeln (out1,'END')
end.
```