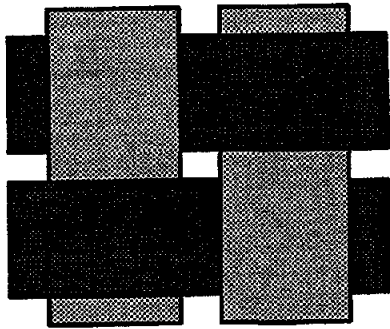




Visual Simulation of Woven, Knitted, and non-Woven Fabrics (using the SAS® System)

Robert E. Allison, Jr., North Carolina State University College of Textiles, Raleigh, NC
and SAS Institute Inc., Cary, NC

Moon W. Suh, North Carolina State University College of Textiles, Raleigh, NC

ABSTRACT

This paper describes techniques that have been used to simulate fabric using SAS/GRAPH® software. The methods can be used to visualize real yarn data, or to visually simulate the effects of various random distributions of yarn or fiber properties on the fabric. The examples are generalized versions of code developed at North Carolina State University College of Textiles for research projects under the direction of Dr. Moon W. Suh.

INTRODUCTION

Fabric simulation is the latest tool helping U.S. textile manufacturers compete with goods produced by inexpensive overseas labor. By simulating the fabric before it is actually produced, fabric with unwanted properties such as weak spots, unevenness, or patterns can be avoided.

To simulate a fabric, yarn and fiber properties such as thickness, strength, length, or evenness can be measured (or simulated), and then visually represented as shades, colors, and geometry using a computer.

Three examples are included here. The first two examples show how to simulate **woven** and **knitted** fabric using the GMAP procedure. The third example demonstrates how to simulate **non-woven** fabric using the GPLOT procedure.

WOVEN FABRIC SIMULATION

The first example demonstrates how to simulate a woven fabric. A woven fabric is composed of interlaced yarns, similar to a basket. There are many different fabric weaves — the simplest (i.e., plain, or 1/1) weave is used here.

There are generally two sets of yarns used in a woven fabric. The **warp** yarns run the entire length of the fabric and are simulated going from the top to the bottom of the page. The **filling** (or weft) yarns run across the fabric and are simulated going across the page.

The basic strategy used to simulate the woven fabric is to create a **SAS "map"** where each visible segment of yarn is a separate object that can be addressed and shaded appropriately. The yarn segments are analogous to states in the U.S. map.

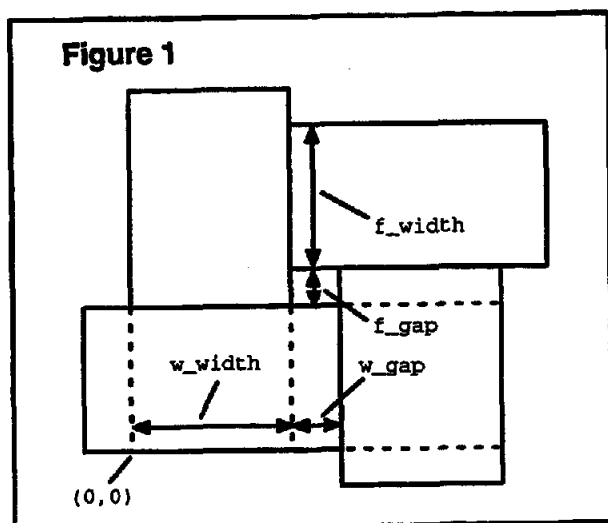
Macro variables allow you to easily specify the **number** of picking and warp yarns, as well as the **width** of the yarns and the **spacing** (or gap) between the yarns. These variables allow you to control every aspect of the geometry and density/looseness of the simulated fabric. A **repeat** variable allows you to specify how many yarn segments in the map each yarn value is to be mapped to — this is useful when the yarn values represent a length of yarn longer than the length of a single segment in your simulated fabric.

```
%let filling=21;
%let f_width=100;
%let f_gap=25;

%let warps=31;
%let w_width=100;
%let w_gap=30;

%let warpstrt=10000;
%let repeat=4;
```

The following plain-weave unit repeat (see Figure 1) illustrates how the macro variables affect the geometry of the fabric.



The coordinate system places 0,0 at the bottom left corner of the map. The order in which the 4 coordinates of each rectangular segment are created is important, because the GMAP procedure will connect the coordinates in the order specified, for example if you sort the dataset, the coordinates will be in an undesirable order.

The **filling yarn segments** are numbered from left to right, starting with the number 0. Each time through the following “do” loops, four x/y pairs representing a yarn segment are output to the fillmap SAS data set.

```
data fillmap;
squ_id=0;

do j=0 to (&filling-1);
  do i=0 to (&warps-1);
    squ_id + 1;
    if (mod((j+1),2)-mod((i+1),2))=0
    then do;

      x=i*(&w_width+&w_gap)-&w_gap;
      y=j*(&f_width+&f_gap);
      output;

      x=(i+1)*(&w_width+&w_gap);
      y=j*(&f_width+&f_gap);
      output;

      x=(i+1)*(&w_width+&w_gap);
      y=(j+1)*(&f_width+&f_gap) - &f_gap;
      output;

      x=i*(&w_width+&w_gap) - &w_gap;
      y=(j+1)*(&f_width+&f_gap) - &f_gap;
      output;
    end;
  end;
end;
run;
```

The **warp yarn segments** are numbered from bottom to top, starting at 10,000. Notice the use of the mod statement to select only alternating warp and picking yarns — this simulates the interlacing of the yarns.

```
data warpmap;
squ_id=0;

do i=0 to (&warps-1);
  do j=0 to (&filling-1);
    squ_id + 1;

    if (mod((j+1),2)-mod((i+1),2))~=0
    then do;
      x=i*(&w_width+&w_gap);
      y=j*(&f_width+&f_gap) - &f_gap;
      output;

      x=(i+1)*(&w_width+&w_gap) - &w_gap;
      y=j*(&f_width+&f_gap) - &f_gap;
      output;
```

```

x=(i+1)*(&w_width+&w_gap) - &w_gap;
y=(j+1)*(&f_width+&f_gap);
output;

x=i*(&w_width+&w_gap);
y=(j+1)*(&f_width+&f_gap);
output;
end;
end;
end;
run;

data warpmap;
set warpmap;
squ_id = squ_id + &warpstrt;
run;

```

Next, the warpmap and fillmap data sets are joined to create the fabric map.

```

data fabric (keep = squ_id x y);
set fillmap warpmap;
run;

```

Here, filling (yarns1) and warp (yarns2) yarn values are simulated using the ranuni function random number generator. Notice that random number seeds are used to allow the simulation to be repeated. Also, a different seed is used for the filling and the warp yarns — this allows the same picking or warp yarns to be simulated, while varying the other. Note that real yarn values, as measured from various yarn testing devices, could be used instead of random numbers.

```

%let v_seed1=4363;
data yarns1;
do i=1 to ((&filling*&warps)/&repeat
+ &repeat);
value = ranuni(&v_seed1)*1000;
do j=1 to &repeat;
squ_id + 1;
output;
end;
end;
run;

%let v_seed2=2912;
data yarns2;
squ_id = &warpstrt;
do i=1 to ((&filling*&warps)/&repeat
+ &repeat);
value = ranuni(&v_seed2)*1000;

```

```

do j=1 to &repeat;
squ_id + 1;
output;
end;
end;
run;

```

Next, the graphics options (goptions) and titles are set.

```

goptions reset=global gunit=pct
noborder rotate=landscape
vsize=7 hsize=9 htitle=7 htext=5
ftitle=zapfb ftext=swissl;

title "Woven Fabric Simulation";
title2 "(from simulated yarn data)";

```

Finally, the yarn values are mapped into three shades. Shade 1 represents thin yarn, 2 represents normal yarn, while category 3 is for thick yarn. These numerical values are then mapped to colors using pattern statements, and the fabric is simulated using PROC GMAP (see Figure 2).

```

data yarn;
set yarns1 yarns2;
if value < 150 then shade=1;
else if value > 850 then shade=3;
else shade=2;
run;

pattern1 v=s c=white;
pattern2 v=s c=graya0;
pattern3 v=s c=gray22;

proc gmap data=yarn map=fabric;
id squ_id;
legend1 value=(j=1 'thin' 'normal' 'thick')
shape=bar(4,4) label=none;
choro shade/discrete
legend=legend1
midpoints = 1 2 3
coutline=black;
run;

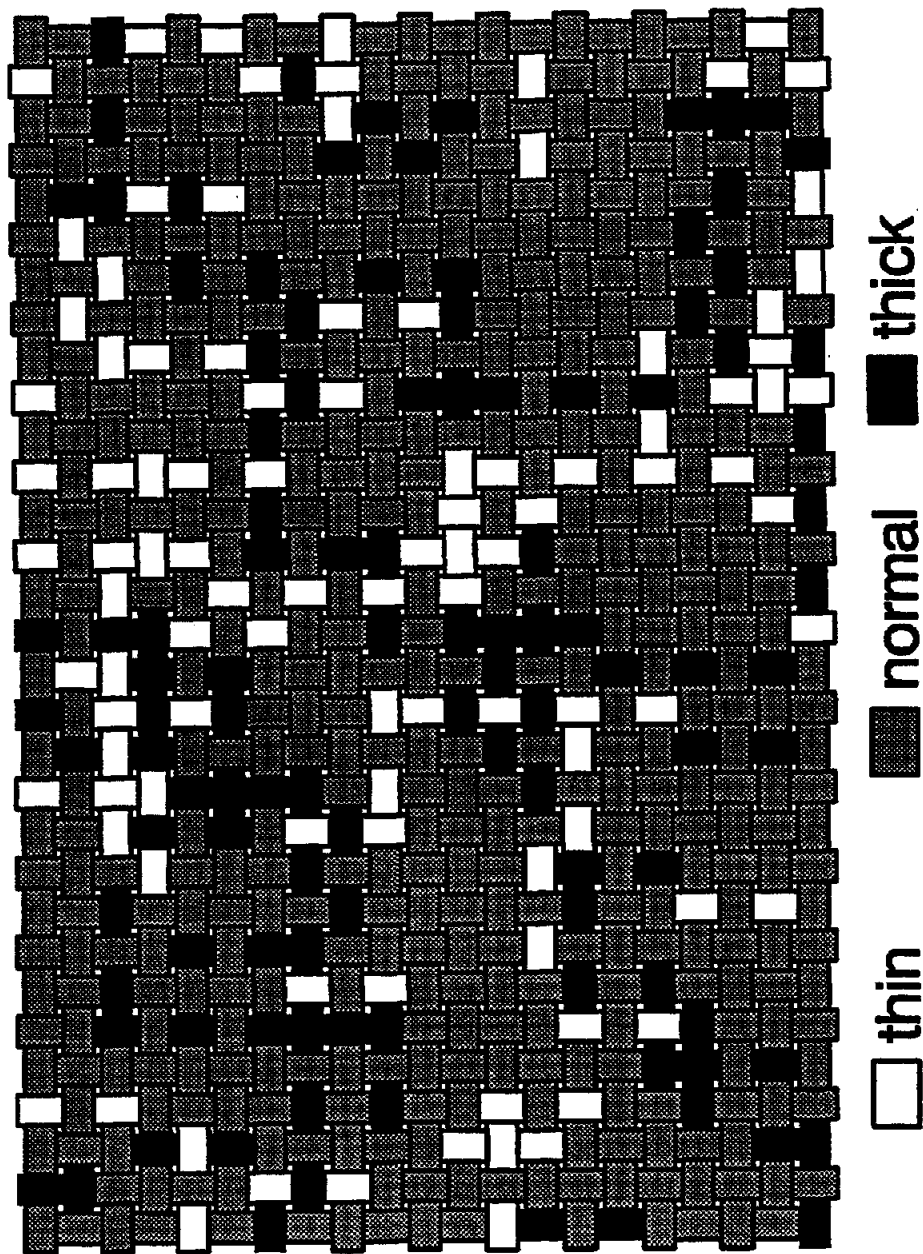
```

A small number of warp and filling yarns were used in this example to demonstrate the method and to produce a fabric simulation with easily visible yarn segments. Much higher numbers of warp and filling yarns can be specified to produce a more realistic fabric simulation.

Figure 2

Woven Fabric Simulation

(from simulated yarn data)



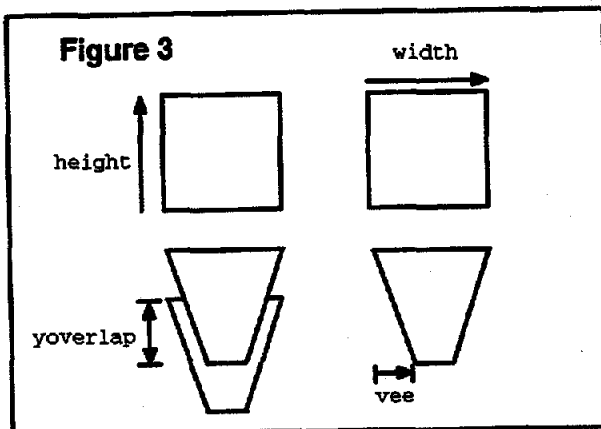
KNITTED FABRIC SIMULATION

This example demonstrates how to simulate a knitted fabric using the SAS System. A knitted fabric is composed of inter-looped yarns. Sweaters, socks, and t-shirts are examples of items made from knitted fabric.

The knitted fabric simulation is different from the woven fabric simulation in two ways. First, a single continuous yarn (yarns1) is simulated instead of two yarns. Second, a different geometry is used for the unit repeat. The main purpose of this example is to show the ease with which the previous technique can be modified and applied to other areas.

The geometry of the knitted fabric unit cell is controlled by the following macro variables, as shown in Figure 3.

```
%let height=1;           %let width=2;
%let yoverlap=.45;       %let xoverlap=.30;
%let vee=.80;
```

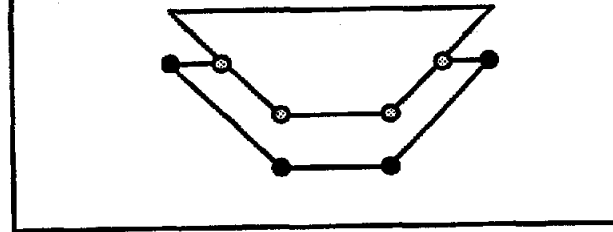


The following macro variables determine the size of the knitted fabric (in unit cells), and how many times to repeat each yarn shade/value.

```
%let x_cells=25;         %let y_cells=60;
%let repeat=3;
```

The coordinates of the unit cells are calculated in much the same manner as in the woven fabric example. Although only four coordinates are calculated for each unit cell, the visual effect of eight coordinates is produced by overlapping the unit cells (see Figure 4).

Figure 4



```
data knit_map;
do j=0 to ((&y_cells-1)*(&height-&yoverlap))
    by (&height-&yoverlap);
do i=0 to ((&x_cells-1)*&width) by &width;
    squ_id + 1;
    x=i+&vee;           y=j; output;
    x=i+&width-&vee;     y=j; output;
    x=i+&width+&xoverlap;
    y=j+&height; output;
    x=i-&xoverlap; y=j+&height; output;
end;
end;
```

Next, random yarn values are generated and mapped to shades, and the knitted fabric simulation is rendered using PROC GMAP. The knitted fabric simulation is shown in Figure 5.

```
%let v_seed1=4363;
data yarns1;
do i=1 to (&x_cells*&y_cells)/&repeat+&repeat;
    value = ranuni(&v_seed1)*1000;
    do j=1 to &repeat;
        squ_id +1; output; end;
end; run;
```

```
/* Use the same goptions as before */
```

```
title "Knitted Fabric Simulation";
title2 "(from simulated yarn data)";
```

```
data yarn; set yarns1;
    if value < 150 then shade=1;
    else if value > 850 then shade=3;
    else shade=2; run;
```

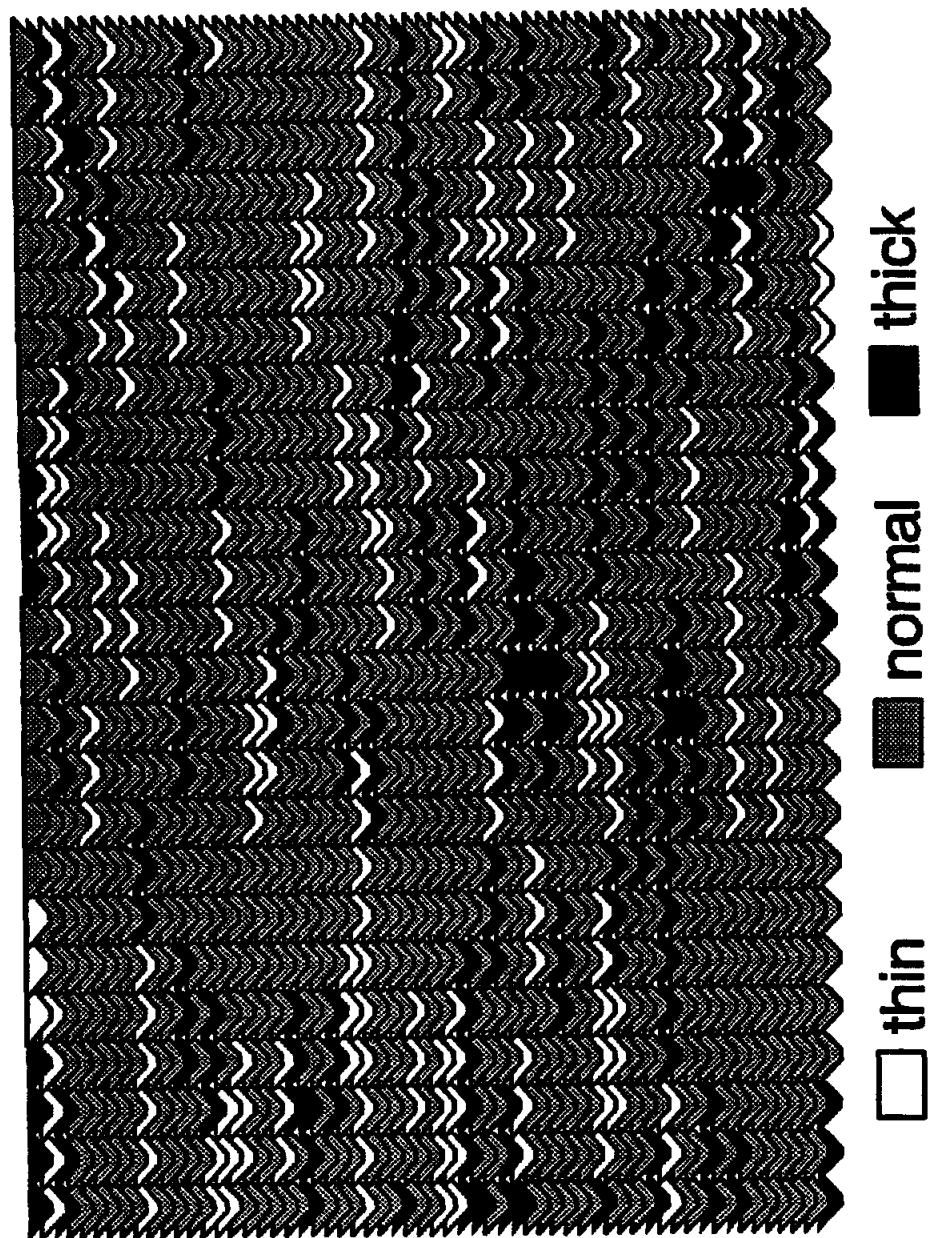
```
/* Use same pattern statements as before */
```

```
proc gmap data=yarn map=knit_map;
    id squ_id;
    legend1 shape=bar(4,4) label=none
    value=(j=1 'thin' 'normal' 'thick');
    choro shade/discrete coutline=black
    legend=legend1 midpoints = 1 2 3;
run;
```

Figure 5

Knitted Fabric Simulation

(from simulated yarn data)



NON-WOVEN FABRIC SIMULATION

A **non-woven fabric** is composed of sheets of **fibers** with varying degrees of organization, held together by adhesives, stitching, or needle punching. Generally, the more intersecting fibers, the stronger the fabric. This example shows how to simulate such a fabric using PROC gplot.

Unlike the yarn used in woven fabric, it is impractical to test each individual fiber used in a non-woven fabric. Instead, random samples are tested, and a characteristic distribution is calculated. These random distributions can be simulated easily using SAS software. The simulated fiber values can then be visualized as the color and thickness of the lines used to simulate the fibers.

In addition to **visual** fabric simulations, **mathematical** simulations are also useful. For example, the fiber end point coordinates were used in one study to calculate the number of fiber intersections and help validate a statistical model of the relationship between the number of fibers and the number of intersections.

The code used to visualize a non-woven fabric is similar to the woven and knitted fabric code, but uses a different graphical procedure (PROC Gplot) to render the fabric.

The following macro variables allow you to easily change the **number** of fibers (numfibers) and their **length** (length), as well as the random number **seeds** for the location of the center point of the fibers (cen_seed), their **rotation** (rot_seed), as well as the seed for the values (v_seed) assigned to the fibers.

```
%let numfiber=1400;      %let length=18;
%let cen_seed=3826;      %let rot_seed=2518;
%let v_seed1=4313;
```

Using the macro variables, center coordinates and rotation angles are randomly selected for the proper number of fibers, and the end points are calculated. A value is also

randomly assigned to each fiber so that it can be mapped to a shade. After each fiber's end points are output, a null value is output — this allows the skipmiss option to be used in PROC Gplot so that each fiber can be plotted individually without a connecting line.

```
data fibers;
  do i=1 to &numfiber;
    fiber=i;
    x_center = ranuni(&cen_seed)*100;
    y_center = ranuni(&cen_seed)*100;
    angle    = ranuni(&rot_seed)*360;

    value = ranuni(&v_seed1)*1000;
    if value < 980 then shade=1;
    else shade=2;

    x = x_center + &length/2*cos(angle);
    y = y_center + &length/2*sin(angle);
    output;

    x = x_center - &length/2*cos(angle);
    y = y_center - &length/2*sin(angle);
    output;

    x=.; y=.; output;
  end;
run;
```

Next, the goptions and titles are set. The targetdevice used in conjunction with the lengths specified in the axis statements will force the axes to be the same size, instead of automatically stretching to fit the page.

```
goptions reset=global gunit=pct gwait=1
  targetdevice=ps rotate=landscape
  noborder cback=white colors=(black)
  ftitle=zapfb ftext=swissl
  htitle=5 htext=3;

title "Non-Woven Fabric Simulation";
title2 h=3 "number of fibers=&numfiber
  fiber length=&length";
title3 h=3 "seeds: center=&cen_seed
  angle=&rot_seed value=&v_seed1";
```

Although the centers of the fibers were placed in a 0–100 grid, the end points can extend beyond those coordinates. Therefore, the following axes are used to show ten extra units in each direction.

```
axis1 label=none length=5.5in
  order=( -10 to 110 by 10);
axis2 label=none length=5.5in
  order=( -10 to 110 by 10);
```

Symbol statements are used to specify both a width and a color for each fiber. Although black and gray are used for this publication, colors are more easily distinguishable. Also, code can easily be added to extend this model to more than two colors.

```
symbol1 v=none i=join w=2 c=black;
symbol2 v=none i=join w=10 c=graybb;
```

Finally, the fibers of each shade are plotted with separate gplots, using the appropriate symbol. The plots are stored using name='shade1' and name='shade2' so that they can later be displayed together.

```
goptions nodisplay;
proc gplot data=fibers; where shade=1;
plot y*x=1 / overlay skipmiss frame
href = 0 100 vref = 0 100 lhref=2 lvref=2
vaxis=axis1 haxis=axis2 name='shade1';
```

```
proc gplot data=fibers; where shade=2;
plot y*x=2 / overlay skipmiss frame
href= 0 100 vref= 0 100 lhref=2 lvref=2
vaxis=axis1 haxis=axis2 name='shade2';
```

The GREPLAY procedure is used to display the shade1 and shade2 plots in the same frame of a template, thereby overlaying them (see Figure 6). The order in which the plots are displayed determines which yarns appear "on top."

```
goptions display; proc greplay igout=gseg
tc=sashelp.templt template=whole nofs;
treplay 1:shade1 1:shade2; end; run;
```

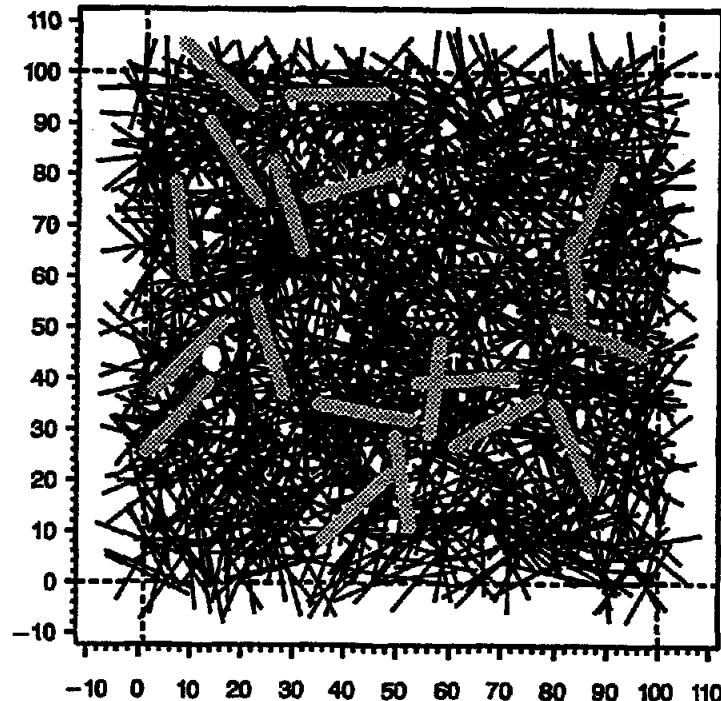
The following code can be used to delete the shade1 and shade2 plots, allowing you to run another simulation using the same graph names.

```
proc greplay nofs igout=gseg;
delete _all_; end; run;
```

Figure 6

Non - Woven Fabric Simulation

number of fibers=1400 fiber length=18
seeds: center=3826 angle=2518 value=4313



FUTURE RESEARCH PLANS

One area of potential research is to extend the **woven fabric model** to include more weaves. For example, twills, satins, and other weaves could be simulated. SAS libraries of maps containing each weave could be created, thereby allowing users to quickly map their data to these weaves without recalculating the maps each time.

Both the **woven** and **knitted fabric models** could be enhanced to represent the actual physical width of the yarns, like the non-woven model. Also, the models could be enhanced to represent the depth, thickness, and texture of the yarns, producing a three-dimensional fabric simulation.

Also, since these simulations use SAS maps, the new SAS/GIS® software could be used to allow the user to click on a certain yarn and trigger hyper-graphic actions that would display drill-down information about the yarn segment.

Another promising area is interfacing the models with real-time data acquisition. For example, if the fabric is simulated as the yarn is spun, problems can be avoided earlier in the manufacturing process.

There are also several enhancements planned for the **non-woven fabric model**. For example, existing FORTRAN code used to count the number of fiber intersections based on the coordinates of the end points could be converted to SAS code. Also, arcs could be used instead of lines to simulate fibers with crimp, and the length could be randomly assigned for each fiber.

In addition, SAS/INSIGHT® software might be used to view the simulation, providing the users with an interface that allows them to rotate the plot in three-dimensions. This software also allows the user to easily select fibers on the plot or in a spreadsheet-like grid of values, and immediately see both the values and graphic segments high-lighted.

In all of the models, the SAS code could be enhanced to allow users to easily select other random distributions to simulate the yarn and fiber properties. For example, a Normal distribution could be used for the fiber rotations to simulate orientation of fibers in the non-woven fabric. Also, distributions representing periodic defects could be easily generated.

There is also great potential for applying these techniques to many other areas of simulation, such as engineering and geology.

REFERENCES

Jeong, Sung H. (1994), *Monitoring and Visualization of Yarn and Fabric Qualities Through Signal Processing*, dissertation at NCSU.

Kim, Jong J. (1992), *Softness Properties of Fabric-Like Tissues*, dissertation at NCSU.

Suh, Moon W., *Geometrical Probability of a Two-Dimensional Random Fiber Composite*, a paper presented to Joint Statistical Meetings.

SAS Institute Inc. (1990), *SAS/GRAPH® Software: Reference, Version 6, First Edition*, Cary, NC: SAS Institute.

SAS, SAS/GRAPH, SAS/INSIGHT, and SAS/GIS are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

The authors welcome questions and comments via e-mail (realliso@tx.ncsu.edu and moon_suh@ncsu.edu). The authors can also be contacted at the following address:

Box 8301
North Carolina State University
College of Textiles
Raleigh, NC 27695-8301 USA