



# Statistical

## COMPUTING & GRAPHICS

### A WORD FROM OUR CHAIRS

---

## Statistical Computing

One of the perks of the Chair of the Statistical Computing Section is writing a column for this newsletter. Imagine: I can write for several thousand members of my profession, without the benefit of refereeing. I get to tell you what I think, rather than what I know.

Maybe I'm a bit odd, but what I've been thinking about is copyrighting and patenting of statistical software and of computer material in general. My interest was piqued by an advertisement I received for a computer package that would display a high dimensional plot using a patented algorithm. The name of the method was jargon, so I could tell nothing from the advertisement about the program and what it did. Mainly, what the advertisement did was make me wonder just what it meant for an algorithm to be patented. Was I supposed to believe that patenting was a substitute for peer review? Did it imply certification? Did it mean that I could not compute this graph, whatever it is, with my own computer code unless I paid a royalty?

Assuming that most of us non-lawyers are as ignorant of copyrighting and patenting as I was, I thought a short summary of what I've learned might be of interest. Finding the information was easy: one call to my university's patent office brought a few relevant papers, particularly [1], and a copy of the law. A review of my university library's on-line card catalog gave nearly 200 references, including a journal, *Software Protection*, which has been published since 1982.

Copyright and patent are very different. The basis for copyright is contained in the U. S. Constitution, which gives to Congress the authority "[t]o promote the Progress of Science and the useful Arts by securing for Limited Times to Authors and Inventors the exclusive Right to their respective Writings and Discoveries" [2]. Exactly what can be covered by copyright was spelled

CONTINUED ON PAGE ??

### FEATURE ARTICLE

---

## Saxpy, gaxpy, LAPACK, and BLAS

Colin Goodall *The Pennsylvania State University*

### Measuring Performance

One of the best understood computational tasks is linear algebra. Considerable effort has gone into fast and accurate code for these manipulations, e.g. LINPACK (Dongarra et al. 1979), EISPACK, and most recently LAPACK (Anderson et al. 1992). The speed of these computations is measured in mega flops (MFLOPS), or millions of floating point instructions per second. Each floating point instruction is a single arithmetic operation, e.g. a multiplication, a divide, an addition or subtraction, performed in full floating point precision arithmetic, usually 64 bits double precision. Two common benchmarks for comparing performance are Dhrystone, which measures speed of integer arithmetic computations, and Whetstone for floating point computations. These measure not just the speed of the CPU but also compiler performance. Dhrystone performance, measured in millions of instructions per second, can equal and possibly exceed the clock speed of the CPU.

A more exacting benchmark involves a practical linear algebra task, specifically the  $100 \times 100$  LINPACK benchmark (Dongarra (1993)). This measures the speed achieved, in MFLOPS, in computing the Cholesky decomposition of an arbitrary  $100 \times 100$  symmetric matrix, using a fixed set of FORTRAN code. *The numbers obtained from this benchmark are surprising: For example a SparcStation II, with a processor rated at around 25 MFLOPS, has a  $100 \times 100$  LINPACK speed of 4.0 MFLOPS (Dongarra (1993)). Put another way, the theoretical requirement for an  $n \times n$  Cholesky decomposition is  $n^3/3 + 2n^2$  flops, or 353,333 flops when  $n = 100$ .*

CONTINUED ON PAGE ??

## EDITORIAL

---

“Three in 93 and four in 94” was proclaimed as the rallying call of the new editors of the joint *Statistical Computing and Statistical Graphics* newsletter. If we have succeeded then this first issue of 1993 should be in your hands before the April Interface meeting. We are planning to mail the second issue before the Joint Statistical Meetings in August, and the third sometime around Thanksgiving. More issues mean the newsletter can be more timely, provide announcements and encourage dialogue. Please use your newsletter to communicate with the membership of the two largest sections of the ASA. Our deadlines for the remaining issues in 1993 are the last day of June and October.

Many regular columns will continue, but we solicit your help with new ideas and offers to write columns or once-only pieces. Please keep those e-cards and e-letters coming!

This issue has two feature articles. The first feature describes the availability and discusses the design of public domain matrix and linear algebra routines. Many academic computer installations can make available high quality subroutines of the standard algorithms without the need to purchase or lease commercial software packages, e.g. IMSL or NAG. Algorithms are available, as described by Colin Goodall, to anyone with FTP software and access to the internet.

The second feature, about graphics and stereoscopic displays, is like the Sunday night mini-movie on network television. It looks like a feature article, but is really the first episode of a new column. Dan Carr has stepped down as editor but couldn't resist the challenge of a regular column. The graphic images which accompany this article should also be seen as the beginning of more innovative graphical material which we would like to print.

The Newsletter is now being set/typeset in  $\text{\LaTeX}$ . After three years of colorful newsletters we have tried to restrain ourselves to typographic spice and a change in format. Neither of us has any sense of color, so we will take the conservative (and probably boring) tack and use black text on a neutral background. We want to credit Kevin Fox, the design artist from Penn State, for the new masthead and for keeping our link to the past with the intersecting circles punctuating the articles.

Submissions should be sent by email to either of the editors. If you can prepare your article in  $\text{\TeX}$  or  $\text{\LaTeX}$  that will make our lives just a little easier. Otherwise plain old ASCII format is fine.

In preparing this issue we have realized yet again the herculian efforts of the newsletter's founding editors, Sallie Keller-McNulty and Dan Carr. We hope we can live up to the standards they have set for us.

James L. Rosenberger  
*Editor Statistical Computing Section*  
JLR@stat.psu.edu

Mike Meyer  
*Editor, Statistical Graphics Section*  
mikem@stat.cmu.edu



## SECOND FEATURE

---

# Production of Stereoscopic Displays for Data Analysis

Daniel B. Carr *George Mason University*

Dedicated to David L. Hall, colleague and friend.

Stereoscopic displays help the analyst escape from the limited domain of 2-D visualization into the natural domain of 3-D visualization. The goal of producing 3-D scatterplots motivates much of the following discussion. The goal has strong implications in terms of selecting a stereo projection. In the every day world, familiarity with objects and many depth cues facilitates fusion of left and right retinal images into a stereo image. Monocular depth cues include linear perspective (objects of equal size transect areas inversely proportional to their distance), interposition or occlusion (when one object is in front of another it obscures the more distant object), shadows (we generally assume light comes from above), detail perspective (no fine detail appears in distant objects due to limited visual acuity), and aerial perspective (greater optical depth through the air leads to a blue shift). For most elementary 3-D scatterplots prior knowledge about the form to be perceived is limited and monocular cues are restricted, so care must be taken in the selection of a stereo projection.

## ***Two Infinite Families of Reasonable Stereo Projections***

Different geometric models lead to different stereo projections. (Geometric models are idealized in that each eye has a blind spot, a region of high resolution, and various imperfections.) A simple model (Newman and Sproull 1979) presumes that the eyes converge on a single focal point and constructs left and right images by

projecting onto left and right projection planes. The projection planes contain the focal point and are orthogonal to the respective lines of site. This fixed-focal-point model is appropriate for advanced dynamic systems that update immediately as the eyes change their focal point. However, the fixed-focal-point projection does not correspond to the data analyst's typical stereoviewing scenario. Valyus (1962) states, "it has been shown experimentally that eye movements performed when stereoscopic pictures are viewed are similar to those performed in observing a real object. As the gaze is transferred from one object to another the eyes perform conjugate movements directed to the subjectively most important regions, and at the same time coordinated convergence movements take place." While the fixed-focal-point projection has proved passable for showing non-updated images of familiar scenes, image fusion problems result when looking at points in the corners of the plot. Thus fixed-focal-point projection is inadequate for 3-D scatterplots.

When the eyes have multiple focal points within the same stereo image, a reasonable compromise uses a single common projection plane that is parallel to a frontal view of the face. In a multiple-focal-point model, a data point projected into the projection plane has the same  $y$  coordinate for both left and right images. This is a fundamental requirement for 3-D scatterplot projections.

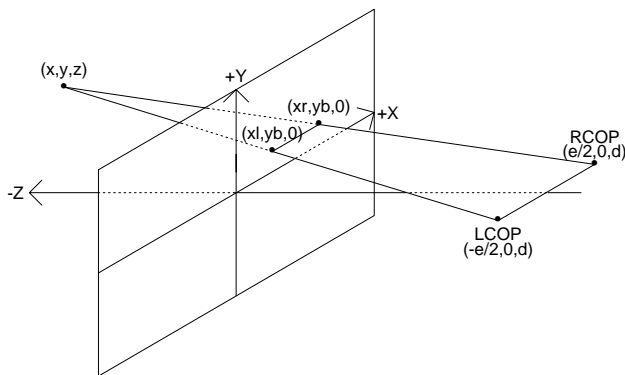


Figure 1

Two classes of projections satisfy the multiple-focal-point (single projective plane) constraint. The first class uses standard projective methods with separate centers of projection for the left and right eyes, denoted LCOP and RCOP respectively. Assume a right handed coordinate system with positive  $z$  toward the viewer. Then the LCOP and RCOP coordinates relative to the center of the workstation screen are  $(-e/2, 0, d)$  and  $(e/2, 0, d)$  where  $e$  is the eye separation (see Figure 1). The projected coordinates can then be found by scaling the vector from the eye to the data point by a constant,  $s$ , so that the scaled vector touches the screen (the  $z$  coordinate is

0). For the right eye this yields

$$s \times [(x, y, z) - (e/2, 0, d)] + (e/2, 0, d) = (x_r, y_r, 0) \quad (1)$$

Solving both right-eye and left-eye equations for  $s$  based on the  $z$  coordinate and substituting yields:

$$\begin{aligned} x_r &= (dx - ez/2)/(d - z) \\ x_l &= (dx + ez/2)/(d - z) \\ y &= dy/(d - z) \end{aligned} \quad (2)$$

The fact the both left and right images have the same  $y$  coordinate is evident from geometric considerations. Consider a data point appearing behind the projection plane (or viewing screen). The two eyes and this data point form a triangle that intersects the screen. If the frontal view of the face is parallel to the screen and the eyes are level, then the twin projected points must also be level.

The second class of multiple-focal-point stereo projections may be called a depth-cued orthogonal projection. In this projection, the LCOP and RCOP are shifted for each data point so that the midpoint between the eyes has the same  $x$  and  $y$  coordinates as the display-scaled data point to be projected. The depth-cued points are then

$$\begin{aligned} x_r &= x - ez/[2 \times (d - z)] \\ x_l &= x + ez/[2 \times (d - z)] \\ y &= y \end{aligned} \quad (3)$$

The two projection classes have continuous projection and viewing parameters. Projection parameters include eye separation ( $e$ ), projection distance ( $d$ ) and the size of the viewing cube into which we translate and scale the data. For convenience consider the workstation screen as a viewing cube that is 20 centimeters on a side with its front face centered and aligned with the screen surface. Viewing conditions may differ from the projection model and that introduces magnification ( $m$ ) and viewing distance ( $d'$ ) parameters. These parameters can be varied over an interval and still lead to comfortable image fusion of a 3-D point cloud. Consequently both stereo projection classes are infinite.

### Projection Parameter Bounds and Parallax

While an infinite number of projections are effective, some parameter bounds should not be violated and some projections are more desirable than others. The following provides some background concerning parameter constraints. The curious reader is referred to Valyus (1962) for additional detail.

For most people, eye separation falls in the interval between 5.2 and 7.4 centimeters. Fortunately, using the

exact eye separation for each individual is not crucial as evidenced by stereo publications that are enjoyed by diverse audiences.

Parallax is a key concept for understanding the projection parameter bounds. The horizontal parallax,  $p$ , of a point refers to the distance between the projected coordinates on the screen,  $x_r$  and  $x_l$ . Then for our multiple-focal-point stereo projection

$$p = x_r - x_l = -ez/(d - z) \quad (4)$$

A point appearing in front of the screen will have a positive  $z$ ,  $z < d$ , and the parallax will be negative. Similarly, a point appearing behind the screen will have positive parallax. Parallax must be limited to provide acceptable stereo fusion. Maintaining image focus constrains the amount of acceptable parallax. Suppose the eyes converge on the twin images of a point as if the point were real. The apparent location of the point is the focal point. Eye convergence is usually coupled with accommodation (lens focusing) so that a region in front and in back of the focal point is in focus. If this region includes the workstation screen the image of the point will be clear. While those experienced in stereo viewing often learn to decouple the convergence and accommodation of their eyes, a mismatch can lead to either fusion or focus problems. Constraining the parallax to keep the perceived image depth close to the screen avoids such problems. The horizontal parallax is related to angular parallax on the retina. Studies (Valyus 1962, Yeh and Silverstein 1990) have related the speed of image fusion to angular parallax and provide guidelines. In short, restricting the viewing cube to 23 centimeters on a side or smaller will generally satisfy the less restrictive (but slower fusion) Valyus bound. Lipton (1982) recommends the equivalent of centering the viewing cube depthwise on the screen to control the parallax. In this case the viewing cube can be made substantially larger before the parallax becomes excessive.

### **Magnification and Viewing Distance**

View related parameters include the viewing distance  $d'$  and the image magnification  $m$ . The equation of parallax can be solved for the apparent distance from a point to the screen,  $z'$ , in terms of the actual viewing distance,  $d'$ .

$$z' = d'/(1 - e/p) \quad (5)$$

If the parallax is fixed then doubling the viewing distance doubles the apparent distance of the point to the screen. Thus the viewing cube frame can be made to appear squashed or elongated by selecting a different viewing distance,  $d'$ , than actual projection distance,  $d$ .

Often a stereo image is shown to a large audience on a big screen. If the image is not designed for the room,

magnification may cause problems. Magnifying the image magnifies the parallax. A Taylor series expansion of the depth equation about  $p = 0$  yields

$$z' = -d' \times (p/e + (p/e)^2 + (p/e)^3 \dots) \quad (6)$$

When the parallax is small relative to eye separation, magnifying the image increases the apparent depth almost linearly. As the parallax to eye-separation ratio approaches one-half, the nonlinear terms contribute equally and magnification distorts the image. As magnified parallax approaches eye separation, the depth of image theoretically approaches minus infinity. This causes both depth distortion and fusion problems before the apparent depth reaches infinity.

### **Control of Perspective Differences**

Under natural conditions the left and right eyes have different views of the world. The field of view for a single human eye is about 150° horizontally and 135° vertically. In binocular vision the field of view covered by both eyes is 120° so 30° or 1/5 of each eye's field of view is unique to the eye. For 3-D scatterplots all points must be seen unless hidden by occlusion, so images must be restricted to the shared field of view. If the viewing cube is sufficiently smaller than available display space on the workstation screen this is not a problem.

Hodges (1992) provides an astute comment on stereo production for workstations. He notes that some hardware systems provide only a single center of projection. Then the standard trick for producing stereo projections is to use the "off-axis" projection. For each eye's view this shifts the data toward midpoint between the eyes, projects from the midpoint and then shifts the result back. While the projections produce coordinates identical to the LCOP and RCOP projections, the fields of view differ. This provides another reason to constrain the size of the viewing cube. For scatterplots all corresponding left and right points must appear on the screen.

Perspective differences within the shared field of view can cause fusion and interpretation problems. Perspective induced fusion problems may be evident when showing the viewing-cube frame in small side-by-side plots. If, for example, the left cube face has the  $x$  coordinate at  $-e/2$ , the left cube face will project as a line in the left eye image but as a trapezoid in the right eye image. This radical perspective-induced discrepancy complicates image fusion. Since the viewing cube frame is simple, avoiding radical differences in perspective is straightforward. For example rotating the viewing cube often suffices. In fact some analysts prefer perspective views of the cube that have two or

Figure 2 - Random Points on a Mobius Strip. The structure is apparent with any 3-D viewing approach. In flatland recognizing the structure is much harder. Piecing scatterplot matrix views together is not so easy without brushing or conditioning. Conditioning or slicing is helpful in moving to higher dimensions. For real data, overplotting is more of a problem and density- based presentations become advantageous.

Figure 3 - Selected Stereo Contours for a 3-D Density Estimate.

Figure 4 - Steepest Ascent Ridge Traces for a 2-D Density Surface. Ridges and their projections in domain space provide another way of viewing density estimates. This example is provided courtesy of Qiang Luo.

Figure 5 - Random Dot Stereogram. This contour completion illusion at different depths takes a while to fuse. This image is provided courtesy of Nathan Carr, who used it in a high school science project that repeated experiments pioneered by Bela Julesz (1971).

three infinity points rather than the common head-on view that has only one infinity point. Another trick bases the projection on a distance substantially larger than the actual viewing conditions. This reduces perspective differences.

The foreshortening (interpoint distances appear smaller as a function of depth) of perspective views complicates the reading of coordinates based on given axes. Equations (3) provide an orthographic stereo projection that removes all perspective differences and allows the  $x$  and  $y$  coordinates to be interpreted directly. Carr and Littlefield (1983) describe a simple implementation that exploits the scaling in standard statistical graphics packages. Plot the left eye and right eye coordinates of a point using  $(X - X_p, Y)$  and  $(X + X_p, Y)$  respectively where the values are in data units. The expression for the half parallax,  $X_p$ , in  $X$  data units, is

$$X_p = k \times \text{range}(X) \times (Z - Z_{\min}) / \text{range}(Z) \quad (7)$$

The constant  $k$  is chosen as a conveniently small value such as .026.

Littlefield provided a color-anaglyph implementation in a statistical package over a decade ago by modifying Minitab (remember when source code was available). This modification allowed drawing of arbitrary glyphs and included color table control to handle light mixing. Red and green points overplotted as yellow as they should and color control allowed subtle depth-based shading. Color polaroids demonstrating that a third variable helps little in group discrimination were shown at the first ASA Graphics Exposition in 1982 and additional examples were published (Carr, Nicholson, Littlefield, and Hall 1986). While color anaglyph stereo is not the most desirable form of stereo, the anaglyph work demonstrated the simplicity of stereo plot production as long as statistical packages provide for control of color mixing (hint).

### **Side by Side Stereo Examples**

An article on graphics should have some graphics. Stereo production methods are incredibly diverse. For workstations, time-multiplexed methods that alternately route images to the left and right eyes are gaining in popularity. Side-by-side methods are common in non-electronic publications and so will be used here. Note that other disciplines often publish stereo images in color (*Editors' note: Maybe someday the newsletter will be able to afford to do this*). For example see the ray-traced image in Hodges (1992). The additional color-based depth cues are especially helpful when showing surfaces. The images here are monochrome point and line drawings and for brevity focus on geometry rather

than data analysis. The legends provide a brief description of the simple examples.

The examples are designed for parallel fusion. To fuse the images look at the left image with the left eye and the right image with the right eye from a distance of about a 50 centimeters. It may be helpful to separate the images with a card or to use an inexpensive magnifying stereopticon. These parallel fusion figures were produced using an S function (stereo.pairs) which is available via anonymous ftp from galaxy.gmu.edu (in subdirectory /submissions/eda).

### **Future Articles**

Separate articles will give background for more advanced examples and provide some discussion of statistical interpretation. The next article will focus on alpha blending and with luck will include a translucent stereo rendering of overlapping contours of a density estimate for 3-D data. While many practicing statisticians do not yet have the requisite hardware, computing environments are changing very rapidly. Soon a high percentage of statisticians will be able to study 3-D structure in the calm of non-rotating images. Future articles will discuss enhancement tools for the representation and study of data.

### **Additional References**

Many additional details are available in the literature, concerning stereoscopic resolution, hyperstereoscopy, and enhancements for statistical images, common production difficulties, etc. Some useful starting references are Wegman and Carr (1992), Hodges (1992), Carr and Nicholson (1988), Papathomas and Julesz (1988), Huber (1987) and Lipton (1982). Valyus (1962) provides one of the most detailed expositions and the fascinating work of Julesz (1971) produces considerable insight into visual processing via study of optical illusions encoded in random dot stereograms.

### **References**

- Carr, D. B. and W. L. Nicholson. (1988), "EX-PLOR4: A Program for Exploring Four-dimensional Data." *Dynamic Graphics for Statistics*, eds. W. S. Cleveland and M. E. McGill, pp. 309-329. Belmont, California: Wadsworth.
- Carr, D. B., W. L. Nicholson, R. J. Littlefield, and D. L. Hall. (1986), "Interactive Color Display Methods for Multivariate Data." *Statistical Image Processing and Graphics*, eds. E. J. Wegman and D. J. Depriest, pp. 215-250, New York: Marcel Dekker.
- Carr, D. B. and R. J. Littlefield. (1983), "Color Anaglyph Stereo Scatterplots - Construction Details." *Computer Science and Statistics: Proceedings*

of the 15th Symposium on the Interface. New York: North Holland Publishing Company.

Hodges, L. R. (1992), "Tutorial: Time-Multiplexed Stereoscopic Computer Graphics." *IEEE Computer Graphics & Applications*, pp. 20-30.

Huber, P. J. (1987), "Experiences With Three-Dimensional Scatterplots." *Journal of the American Statistical Association*, **82**(398), 448-453.

Julesz, B. (1971), *The Foundations of Cyclopean Perception*. Chicago: University of Chicago Press.

Lipton, L. (1982), *Foundations of the Stereo-Scopic Cinema, A Study in Depth*. New York: Van Nostrand Reinhold.

Newman, W. M. and R. F. Sproull. (1979), *Principles of Interactive Computer Graphics, Second Edition*. New York: McGraw-Hill.

Papathomas, T. V. and B. Julesz. (1988), "The Application of Depth Separation to the Display of Large Data Sets." *Dynamic Graphics for Statistics*, eds. W. S. Cleveland and M. E. McGill, pp. 353-377. Belmont, California: Wadsworth.

Valyus, N. A. (1962), *Stereoscopy*. New York: The Focal Press.

Wegman, E. J. and D. B. Carr. (1992), "Statistical Graphics and Visualization." Technical Report No. 84. Center for Computational Statistics, George Mason University, Fairfax, VA 22030.

Yeh, Y. Y. and Silverstein, L. D. (1990), "Limits of Fusion and Depth Judgement in Stereoscopic Color Displays." *Human Factors*, **32**, 45-60.

\*Minitab is a trademark of Minitab, Inc.  
S is a trademark of AT&T.

Daniel B. Carr  
George Mason University  
dcarr@galaxy.gmu.edu



## DEPARTMENTAL COMPUTING

# Not just hardware and software

A computer system has four major components—hardware, software, communications and people. The hardware and software angles of computing get discussed at length. In this column I will try to focus on issues in computing that have to do with getting a coordinated computing system going and maintained. The issues cut across types of hardware and software, operating systems and organizations. While my personal experience has been in developing a large (100+

computers) UNIX-based environment in academia, the issues surrounding departmental computing appear in business and government and across platforms. My department has networked PCs, Macs and multiple Vaxes in addition to the UNIX network. I'll try to keep the presentations as independent as I can of any particular computing platform. We'll discuss platform specifics when necessary for examples of general principles.

I confess to having several basic premises that guide my thinking about departmental systems.

- Departmental computing is the right level for support of statistical activities. Reliance on University or corporate level computing systems naturally leads to inadequate resources.
- A coordinated system in which users can send and receive mail, share files, share printers and other peripherals, share software installations, share documentation and share user support is a reasonable goal.
- Planning is possible. Despite budget uncertainties and power politics, planning can and should be done.
- Diversity happens. We can't stop it and we have to work to accept it. The best system for a fiscal person may not be the best system for a statistical scientist. Should all the statistical scientists in a department have the same platform? Let's discuss this. Let me know what you think. See the end of this column for my e-mail address.
- A network connection to the Internet is essential for proper access to the information needed to do one's job. Again, let me know what you think.
- Larger systems require full-time systems management. We'll start here (see the next section).

In future columns I hope to cover issues that hinder acceptance and development of departmental computing resources. Here are some of the issues I see as important (in no particular order):

- Acquisition of systems. Choosing platform(s), fund raising, planning, purchasing. Breaking in—how does one get started? What about solo machines?
- Administration. System management, user support.
- Maintenance (hardware, software, network gear) of modern departmental computing networks.
- Resource sharing. Coordinating funds as well as equipment.

- Specifications. What should the system be able to do. Performance, software, licensing, vendors.
- Diversity and innovation. Supporting diversity, planning and implementing innovation.
- The computer life-cycle. Trickle-down computing and opportunities for old equipment.
- Migration to new systems and new ways of working. Acceptance and training.
- Institutional barriers to progress. Technical, physical, financial and bureaucratic impediments.

As a first topic, let's consider the often neglected requirements for system management.

## **System Management**

Providing system management for a new or evolving system can be a real "catch-22". How can you justify system management if you don't yet have a complex running system? How can you build a complex running system without system management?

Most departments recruit a knowledgeable faculty member into the role of system administrator or manager. If things go well, the system grows and the faculty member gets burned out trying to do system management in his or her "spare time". Graduate students may be called in to assist or a scientific programmer may become involved. If things go very well, the system becomes complex and the case is made to hire full-time help. The faculty member is relieved of system management duties. The start-up phase from no system to a system requiring a system manager may last years.

Just as a department needs secretarial help to support the work of its faculty, a department with a complex modern computing system will need a full-time system manager to support the systems and its use by students, staff and faculty. Twenty systems is more than a part-time job and 100 systems is more than a full-time job. Check with other departments at your institution to see what staffing commitments they have made to system management.

Centralized support organizations may be available at your institution to relieve some system management duties, but these rarely compensate for having someone locally.

System managers are responsible for the day to day operation of the system. They perform backups, install software, install revisions, install new hardware, send hardware out for repair, make minor repairs in-house, prepare strategic plans, keep up-to-date on hardware and software trends, manage the department's software licenses, track down bugs in hardware and software,

monitor system usage, provide and maintain basic security for data, interact with the institution's networking organizations, respond to hardware, software and network failures, design and implement solutions to work-group problems, train users in the operation of the system, write documentation regarding system procedures, use of software, and local conditions, maintain the various operating system software subsystems, respond to changing external network conditions, simplify system use by providing new tools, negotiate with vendors, and generally make sure that everyone can use the computers to do their jobs.

Why would anyone think this is not a full-time, difficult job? It reminds me of the problems we have convincing some non-statisticians that statisticians are necessary. "Why can't I do it myself with a computer program", they ask. And we stand ready to tell them. We must be equally clear that system management should not be done as a hobby.

In planning for system management or making the case for system management there are additional issues. Who will provide backup for the system manager so that there is someone to turn to when the system manager is on vacation and things break? The obvious choice is the faculty member who recently retired from creating the system in the first place. A second issue involves hiring and retention. System managers can be hard to find. Senior system managers have large salaries and may have tired of the daily grind of responding to problems. Junior people may lack either the technical knowledge necessary to do a thorough job, or lack the maturity and responsibility that comes with experience. In any case, system managers are in very great demand, and rightly so. One needs to expect turnover in the system manager's position.

## **What's Next?**

I would very much like to hear from the readers. We'll all get bored hearing what I did, what I'm doing, and what I think is important. What issues have you faced in the construction of your departmental system? How were problems solved? What problems remain? Which issues that I plan to discuss are the most important to you? Which are irrelevant? What should I discuss that I haven't mentioned?

The best way for me to communicate with you is via electronic mail. My address is at the end of the column. I will collect comments, questions, criticisms and anecdotes and use them in future columns (with your prior permission, of course).

At the ASA meeting in San Francisco this Au-



gust, I will be hosting a roundtable luncheon on the topic of Departmental Computing. Please join us!

Michael Conlon  
University of Florida  
mconlon@stat.ufl.edu



## FROM OUR CHAIRS...

---

CONTINUED FROM PAGE 1

out in the 1976 copyright law: protection is available for "original works of authorship. . . fixed in any tangible medium of expression, now known or later developed, from which they can be perceived, reproduced, or otherwise communicated, either directly or with the aid of a machine or device" [3]. *Originality* does not imply that the work must be novel or unique, only that it originate with the author. Thus, for example, a computer program written as a class assignment is presumably a candidate for copyright even though many other students would produce similar programs to perform the same task.

Some things are specifically excluded from copyright, such as "any idea, procedure, process, system, method of operation, concept, principle, or discovery, regardless of the form in which it is described, explained, illustrated, or embodied. . ." [4]. I would interpret this to mean that one could copyright a program that computes a singular value decomposition, for example, but one could not copyright the singular value decomposition itself.

The owner of a copyright has authority to sell or otherwise distribute copies of the program. If I buy a copy of *Systat*, for example, I have whatever rights are laid out in the agreement between me and the copyright owner. The license for *Systat* allows me to use the program on one CPU for its intended purpose of statistical analysis. I do not have the right to reverse engineer the software to find out how it works. I can sell my copy, but the license agreement will apply to the buyer as well. If I were to use *Systat* with my own data, output will be produced in a format that may be unique to *Systat*; for example, I might get a graph that could not be produced in any other way. Copyright law specifically excludes forms such as bank checks or date books from copyright, so presumably the graph produced in *Systat* would be owned by me, not by Systat, Inc.

Whereas a copyright gives its holder exclusive use of a creation, a patent gives its holder the right to claim royalties to an *invention*. Patents are generally not intended to limit use of an invention, only to require payment for use. For example, a patent could be issued for an inven-

tion such as a cassette tape. Many manufacturers can make cassette tapes, and they would pay a royalty to the patent holder.

Under U.S. patent law, it appears that a mathematical algorithm, defined as a "procedure for solving a given type of mathematical problem," [5] is not patentable, while a particular *application* of an algorithm may be patentable. For example, a patent would not be issued for a factor analysis, but a patent could be issued for a specific implementation in computer code that does factor analysis. Different computer code using the same computational algorithm may not be in violation of the patent.

According to Greg Aharonian, over 9000 software patents have been issued, with over 1300 issued in 1992 alone, mostly to a few large companies. Many of the patents are issued for code that serves a specific purpose. For example, IBM was issued patents for a "Computer user interface with window title bar icons" and for a "Method for deleting a marked portion of a structured document." Again according to Aharonian, patent infringement lawsuits are rare. He believes that most patents are obtained as a defensive measure to protect against future law suits, not to prosecute others.

After this bit of research, I guess I now understand a program that uses a patented algorithm for computing a graph: the statement is advertising hyperbole. Patenting is no substitute for peer review, and if the graph is of use, and I understand it, I am free to write my own code to compute it.

## References

- [1] Wadley, James B. (1986), "An Introduction to copyright protection of computer programs and the semiconductor chip protection act of 1984," *Journal of the Kansas Bar Association*, July, 185-190.
- [2] United States Constitution, Article 1, Section 8.
- [3] 17 U.S.C. 102(a).
- [4] 17 U.S.C. 102(b).
- [5] 1106 OG 6

Sanford Weisberg  
Chair, Statistical Computing Section  
sandy@umnstat.stat.umn.edu

## Statistical Graphics

Hello. I know you're busy, so I won't run on too long here at the keyboard. I just wanted to let you know that the Statistical Graphics Section has many activities planned for 1993 and that you are invited to take part. The easiest way to participate is to come to the Joint Sta-

tistical Meetings in San Francisco; it will be a prime focus for graphics events. I'd like to tell you about a few of the things we have planned.

Research contributions are always an important component of the meetings, and David Scott has put together a strong set of invited paper sessions—the list appears on page 21. If you would enjoy stimulating conversation on graphical topics, try one of the luncheon roundtables. Sally Morton has been working to catalog, distribute, and display our growing collection of graphics video tapes. The newest and best of these will be shown in a screening room at the meetings. Another feature of this year's graphics program is a continuing education course that we are sponsoring: Bill Cleveland on "Visualizing Data".

Every other year the graphics section sponsors a poster session where any interested person or group is invited to analyze a common dataset. As a participant, it's always stimulating to be able to discuss a project with others who have looked carefully at the same data; as a spectator, it is interesting to compare and contrast the different approaches to the data. This year, David Coleman has provided two datasets for the session: one on nutritional content of breakfast cereals and another one a time-series with surprises. Make sure you show up to see the posters this year, and make a note to do the analysis and contribute a poster of your own for the 1995 meetings.

Keep on the lookout for the student poster competition that we sponsor with the Statistical Education section. The work of these school children is always refreshing and often displays data in a very creative way.

Under the new ASA constitution, the sections have much more autonomy to provide services for their members. The statistical graphics section can accomplish a lot, but we need to know what activities would be of most interest to you. Let me or any of the other section officers know if you have a good idea that the section can pursue. Also, please get involved—we are always looking for volunteers who would like to play a role in section activities.

Rick Becker  
*Chair, Statistical Graphics Section*  
 rab@research.att.com



## SAXPY, GAXPY, ...

CONTINUED FROM PAGE 1

Nominally, around 71  $100 \times 100$  Cholesky decompositions can be calculated each second at 25 MFLOPS,

but in actuality only 11 per second are possible at 4.0 MFLOPS. The performance of other machines is given in Table 1, selected from 671 entries in Dongarra (1993).

The  $100 \times 100$  LINPACK benchmark uses standard compiler options. For example, the `-dalign` options on SUNs, which forces the alignment of each double precision datum on an 8-byte boundary, can yield faster speeds. Significantly faster speeds can be achieved if the code itself is modified. These attempts are recorded against the  $1000 \times 1000$  LINPACK benchmark in Table 1 (col.2). Considerable ingenuity has gone into optimizing  $1000 \times 1000$  LINPACK performance, and it is apparent from Table 1 that a two to three-fold increase in performance or better is common, which then approaches nominal performance (col.3).

Table 1: LINPACK (Cholesky decomposition) performance in MFLOPS for  $100 \times 100$  and  $1000 \times 1000$  matrices

Computer	100	1000	CPU
CRAY Y-MP C90 (16 proc.)	479	9715	15238
CRAY Y-MP C90 (1 proc.)	387	874	952
NEC SX-3/14R (1 proc.)	368	5199	6400
DEC 10000-610 (200 MHz)	43	112	200
HP 9000/735 (99 MHz)	41	107	198
IBM 6000-560 (50 MHz)	31	84	100
DEC 3000-500 (150 MHz)	30	80	150
SGI R4000 (50 MHz)	16	32	50
IBM 6000-530 (25MHz)	15	42	50
SUN SPARC 10/30 (36MHz)	9.3		
SUN SPARC 2	4.0		
Apple MAC Quadra 700	1.4		
NeXTCube	1.4		
Compaq Deskpro 486	1.3		
nCube 2 (1024 proc.)		258	2409
nCube 2 (32 proc.)		46	75
nCube 2 (1 proc.)	0.78	2.02	2.35
Apple Mac IIfx	0.37		
Apple Mac IIsi	0.19		
Compaq 386/20 w/387	0.16		
microVAX II	0.12		
IBM AT w/80287	0.012		

The initial goal of this article is to elaborate further on the gap between nominal and actual performance. The key issue is memory management, discussed in the next section. The efficient use of memory is illustrated for the apparently straightforward task of matrix multiplication in the *Matrix Multiplication* section. The details are illuminating, and include description of the "elementary operations" of vector and matrix arithmetic, *saxpy* and *gaxpy*. Two other, very effective techniques for speeding computations are block algorithms and loop unrolling, discussed in the *Performance Issues* section.

In programming the LINPACK subroutines for linear algebra, Dongarra et al. (1979) noted that efficient code could be achieved through a modular approach. That is, an efficient and numerically accurate algorithm for, say, the QR decomposition or singular value decomposition, is implemented through a series of calls to a standard library of basic subroutines for vector and matrix manipulation, the BLAS (Basic Linear Algebra Subroutines). The same is true for LAPACK (Demmel et al. (1991)), but in a more sophisticated sense, reflecting the memory management issues discussed in this article. The BLAS encompass many operations, from the most simple such as vector copy or vector inner product (BLAS level 1, used in LINPACK), to matrix multiplication and the solution of triangular systems of linear equations (BLAS level 3, used in LAPACK).

The portable FORTRAN public-domain implementation of BLAS has been improved by several hardware vendors, including IBM and Silicon Graphics Inc., and can include some assembly coding. Dyad Software markets an assembly code version of BLAS for DOS. The BLAS routines are useful very generally: While it is quite easy to code a routine to handle such tasks as copying an array to another array, computing the inner product of two vectors, or matrix multiplication, BLAS provides these routines also. A tabular summary of the BLAS routines is given on page ???. They are carefully coded for numerical accuracy and speed, and can often be called from C.

LINPACK, LAPACK, and the portable implementation of BLAS can be obtained by "mail order" from netlib, using e-mail, ftp, or the X-based graphical interface program xnetlib. The xnetlib software is particularly sophisticated, as it includes the library contents hierarchically by name and by subject classification, includes various search options, and follows dependencies to facilitate downloading all relevant files. Calls to the BLAS library can then be substituted for the BLAS routines distributed with individual LINPACK routines; LAPACK distribution keeps the two separate. The xnetlib software can be obtained by ftp from netlib.

This section concludes with a caveat. In a user application, such as graphics, combinatorial optimization, or data base manipulation, even if the code is built on the BLAS library, the usage of memory may be substantially more non-local, and include much more indirect addressing, than is found in typical linear algebra applications. Then the gap between nominal and actual performance is wider still. For example, the IBM RISC 6000 architecture provides exceptional performance on the LINPACK benchmarks, especially the  $1000 \times 1000$

benchmark which utilizes IBM's ESSL library implementation of BLAS. On the other hand, on a molecular dynamics benchmark, the price/performance ratio of a DEC, SGI, or HP workstation can be better than that of the IBM.

## **Memory Management**

The fast cycle time of a CPU can only be utilized if the data and instructions can be placed in CPU quickly enough. The first concern is to configure the system with sufficient RAM, so that paging between RAM and disk is eliminated. As 16MB of RAM on a workstation is now standard, and 32MB is common, this will be taken for granted. The second concern is data transfer between RAM and CPU. Whether the memory and CPU boards connect by the system bus or a dedicated bus, the transfer rate is slow compared to the CPU speed. Thus some memory, so-called primary cache memory, is typically placed on the CPU board and integrated tightly with the CPU.

Figure 1 Typical Architecture

Figure 2 Simplified Architecture

Typical cache memory may comprise 8KB for data and 8KB for instructions. (A greater amount of secondary cache memory, 1 MB say, may be available, but access speeds to it are not comparable to those for primary cache access.) Figure 1 shows a typical configuration; for our purposes the simplified version, Figure 2, suffices. The size of the primary cache (from now on, simply 'cache') is small. For example, the data for the  $100 \times 100$  LINPACK benchmark greatly exceeds the 1024 double precision number ( $32 \times 32$  matrix) capacity of an 8KB cache. Larger primary caches are included in some recently introduced workstations, but the amount is still nowhere the size of a modest application.

An efficient program is one that minimizes the transfer of data between RAM and cache, through maximizing the number of floating point operations for each copy

between RAM and cache. Vector-vector and matrix-vector routines in levels 1 and 2 BLAS respectively perform  $O(n)$  operations on an  $O(n)$  amount of data, but the matrix-matrix routines in level 3 BLAS perform  $O(n^3)$  floating point operations on an  $O(n^2)$  amount of data.

### Matrix Multiplication; saxpy and gaxpy

Consider the  $3 \times 3$  by  $3 \times 3$  matrix multiplication

$$AB = C \quad (8)$$

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix} \begin{pmatrix} .1 & .2 & .3 \\ .4 & .5 & .6 \\ .7 & .8 & .9 \end{pmatrix} = C \quad (9)$$

The conventional approach is to compute the elements of  $C$  one by one, by the dot (inner) products

$$c_{ij} = \sum_k a_{ik} b_{kj}. \quad (10)$$

In more detail, first initialize  $c_{ij} = 0$  and then sequentially add  $a_{ik} \times b_{kj}$  to the current  $c_{ij}$ , for  $k = 1, \dots, 3$ . For  $c_{11}$  this may be depicted

$$\begin{pmatrix} \times & \times & \times \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \end{pmatrix} \begin{pmatrix} \times & \cdot & \cdot \\ \times & \cdot & \cdot \\ \times & \cdot & \cdot \end{pmatrix} = \begin{pmatrix} \times & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \end{pmatrix}. \quad (11)$$

Another approach is, after initializing the elements of  $C$  to 0, to multiply the first column of  $A$  by  $b_{11}$  and add the result to the current first column of  $C$ , then multiply the second column of  $A$  by  $b_{21}$  and add the result to the current first column of  $C$ , then multiply the third column of  $A$  by  $b_{31}$  and add the result to the current first column of  $C$  to obtain the final first column of  $C$ . The first step is depicted

$$\begin{pmatrix} \times & \cdot & \cdot \\ \times & \cdot & \cdot \\ \times & \cdot & \cdot \end{pmatrix} \begin{pmatrix} \times & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \end{pmatrix} = \begin{pmatrix} \circ & \cdot & \cdot \\ \circ & \cdot & \cdot \\ \circ & \cdot & \cdot \end{pmatrix}, \quad (12)$$

where  $\circ$  indicates that the element of  $C$  is partially computed after this step, instead of fully computed ( $\times$ ). The three steps may be depicted

$$\begin{pmatrix} \times & \star & \diamond \\ \times & \star & \diamond \\ \times & \star & \diamond \end{pmatrix} \begin{pmatrix} \times & \cdot & \cdot \\ \star & \cdot & \cdot \\ \diamond & \cdot & \cdot \end{pmatrix} = \begin{pmatrix} \times & \cdot & \cdot \\ \times & \cdot & \cdot \\ \times & \cdot & \cdot \end{pmatrix}. \quad (13)$$

Each of the three steps is called a *saxpy*, for “scalar times (vector)  $x$  plus  $y$ .” Algebraically, a saxpy operation is

$$\alpha x + y \mapsto y. \quad (14)$$

The first column of  $C$  is constructed from three saxpy operations. Together they are known as a *gaxpy*, or

generalized saxpy,

$$Ax + y \mapsto y. \quad (15)$$

(In the saxpy the vector  $x$  is a column of  $A$ , in the gaxpy the vector  $x$  is the first column of  $B$ , which contains the scalars  $\alpha$ .) In their Second Edition (1989), Golub and van Loan strongly emphasize saxpy and gaxpy, and write of saxpy as a fifth elementary operation, along with addition, subtraction, multiplication, and division. They point to the increasing role that saxpy and gaxpy has played in numerical linear algebra in the past 10-15 years, beginning with the use of saxpy in LINPACK (Dongarra et al. 1979).

A third approach to multiplying two matrices is by an outer product, building the matrix in three steps,

$$a_1 b_1^t + a_2 b_2^t + a_3 b_3^t = C, \quad (16)$$

where  $a_j$  is the  $j$ th column of  $A$  and  $b_j^t$  is the  $j$ th row of  $B$ . The first outer product, which can be viewed as 3 saxpy’s, is depicted

$$\begin{pmatrix} \times & \cdot & \cdot \\ \times & \cdot & \cdot \\ \times & \cdot & \cdot \end{pmatrix} \begin{pmatrix} \times & \times & \times \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \end{pmatrix} = \begin{pmatrix} \circ & \circ & \circ \\ \circ & \circ & \circ \\ \circ & \circ & \circ \end{pmatrix}. \quad (17)$$

The dot product, saxpy-gaxpy, and outer product algorithms are the principal alternatives from among  $3! = 6$  arrangements of the computation, eq. (??), as three nested loops indexed by the subscripts  $i$ ,  $j$ , and  $k$  in every order. Each saxpy in eq. (??) and eq. (??) is a column saxpy, because the  $x$  is a column vector. The 6 algorithms comprise 2 algorithms each based on dot products, 2 on column saxpy’s, and 2 on row saxpy’s.

### Performance Issues

Although each of the six algorithms for matrix multiplication involves precisely  $3^3$  multiplications and  $2 \times 3^2$  additions, they differ markedly in the manner in which data is transferred to and from memory. For the efficient use of cache and also in vector processing, we prefer algorithms that both minimize data transfer and operate on vectors of elements that are contiguous in memory. Given that matrices are stored by column, each dot product accesses a vector of contiguous entries and a vector of non-contiguous entries. Column saxpy’s access two vectors of contiguous entries ( $x$  and  $y$ , and a scalar), and are preferable to row saxpy’s, which access two vectors of *non-contiguous* entries. When three column saxpy’s are arranged as a gaxpy, the  $x$  differ but  $y$  does not. When three column saxpy’s are arranged as an outer product, the  $y$  differ, but  $x$  does not. The former (gaxpy) is preferable, as the  $x$  need only be loaded from memory, whereas in computing the outer product

the  $y$  must be loaded from memory, incremented, and replaced in memory.

These memory management issues are known as unit stride and vector touches. Two additional topics, block algorithms and loop unrolling, are discussed shortly. In summary:

**Unit stride.** The stride of a vector is the distance between successive elements of the vector in memory. Unit stride is preferred. A column saxpy has unit stride, a row saxpy does not.

**Vector touches.** A vector touch is the loading of a vector of data, to or from RAM, cache or a vector processor. A gaxpy requires approximately half the vector touches of an outer product.

**Block algorithms.** When a computation is divided into blocks, a greater fraction of the computation can be performed using only data transfer between cache and the CPU, without the use of main memory.

**Loop unrolling.** Utilization of data currently in memory is enhanced when several successive iterations, 4 say, of a loop are written out explicitly (The number of iterations of the loop is decreased by the same factor.)

Interestingly, very similar issues arise in the design of algorithms for parallel processing. Much of the excitement surrounding coding for parallel processing can be found in efficient coding for single processors. It is widely known that it is essential to structure code carefully if it is to be vectorized for a multiple processor, or vector, computer. But similar concerns apply to single processor architecture, as vector operations may be used to move data into and out of cache.

### Block algorithms

In general the full  $O(n^3)$  operations to multiply two  $n \times n$  matrices cannot be accomplished with all three matrices,  $A$ ,  $B$ , and  $C$  entirely in cache, at least for large matrices. However, it is possible to substantially reduce the number of memory loads and stores through the use of operations on submatrices, that is, block algorithms. The floating point operations are unchanged in number, but their order is rearranged.

Let  $A_{IK}$ ,  $B_{KJ}$  and  $C_{IJ}$  denote the  $IK$ th,  $KJ$ th and  $IJ$ th block of  $A$ ,  $B$  and  $C$  respectively. Then

$$C_{IJ} = \sum_K A_{IK} B_{KJ}. \quad (18)$$

Apart from the last row and column of submatrices, each submatrix  $A_{IK}$  of  $A$ , say, will have the same number of rows and of columns, the *block size*. The block size is

often 32, but can be 64, for example for operations on triangular matrices.

As is the case for ordinary matrix multiplication, block matrix multiplication can be arranged in  $3! = 6$  ways. The `block_gaxpy` approach is

$$C_{IJ} + A_{IK} B_{KJ} \mapsto C_{IJ} \quad (19)$$

where the subscript  $I$  varies fastest,  $K$  second, and  $J$  slowest. When each matrix multiplication  $A_{IK} B_{KJ}$  is computed using a gaxpy, there are six subscripts, which vary in the order  $i$  (fastest),  $k$ ,  $j$ ,  $I$ ,  $K$ , to  $J$  (slowest). The cache hit rate is particularly high if, for each  $I$ ,  $J$ , and  $K$ , the entire matrices  $A_{IK}$ ,  $B_{KJ}$ , and  $C_{IJ}$  are contained in cache. If each matrix is  $16 \times 16$ , 6144 bytes of cache are used. In fact, though, a block size of 32 is feasible with an 8KB cache: For given  $I$ ,  $J$ , and  $K$ , and using gaxpy multiplication, the matrix  $A_{IK}$  is used repeatedly, once for each column of  $B_{KJ}$ . The matrix  $C_{IJ}$  is used repeatedly, but one column at a time, and the elements of  $B_{KJ}$  are referenced once only. Therefore cache need contain only the entire matrix  $A_{IK}$  and a column of  $C_{IJ}$ .

### Loop unrolling

Panzeira (1992) describes the use of loop unrolling to further reduce the ratio of memory accesses to floating point operations, Consider the gaxpy

```
do j = 1, 32, 1; do k = 1, 32, 1
  t = bkj
  do i = 1, 32, 1
    cij = cij + aik * t
  end do
end do; end do
```

The inner loop is unrolled four times as follows:

```
do j = 1, 32, 1; do k = 1, 32, 1
  t = bkj
  do i = 1, 29, 4
    ci+0 j = ci+0 j + ai+0 k t
    ci+1 j = ci+1 j + ai+1 k t
    ci+2 j = ci+2 j + ai+2 k t
    ci+3 j = ci+3 j + ai+3 k t
  end do
end do; end do
```

Inner loop unrolling reduces the looping overhead but not memory accesses. Thus the performance gains are

not significant. Panzeira reports that for  $50 \times 50$  matrices actual gaxpy performance is 31% of theoretical (CPU) speed. With inner loop unrolling efficiency is 33% of theoretical speed. For matrix multiplication with matrices of 250 or more rows and columns, efficiency increases from 24% to 25%. With blocking, the increase in efficiency is from 31% to 33% for matrices of all sizes.

Middle and outer loop unrolling is much more successful. Unrolling the inner loop twice, the middle loop four times, and the outer loop twice, the code becomes:

```
do j = 1, 31, 2; do k = 1, 29, 4
  t00 = b_{k+0 j}; t10 = b_{k+1 j}
  t20 = b_{k+2 j}; t30 = b_{k+3 j}
  t01 = b_{k+0 j+1}; t11 = b_{k+1 j+1}
  t21 = b_{k+2 j+1}; t31 = b_{k+3 j+1}
  do i = 1, 31, 2
    c_{i j} = c_{i j} + a_{i k} t_{00} + a_{i k+1} t_{10} + a_{i k+2} t_{20} +
      a_{i k+3} t_{30}
    c_{i j+1} = c_{i j+1} + a_{i k} t_{01} + a_{i k+1} t_{11} + a_{i k+2} t_{21} +
      a_{i k+3} t_{31}
    c_{i+1 j} = c_{i+1 j} + a_{i+1 k} t_{00} + a_{i+1 k+1} t_{10} +
      a_{i+1 k+2} t_{20} + a_{i+1 k+3} t_{30}
    c_{i+1 j+1} = c_{i+1 j+1} + a_{i+1 k} t_{01} + a_{i+1 k+1} t_{11} +
      a_{i+1 k+2} t_{21} + a_{i+1 k+3} t_{31}
  end do
end do; end do
```

Efficiency is now around 70% and 51% for the  $50 \times 50$  and  $250+ \times 250+$  matrices respectively, and 69% and 65% respectively with blocking. Thus using loop unrolling, actual speeds can be at least doubled, e.g. from 31% to 65% efficiency. Automatic innermost loop unrolling has been implemented in compilers, e.g. Silicon Graphics FORTRAN compiler (IRIX release 3.3 and above). More specialized compiler technology is needed to implement other types of automatic loop unrolling. For example, the SGI Power FORTRAN Analyzer implements automatic outermost loop unrolling.

### Software for Linear Algebra: An Overview

Software for linear algebra includes libraries of high level routines, such as LINPACK and LAPACK, and also lower level routines, the BLAS library. CORE-MATH and netlib are repositories for sets of routines, such as EISPACK, LINPACK, LAPACK, the generic version of BLAS, optimizers, and nonlinear equation

solvers. Electronic mail requests can be sent to netlib netlib@ornl.gov, with a message such as

```
send index
send index from LAPACK
send index from BLAS
send index from benchmark
```

Alternatively software can be downloaded interactively using ftp to research.att.com (login as user ftp), or the utility xnetlib.

BLAS routines are in sets of 2 or 4, with suffices S and D for single and double precision real data, and C and Z for single and double precision complex data. Tables 2-4 show the contents of BLAS at level 1 (vector-vector operations), level 2 (matrix-vector operations), and level 3 (matrix-matrix operations) respectively. These summaries are intended to show what routine programming tasks can be circumvented using the BLAS library.

copy	scopy	$\mathbf{x} \mapsto \mathbf{y}$
swap	sswap	$\mathbf{x} \leftrightarrow \mathbf{y}$
scale	sscal	$\alpha \mathbf{x} \mapsto \mathbf{x}$
saxpy	saxpy	$\alpha \mathbf{x} + \mathbf{y} \mapsto \mathbf{x}$
dot product	sdot	$\mathbf{x}^t \mathbf{y}$
Euclidean norm	snrm2	$\sqrt{\mathbf{x}^t \mathbf{x}}$
$L_1$ norm	sasum	$\sum  x_i $
maximum index	isamax	index of max
planar rotations	drot	planar rotations

multiplication (gaxpy)	dgemv	$\alpha A \mathbf{x} + \beta \mathbf{y} \mapsto \mathbf{y}$
triangular matrices	dtrmv	$T \mathbf{x} \mapsto \mathbf{x}$
back substitution	dtrsv	solve $T \mathbf{x} = \mathbf{b}$
rank 1 matrix update	dger	$A + \alpha \mathbf{x} \mathbf{y}^t \mapsto A$
symmetric rank 1 update	dsyr	$A + \alpha \mathbf{x} \mathbf{x}^t \mapsto A$
symmetric rank 2 update	dsyr2	$A + \alpha \mathbf{x} \mathbf{y}^t + \alpha \mathbf{y} \mathbf{x}^t \mapsto A$

Matrix Multiplication		
general matrices	dgemm	$\alpha AB + \beta C \mapsto C$
one symmetric matrix	dsymm	$\alpha AB + \beta C \mapsto C$
one triangular matrix	dtrmm	$\alpha TB + \beta C \mapsto C$
Solving Triangular Systems		
back substitution	dtrsm	solve $T X = \alpha B$
Rank $k$ and Rank $2k$ Matrix Updates		
rank $k$ matrix update	dsyrk	$\beta C + \alpha A A^t \mapsto C$
symmetric rank $2k$ update	dsyr2k	$\beta C + \alpha A B^t + \alpha B A^t \mapsto C$

The BLAS routines are carefully coded, sometimes in assembly code, and, particularly for BLAS level 3, may utilize loop unrolling, blocking, and parallel processing — as well as saxpy's and gaxpy's — to take advantage of modern machine architecture. (On rare occasions the use of higher level BLAS routines is not optimal, e.g. to invert a  $p \times p$  triangular matrix, `dtrsm` from BLAS is inferior to `dtrtri` from LAPACK.) Original sources for BLAS are Lawson et al. (1979) and Dongarra et al. (1988, 1989). Coleman and van Loan (1988) give examples of the use of the BLAS, emphasizing level 1. Anderson et al. (1992, Appendix C) give a summary of the BLAS. Other summaries can be found in UNIX man pages distributed with the software. The summary given here includes only some of the routine names, but other routines are closely related and their names can be found from the references, or the man pages (which may combine descriptions of several routines on a single page).

LINPACK uses only BLAS level 1 routines, but can be re-coded quite easily to take advantage of BLAS level 2. However, performance can then diminish as the size of the problem increases. LAPACK uses the more efficient BLAS level 3 routines, although some tuning of LAPACK is possible for optimal performance. The LAPACK auxiliary enquiry function `ILAENV`, provides a lookup table of parameters, such as block size, for each primary function. The generic `ILAENV` in the LAPACK distribution may be modified for better performance on individual systems, but, as LAPACK itself is new, departures from the generic version are uncommon. This is perhaps an opportunity for an excellent and worthwhile class exercise in experimental design. See Demmel et al. (1991), Section 4.

The necessary BLAS routines are included when LINPACK routines are sent from netlib; LAPACK requires that the BLAS routines have been installed separately. LAPACK is intended to be a replacement to LINPACK. LAPACK, like BLAS and unlike LINPACK and EISPACK, consistently includes routines for all four data types. A little of LINPACK's functionality is absent from LAPACK, for example, the LINPACK routines `DCHUD` and `DCHDD` to update the Cholesky factor of a symmetric matrix when a rank 1 matrix is added or subtracted; a more general alternative is TOMS algorithm 686, Reichel and Gragg (1990). The LAPACK release notes (`send release_notes` from `lapack`) include practical guidance on using LAPACK on a number of machines, e.g. by suggesting compiler options.

### Acknowledgements

I wish to thank Jim Rosenberger for suggesting I write this article, Jack Dongarra for advice on the use of

LAPACK, and Olivier Schreiber at Silicon Graphics Inc. for information on their work with the LINPACK benchmarks and compilers. This work is based in part on Goodall (1993), and was supported by NSF Grant DMS-9208656.

### References

- Anderson, E., Bai, Z., Bischof, C., Demmel, J., Dongarra, J., Du Croz, J., Greenbaum, A., Hammarling, S., McKenney, A., Ostrouchov, S., and Sorensen, D. (1992). *LAPACK Users' Guide*. Philadelphia, PA: SIAM.
- Coleman, T.F. and Van Loan, C. (1988). *Handbook for Matrix Computations*. Philadelphia, PA: SIAM.
- Demmel, J.W., Dongarra, J.J., and Kahan, W. (1991). "On designing portable high performance numerical libraries. LAPACK working notes no. 39. University of Tennessee. (to netlib: `send lawn39` from `lapack`)
- Dongarra, J.J. (1993). "Performance of various computers using standard linear equations software." Technical report CS-89-85, University of Tennessee and Oak Ridge National Laboratory. 1 March 1993. (to netlib: `send performance` from `benchmark`)
- Dongarra, J.J., Moler, C.B., Bunch, J.R., and Stewart, G.W. (1979). *LINPACK Users' Guide*. Philadelphia, PA: SIAM.
- Dongarra, J.J., DuCroz, J., Hammarling, S., and Hanson, R.J. (1988). "An extended set of Fortran basic linear algebra subprograms." *ACM Trans. on Math. Software*, **14**, 1-17. (See also pp. 18-32.)
- Dongarra, J.J., DuCroz, J., Duff, I.S., and Hammarling, S. (1989). "A set of level 3 basic algebra subprograms." *ACM Trans. on Math. Software*
- Golub, G.H. and van Loan, C.F. (1983, 1989). *Matrix Computations*. Baltimore, MD: Johns Hopkins University Press. Second edition, 1989.
- Goodall, C.R. (1993), "Computation Using the QR Decomposition." In: *Handbook in Statistics, Volume 9: Statistical Computing* (C.R. Rao, ed.). Amsterdam, NL: Elsevier/North-Holland.
- Lawson, C.L., Hanson, R.J., Kincaid, D., and Krough, F. (1979) "Basic linear algebra subprograms for Fortran usage." *ACM Trans. Math. Software*, **5**, 308-325.
- Panzeria, (1992). "Nested loops optimization." SGI technical report.
- Reichel, L. and Gragg, W.B. (1990). "Algorithm 686: Fortran subroutines for updating the QR decomposition," *ACM Trans. Math. Software*, **16**, 369-377.

Colin Goodall  
*The Pennsylvania State University*  
 colin@stat.psu.edu

## **Gopher and other resource discovery tools**

The tremendous growth of services and information available over the Internet has prompted the development of several tools that ease the search for these resources. Recently, I described *archie* in this column. *Archie* is a tool that maintains and searches an inventory of information available at FTP sites. One disadvantage of *archie* is that one has to know at least a substring of the file name or directory in which the information is located to find it. Recently, a reader of this column brought *gopher* to my attention. I used *archie* to find *gopher*, installed a *gopher*-client on my machine (*xgopher* in my case), and then I used *gopher* to find and retrieve information about itself and other related tools.

I found that most of the Internet services that I described in previous issues of this newsletter are available through *gopher*. Some of these are: the statlib archive, the netlib archive, whois servers, *archie*, and many more. Some new resources that I discovered with *gopher* include: other white page servers, databases (such as census data, weather data, and geographical data), library catalogs of many universities around the world, electronic books (such as the King James Bible, Moby Dick, and Peter Pan), a wavelet archive, the AMS combined membership list, and many more.

*Gopher* is really more difficult to describe than it is to use, but I will try anyway. *Gopher* is a menu driven interface that allows browsing through Internet resources regardless of their type and without having to worry about Internet addresses. By selecting menu items one can obtain programs, documents, pictures, and even sounds, look up addresses, do keyword searches, and use many other quite different Internet resources. *Gopher* originated as a distributed campus information service at the University of Minnesota, the "Golden Gophers," and because it evolved to "go fer" things on the Internet, the name was coined. A recent book that contains a whole chapter on *gopher* as well as a wealth of other information about the Internet is Kroll (1992). The best way to learn about *gopher* is to find a *gopher*-client and start using it. Telnet to *consultant.micro.umn.edu*, login as *gopher* and "go fer" it! This telnet-accessible anonymous *gopher*-client can help you obtain the software required to install your own *gopher*-client. There are now versions available for just about any hardware and operating system combination. You only need to be

connected to the Internet.

The collection of all information and services available through *gopher* is called gopherspace. Gopherspace is still growing but is already vast with hundreds of *gopher*-servers worldwide. Once you are connected to one *gopher*-server you can reach other *gopher*-servers by selecting the menu item "Other Gopher and Information Servers" and then selecting the server you want. Most *gopher*-servers are very general and have similar menu entries, while others can be focused on particular information available at a given site.

### ***Gopher is a menu driven interface that allows browsing through Internet resources...***

*Gopher* is great for browsing through information on the Internet, especially if you have some idea which *gopher*-servers have information that interests you. Because gopherspace is now so vast, browsing is not an efficient way to look for information. A recent answer to this problem, available through *gopher* on most *gopher*-servers, is *veronica* (very easy rodent-oriented net-wide index to computerized archives). It was developed at the University of Nevada and provides a keyword search of most *gopher*-server menus in the entire gopherspace. Searching the menus with *veronica* is not a replacement for browsing because the menus are not always very descriptive of the contents and represent only one of many interpretations of the contents. For example, the same information could be listed as "geological databases," "rock data," or even "Jack's favorite data" because Jack is a geologist.

WAIS (Wide Area Information Server) is another tool for finding resources on the Internet. Unlike guided browsing through Internet resources with *gopher* or *veronica* searches of *gopher* menus, WAIS provides a facility for keyword search of documents over the Internet. If the Internet is a collection of libraries of documents, WAIS is an attempt to automate the interaction between a library patron and a reference librarian. WAIS does not search through the actual documents, rather it searches through indexes built from the documents. Only those libraries that have a WAIS index can be searched. The uniqueness and strength of WAIS is that it can rate the relevance of documents in terms of similarity to other user specified documents. You can get to a WAIS server with *gopher*, but there are some advantages to running your own WAIS client program. Many are available from *think.com*, because much of the development of WAIS is done at Thinking Machines. Of course, you can also use *gopher* or *archie* to find them.



## Reference

Kroll, Ed (1992), *The Whole Internet User's Guide and Catalog*, O'Reilly & Associates.

## The Last Word . . .

Let me end this column by saying that it is time for me to sign off and let someone else continue in my place. While writing this column for the past three years, I have learned much about computer communication and various Internet services. Please contact me `ost@azure.epm.ornl.gov` or the computing editor `jlr@stat.psu.edu` of this newsletter if you are interested in writing or editing a similar column or simply contributing to it. Also, please send any comments or suggestions regarding this column to me and I will forward them to the next column editor.

George Ostrouchov  
Oak Ridge National Laboratory  
`ost@azure.epm.ornl.gov`

## Getting your Facts from FAQs

This is the first of an occasional series about navigating the Internet. Since there are entire books devoted to this subject (see above), I won't even try to be comprehensive. Instead, I'll provide some clues on Internet snooping for the statistician.

The Internet and bitnet mailing lists and netnews system is an amazing resource. It contains technical material and discussions about all sorts of other things. At my university the campus electronic bulletin board system contains a wide collection of the Internet bitnet mailing lists (including things like the s-news mailing list), an electronic newswire, netnews, and a gaggle of local discussion boards. I find it very useful to be able to watch netnews and the Internet mailing lists without getting my personal mailbox even more clogged than it already is. I'll use the generic term "bboard" to refer to both netnews and the various Internet mailing lists. I regularly read a large number of bboards (I won't admit to how many). Four of them are devoted to statistics. The three high volume bboards are `netnews.sci.math.stat`, the Internet S-news mailing list, and the Bitnet STAT-L mailing list. Many readers will already know about these two mailing lists, but if you do not, here is how to subscribe. To subscribe to S-news send a message to `s-news-request@utstat.toronto.edu`. A human will either add you to the list or suggest a local

re-distribution point for the list. I strongly suggest that you find a local feed for the S-news mailing list, or set one up yourself. To subscribe to STAT-L, send a one line e-mail message containing just the line

```
SUBSCRIBE STAT-L
```

to `LISTSERV@MCGILL.BITNET`. My fourth statistical related bboard is the `xlisp-stat-news` mailing list. There is also a fledgling mailing list for those interested in Bayesian statistics. If you are interested in that, contact me via e-mail.

Finding your way around netnews and the mailing lists can be daunting both for novices and experienced users alike. One way of making quick headway is to read the FAQs (Frequently Asked Questions) postings.

Many bboards have a periodic posting (often monthly) containing the FAQs. These posts are often long messages containing answers to many questions which are frequently discussed on the bboard. For example there are several forums for discussing the  $\text{\TeX}$  typesetting language. One of the more common ones is `netnews.comp.text.tex`. The FAQ posting contains information about FTP sites that carry collections of  $\text{\TeX}$  macros, various public domain implementations of  $\text{\TeX}$  for different platforms (e.g.,  $\text{\EmTeX}$  for DOS/Windows,  $\text{\OzTeX}$  for the Mac) and so on. For new users the FAQ posting will often contain answers to many of the simple (and not-so-simple) questions. For aficionados the FAQ file often contains pointers to sources of detailed information. If you are considering posting a question to a bulletin board or newsgroup (particularly the more technical newsgroups) it is always a good idea to read the FAQ file first. At the least it saves you getting a number indignant "read the FAQ" replies and in the best of circumstances it gets you an answer in minutes rather than hours or days.

***One way of making quick headway is to read the FAQs (Frequently Asked Questions) postings.***

Not every bulletin board or newsgroup contains a FAQ post and, even worse, many campus bulletin board system generally don't keep a months worth of back posts. So what should you do when you can't find a FAQ post? Suppose you are interested in the LISP programming language and you can't find the FAQ file. (There is one, it is posted around the middle of every month, in about 6 parts, with a total of over 200Kb of text, to several lisp-related forums). A good strategy is to start out by subscribing to the appropriate newsgroup (in this case `netnews.comp.lang.lisp` is a good place to start) and become a passive reader/scanner of the messages. If the

FAQ file is posted every month, and it is not in the old messages available on your bulletin board system, then it is likely that you will see the post in the next week or two. While you are waiting you can sometimes learn a lot by just reading the other discussions.

A final tidbit. The USENET community does quite a good job of collecting FAQ files in one place. There are several USENET newsgroups that consist solely of FAQs. Some that I know of are netnews.comp.answers (many FAQs for newsgroups in the netnews comp hierarchy), netnews.misc.answers, netnews.news.answers, netnews.rec.answers, and netnews.sci.answers.

Of the statistics related boards, I'm only aware of a FAQ for the S-news mailing list. This is archived in StatLib. To get a copy, send the message

send faq from S

to statlib@lib.stat.cmu.edu, or use gopher or FTP to access StatLib.

By adroitly using your bulletin board system you can quickly learn a lot about many technical areas. Of course, the trick is to limit the amount of time you waste reading pointless material. Again, this is where it is useful to read the FAQs. In a few minutes one can often find answers to most of the interesting and common questions. So go forth and explore the bulletin boards and be a smart reader by sticking to the FAQs.

Mike Meyer  
Carnegie Mellon University  
mikem@stat.cmu.edu



## ***BITS FROM THE PITS***

# **Statistical Computing and Graphics in Science and Industry**

This column features statistical computing and statistical graphics activities in science and industry. I invite your comments and suggestions for future columns. Please send comments, inquiries, and suggestions to Albert M. Liebetrau, Analytic Sciences Department, Battelle-Northwest, MS K7-34, P.O. Box 999, Richland, WA 99352, AM\_Liebetrau@pnl.gov, 509-375-2694.

## ***Uncertainty Analysis for Computationally-Demanding System Codes***

*Part one of a two-part series*

Some degree of anthropogenic environmental change is an inevitable consequence of global-scale human activity. Both policy makers and the public are extremely interested in the possible consequences of environmental changes. The Department of Energy is particularly interested in the effects on energy consumption, both of naturally occurring environmental changes and those resulting from human activity. Large-scale modeling efforts are currently underway to assess the possible effects of potential environmental change scenarios. Global estimates of energy consumption, for example, are obtained by integrating the results from several regional models. The regional models incorporate factors such as regional climatology and hydrology, and include models of terrestrial ecosystems and human activities (agriculture, energy use, adaptation, etc.). Boundary conditions for regional models are derived from general circulation models, which are run for the climate-change scenarios of interest. (We can debate the usefulness of models, but such models are in fact being used to make substantive policy decisions and their use for this purpose will only increase with time.)

One important use of models such as those described above is uncertainty analysis, which refers to methods for estimating the probability distribution of model response resulting from variability in its input variables. However, traditional methods of uncertainty analysis (Monte Carlo simulation) fail for system models such as these because each component may in its own right be a complex code with heavy computing demands. A new approach to uncertainty analysis is required. To address this problem, McKay et al. (1979) developed a sampling method (Latin hypercube sampling) that is more efficient for simulation than simple random sampling. Taking another approach, Downing et al. (1985) used response surface approximations to reduce computing requirements. In Liebetrau and Scott (1991) and Liebetrau et al. (1993), we propose a strategy for complex system codes that employs both response-surface approximations and efficient sampling strategies. The basic idea is to develop a simplified analog to the complex system code, which we shall call a performance assessment (PA) code, by approximating its computationally demanding components. The PA code is abstracted from the underlying code so as to preserve the essential features of component processes and the interactions among them. The PA code can be used for uncertainty analysis because it requires less computing

resources than its underlying analog.

Our strategy for developing PA codes has three basic elements. These are (i) efficient selection of an initial set of model inputs (realizations), (ii) development of an approximation to the response surface of the underlying codes, and (iii) an updating algorithm that uses existing information to determine the locations (inputs) for additional runs. The overall idea is to develop a two-tiered system model that consists of a performance assessment model that “sits atop” an underlying model which is made up of the detailed component models. After response surfaces for the component models have been approximated, most computations for uncertainty analysis are done at the PA level. When it is necessary to drop down to the lower level, results of these new runs are used systematically to improve approximations at the upper level.

In previous columns, I have described some of the tools available to implement the approximating strategy outlined here. In the second part of this article, I will describe our experiences in attempting to implement this strategy for two real-world examples.

## References

- Downing, D. J., R. H. Gardner and F. O. Hoffman. (1985). “An Examination of Response-Surface Methodologies for Uncertainty Analysis in Assessment Models.” *Technometrics* 27(2), 151-163.
- Liebetrau, A. M. and M. J. Scott. (1991). “Strategies for Modeling the Uncertain Impacts of Climate Change.” *Journal of Policy Modeling* 13(2), 185-204.
- Liebetrau, A. M., P. D. Whitney, D. W. Engel and C. A. LoPresti. (1993). “Computational Analogues to Complex Computer-Based Codes.” Technical Report.
- McKay, M. D., R. J. Beckman, and W. J. Conover. 1979. “A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code.” *Technometrics* 21(2), 239-245.

Albert M. Liebetrau  
*Battelle Pacific Northwest Laboratories*  
AM.Liebetrau@pnl.gov



## GEOGRAPHIC INFORMATION SYSTEMS

# Designing the GIS Interface

In my last column, I promised to describe the use of focus groups if a significant number of readers expressed

interest. Focus groups might appear inappropriately distant a topic for a column on geographic information systems (GIS). But anyone who has tried to use a GIS intended for more than pedagogic purposes will appreciate its relevance. Developers of GIS software have paid scant attention to both interface design and the experiences and opinions of actual and potential users, and their products reflect an appalling ignorance of human factors equaled only, one might argue, by the developers of videocassette recorders. That GIS developers still rely almost exclusively on after-the-fact feedback from users and disgruntled letter-writers reflects an immature technology inadequately exploited by specialized developers facing limited competition in a fragmented marketplace. That people struggle to use their products is convincing evidence of human persistence in exploiting a promising technology—not to mention the profitability of vendor-run workshops and short courses. I had no delusions that the comparatively infantile graphic narratives discussed in my last column were any better than the typical GIS. But group interviews seemed a useful way to solicit critical advice in a synergetic setting.

## Focus Groups

Although the focus group is not a full-fledged human-factors technique, human-factors experts use group interviews (as they are also called) to pretest product designs as well as to identify issues worth addressing with more exacting testing procedures. Employed principally in marketing and media research, the group interview is a systematic method for identifying the range of attitudes and preferences that buyers, readers, viewers, and voters have about a product, publication, TV program, or political candidate. Social scientists have little use for focus groups, which are wholly unreliable for estimating means, variances, covariances and other fodder for the variation-explaining paradigm. But marketing experts and product designers have found group interviews an effective strategy for identifying flaws and gaining insight about user needs. A synergy develops in which participants stimulate each other to recognize problems and propose alternatives. It seems obvious but I'll say it anyway: to generate meaningful results, focus-group participants must reflect the target population.

To gather useful opinions about my prototype Atlas Touring scripts, I needed groups of more or less typical users of geographic data or graphic software. So instead of coercing undergraduates or bribing church groups, I wrote some letters, made some calls, and arranged for sessions with groups of six to ten professionals at four sites: Syracuse University's School of

Information Studies (faculty and doctoral students only, please), the Cartographic Division of the National Geographical Society (in Washington, D.C.), the Office of Geographic and Cartographic Research in the U.S. Geological Survey's National Mapping Division (in Reston, VA), and the GIS applications unit at IBM Corporation's research and development center at Kingston, NY. The managers with whom I dealt were cooperative and enthusiastic, and graciously recruited staff with a range of experiences in geographic analysis, map use, and interface design. They bought the argument that their commitment of valuable staff time might be treated as a colloquium or training seminar. At IBM I even sweetened the offer by lecturing later in the day to a somewhat larger group on the theme of *How to Lie with Maps* (University of Chicago Press, 1991)—an opportunity to plug my book proved difficult to resist, as it does now.

***Developers of GIS software have paid scant attention to both interface design and the experiences and opinions of actual and potential users, . . .***

I didn't conduct the focus groups myself; to do so would have tainted the results. I wanted honest, frank opinions about the prototype graphic narratives, not nice, polite comments inhibited by the designer's presence. Although I could have tried to avoid defensive responses to criticism, I find body language—the crossed arms, the shocked look, the odd twitch, the higher pitch, the long pause—nearly impossible to control. So early in the study I engaged an experienced focus-group “facilitator,” Myke Gluck, then a doctoral student in information studies. (Myke defended his dissertation on December, and moved on to a faculty position in the School of Library and Information Studies at Florida State University, in Tallahassee.) I accompanied Myke only on the visit to IBM, and while he ran the session, I hid out in an office. Myke and I collaborated in developing a participant questionnaire and a focus-group protocol. The questionnaire helped both to verify that the participants did in fact reflect our target population and to reveal backgrounds and viewpoints useful in interpreting individual responses and group dynamics. The protocol promoted uniformity among the four groups and guaranteed efficient use of the hour or so participants and their supervisors could commit to the session.

### ***Group Interviews***

We divided the group interview into two parts, the first to address a demonstration of the full “correlation script” (about 11 minutes) and the second to discuss a mixed demonstration (about 10 minutes) consisting of two dy-

amic spatial-temporal maps from the “historical script” and a new, interactive version of a key introductory graphic phrase from the correlation script. Myke narrated each demonstration—we rehearsed this carefully, for consistency—and then presented a number of issues for discussion. After the first demonstration, the participants were asked to discuss the issues of informativeness and coherence as well as the script's good points and bad points, and after the second demo they addressed issues of user interaction and customization. For each issue the protocol included a sequence of discussion questions. When addressing the issue of coherence, for instance, Myke asked questions such as: Did the progression of graphic phrases seem logical? Did the maps and graphs fit together? Were the text and labels useful or helpful? Our protocol also included several “probes” with which Myke coaxed further criticisms or suggestions from the participants. The probes included: Can you give me a specific example? Why do you feel that way? Does anyone feel differently? How do you see it? Myke taped the entire session with two recorders, both in full view. As part of the university's human-subjects review process, we prepared a statement, read to participants at the outset, agreeing to hide their identity and to destroy the tapes when finished with the analysis. Our statement also noted that they should feel free to leave at any time for any reason. Fortunately for us no one did. Then came the hard part. Interpreting the results required hours of listening to tapes and taking and comparing notes. We found it helpful to listen to the tapes separately, record our own observations, exchange notes, listen to the tapes again, and discuss the results. One of us then wrote an initial draft, which the other modified extensively.

### ***Findings***

Although communications and marketing researchers often carry out a more systematic content analysis—after having their tapes transcribed and coded by specially trained assistants—that level of rigor is more appropriate to studies comparing two or more designs or products than to exploratory studies concerned, like ours, with identifying issues and pinpointing flaws. Our intermediate goal was two lists, one registering widely shared complaints or suggestions to be addressed in the next design cycle and the other inventorying less frequent yet nonetheless worthwhile hints and options. In addition to suggestions explicitly stated by group participants, this second list typically includes the investigators' own ideas spontaneously triggered by the taped discussion. More important, though, are shortcomings and insights registered by two or more groups: in addition to blatantly obvious problems, the primary list

commonly includes design deficiencies and strategies affecting only a minority of potential users. How effectively a software designer appreciates and addresses these not-widely-shared concerns can be crucial in developing an efficient, effective, and broadly inclusive system. Findings based on focus groups are much closer to the soft end of the measurement-rigor-reliability continuum, and the risk of bias seems markedly greater than with the more tightly controlled research strategies of confirmatory analysis. Whoever reads our report (still under editorial review) might well question whether Myke and I have carefully listened to participants' complaints about the demonstration software and openly reported their doubts and reservations about narrative graphics. Conscious or unconscious bias is always a problem when researchers rely heavily on interviews, and our study is no exception. But the possibility of bias is cause only for caution and skepticism, not outright rejection. The value of our study can be judged by the apparent reasonableness and thoroughness of the collective and constructive criticisms we report.

What did we learn? Lots—far too much to describe in the current column. Would we do use focus groups again? You bet—despite the difficulties of pleading with managers, coping with D.C. traffic to meet a tight schedule, and taking our own Macintosh system into and out of an IBM facility. The exercise yielded valuable insights, and I can't wait for the opportunity this summer—NSF willing—to plunge in, implement and refine some new strategies, and generate vastly improved demos for another round of focus-group evaluation. The next iteration might not yield a satisfactory solution, but I surely have a clearer sense of where the Atlas Touring project is headed.

Mark Monmonier  
Syracuse University  
MON2IER@Sunrise.Syr.Edu



## **BOOK REVIEW BEAT**

---

This column returns, with the same proprietorship, because the new editors invited it, and because I was willing to continue. I had reservations, however. The previous editors always asked for the next column, but there was never any feedback from anyone about whether the column was good or bad, useful or wasted space. I will need to have positive feedback before I will feel that it is worth continuing. Actually, I will need more than that. Cost cutbacks in my corporation have taken away from me all the statistics journals, except the ones that I

buy. So, I am looking for some contributing assistance, correspondents who can report on reviews of statistical computing and graphics books that are reviewed in other journals. Particularly I am interested in the proliferating crop of statistical computing journals, none of which I will see any more, plus *Biometrics*, *Applied Statistics*, and *Shorter Book Reviews*.

There were two other problems that I encountered in preparing this column. First, there have been fewer reviews of statistical computing and graphics books recently. Second, some of the ones that have appeared have been written by me in the report section of *Technometrics* Book Reviews. It seems rather ridiculous to quote myself, so I shall just remark briefly on those reports. In November I commented on new editions of two books on statistical analysis using SAS, *Sas System for Regression*, Freund and Littell, and *Sas System for Linear Models*, Littell, Freund, and Spector. I found each to be significantly improved over its previous edition and an essential personal library item for any statistician who uses SAS for doing statistical analysis. In February I commented on a newcomer to this collection, *Sas System for Statistical Graphics*, M. Friendly. For me this much larger book was equally valuable, especially for its providing of extensive SAS code and macros, also for its comprehensive coverage of graphics in all types of statistical analyses.

Other statistical packages were included as central features of books, too. In *JASA* for December is a review for the 2nd Edition of *Data Analysis for Managers with Minitab*, H. Roberts, Scientific Press. John McKenzie praises the book's use of real data, effort at making statistics interesting, inclusion of statistics as part of business, and use of software to analyze data, stating that "the author has written a book that should be examined by anyone dealing with elementary business statistics". In comments in the *Technometrics* Editor Reports in February, I opine that the book is better suited for teaching statisticians about business applications than managers about statistics. GENSTAT also gets some textbook support with *Applied Statistics: A Handbook of Genstat Analyses*, E. Snell and H. Simpson (Chapman and Hall). Mike Driscoll, in *Technometrics* for November, notes that the "handbook details the use of GENSTAT to analyze the data used for the examples in Snell's book, *Applied Statistics: Principles and Examples*". For that book, he finds this one a "useful supplement" with "limited value otherwise".

There are some reviews of books not about statistical packages. In *JASA* for December, Martin David reviews a collection of papers, *Statistical and Scientific Data*

*Bases*, Z. Michalewicz, editor (Ellis Horwood). Noting the proliferation of persons who are embedding scientific data in data bases, he says that "this volume will leave them inadequately informed on the conceptual structure to use in their work and on the strategic trade-offs that are possible with powerful off-the-shelf Relational Data Base Management Systems (RDBMS)". In *Technometrics* for February, Minoos Niknian reviews *Randomization and Monte Carlo Methods in Biology*, B. Manly (Chapman and Hall), and finds the book "appropriate for master's level students of statistics and practicing statisticians, as well as subject matter investigators with good backgrounds in statistics and experience in computing". He calls the book "welcome in applied statistics". In *Biometrics* for September, Jim Gentle reviews the same book, saying that "the book is generally well written, and the examples are interesting and easy to understand". He also notes that "the word 'biology' in the title should not limit the readership".

The Editors or this columnist need some feedback to determine whether a look at reviews of statistical computing and graphics books is useful, or whether some other format should be used to deal with the books. If the overview of reviews is to remain reasonably comprehensive, some contributors for the aforementioned journals are needed. Contact Eric Ziegel or the editors.

Eric Ziegel  
*Technometrics Book Reviews Editor*  
 erziegel@nap.amoco.com



## NEWS CLIPPINGS

# Joint Statistical Meetings

## Statistical Graphics Invited Program

Section: Statistical Graphics  
 Session Title: Young Researchers In Statistical Graphics  
 Session Time: Tuesday, August 10, 8:30 - 10:20  
 Organizer: David W. Scott  
 Session Chair: David Scott  
 Speakers:

- "Graphics Keys—Misha's Resource Database Approach to Extensible Graphics," James Hardin, Battelle Lab, hardin@gandalf.pnl.gov
- "The Mode Tree: Nonparametric Visualization of Density Features," Michael Minnotte, Utah State University, minnotte@math.usu.edu
- "Assisting Inductive Modeling With Visualization," John Elder, University of Virginia, jfe7w@virginia.edu

"Graphical Methods for Finding Structure in Multivariate Data," Qiang Luo, George Mason University, qluo@galaxy.gmu.edu

Section: Statistical Graphics—co-sponsor Stat Ed, Teaching Health Stat  
 Session Title: Statistical Graphics and Animation For Instruction/Classroom  
 Session Time: Wednesday, August 11, 10:30 - 12:20  
 Organizer: Joseph Newton  
 Session Chair: Joseph Newton  
 Speakers:

- "Integrating Dynamic Graphics into a Linear Regression Course," Sandy Weisberg/Dennis Cook, University of Minnesota, sandy@umnstat.stat.umn.edu
- "Exploratory Dynamic Graphics for Survival Data," Neely Atkinson, Texas Medical Center, neely@biomath.mda.uth.tmc.edu
- "Methods for the Analysis of Data From Designed Experiments," Wei-Yin Loh, University of Wisconsin, loh@stat.wisc.edu

Section: ACM SIGGRAPH [subj to final signing]—Statistical Graphics, Stat Ed  
 Session Title: Multimedia: Past, Present, and Future  
 Session Time: Tuesday, August 10, 2:00 - 3:50  
 Organizer: Jan Pedersen, Xerox Parc  
 Chair: Jan Pedersen

- TBA [Current Multimedia Environment], Dan Russell, Xerox PARC, dmrussell@parc.xerox.com
- "Intelligent Agents as a User-Interface Metaphor," Tim Oren, Kaleida Labs, Inc., oren@kaleida.com
- "Multimedia Futures," Enrique Godreau, Aldus Corporation, enrique-godreau@aldus.com
- Discussant: William Cleveland, AT&T Bell Labs, wsc@research.att.com

Section: Statistical Graphics  
 Session Title: Novel Applications of Multivariate Statistical Graphics  
 Session Time: Tuesday, August 10, 4:00 - 5:50  
 Organizer: David Scott  
 Chair: TO BE NAMED

- "Visualizing Speech Data and Hidden Markov Models," Jim Schimert [speaker], Andreas Buja, Werner Stuetzle, Statistical Sciences, Inc., schimert@statsci.com
- "Visualizing Multivariate Rank Data," G. Thompson and K. Baggerly, Southern Methodist University
- "Variable Resolution Bivariate Plots," Chisheng Huang, Univ. of Washington, cph@stat.washington.edu

## **Statistical Graphics Roundtable Luncheon Discussions**

1. "Software for statistical graphics," James E. Gentle, George Mason University, [jgentle@gmu.edu](mailto:jgentle@gmu.edu)
2. "Software for Multivariate Functional Estimation and Visualization," David W. Scott, Rice University, [scottdw@rice.edu](mailto:scottdw@rice.edu)
3. "Issues in implementing graphical functions," Linda Clark, AT&T Bell Labs, [lac@research.att.com](mailto:lac@research.att.com)
4. "Meaningful Statistical Graphics for Manufacturing," Karen Kafadar, National Cancer Institute, [kk@helix.nih.gov](mailto:kk@helix.nih.gov)
5. "Single Frame Video Recording: The Wave of the Future?," William F. Eddy, Carnegie Mellon University, [bill@stat.cmu.edu](mailto:bill@stat.cmu.edu)
6. "Graphics in Bayesian Statistical Computing," William DuMouchel, New England Biomedical Research Foundation, [dumouche@jimmy.harvard.edu](mailto:dumouche@jimmy.harvard.edu)

## **Statistical Computing Invited Program**

The Statistical Computing Section has an exciting collection of eight invited and special contributed sessions for the annual ASA meeting in San Francisco.

The session that received the top number of votes in the ASA competition for extra sessions was organized by Ed Wegman of George Mason University and titled "Virtual Reality for Exploratory Analysis." The session is a tutorial on using virtual reality to explore high dimensional data. Rotating a point cloud on a screen to "see" in three dimensions is an example of the kind of advances in graphical data analysis we saw in the 1980's. Virtual reality appears to offer the next step for the 1990's: it can be used to "immerse" the statistician in the data world.

A second session entitled "Software Statistics" organized by David James, AT&T Bell Laboratories, examines the role of statistics in the relatively new field of software engineering. Scott Vander Wiel, AT&T Bell Laboratories, will discuss "Capture-Recapture and Other Statistical Methods for Software Inspection Data," while Wendell Jones, Bell Northern Research, will discuss "Modeling for Software Systems." Also Bill Curtis, Carnegie-Mellon University, will discuss "Conundrums in Applying Statistics to Software Engineering."

Getting up to speed on high performance computers is the topic of the session "High Performance Computing"

organized by Janis Hardwick, University of Michigan, who will present "Computational Analyses of Sequential Allocation Problems." Phil Spector, University of California, Berkeley, will present "Parallelizing CART" and George Ostrouchov, Oak Ridge National Laboratories, will present "Statistical Modeling Applications on Parallel Computers." Quentin Stout, University of Michigan, will be the discussant.

Sallie Keller-McNulty, Kansas State University, organized a session that looks at nontraditional models entitled "Statistics Driven by Non-Classical Assumptions: Spatial Sampling, Estimation, and Modeling." She will present "Mean Field Estimation for Probability Distribution Function Methods of Turbulent Reactive Flows." Katherine Campbell, Los Alamos National Laboratories, will discuss spatial considerations in environmental sampling in "Sampling the Continuum: The Spatial Support of Environmental Data." Douglas Nychka, North Carolina State University, will discuss "Stretching and Translating Surfaces to Model the Drain Current in a Semiconductor."

G. Pliego of ITESL, Mexico, has organized a session "Wavelet Software and Algorithms in Statistics." that includes an introduction to wavelets. "Computational Techniques for Wavelet Applications" is the title of Wim Sweldens, University of South Carolina and Katholieke Universiteit Leuven, Belgium. Z. Wang, Purdue University, will present "Software Examples for Statistical Wavelet Analysis." Carolyn Carroll, IBM, will discuss "Parallelizing Wavelets."

John Maryak, The Johns Hopkins University, organized "Recursive Methods for Optimization in Time Series" with Larry Goldstein, University of Southern California, as discussant. T.L. Lai, Stanford University, will present "Parallel Recursive Algorithms in Time Series Estimation and Control." James Spall, The Johns Hopkins University, discusses "Accelerated Stochastic Optimization in Multivariate Problems," and John Monahan, North Carolina State University, presents "Applications of Stationary Stochastic Approximation."

The session "Smoothing in Data Analysis: Tutorial on Methods and Applications" is organized by Karen Kafadar, National Cancer Institute, who will present "Choosing among Two-dimensional Smoothers in Practice." Trevor Hastie, AT&T Bell Laboratories, will present "What Smoothers Do" and Colin Goodall, Penn State University, will present "Nonlinear Smoothers for One-Dimensional Data." Katherine Hansen, Sandia National Laboratories, will present "Smoothing Multidimensional Data," and Owen Devine, Center for Disease Control, will act as discussant.

Tim Hesterberg, Franklin and Marshall College, organized the session "Weighted Samples in Simulation," and he will present "Importance Sampling and Control Variates for the Bootstrap." George Easton, University of Chicago, will discuss "Configural Polysampling and Some Related Research" while Peter Glynn, Stanford University, will discuss "Rare Event Simulation for Queues."

Mary Ellen Bock, Program Chairman 1993  
Statistical Computing Section

### **Statistical Computing Roundtable Luncheon Discussions**

Five interesting Luncheon Discussions are planned in San Francisco. The topics and a brief description of each are given below. Feel free to contact the Discussion Leaders for more information. To participate in these discussions, you need to sign-up for the luncheon when you register for the conference. All of these luncheons will be scheduled for the same day

1. "Impact of Computing on Industrial Experimental Design," Perry D. Haaland from Becton Dickinson Research Center (phd@bdrc.bd.com) will be leading a discussion on a variety of issues including new results in optimal design, inference based on robust methods, graphical analysis, new results on dispersion modeling, Taguchi or "classical" approaches, industrial training, and practical experiences. Perry also plans to leave some time for the sharing of "war stories."
2. "Using the Statistical Computing and Statistical Graphics Newsletter as a Vehicle to Keep Up with Recent Developments in Computing" Newsletter Editor Jim Rosenberger from Penn State University (jlr@psuvm.psu.edu) will lead this discussion. This roundtable will be the perfect vehicle for those interested in providing input to our ever growing and evolving Section Joint Newsletter.
3. "The Design and Analysis of Computer-Based Experiments" Albert M. Liebetrau from Battelle-Northwest (AM.Liebetrau@pnl.gov) plans to cover this topic by encompassing the work of the "Sandia" School (Iman and Conover latin hypercube sampling approach), the "Oak Ridge" School (Worley, Oblow, Pin and others), and the "Second Oak Ridge" School (Sacks, Welsh, Morris, and Mitchell). Discussion will also cover the recent theoretical work by Art Owen on latin hypercube designs and orthogonal arrays.

4. "Developing Departmental Computing" Mike Conlon from University of Florida (mconlon@stat.ufl.edu) will discuss the development, administration and maintenance of modern departmental computing networks. Issues of resource sharing (coordinating funds as well as equipment), specifications (what should the system be able to do), performance, software, vendors, migration (acceptance and training), institutional barriers (technical and bureaucratic) and more will be discussed. Mike will consult on how to get from various starting points (stand alone PCs, mainframes) to operational networked systems.

5. "Future Directions in Statistical Computing" John Chambers from AT&T Bell Laboratories (jmc@research.att.com) will lead this discussion. This luncheon parallels an Invited Speaker Session with John, Wayne Oldford, Werner Stuetzle, and Dave Andrews on the same topic. Lively discussion on our future is expected to be found here!



## **Interface '93 Meeting Highlights**

**25th Symposium on the Interface: Computing Science and Statistics**

**Statistical Applications and Expanding Computer Capabilities**

April 14-17, 1993  
San Diego, California  
Pan Pacific Hotel

*Keynote Speaker:* David Brillinger "Statistics and Computing in Science"

*Sponsor:* Interface Foundation of North America

*Hosts:* Precision Data Group and University of California, Berkeley

*Cooperating Societies and Institutions:*

American Statistical Association (ASA); Institute of Mathematical Statistics (IMS); Society for Industrial & Applied Mathematics (SIAM); Operations Research Society of America (ORSA); The Biometrics Society (WNAS); University of California, Berkeley; San Diego State University, San Diego; Northern and Southern California Chapters of the American Statistical Association.



*Invited Sessions include:*

Data Compression; Computing with Environmental Data; Biopharmaceutical Maps and Graphics; Clinical Trials; Protein Structure; Digital Networks; User Interfaces; Geosciences; Software Engineering and Statistical Methods; The Interface at 25; Likelihood Applications; Library Systems; Medical Applications; Multivariate Function Estimation; Networked Information Systems; Supercomputers; Time Series Analysis; Wavelets; Computers and Statistics in Drug Discovery; Quality Data Bases.

Inquiries should be sent to:

Interface '93  
Michael E. Tarter, Program Chair  
140 Warren Hall  
University of California, Berkeley  
Berkeley, CA 94720  
(510) 642-4601  
tarter@stat.Berkeley.EDU

Conference proceedings of invited and contributed papers will be published. Camera-ready copy will be due on June 1, 1993.

*Conference Schedule:* The conference begins on Wednesday evening, April 14, with a get-acquainted reception. Technical sessions will be held Thursday and Friday with a banquet Thursday evening. There will be final technical sessions Saturday morning.

*Short Courses:*

- SYSTAT: Data Analysis – 2 days prior to the conference, Monday, April 12 and Tuesday, April 13. SYSTAT produces statistical software designed for data analysis. Registration for this short course will be handled separately.
- BMDP: A New Windows-Based Statistical Analysis Environment will be presented Wednesday afternoon, April 14. There is no fee for this course, however, please note on the registration card if you plan to attend.
- Randomization Tests: Jenny Baglivo, Marcello Pagano, and Cathie Spino will offer a short course in Randomization Tests: Theory and Practice. There is no fee for this course. Please note on the registration card if you plan to attend.

*Registration:* The registration fee is \$165 for members of the cooperating societies, ASA, IMS, SIAM, ORSA, the Biometrics Society (ENAR and WNAR), and for persons affiliated with University of California, Berkeley. The fee is \$55 for students; for others it is \$185. Registrations received after February 15, 1993 will be

charged a late fee of \$20. The registration fee covers the reception, coffee breaks, banquet, and the proceedings. Please make checks payable to Interface '93.

## GENERAL INFORMATION

*Accommodations:* All meetings will be held at The Pan Pacific Hotel in downtown San Diego, California. Conference room rates of \$99 per room have been arranged for all attendees. Early registration is strongly recommended to assure a room. Please contact the hotel directly for reservations.

THE PAN PACIFIC HOTEL, 402 West Broadway, San Diego, CA 92102-3580. Telephone (619)239-4500 or (800)626-3988.

Conference rate for Interface '93 participants honored until March 14, 1993. Rate can be extended 3 days prior and/or following the conference. Rates are \$99 for either single or double room. Reservations held until 4 p.m. without deposit or accepted credit card.

*Air Fare:* At the present time, no airlines are offering discount association fares. Vineyard Travel in San Diego will be happy to assist you with your travel plans. Their telephone number is (619) 741-6669. Rosemary Nigro is our Corporate Travel Consultant.

*San Diego International Airport:* San Diego International Airport is served by most major airlines. Shuttle bus and limousine service are available to the Pan Pacific Hotel in downtown San Diego.

*Program Committee:* Lynne Billard, Mary Ellen Bock, Noel Cressie, Arnold Goodman, Sam Greenhouse, Jon Kettenring, Diane Lampert, Michael Lock, Bob Newcomb, Joseph Newton, John Rice, Ernest Scheuer, Bob Shumway, Michael Tarter, Grace Wahba, Edward Wegman

*Interface Foundation:* The Interface Symposium is an activity of the Interface Foundation of North America, a nonprofit, educational corporation founded in August of 1987. The aim of the IFNA is to promote the Interface Symposium and related activities at the interface of computing science and statistics. The 25th Symposium is the sixth held under the auspices of the Interface Foundation. The next symposium will be hosted by the SAS institute in North Carolina in 1994.



# News from NSF

February 16, 1993

Dear Colleague:

## Announcement of Proposal Target Dates

In order to improve the Division's proposal management, and possibly to employ disciplinary panels in the merit review of proposals, the Division of Mathematical Sciences plans to introduce target dates for proposal submission for disciplinary research activities for FY 1994 NSF funds.

Beginning in the fall of 1993 the Division will introduce two target dates for proposals submitted to the following programs:

Oct. 22, 1993    Algebra and Number Theory Program  
                  Classical Analysis Program  
                  Modern Analysis Program  
                  Topology and Foundations Program

Nov. 19, 1993    Applied Mathematics Program  
                  Computational Mathematics Program  
                  Geometric Analysis Program  
                  Statistics and Probability Program

These dates do not overlap substantially with other known Foundation target dates, mesh reasonably well with academic calendars, and cluster the programs so as to provide a balance with respect to both overlapping scientific content and anticipated program proposal loads.

Proposals which miss the target dates will be handled as time permits. Priority will be given to proposals arriving on or before the above target dates.

The above dates do not apply to the activities of the Division's Office of Special Projects. These activities already have specified target or deadline dates.

Sincerely,

Frederic Y. M. Wan  
Division Director  
Division of Mathematical Sciences



# News from NSA

## ANNOUNCEMENT

The NSA Mathematical Sciences Program continues its efforts at funding high quality mathematical research in the areas of Algebra, Number Theory, Discrete Mathematics, **Probability**, **Statistics** and Cryptology. The program, in its present form, had its beginning in 1987 when the then director of the National Security Agency,

Lieutenant General William E. Odom, announced the expansion and redirection of Program OCREAE, NSA's grants program for research in Cryptology and related areas. This effort is currently being vigorously supported by the Agency.

The grant proposals submitted to the program are reviewed by the NSA Mathematics Review Panel which is appointed and administered by the American Mathematical Society. Under the guidance of this panel, the program has been particularly interested in supporting promising young investigators with small summer salary grants. The program has also directed a significant portion of its funds to senior investigators for the support of their graduate students and to university departments for the support of special conferences and workshops. In an attempt to formalize these objectives, the MSP program now offers funding in four distinct categories; the **Young Investigators Grant**, the **Standard Grant**, the **Senior Investigators Grant**, and the **Conferences, Workshops, and Special Situations Grants**.

Firm deadline of October 15, 1993, for all proposals except conferences. Funding as soon as possible after October 1, 1994.

Conference proposals accepted any time. Allow 8 months for funding.

Further information on grants is available by calling (301) 688-0400 or writing:

Dr. Charles F. Osgood, Director  
NSA Mathematical Sciences Program  
National Security Agency  
Attention: R51A  
Ft. George G. Meade, MD 20755-6000  
msp@titan1.math.umbc.edu



# Electronic White House

There has been a lot of recent information and disinformation about the Clinton-Gore electronic White House. We have seen various postulated e-mail addresses for contacting the new president of the United States, but none of the addresses is completely satisfactory. One common outcome of e-mail to the President is a reply, via regular paper mail—not quite what one would expect! The most common explanation? . . . lack of funds to buy computers and connect to the outside world. With that cleared up, it is still worth reporting some of the ways of contacting the white cottage.

The new Clinton-Gore Administration has several elec-

tronic mail addresses. The MCI Mail box address (see below) and bulletin board have received the most publicity.

GEnie has a PF (Public Forum) section which carries MIT-generated files from White House press briefings and speeches, but that forum offers no e-mail feedback to the Clinton Administration. It serves the public interest by making important text files and other public information widely available.

What if you really want to get a message through to Washington? In the past this meant letters or telegrams, but now you can save paper and attempt direct electronic contact in several ways.

The following mailbox addresses are reported to work for sending e-mail to the White House and entering dis-

cussion/file areas related to the new administration:

- Compuserve: 75300,3115 (e-mail); GO: WHITE HOUSE (White House forum)
- America OnLine: clinton pz (e-mail); KEYWORD: WHITEHOUSE (White House area)
- MCI; WHITE HOUSE (e-mail); VIEW WHITE HOUSE (views bulletin boards)
- Internet e-mail address:  
clinton-hq@Campaign92.Org;  
75300.3115@compuserve.com;  
clintonpz@aol.com

We haven't personally tried any of these addresses or discussion forums. Perhaps our readers could give us some feedback!

### ***1994 Joint Statistical Meetings in Toronto Canada***

As Statistical Computing Program Chair for 1994, I am soliciting your help. Now is the time to give me suggestions for topics, speakers, and organizers for our invited sessions. If you have any ideas along these lines please contact me before or at this year's Joint Meetings in San Francisco.

Sallie Keller-McNulty,  
*Kansas State University*  
sallie@cecil.stat.ksu.edu  
913-532-6883

The *Statistical Computing and Statistical Graphics Newsletter* is a publication of the Statistical Computing and Statistical Graphics Sections of the ASA. All communications regarding this publication should be addressed to:

James L. Rosenberger  
 Editor, Statistical Computing Section  
 Department of Statistics  
 The Pennsylvania State University  
 University Park, PA 16802-2111  
 (814) 865-1348  
 JLR@stat.psu.edu

Michael M. Meyer  
 Editor, Statistical Graphics Section  
 Department of Statistics  
 Carnegie Mellon University  
 Pittsburgh, PA 15213-1380  
 (412) 268-3108  
 mikem@stat.cmu.edu

All communications regarding membership in the ASA and the Statistical Computing or Statistical Graphics Sections, including change of address, should be sent to:

American Statistical Association  
 1429 Duke Street, Alexandria, VA 22314-3402  
 (703) 684-1221

### ***Where to find it***

A WORD FROM OUR CHAIRS . . . . .	1	Getting your Facts from FAQs . . . . .	17
Statistical Computing . . . . .	1	BITS FROM THE PITS . . . . .	18
Statistical Graphics . . . . .	9	Statistical Computing and Graphics in Science and	
FEATURE ARTICLE . . . . .	1	Industry . . . . .	18
Saxpy, gaxpy, LAPACK, and BLAS . . . . .	1	GEOGRAPHIC INFORMATION SYSTEMS . . . . .	19
EDITORIAL . . . . .	2	Designing the GIS Interface . . . . .	19
SECOND FEATURE . . . . .	2	BOOK REVIEW BEAT . . . . .	21
Production of Stereoscopic Displays for Data Anal-		NEWS CLIPPINGS . . . . .	22
ysis . . . . .	2	Joint Statistical Meetings . . . . .	22
DEPARTMENTAL COMPUTING . . . . .	7	Interface 93 – Meeting Highlights . . . . .	24
Not just hardware and software . . . . .	7	News from NSF . . . . .	26
COMPUTER COMMUNICATION AND NET		News from NSA . . . . .	26
SNOOPING . . . . .	16	Electronic White House . . . . .	26
Gopher and other resource discovery tools . . . . .	16		