Newsletter of the NeXT Programmer's Community

NeXT Users' journal

SEPTEMBER 1990 • ISSUE 9

WELCOME

Erica J. Liebman

This issue will be more "September version a" than just September. Another issue will be out in a few weeks with Andrew Stone's Stone Age, a delightful new column called "the Master Programmer" care of our friends in the Dallas User Group (thanks Jeff!), and a whole lot of information (I hope!) about the new line of 68040 and Color NeXT machines.

This is such an exciting and crucial time as far as the NeXT goes. Within the next few months, we'll be seeing how close the pessimist nay-sayers are predicting reality. Tidings of doom have sprung from the page (and been sent to me) from as near as Florida and as far as New York and California. The Wall Street Journal and Businessweek have both been on the "going-to-fail, going-to-die" bandwagon.

The thing that frustrates me the most, the thing that keeps bothering me as everyone is knelling gloom is that I *really* like the NeXT. I do. I really do. I know that I'm preaching to the converted, but these nay-sayers have for the most part never *used* a NeXT for more than five minutes in their lives.

Let me give you a case in point. I was recently invited to speak at an Apple Developers meeting. The main thing these programmers wanted to know was could they be guaranteed to sell five hundred units a month at $395 or at least $500K per year or other such criteria.

"Frankly", I responded as honestly as I could, "no".

"Well," they said, "we're simply not going to develop for the machine, and no one else is either." This is the mindset from people who in 1984 were programming for the 128k Mac. (I beg indulgence if I have gotten the year or the memory size wrong...I was programming on Vaxen and Symbolics in '84). These, in fact, were the Macintosh Pioneers, the ones whose dedication and stubborness had brought them through the maturing of an exciting new machine.

With all that experience under their belt, these were people demanding guaranteed hundreds of large purchases in an uncertain market. "It can't survive," they said to me. "It's not innovative, it's got a stupid medium for data exchange, no color, its too expensive, and everyone knows that Jobs is just trying to relive his youth." As an aside, one of the strangest points brought up by this group was that the NeXT had no *resedit* -- endusers couldn't modify object code. You figure it out.

"What would entice you to start porting applications to the NeXT, or (better yet) write some there from the ground up?" Guaranteed money, they replied, guaranteed audience and purchases. Risk was okay in 1984, they told me, but the industry of software production has little room these days for pioneers.

Where, I asked myself, can one pick up some spare pioneers?

**Copyright 1990 by BuzzNUG
Individual Articles Copyright 1990 by their Authors. This issue of NUJ may be freely copied and distributed but not altered. Authors are free to resubmit articles for publication in other media.
This issue of NUJ may not be sold for profit.**

HEADLINE NEWS

• NeXT Introduces new 68040 machines including the new NeXT Dimension color system. The 68040 board upgrades will be shortly available for the current owners of the cube with an estimated cost of $1495. System software 2.0 will be required for the 68040 and compatible with the 68030. Most applications are being unbundled to decrease cost and limit disk size. Developer software (IB, etc) will be unbundled and delivered as part of a special developers package.

INDEX

Welcome • 1
Headline News • 1
Editorial Stuff • 2
Feedback from the Trenches • 2
TopDraw ABC's • 3
A Mathematica Algorithm • 5
A Mathematica Primer • 17
Installing Fujitsu Drives • 38
Market View • 41
User Groups • 44
Buzz's Hint Corner • 49
NeXT Support Answers • 49
The Way I See It • 51

Contributors:

Erica J. Liebman, Bruce Henderson, Dick Silbar, Kris Kamisetty, Dave Hieb (thank you for your patience, Dave!), Izumi Ohzawa, Conrad Geiger, Doug Keislar, Roger Kirschner, Christopher Lane

•Editorial Matters•
1150 Collier Road NW/L-12
Atlanta, GA
30318

•Editorial Stuff

Articles for Buzzings are accepted in various forms, NeXT mail enclosures and Internet .wn.tar.Z forms are preferred but ascii text via the net, IBM and Mac Disks via USMail and (yikes) written text via the same USMail are happily accepted. We can guarantee no return of materials without SASEs, sorry. Our focus is how-to articles, especially with sample code. All articles are subject to editorial review.

We also welcome copies of new (and old) software for review from third party vendors. Again, we can not guarantee material return without SASE or guarantee publication dates, if at all, although we try to be prompt.

"Feedback from the Trenches" is open for comments/ letters of limited length from all readers. Please write and tell us what you liked and what you disliked.

Mailed subscriptions should start real soon now, with any luck. Please write for information.

Feature Submission Guidelines

If you have NeXT mail, simply drag your icon into next mail and post it off to me. Don't worry about tar'ing and compressing.

•Avoid passive voice. Please.

•Spellcheck.

•Wit is welcome, obscenity is not.

•Preformatting with the following guides will aid us ENORMOUSLY:

•Paragraphs: No tabs, single return at end of paragraph, no hard-returns at ends of lines

•Title on one line, author information on one line.

•Avoid blank lines between paragraphs. Use blank lines to denote sections.

•Start sections on first line after section header.

Editorial matters to:

BuzzNUG c/o EJ Liebman
1150 Collier Rd./NW L-12
Atlanta, GA 30318.
1-404-352-5551.

There is always an answering machine, but please respect relatively normal hours. Long distance phone calls may not be returned by the impoverished student at the other end -- if you leave a message, indicate if it is acceptable to call you collect. Please send any deliveries of items that will not fit within a tiny mailbox care of the Leasing Office. To contact me directly for subscription information, corrections, requests, or just to say hi, write via internet: erica@kong.gatech.edu

This issue of NeXT Users' Journal was prepared in FrameMaker, generously donated by Frame.

Feedback from the Trenches

•Erica -How are you doing? I've been working at Canon for most of the summer and they've kept me too busy for me to write any journal articles. I'll write up a discussion of a new object I created at Canon and that's being made freely available for you Real Soon Now (i.e., before Christmas).

I'm looking at the Issue 8 right now. You have quite a following all the way over in Japan too. :-)- jiro

• Hi!1) First, nice work; I can see how the issues is shaping up.

2) Stick with one, just one, column/font size combination. In #9, I see two column small font (p. 2), and two column big (p. 14). Stick with all small, it makes for better reading, IMHO. Using bars for section delimiting helps, too, as on p.2. If you can stick with a global style, it will go a long way towards making a classier publication.

3) Develop a two column code layout. Most code is less than 40 characters long - comments are what wastes width. I even question putting the code IN the issues - reading a one or two page example, interspersed with non-code comments is informative. _Reading_ a five page example of just code is a waste, no matter how good the code is. At that point, I just want to get the code on-line and try it out. Check out things like UNIX Review to see what I mean. Then, take all the code for each issue and make another file, say NUJcoden, where n is the issue number. That way people won't have to 1) type it all in or 2) cut and paste till their wrists ache.

There is the issue of people who only get NUJ in hardcopy though...

3a) Never use variable width font for code, as in #9 p.36-44 - it just doesn't scan well. 4) Either use a column head graphic for all sections, or not at all. The ones in #9, like Robert Lins' Tao and Ian Smiths' HardCore are nice and somewhat in the same style. Whay not ask for submissions for all the 'standard' columns like Headline/Editorial/Trench Stuff and just random submitted product reviews. They may take up space, but now that you're up to 30+ pages, flipping through an issue with large graphics can help find an article quickly.

TopDraw ABC's

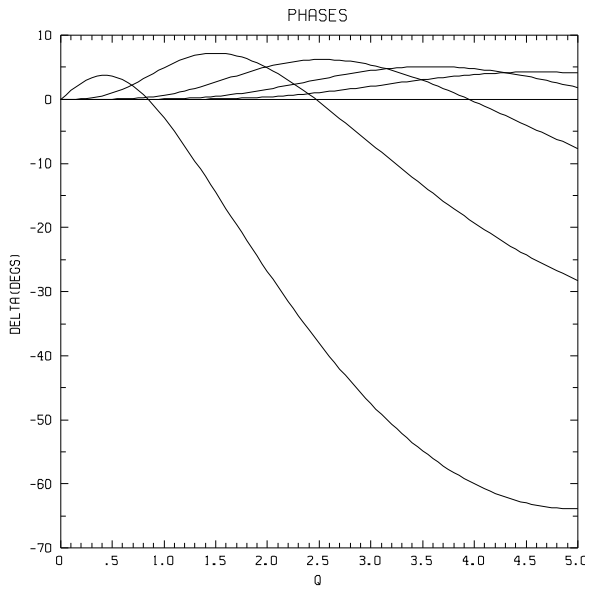
Dick Silbar, Los Alamos National Laboratory

(silbar@whistler.lanl.gov)

Prompted by Erica Liebman's review of TopDraw‰ (by Media Logic, Incorporated) in NUJ #7, I would like to share with you some of my experiences (over all of two days!) in using this nice software package. It is nice, but the "ABC's" in the title stands for "Annoyances, Bugs, and Circumventions". Here are three anecdotes, which may also have some value as Buzz Hints (of an extended nature).

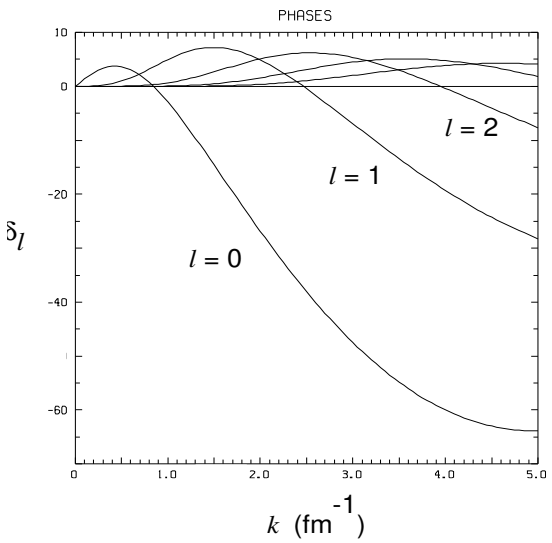
Cut-and Paste

Like Erica, I too felt a little stung by not being able to cut and paste conveniently between TopDraw and WriteNow‰ documents. To explain further, the sort of thing I had in mind for this package was as follows. I have some Fortran programs (on a nearby VAX) that generate output graphs on a Tektronix terminal. That's OK -- I talk to the VAX from Whistler (a NeXT) using Communicae‰, which has a Tektronix emulator. It all works quite well, and the window with the Tektronix graphics can be saved as an Encapsulated PostScript (or .eps) file. This in turn can be imported into a WriteNow document (which has the suggestive name of "Transparencies.wn") by using the "Place Graph..." item in the Edit sub-menu. The trouble is that the graph itself is kind of busy around the edges, its text is in an ugly font (actually, pretty primitive), and the curves on it *cry out* for labeling:



How to annotate and clean up the graphic? Well, with a lot of work, one could conceive of going into YAP and editting the bejabbers out of it. (It consists of a zillion little vectors drawn with "lineto" and "moveto" commands, boringly explicit but not very intuitive.) I in fact tried something else -- to see how I might work over the thing with the Draw App -- but was not very successful. Enter TopDraw (ta-DAH!).

Well, as EJL says, you can't simply cut-and-paste between the two Apps. In thinking about it, that's not so surprising (since most word proccessing packages can only communicate with each other by means of an ASCII intermediary). However, one can import and export EPS into and out of TopDraw. So, my first task (after sitting through the 18 MByte, British female tutorial) was to do just this and clean it up:



Gee, not too bad! Imagine my horror, however, when I tried to *print out* what I could see on my screen. Everything I added using the TopDraw tools was there *but nothing of the original graph showed up on the paper*!

I won't tell you about the terrible kludge I put together (TIFF files do print out) to get something like the above onto paper. The true resolution of the puzzle came from the response of Hal Roseman of Media Logic. He was well aware of circumstances in which the Display Post Script shows up fine but the Post Script printer sees nothing. ("It is fixed in the next release.") Certain applications that write EPS files miss an important line in the "prolog". The fix is to Edit the EPS file (here, "phases.eps") and insert as the first executable line

```
0 setgray
```

That done, it works like a charm.

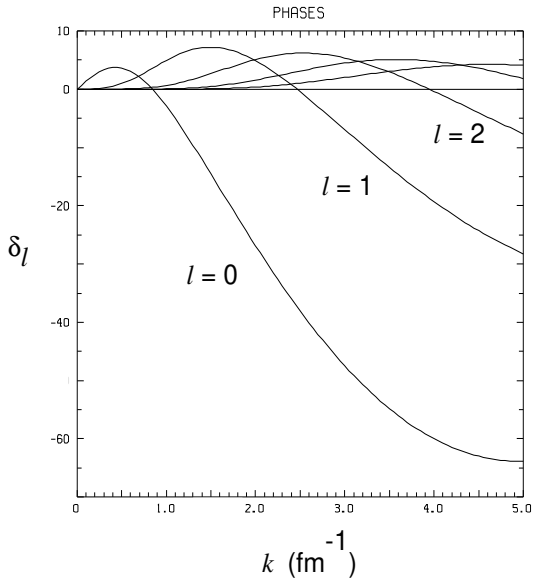
Disappearing Text

WriteNow allows you to write things like k2 = w2 - m2, but it ain't easy. Lots of mouse clicks needed in both the Font Menu and the Font Panel are needed to get the greeks and superscripts in the right places. A lot of the work is in remembering to always hit the "Set" button every time you change fonts.

You can write the same kind of text in TopDraw as well, and it isn't much different in ease of use. (The

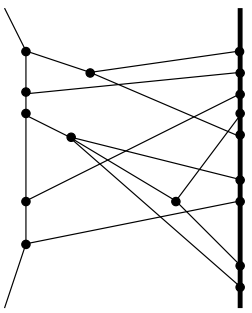
TopDraw Font Panel at least includes sub/superscript buttons, but there is just as much banging on the "Set" button.) The surprise, however, is that when you do the next letter in a new font, sometimes the whole line ahead of your inserted letter just disappears! Another bug (but one I haven't discussed with Media Logic).

Fortunately, the earlier text hasn't really disappeared. You get it back as soon as you hit "Set" again. And lose it all once more as soon as you type your next character. Pretty eerie stuff. This too has a simple cure, in fact two. For one thing, you can click the mouse to end the text insertion, and then highlight the text field again and hit "Set" one more time. Simpler, however, is to type one more (extraneous) character (such as a space) and then delete it at once. You get it all back (on the screen) either way.



A Small, Super Story

Not a TopDraw bug (as far as I know): WriteNow has another curious feature. If you "Place Graphic..." in front of a line, sometimes the App will crash entirely. (Moral: Save Early and Often, and especially so before placing a graphic.) Or, sometimes WriteNow will just flash the inserted graphic and then disappear it. In the two cases this happened to me, I ended up making the transparencies directly as TopDraw documents. However, sometimes the graphic is accepted and the result looks like



+ complications.

The tag-line "+ complications", which was there in the WriteNow file before the graphic was placed, ends up low.

How can one move the "+ complications" up to vertical center? There is a simple solution: Highlight it and click the "Superscript" item in the Font Menu an appropriate number of times. I won't bother showing you the improvement.

Mathematica Implementation of The Hungarian Algorithm

Kris Kamisetty, Research Assitant, Industrial Engg., CIM Systems Research Center, Arizona State University, Tempe, AZ 85287-5106

Email: kamisett@enuxha.eas.asu.edu (nonnextt) kris%enws34.cim.eas.asu.edu@enuxha.eas.asu.edu (NeXT)

1. Introduction

The *assignment problem* consists of allocating some resources to specific operating patterns in such a way that the costs of the assignments are minimized

Each resource is allocated only once and each operating pattern is performed exactly once. This is a special case of the Transportation problem. This is an extremely old technique found in any Operations Research textbook.

There are many implementations of this algorithm in many traditional programming languages. I have implemented the popular *Hungarian* Algorithm in *Mathematica*. This is to give the students of Operations Research Techniques, a feel of both the algorithm and *Mathematica* which is a pretty good tool to develop neat, user-friendly and live notebooks. I was toying with the idea of choosing the Time Crash algorithm or Out of Kilter algorithm for implementation. But, keeping the concept of "Incremental Delivery" in mind, I decided to go along with *Hungarian algorithm*, which, I thought was the simplest of all. In hindsight, I have done probably, the right thing under the circumstances since that itself proved quite a challenge.

2. Program Structure and Operation

The *Mathematica* program for the *Hungarian* Algorithm utilizes a lot of C-like constructs and loops. This program is structured into the well-known 3 phases. They are:

* Row column reduction
* Optimal checking and assignment
* Modification in case of non-optimality

I have used a lot of built -in functions related to lists like *Count*, *Length*, *RotateLeft*, *Drop*, *Append*, *Intersection* etc. I have also used many For, Do, While control loops.

It is important to start out with implementing the basic mathematics of the program instead of worrying about the loop structure, user interface or the optimization of variables and the program etc. I started with some really simple statements, seeing how I can implement the algorithm. This is so easy in *Mathematica* because, we can execute statements line by line like in interpreters. I used Block statements wherever I felt I needed a lot of local variables and encapsulation of capabilities.

Once the basic segments of the program started to work reliably after about 50 hours of working at the computer sometimes in joy and sometimes in anger, I started to give thought to the input phase of the program, the overall structure of the program etc.

I have grouped various statements which at first seemed like they were strewn all over the place into function definitions that achieve a 'logical milestone' in the algorithm. There are about 7 of them.

1. Initialization
2. Row reduction
3. Column reduction
4. Zero check
5. Whatdoithink
6. Optimal & Assignment
7. Nonoptimal

I saved all the function definitions in a file called hungarian.m. That needs to be kept in the home directory. I have also included usage statements for all functions. So, the user can find out about the purpose of the function by executing statement ?function , where function should be replaced by the name of the function you would like information about.

To run the program

1. Login to the Next Computer.
2. Start *Mathematica* by double clicking on its icon.
3. Open the file containing the notebook. (open *.ma file please. *.mb files are supposed to act as buffers for speeding up graphics processing.)
4. Click anywhere on the statement: "<<hungarian.m" followed by ENTER, not <cr>.
5. Click on the statements that are already there in the file in the following order.

    initialization
    inputphase
    rowreduction
    columnreduction
    zerocheck
    whatdoithink

If the message you get at this stage says,

Possible Optimum go to the optimal function,

Click on optimal. If you get OPTIMAL solution, you are done. If you get a message, *nonoptimail still* , click on the following statements in this order:

    nonoptimal
    zerocheck
    optimal

go through the iterations until you get the final optimal solution.

If you get the message (after whatdoithink)

Nonoptimal yet, go to nonoptimal function

Then, click on the following statements in this order:

nonoptimal
zerocheck
optimal

Again, go through these steps, until you arrive at the final **optimal** solution.

3. Future Directions

1. A package can be developed for the *Hungarian* Algorithm. This needs some modification of the program. I basically need to enclose the entire function definitions between Begin Package & End Package along with segmenting a part of the program involving internal calculations in a private portion of the program. Then, any user can use the package in his program and run the algorithm. Right now, the user needs to get into the specified file and use the algorithm.

2. I plan to develop an Interface Builder interface for this program. Then, I can replace the present user interface and replace it with a much better graphical interface.

4. Conclusions

It was a great learning experience to work on *Mathematica* and I acquired a little more insight into the operation of the *Mathematica*. It is both a language and a tool at the same time.

One can notice the tranditional C procedural programming like functions in thisnotebook's statements. I can explain this only by the fact that this is my first notebook in *Mathematica* and it is pretty darned difficult to jump into a new paradigm without bringing in some old habits!

Though, the same program could have been done in C or any other language of the same genre, it is much easier to work in *Mathematica* because we can start executing statements and getting results without any basic main() loop or even without having to define variables before using them.

It is easy to work with single statements first and then group them into a few functions later.

When I finished this project after long hours, the feeling was one of great satisfaction.

```
(*^

::[paletteColors = 128;
	fontset = title, "Times-Bold", 24, L2, center, bold, nohscroll;
	fontset = subtitle, "Times-Bold", 18, L2, center, bold, nohscroll;
	fontset = subsubtitle, "Times-Bold", 14, L2, center, bold, nohscroll;
	fontset = section, "Times-Bold", 18, L2, bold, nohscroll, grayBox;
	fontset = subsection, "Times-Bold", 14, L2, bold, nohscroll, blackBox;
	fontset = subsubsection, "Times-Bold", 12, L2, bold, nohscroll, whiteBox;
	fontset = text, "Times-Roman", 12, L2, nohscroll;
	fontset = smalltext, "Times-Roman", 10, L2, nohscroll;
	fontset = input, "Courier-Bold", 12, L2, bold, nowordwrap;
	fontset = output, "Courier", 12, L2, nowordwrap;
	fontset = message, "Courier", 12, L2, R21845, G21845, B21845, nowordwrap;
	fontset = print, "Courier", 12, L2, nowordwrap;
	fontset = info, "Courier", 12, L2, nowordwrap;
	fontset = postscript, "Courier", 12, L2, nowordwrap;
	fontset = name, "Times-Italic", 11, L2, italic, R21845, G21845, B21845, nowordwrap, nohscroll;
	fontset = header, "Times", 10, L2;
	fontset = footer, "Times", 12, L2, center;
	fontset = help, "Times-Roman", 14, L2, nohscroll;
	fontset = clipboard, "New York", 12, L2;
	fontset = completions, "Courier", 16, L2, nowordwrap;
	fontset = network, "Courier", 10, L2, nowordwrap;
	fontset = graphlabel, "Courier", 12, L2, nowordwrap;
	fontset = special1, "New York", 12, L2, nowordwrap;
	fontset = special2, "New York", 12, L2, center, nowordwrap;
	fontset = special3, "New York", 12, L2, right, nowordwrap;
	fontset = special4, "New York", 12, L2, nowordwrap;
```

fontset = special5, "New York", 12, L2, nowordwrap;]
:[font = subtitle; inactive; startGroup;]
A Tutorial For Solving
:[font = title; inactive;]
Problems in Operations Research
:[font = text; inactive; endGroup; center;]
by
Kris Kamisetty
CIM Center
:[font = section; inactive; startGroup; Cclosed;]
The Assignment Model
:[font = subsection; inactive; startGroup;]
Introduction
:[font = text; inactive; endGroup;]
This problem consists of allocating some resources to some operating patterns in such a way that the costs of the assignments are minimized, each resource is assigned exactly once. Hungarian algorithm is used to solve the problem.

This tutorial assumes that the given problem will be in a square matrix form and that the values given in the matrix are COSTS. If the original problem is given in terms of profits, the user will have to multiply the profit numbers by -1 and then use the tutorial.
:[font = subsection; inactive; startGroup; Cclosed;]
Example1
:[font = text; inactive;]
This section steps you through an example problem. The following subsections are arranged logically so that you can perceive the operations performed to arrive at an optimum solution. Please note that there may typically be many more alternate optimum solutions.
:[font = subsubsection; inactive; startGroup; Cclosed;]
Loading the Function Definitions
:[font = input; endGroup;]
<<hungarian.m
:[font = subsubsection; inactive; startGroup; Cclosed;]
Initialization
:[font = input; startGroup;]
initialization
:[font = print; inactive; endGroup; endGroup;]
INITIALIZATION DONE
:[font = subsubsection; inactive; startGroup; Cclosed;]
Data Input Process
:[font = input; startGroup;]
inputphase
:[font = print; inactive; endGroup; endGroup;]
2 10 9

15 4 14

13 14 16
:[font = subsubsection; inactive; startGroup; Cclosed;]
Row Reduction by the minimum of the row elements
:[font = input; startGroup;]
rowreduction

:[font = print; inactive; endGroup; endGroup;]
0 8 7

11 0 10

0 1 3
:[font = subsubsection; inactive; startGroup; Cclosed;]
Column Reduction by the minimum of the column elements
:[font = input; startGroup;]
columnreduction
:[font = print; inactive; endGroup; endGroup;]
0 8 4

11 0 7

0 1 0
:[font = subsubsection; inactive; startGroup; Cclosed;]
Checking for all the zeroes that the matrix has at this point
:[font = input; startGroup;]
zerocheck
:[font = print; inactive; endGroup; endGroup;]
11
22
31
33
:[font = subsubsection; inactive; startGroup; Cclosed;]
What kind of solution do we have now?
:[font = input; startGroup;]
whatdoithink
:[font = print; inactive; endGroup; endGroup;]
POSSIBLE OPTIMUM goto OPTIMAL function
:[font = subsubsection; inactive; startGroup; Cclosed;]
Check for Optimum solution and find the SOLUTION
:[font = input; startGroup;]
optimal
:[font = print; inactive; endGroup; endGroup; endGroup;]
OPTIMAL
11
22
33
:[font = subsection; inactive; startGroup; Cclosed;]
Example2
:[font = subsubsection; inactive; startGroup; Cclosed;]
Loading the Function Definitions
:[font = input; endGroup;]
<<hungarian.m
:[font = subsubsection; inactive; startGroup; Cclosed;]
Initialization
:[font = input; startGroup;]
initialization

:[font = print; inactive; endGroup; endGroup;]
INITIALIZATION DONE
:[font = subsubsection; inactive; startGroup; Cclosed;]
Data Input Process
:[font = input; startGroup;]
inputphase
:[font = print; inactive; endGroup; endGroup;]
2 10 9 7

15 4 14 8

13 14 16 11

4 15 13 19
:[font = subsubsection; inactive; startGroup; Cclosed;]
Row Reduction by the minimum of the row elements
:[font = input; startGroup;]
rowreduction
:[font = print; inactive; endGroup; endGroup;]
0 8 7 5

11 0 10 4

2 3 5 0

0 11 9 15
:[font = subsubsection; inactive; startGroup; Cclosed;]
Column Reduction by the minimum of the column elements
:[font = input; startGroup;]
columnreduction
:[font = print; inactive; endGroup; endGroup;]
0 8 2 5

11 0 5 4

2 3 0 0

0 11 4 15
:[font = subsubsection; inactive; startGroup; Cclosed;]
Checking for all the zeroes that the matrix has at this point
:[font = input; startGroup;]
zerocheck
:[font = print; inactive; endGroup; endGroup;]
11
22
33
34
41
:[font = subsubsection; inactive; startGroup; Cclosed;]
What kind of solution do we have now?

```
:[font = input; startGroup; ]
whatdoithink
:[font = print; inactive; endGroup; endGroup; ]
POSSIBLE OPTIMUM goto OPTIMAL function
:[font = subsubsection; inactive; startGroup; Cclosed; ]
Check for Optimum solution and find the optimal solution
:[font = input; startGroup; ]
optimal
:[font = print; inactive; endGroup; endGroup; ]
NONOPTIMAL still go to Nonoptimal
:[font = subsubsection; inactive; startGroup; Cclosed; ]
Modify the matrix by the HUNGARIAN way
:[font = input; startGroup; ]
nonoptimal
:[font = output; inactive; output; endGroup; endGroup; ]
TableForm[{{0, 6, 0, 3}, {13, 0, 5, 4}, {4, 3, 0, 0},
  {0, 9, 2, 13}}]
;[o]
0   6   0   3

13  0   5   4

4   3   0   0

0   9   2   13
:[font = subsubsection; inactive; startGroup; Cclosed; ]
Check for the zeroes again in the new matrix
:[font = input; startGroup; ]
zerocheck
:[font = print; inactive; endGroup; endGroup; ]
11
13
22
33
34
41
:[font = subsubsection; inactive; startGroup; Cclosed; ]
Check for the optimum solution again
:[font = input; startGroup; ]
optimal
:[font = print; inactive; endGroup; endGroup; endGroup; ]
OPTIMAL
13
22
34
41
:[font = subsection; inactive; startGroup; Cclosed; ]
Tutorial for solving a given problem
:[font = subsubsection; inactive; startGroup; Cclosed; ]
Introduction
```

:[font = text; inactive; endGroup;]
This tutorial allows you to solve any assignment problem you have, of any order. You will have to start the program by first clicking the INITIALIZATION cell. This clears out all the previous results stored. Then you should click the DATA INPUT cell. First, you will be asked to give the order of the matrix. i.e. 3,4,5 etc.

Then, click the ROW REDUCTION, COLUMN REDUCTION, ZEROCHECK, WHATDOITHINK cells in that order. Depending on the message given by the last function, you will want to go either to OPTIMAL or NONOPTIMAL cell. If you are directed to OPTIMAL and OPTIMAL says "optimal", you will get the final solution and you are done. But, if you get the message "Non optimal still...go to NONOPTIMAL", then click NONOPTIMAL, ZEROCHECK, OPTIMAL in that order. Repeat the 3 function process until you are done. The Tutorial has appropriate messages at every point and so, please rest assured that you WILL get the solution even if you do not clearly comprehend what was just said in these two paragraphs.

Whenever the tutorial says click the command etc., you are expected to place the cursor anywhere in that command line and press "Enter" ...not "Return" . Good Luck!
:[font = subsubsection; inactive; startGroup; Cclosed;]
STEP1 : LOAD THE DEFINITIONS
:[font = text; inactive;]
This is an essential step before solving any problem using this tutorial. This step loads the necessary function definitions stored elsewhere.
:[font = input; endGroup;]
<<hungarian.m
:[font = subsubsection; inactive; startGroup; Cclosed;]
STEP2 : INITIALIZATION
:[font = text; inactive;]
This step lets you clear out all the previous definitions and results and start afresh. This will have to be performed every time you would like to solve a new problem or if you would like to re-solve the problem that has been just solved.
:[font = input; endGroup;]
initialization
:[font = subsubsection; inactive; startGroup; Cclosed;]
STEP3 : DATA INPUT
:[font = text; inactive;]
This step when performed, opens a dialog box for you to input the order of the matrix and the cost elements of the matrix. You will have to press "Return" and not "Enter" after each input. The cost elements should be entered row by row. i.e. If you have matrix of order 2, then you will enter the elements in the following order:

row 1 & Column 1
row 1 & Column 2
row 2 & Column 1
row 2 & Column 2

If you make a mistake in inputting the numbers, go back to INITIALIZATION and perform that and the following steps once again. You don't have to LOAD the definitions again, though.
:[font = input; endGroup;]
inputphase
:[font = subsubsection; inactive; startGroup; Cclosed;]
STEP4 : ROW REDUCTION
:[font = text; inactive;]
This step finds the minimum element in each row and subtracts that number from each element of the respective row.These are nothing but valid matrix row operations which point at a possible minimum cost optimum solution.
:[font = input; endGroup;]
rowreduction

:[font = subsubsection; inactive; startGroup; Cclosed;]

STEP5 : COLUMN REDUCTION

:[font = text; inactive;]

This step finds the minimum element in each column and subtracts that number from each element of the respective column.These are nothing but valid matrix column operations which point at a possible minimum cost optimum solution.

:[font = input; endGroup;]

columnreduction

:[font = subsubsection; inactive; startGroup; Cclosed;]

STEP6 : CHECKING FOR ZEROES

:[font = text; inactive;]

This step when performed, enumerates all the zeroes in the matrix resulting from the row and column operations in steps 4 and 5. The rest of the tutorial involves manipulating zeroes to see if we have an optimum solution or to tell us what to do if we still have a nonoptimum solution.

:[font = input; endGroup;]

zerocheck

:[font = subsubsection; inactive; startGroup; Cclosed;]

STEP7 : WHERE DO WE GO NEXT?

:[font = text; inactive;]

This step is designed to aid you to know where to go next. When performed, you will get a message. If you get, "Possible Optimum go to OPTIMAL function" then click the input command in STEP8. If you get, "Non Optimal yet..go to NONOPTIMAL function", then click the input command in STEP9. You will be directed further by those steps till you arrive at a final OPTIMAL solution.

:[font = input; endGroup;]

whatdoithink

:[font = subsubsection; inactive; startGroup; Cclosed;]

STEP8 : CHECKING FOR OPTIMUM SOLUTION

:[font = text; inactive;]

This step checks for an optimum solution and if it finds one comes back and tells you

"OPTIMAL" followed by the solution in terms of the row and column numbers. If it tells you "nonoptimal still...go to NONOPTIMAL", then you should click the command in STEP9.

:[font = input; endGroup;]

optimal

:[font = subsubsection; inactive; startGroup; Cclosed;]

STEP9 : MODIFICATION OF THE MATRIX

:[font = text; inactive;]

This step modifies the matrix from the STEP5 according to the HUNGARIAN Algorithm. At the end of the process, you will see the resulting matrix. You will have to click the command in STEP6 again followed by STEP7. Just follow the directions from there on in an iterative fashion until you arrive at an optimum solution.

:[font = input; endGroup;]

nonoptimal

:[font = subsubsection; inactive; startGroup; Cclosed;]

FINAL STEP

:[font = text; inactive; endGroup; endGroup;]

At this point, you will have an OPTIMUM solution in your hands. If you want to save the calculations for yourself, then choose "SAVE AS" in the "WINDOW" menu and give the file a new name of your choice.

If you just want to quit without saving anything, then just choose "QUIT" in the main menu OR close the window by clicking on X and clicking NO in the dialog box that appears.

Thank you for using this tutorial!

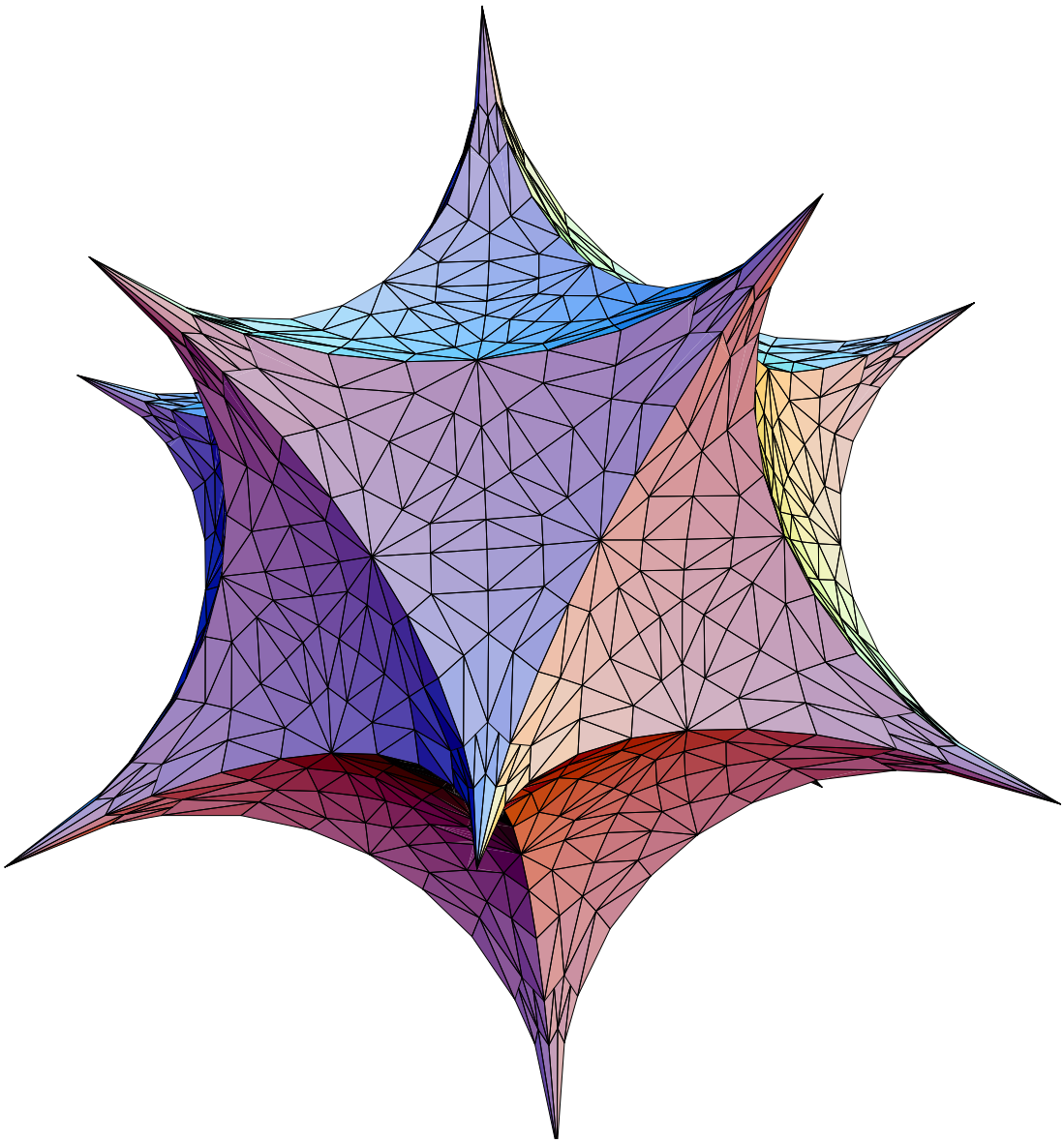
```
:[font = subsection; inactive; startGroup; Cclosed; ]
Implementation Aspects
:[font = subsubsection; inactive; startGroup; Cclosed; ]
Initialization Function
:[font = input; endGroup; ]

initialization := Block[{},
                  k=.;v=.;g=.;h=.;i=1;j=1;
                  Clear[cost,min,rred,cmin,cred,check,check1];

                  Print[INITIALIZATION," ",DONE]
                 ]
:[font = subsubsection; inactive; startGroup; Cclosed; ]
Input Data Function
:[font = input; endGroup; ]
inputphase := Block[{},
                 n=Input["What is the square matrix order?"];
                 Do[cost[xx,yy]=Input["Enter row by row,one
                                      element at a time"],
                                      {xx,n},{yy,n}
                  ];
                 costmatrix = Table[cost[y,z], {y,n},{z,n}];
                 Print[TableForm[costmatrix]]
               ]
:[font = subsubsection; inactive; startGroup; Cclosed; ]
Row Reduction Function
:[font = input; endGroup; ]
rowreduction := Block[{},
                  Clear[min,rred];
                  Do[min[k] = Min[costmatrix[[k]]],{k,1,n}];
                  Do[ rred[k] = costmatrix[[k]] - min[k],{k,1,n}];
                  rowreducedmatrix = Table[rred[k],{k,n}];
                  Print[TableForm[rowreducedmatrix]]
                ]
:[font = subsubsection; inactive; startGroup; Cclosed; ]
Column Reduction Function
:[font = input; endGroup; ]
columnreduction := Block[{},
                    Clear[cmin,cred];
                    p = Transpose[rowreducedmatrix];
                    Do[cmin[k] = Min[p[[k]]],{k,1,n}];
                    Do[cred[k] = p[[k]] - cmin[k],{k,1,n}];
                    q = Table[cred[k],{k,n}];
                    colreducedmatrix=Transpose[q];
                    Print[TableForm[colreducedmatrix]]
                  ]
:[font = subsubsection; inactive; startGroup; Cclosed; ]
Zero Check Function
:[font = input; endGroup; ]
zerocheck := Block[{},
```

```
                Clear[check,check1];v=.;k=.;g=.;h=.;i=1;j=1;
                Do[
                  If[ colreducedmatrix[ [k,v] ] == 0, check[i]=k;
                     check1[j]=v;Print[k,v];i++;j++
                   ],{k,n},{v,n}
                  ];
                zerorows=Table[check[g],{g,1,i-1}];
                zerocols=Table[check1[h],{h,i-1}];
               ]
:[font = subsubsection; inactive; startGroup; Cclosed; ]
Where do we go next from here?
:[font = input; endGroup; ]
whatdoithink :=
          If[Length[zerorows]>=n,
            Print[POSSIBLE," ",OPTIMUM," ",goto," ",OPTIMAL," ",
            function],
            Print[NOT," ",OPTIMUM," ",YET," ",goto," ",
            NONOPTIMAL," ",function]
           ]
:[font = subsubsection; inactive; startGroup; Cclosed; ]
Assignment Function
:[font = input; endGroup; ]
assignment := Block[{l1,l2,s,x={},y={}},
                If[iter != 0,
                   t=Append[t,e[[iter]]];
                   u=Append[u,f[[iter]]];
                   l1=Length[e];
                   x={};y={};x=e;y=f;
                   Do[If[First[x] == Last[t],
                     e=Drop[e,{1,1}];x=RotateLeft[x];
                     f=Drop[f,{1,1}];y=RotateLeft[y],
                     x=RotateLeft[x];y=RotateLeft[y];
                     e=RotateLeft[e];f=RotateLeft[f] ],{s,l1}
                     ];
                   x={};y={};x=e;y=f;l2=Length[e];
                   Do[If[First[y] == Last[u],
                      f=Drop[f,{1,1}];y=RotateLeft[y];
                      e=Drop[e,{1,1}];x=RotateLeft[x],
                      y=RotateLeft[y];x=RotateLeft[x];
                      f=RotateLeft[f];e=RotateLeft[e] ],{s,l2}
                     ]
                ]; (* end of outermost IF *)
                If[Length[e] == 0,iter=0,iter=1]
               ] (* end of BLOCK *)
:[font = subsubsection; inactive; startGroup; Cclosed; ]
Optimal Check Function
:[font = input; endGroup; ]
optimal := Block[{soln,l,r},
              l=Length[zerorows];objective=0;
              For[i=1,i<=l,i++,
```

```
                t={};u={};tu={};
                e={};f={};e=zerorows;f=zerocols;iter=i;
                Do[assignment,{soln,n}];
                tu=Intersection[t,u];
                If[Length[tu] == n, Print[OPTIMAL];
                        Do[Print[t[[r]],u[[r]]];
                objective += costmatrix[[t[[r]],u[[r]]]],{r,n}];
                Print[OBJECTIVE," ",COST," ",IS," ",objective];
                        Break[]
                  ]
                ];
              If[i==l+1,Print[NONOPTIMAL," ",still," ",go," ",
                    to," ",Nonoptimal]]
            ]

:[font = subsubsection; inactive; startGroup; Cclosed; ]
Non Optimal Modification Function
:[font = input; endGroup; ]
nonoptimal :=
Block[ {z,rows={},cols={},uncovered={},common={},iteru,iterc,q2={}},
      z=n;q2=colreducedmatrix;timen=1;timen++;
      While[z != 0,
        For[row=1,row <= n,row++,
         If[FreeQ[rows,row],
          If[Count[q2[[row]],0] ==z, rows=Append[rows,row];
                        q2[[row]]=q2[[row]]+1
                            ] (*endif*)
          ] (*end outer If *)
         ]; (* end of For loop *)

         q2=Transpose[q2];
         For[col=1,col<=n,col++,
          If[FreeQ[cols,col],
           If[Count[q2[[col]],0] ==z, cols=Append[cols,col];
                          q2[[col]]=q2[[col]]+1
                           ](*endif*)
           ] (*end outer If *)
          ]; (*end of FOR loop *)
         q2=Transpose[q2];
       z-- ]; (*end of WHILE loop *)
   For[row=1,row<=n,row++,
     For[col=1,col<=n,col++,
       If[FreeQ[rows,row] && FreeQ[cols,col],
         uncovered=Append[uncovered,colreducedmatrix[[row,col]]]
        ];
       If[MemberQ[rows,row] && MemberQ[cols,col],
         common=Append[common,colreducedmatrix[[row,col]]]
        ]
      ] (* end of inner For loop *)
     ]; (* end of outer For loop *)
```

```
minuncov=Min[uncovered];iteru=1;iterc=1;
For[row=1,row<=n,row++,
 For[col=1,col<=n,col++,
  If[FreeQ[rows,row] && FreeQ[cols,col] && iteru<=Length[uncovered],
      colreducedmatrix[[row,col]] = uncovered[[iteru]]-minuncov;
      iteru++
    ];
  If[MemberQ[rows,row] && MemberQ[cols,col] && iterc <= Length[common],
      colreducedmatrix[[row,col]] = common[[iterc]]+minuncov;
      iterc++
    ]
   ] (* end of inner for loop *)
  ];  (* end of outer for loop *)
 TableForm[colreducedmatrix]

 ] (*end of Block *)
:[font = subsubsection; inactive; startGroup; Cclosed; ]
Command for Saving the function definitions in a FILE.m (Change the filename)
:[font = input; endGroup; ]
Save["hungarian.m",initialization,inputphase,rowreduction,
      columnreduction,zerocheck,whatdoithink,assignment,
      optimal,nonoptimal]
:[font = subsubsection; inactive; startGroup; Cclosed; ]
 Command for CLEARING the function definitions
:[font = input; endGroup; endGroup; endGroup; ]
Clear[initialization,inputphase,rowreduction,
      columnreduction,zerocheck,whatdoithink,assignment,
      optimal,nonoptimal]
^*)
```

A Mathematica Primer



David R. Hieb, University of Colorado, Boulder.
(davehieb@boulder.Colorado.EDU)

[Mathematica? You mean that math program from Steven what-is-his-name that I still haven't taken the time to look at?]
[*Ed Note : Pictures have been resized to fit FrameMaker Columns. Sorry.*]

If you find yourself uttering those exact words or something very close, this article is for you. And for those of you that have actually taken the plunge, double clicked on the Stellated Icosahedron, and played around for a while - keep reading, you might learn something new.

What I would like to do in the next several pages is present to you the power of Mathematica in a simple and clean format that will allow you to sit down and get results in the least amount of time. I plan to detail the highlights as well as the stumbling blocks and at the same time bestow upon you a working knowledge of this rich and elegant program.

What is Mathematica?

Mathematica is a highly interactive system for doing mathematics in the numerical, symbolic and graphical domain. Mathematica provides the casual user with an incredible selection of functions and capabilities and yet goes beyond that to supply the advanced user with a system to represent mathematical knowledge, define functions and program in a high-level interpreted language (much like C or Pascal).

And as one Computer Entrepreneur put it:

"Mathematica will revolutionize the teaching and learning of math by focusing on the prose of mathematics, without getting lost in the grammar"

Steven Jobs
President, NeXT, Inc.

Background of Mathematica

Mathematica, as implemented on the NeXT, consists of a C program of approximately 200,000 lines in length. Initially the code was developed in Wolfram Research's object-oriented extension of C, and then pre-compiled into standard C. Stephen Wolfram is the principal designer and architect of Mathematica and is the president of Wolfram Research Inc. Wolfram's contributions to Mathematica also include much of the kernel code development and the text for the accompanying manual "Mathematica, A System for Doing Mathematics by Computer".

Wolfram, in his colorful and fast paced life, has held faculty positions at Caltech and Princeton and is now a Professor of Physics, Mathematics and Computer Science at the University of Illinois. By utilizing the expertise of a core of brilliant mathematicians and scientists, Wolfram was able to produce truly one of the finest and most complete mathematical systems available today.

Let's get started

Mathematica has both a command line and graphical interactive interface. To use the command line interface just type 'math' from a Terminal/Shell window.

```
% math
In[1]:= 2+2
    Out[1]= 4
In[2]:= ^D
%
```

And you have just completed the shortest (and most boring) session of Mathematica possible. For more information on the command line interface to Mathematica, look at the math(1) man page.

Upon starting the graphical interface to Mathematica (/NextApps/Mathematica.app) you are presented with a new window and are ready to begin typing. Type in '8 * 8' and press **SHIFT RETURN** (or **ENTER** on the numeric keypad) to evaluate the expression. The **RETURN** key is used as a new line character so that long complicated expressions can wrap around on as many lines as is necessary. The excessive amount of time that Mathematica takes to compute this trivial expression is normal for the first computation of each new session and is due to having to load the kernel. The kernel is the core computational engine buried deep underneath the user interface and is responsible for the actual calculations. As you will notice, several things happened.

```
In[1]:= 8 * 8
Out[1]= 64
```

The input expression '8 * 8' was preceded with a 'In[1]:=' indicating that it was the 1st input expression and the output '64' was preceded with a 'Out[1]' indicating that it was the 1st output expression. Also, a few large right square braces started to pile up on the right of the screen, giving the input and output expressions a common but yet unique identity. These braces identify a "cell", where a cell can contain Mathematica text, commands or graphics.

Now let's take the square root of 'e' raised to the output of the last expression. Typing 'Sqrt[E ^ %]' and pressing **SHIFT RETURN** yields 'E^32'. The '%' symbol represents the output of the last expression and '%n' represents the output of expression 'n'. The carat is self explanatory and yes, 'E' is Mathematica's symbol for 2.71828.... (there are many others such as Pi, I and Infinity). All Mathematica functions start with a capital letter, spell out the complete word (almost always) and use square braces for the arguments.

Unless told otherwise, Mathematica will always return expressions in symbolic form if the input expression contains symbolic constants. We can use the numerical function N[] to get the numerical value of 'E^32'. 'N[%]' yields 7.8963 10^13. The N[] function can also have an optional n value that specifies the number of significant digits that you want displayed. I don't know about you, but the first thing I wanted to know was just how big can this n be. Here is an example of displaying Pi with a small and very large value of n. Be ready to take a coffee break if you try the later example with n == 10000 since this is not a trivial computation. Although the documentation states that you can interrupt a calculation with <**CTRL-C**> or <**CTRL-.**>, I have not been able to effectively do so and have had to resort to quiting the kernel (which means that a new kernel does not know about the previous definitions).

```
In[2]:= N[ Pi, 10 ]
Out[2]= 3.141592654
In[3]:= N[ Pi, 10000 ]
Out[3]= 3.14159265358979323846264338327950 ..... 229552498872
```

Let's define some variables for the next expression. To define a variable you simply state that "variable" = "expression". Although a variable name can contain any combination of upper/lower case letters, it is recommended that you avoid confusion with the built-in Mathematica object names by not starting a variable name with an upper case letter. Variable names can also contain (but not begin with) digits.

```
In[4]:= a = (x + y)
Out[4]= x + y
In[5]:= b = (x*y + y)
Out[5]= y + x y
```

As you might have noticed, Mathematica expresses the product of x and y as 'x y' (with a space denoting the multiplication operator) instead of 'x * y'. It might appear at first that in trying to reduce the amount of typing Mathematica has allowed a grammar that introduces confusion. To make sure that this does not become a problem, here is my caveat on grammar: 'x y' really means 'x*y' whereas 'xy' refers to the variable 'xy'. While both '5 y' and '5y' mean '5*y' (variables cannot begin with a digit). Please note that for the rest of this article I will explicitly show all operators.

```
In[6]:= c = a * b
Out[6]= (x + y) (y + x*y)
```

Since x and y have no value yet, Mathematica leaves the result in the form of a symbolic expression. You can always check the value of any object (variable, function, definition) with the following '?name' syntax.

```
In[7]:= ?a
    a
    a = x + y
In[8]:= ?c
    c
    c = (x + y) (y + x*y)
```

If we want to expand 'c' we can determine if an expand function exists and if so, what is the correct syntax.

```
In[9]:= ?Expand
    Expand[expr] expands out products and positive powers in expr.
In[10]:= Expand[ c ]
Out[10]= x*y + x^2*y + y^2 + x*y^2
```

Variables (and any objects that we have yet to discuss) can be saved in a definition file by using the Save[] function. 'Save["abcVarDef", a, b, c]' would save variables a, b and c in the file "abcVarDef" (in your home directory). To instantiate variables a, b and c from their definition file use the '<<name' syntax, where name in this case is "abcVarDef". Mathematica also supports other Operating System operations such as cat'ing a file with the '!!name' syntax and executing a UNIX command with the '!command' syntax.

Let me test your confidence with a complicated example to make sure you know how Mathematica evaluates expressions.

'2mn / 6Pi / 5E / 3!' evaluates to (E*Pi*mn)/90. Right?????

Division '/' has higher precedence than multiplication '*' such that '/' effectively binds it's operands more tightly than '*'. Also, division is left associative which means that the expression is evaluated from left to right. Therefore the above expression actually turns out to be '2 * (mn / 6) * (Pi / 5) * (E / 3!)'. The '2mn' is actually 2 times the variable 'mn' after 'mn' has been divided by 6. Let me note that as usual, the factorial operator '!' has higher precedence than both '/' and '*'. If remembering the precedence rules seem like too much of a hassle, as always, you can use explicit parenthesis.

One of the very important methods in which Mathematica deals with user input is through the use of symbolic expressions. The Calculus student should be quite happy with a full range of functions that are provided.

D[], Dt[]	Partial and total derivatives.
Integrate[]	Definite and indefinite integrals.
Sum[], Product[]	Sum and product of a function.
Series[]	Power series expansion of a function.

Limit[] Limit of a function.

Here is an example that uses the Integrate[] function to determine the Flux (in Amperes) out of a volume defined by the surface (in spherical coordinates):

0 <= r <= 2, Pi/3 <= theta <= Pi, 0 <= phi <= 2Pi

and the current density function (vector field) **F**:

F = <r*Cos[theta], r^3, r^2>

The Flux through a surface is calculated by taking the surface integral of the dot product of the current density function **F** and the surface-element vector d**S.** In this example the current density function **F** is a measure of the instantaneous rate of flow of charge per unit cross-sectional area in the vicinity of any point P in space. The surface-element vector d**S** is actually composed of two separate vectors.

$d\mathbf{S}_1$ = <r*Dtheta * r*Sin[theta]*Dphi, 0, 0>

$d\mathbf{S}_2$ = <0, -Dr * r*Sin[theta]*Dphi, 0>

Where Dr, Dtheta and Dphi are the differential elements in their respective spherical variables.

To determine a resultant expression for the problem you would normally have to tediously evaluate two separate double integrals and use the "Integration by Parts" substitution on one of them. The Integrate[] function can effortlessly provide the answer.

```
In[11]:= surfaceFluxA = Integrate[r*Cos[theta] * r^2*Sin[theta],
          theta, Pi/3, Pi, phi, 0, 2Pi]
Out[11]= (-3*Pi*r^3)/4
In[12]:= surfaceFluxB = Integrate[r^3 * -r*Sin[theta], r, 0, 2, phi, 0, 2Pi]
Out[12]= (-64*Pi*Sin[theta])/5
In[13]:= r = 2
In[14]:= theta = Pi/3
In[15]:= totalFlux = surfaceFluxA + surfaceFluxB
Out[15]= -6*Pi - (32*3^(1/2)*Pi)/5
```

Mathematica also provides a suite of equation solving functions such as Solve[], Eliminate[] and Reduce[]. Let's solve an equation like 'x^9 + 1 = 0'.

```
In[16]:= equ1 =  x^9 + 1 == 0
```

Note the equality sign '==' which asks a question verses the assignment sign '=' which is declarative (just think C).

```
In[17]:= Solve[equ1, x]
```

```
Out[17]= x -> -1, x -> -E^((2*I)/9*Pi),
         x -> -E^((4*I)/9*Pi), x -> -E^((2*I)/3*Pi),
         x -> -E^((8*I)/9*Pi), x -> -E^((10*I)/9*Pi),
         x -> -E^((4*I)/3*Pi), x -> -E^((14*I)/9*Pi),
         x -> -E^((16*I)/9*Pi)
In[18]:= N[%]
Out[18]= x -> -1.,
        x -> -0.766044 - 0.642788*I,
        x -> -0.173648- 0.984808*I,
        x -> 0.5 - 0.866025*I,
        x -> 0.939693 - 0.34202*I,
        x -> 0.939693 + 0.34202*I,
        x -> 0.5 + 0.866025*I,
        x -> -0.173648 + 0.984808*I,
        x -> -0.766044 + 0.642788*I
```

Mathematica will always be able to return an explicit formula to an algebraic equation if the equation is in one variable and the highest powered variable is less than 5. If these conditions are not met then Mathematica will try it's best and either succeed (as it did with the above equation) or present the roots in a symbolic form such that you can still find numerical solutions.

The Solve[] function can also solve simultaneous equations.

```
In[19]:= equ1 = 2x - 3y + 4z == 8
In[20]:= equ2 = x + 2y - 6z == -13
In[21]:= equ3 = -3x - y - z == -8

In[22]:= Solve[equ1, equ2, equ3, x, y, z]
Out[22]= x -> 1, y -> 2, z -> 3
```

Given n equations, the Eliminate[] function can eliminate n-1 (or fewer) variables.

```
In[23]:= equ4 = 1.2w - 3.4x + 5.6y + 7.8z == 9.0
In[24]:= equ5 = 0.9w + 8.7x - 6.5y - 4.3z == 2.1
In[25]:= Eliminate[equ4, equ5, w]
Out[25]= 1.*x == -0.41333 + 0.95111*y + 0.90222*z
```

The Reduce[] function can be used to generate a set of simplified equations which include all possible solutions. Many times this can provide a more complete answer than Solve[]. The following example uses both functions to solve the formula for the slope of a line.

```
In[26]:= Solve[m*x + b == 0, x]
```

```
Out[26]= x -> -(b/m)

In[27]:= Reduce[m*x + b == 0, x]
Out[27]= m != 0 && x == -(b/m) || m == 0 && b == 0
```

As you will notice the above syntax is very C-like, and may be read as follows: m times x plus b equals zero when:

* m is not equal to zero AND x equals -(b/m)
  OR
* m equals zero AND b equals zero

Mathematica has implemented all the C (programming language) logical operations such as: x == y, x != y, !x, x && y, x || y, x >= y, etc.

A list is a comma separated collection of one or more expression(s) that are enclosed by curly braces. In Mathematica, the first element of a list has an "index" of 1 (unlike C). Mathematica functions can treat the entire list as a single object or refer to any specific element(s). There are more list processing functions than you will probably ever need (however Lisp users will appreciate the rich set of primitives). Here is a taste of some of the list processing functions: Length, First, Last, Take, Drop, Prepend, Append, Insert, Join, Union, Intersection, Complement, Sort, Reverse, RotateLeft,Right, Transpose, Flatten, Partition, Apply and there are still more. For those of you who like working with lists, here is an example that demonstrates some of the list processing capabilities.

```
In[28]:= listA = x, x^2, x^3, 8, 10, 10, 3, 5
In[29]:= listB = a+b, y, 2, 2, 8, 6, 4, 4, y^2, x+y
In[30]:= listC = w!, 1, x!, 2, y!, 3, z!, 4
In[31]:= listD = 11, 12, 13, 14
In[32]:= listE = 12, 13, 14, 11

In[33] = answer = Append[Insert[Insert[Union[Join[First[Partition[
    Flatten[Reverse[Join[listD, listE]]], 4]], First[Sort[listC]],
   Take[RotateLeft[listB, 2], 2, 5], Take[listA, 4, 8]]]], 7, 7], 9, -6], 15]

Out[33] = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15
```

By the way, there is no explicit function for turning an element into a list. To do so just enclose the element in curly braces. This indeed turns out to be more convenient than using a function to turn an element (that some function just returned to you) into a list so that an enclosing function (that only accepts list(s)) will not scream.

In Mathematica, a vector is represented by a single list and a matrix is represented by lists of lists. Mathematica has gone to great lengths to provide easy manipulation of vectors and matrices.

```
In[34]:= vecA = 2x, y, 5z
In[35]:= vecB = -x, 2y, -3z
In[36]:= vecC = x, y

In[37]:= vecA + vecB
Out[37] = x, 3*y, 2*z

In[38]:= (2*vecA) . (-3*vecB)
Out[38] = 12*x^2 - 12*y^2 + 90*z^2

In[39]:= vecA * vecB
Out[39] = -2*x^2, 2*y^2, -15*z^2
```

Note that the symbol for the dot (scalar) product of vectors is "." and the symbol for the multiplicative (vector) product of vectors is "*".

It was sure nice to see that Mathematica gave me a descriptive warning (instead of a "core dump") which identified the problem when I tried to add two vectors of unequal length.

```
In[40]:= vecA + vecC
    Thread::badlen:
    Objects of unequal length in 2 x, y, 5 z + x, y cannot be combined.
Out[40] = x, y + 2*x, y, 5*z
```

Let's use the vector/matrix facilities to solve the set of simultaneous equations with which we previously used the Solve[] function. Given the vector/matrix representations, we know that matrixA . **x** == **c**.

```
In[41]:= matrixA = 2, -3, 4, 1, 2, -6, -3, -1, -1
In[42]:= c = 8, -13, -8

In[43]:= x = Inverse[matrixA] . c
Out[43] = 1, 2, 3
```

Of course Mathematica has already anticipated the situation and provides LinearSolve[] as a single function solution. LinearSolve[] returns the vector **x** which solves the matrix equation matrixA . **x** == **c**.

```
In[44]:= LinearSolve[matrixA, c]
Out[44] = 1, 2, 3
```

There are many more functions that deal with vectors and matrices including a whole suite on eigenvalues/eigenvectors. From Linear Algebra we know that for a given n ¥ n matrix matrixA, an eigenvalue, eValue, and eigenvector, eVector, satisfy the following equation:

```
matrixA . eVector = eValue * eVector
```

To determine the eValues we would solve the "characteristic equation" Determinant(eValue * I - matrixA) == 0.

```
In[45]:= matrixA = 2, -2, 2, 0, 1, 1, -4, 8, 3
In[46]:= eValues =
              Solve[ Det[eValue * IdentityMatrix[3] - matrixA] == 0, eValue]
Out[46] = eValue -> 3, eValue -> 2, eValue -> 1
```

We can now determine eVectors that correspond to each $eValue_i$ from the following equation:

$eValue_i$ * I - matrixA == 0.

The first thing you will probably try is the LinearSolve[] function:

LinearSolve[$eValue_i$ * IdentityMatrix[3] - matrixA, 0, 0, 0].

But remember that the determinant of 'eValue * I - matrixA' equals zero and for a particular value of **c**, there cannot be a vector **x** which satisfies m . **x** = **c** when the determinant(m) == 0. When this type of situation occurs it is still possible to find a set of vectors which satisfy m . **x** == 0. These vectors can be obtained with the NullSpace[] function and are called the 'kernel' or 'null space' of the matrix m.

```
In[47]:= eVector1 = NullSpace[1 * IdentityMatrix[3] - matrixA ]
Out[47] = 2, 1, 0
In[48]:= eVector2 = NullSpace[2 * IdentityMatrix[3] - matrixA ]
Out[48] = 9, 4, 4
In[49]:= eVector3 = NullSpace[3 * IdentityMatrix[3] - matrixA ]
Out[49] = 2, 1, 2
```

If, however, you're more interested in the quickest method of finding the Eigenvalues and Eigenvectors, the Eigenvalue[] and Eigenvector[] functions work quite well.

```
In[50]:= Eigenvalues[matrixA]
Out[50] = 3, 2, 1
In[51]:= Eigenvectors[matrixA]
Out[51] = 2, 1, 0, 9, 4, 4, 2, 1, 2
```

All input into Mathematica is considered an expression and all operations on expressions are really just the application of "transformation rules". Mathematica repeatedly subjects the input expression to all the applicable transformation rules that it knows about and returns the result once there are no more transformation rules to apply and/or the result ceases to change. As you have already seen, Mathematica has a voluminous portfolio of built-in transformation rules and on top of that allows the user to augment and even over-ride the environment with his/her own transformation rules such as mathematical relations and functions.

For example, it is trivial to define a series of mathematical relationships which show the simplification rules for the logarithm function, log(x).

```
In[52]:= log[1] = 0
In[53]:= log[x_ y_] := log[x] + log[y]
In[54]:= log[x_ / y_] := log[x] - log[y]
In[55]:= log[x_ ^ n_] := n * log[x]
```

Now the log[] (lowercase) function intelligently performs the following transformation:

```
In[56]:= log[g * (3 numberA) / h ^ (8 - numberB)]
Out[56]= log[3] + log[g] + log[numberA] - (8 - numberB)*log[h]
```

whereas Log[] (uppercase, Mathematica's built-in function) does not know of any such simplification.

```
In[57]:= Log[g * (3 numberA) / h ^ (8 - numberB)]
Out[57]= Log[(3*g*numberA)/h^(8 - numberB)]
```

You will notice that the first definition uses an equals "=" syntax and the second definition uses a colon-equals ":=" syntax. For a function, the equals syntax states that the right hand side (rhs) is to be evaluated only when the assignment is made, whereas the colon-equals syntax states that the rhs is to be evaluated each and every time the left hand side (lhs) is requested. Obviously a point that needs to be well understood. It is also very important to distinguish whether or not to use an underscore and what is the meaning of such syntax. The underscore (on the lhs) is a wildcard that means "any expression" and the presence of a variable (x_, y_, n_ in this example) provides a handle by which the wildcard can be referenced (on the rhs). Note that if you forget the wildcard in the lhs, the expression will be syntactically correct, but will only match an exact instance of the rule-parameter combination.

It's always a good idea to double check your definitions by using the '?name' syntax just to make sure that you get what you want.

```
In[58]:= ?log
    log
    log/: log[1] = 0
    log/: log[(x_)/(y_)] := log[x] - log[y]
    log/: log[(x_) (y_)] := log[x] + log[y]
    log/: log[(x_)^(n_) ] := n log[x]
```

The Fibonacci sequence provides a great example for user defined transformation rules (functions). The sequence is produced by the relationship:

$F_0 = 0, F_1 = 1, F_n = F_{n-1} + F_{n-2}$.

In Mathematica you would use the following syntax:

```
In[59]:= fib[0] = 0
In[60]:= fib[1] = 1
In[61]:= fib[n_] := fib[n-1] + fib[n-2]
```

Since this is a recursive problem, for fib[largeValue], we need to increase the default recursion limit of 256. Setting system values are accomplished with the "SystemParameter" = "Value" syntax.

```
In[62]:= $RecursionLimit = 65536
```

The 20th Fibonacci number would thus be generated by.

```
In[63]:= fib[20]
Out[63] = 6765
```

In looking at the defining relationship for the Fibonacci sequence, it is apparent that in computing some fib[largeValue], several Fibonacci numbers fib[m] (where m < largeValue), will be computed multiple times. This situation can be improved (speed wise) by defining a relationship that remembers it's values, therefore each value need be computed only once. The syntax is as follows:

```
In[64]:= fib[n_] := fib[n] = fib[n-1] + fib[n-2]
```

It is quite instructive to recompute 'fib[25]' and compare the astounding results. A good example to try on your own is Ackermann's function:

```
ack(i, j) =          2j:      i = 0,
                     0:       i >= 1 && j = 0,
                     2:       i >= 1 && j = 1,
ack(i-1, ack(i, j-1)):       i >= 1 && j >= 2.
```

Please note that if you are planning on entering 'ack[3,4]' (or something larger) for your first expression, you need to think about the defining relationship

a little bit longer.

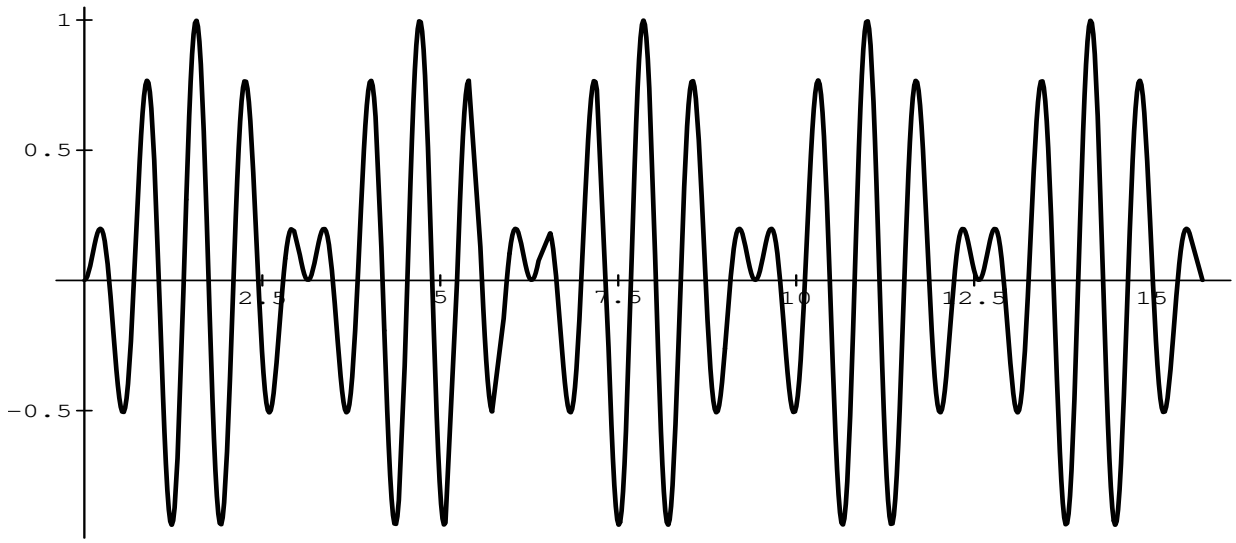
Special Topic: Graphics

Let's start with a simple 2-Dimensional graphic and then make our way to 3-D land. The most simple plotting function is intuitively named Plot[].

```
In[65]:= ?Plot
    Plot[f, x, xmin, xmax] generates a plot of f as a function of x from xmin
  to xmax. Plot[f1, f2, ..., x, xmin, xmax] plots several functions fi.
```

To plot the function 'Sin[x] * Sin[9x]' from 0 <= x <= 5Pi you would use something like the following:

```
In[66]:= Plot[ Sin[x] * Sin[9x], x, 0, 5Pi]
```

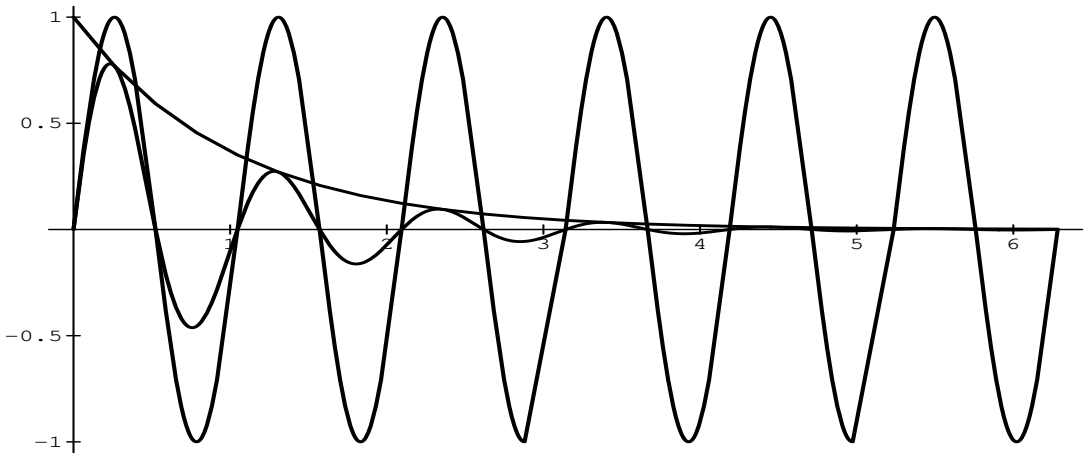


The syntax for Plot[] indicates that we can plot multiple functions by enclosing the individual comma-separated functions in a single pair of curly braces.

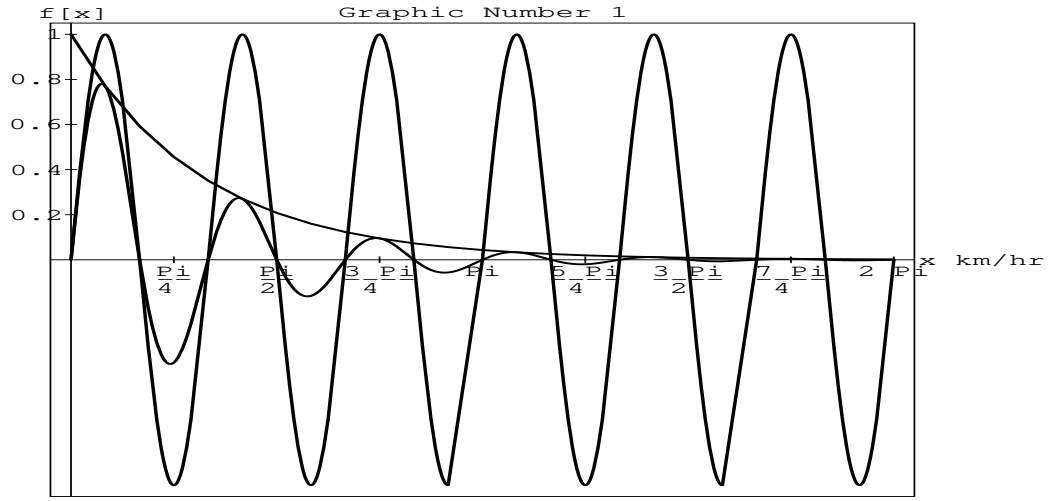
```
In[67]:= Plot[ E^-x, Sin[6x], E^-x * Sin[6x], x, 0, 2Pi]
```

As you can see, Mathematica has automatically scaled the graphic and also used several other default parameters. To manually override the default settings you can include an optional 'name -> value' set of rule(s) in the Plot[] function. To get a listing of all the options and their default values you can use the help facility '??Plot' or use the 'Options[Plot]' function. The following example illustrates some of the Plot[] options when used in conjunction with the last graphic.

```
In[68]:= Plot[ E^-x, Sin[6x], E^-x * Sin[6x], x, 0, 2Pi,
      PlotLabel -> "Graphic Number 1",
      Framed -> True,
```




```
AxesLabel -> "x km/hr", "f[x]",
AspectRatio -> 1,
Ticks -> Range[0, 2Pi, Pi/4], 0.2, 0.4, 0.6, 0.8, 1.0]
```



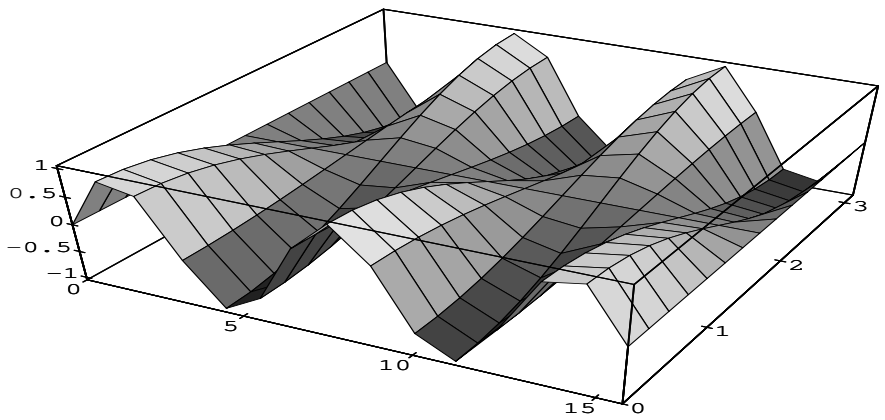
The AspectRatio option is the height-to-width ratio of the plot and by default is the inverse of the Golden Ratio - the most pleasing shape for a rectangle. The Ticks option was used to specify that the x-axis should have tick marks every Pi/4 units (the Range[] function provides the datum points every Pi/4 units, from 0 to 2Pi) and the y-axis should have tick marks at the explicit values in the curly braces.[*Ed: Aspect Ratios have been distored by Frame*]

Mathematica provides a more realistic method of manipulating these graphics (and text). Hold down the Alternate key and you will see a little right hand replace the current shape of the cursor. By simultaneously pressing on the left mouse key you can thus move an object (graphic or text) in any 2-Di-

mensional direction. You can also scale the graphics by double clicking and dragging an appropriate side or corner of the enclosing box that appears.

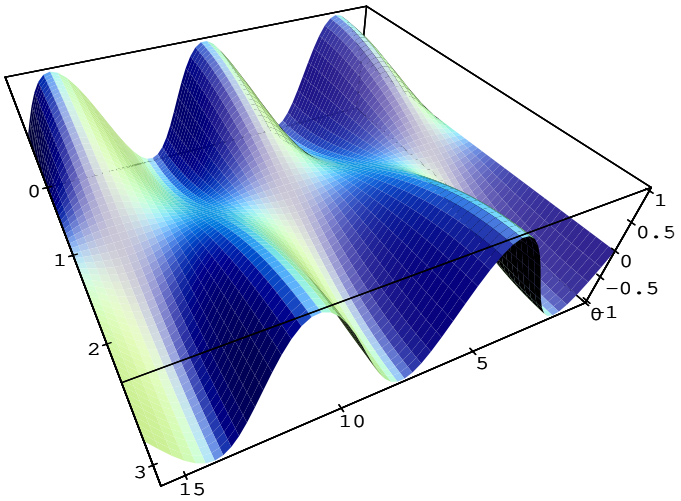
The 3-D plot function, Plot3D[], is very similar to Plot[] except that Plot3D[] allows you to specify both a x and y range and also includes a lot of fun options that it's 2-D counterpart cannot support. Here is a 3-D Sinusoidal plot using the default Plot3D[] options.

```
In[69]:= Plot3D[ Sin[x] * Cos[y], x, 0, 5Pi, y, 0, Pi]
```



Here is the same graphic with a few of the Plot3D[] options defined.

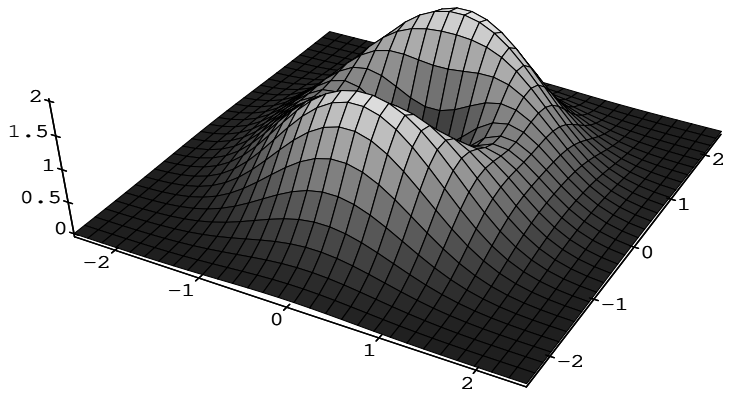
```
In[70]:= Plot3D[ Sin[x] * Cos[y], x, 0, 5Pi, y, 0, Pi,
    Mesh -> False,
    PlotPoints -> 60,
    Lighting -> True,
    ViewPoint -> 0.5, 1, 1]
```



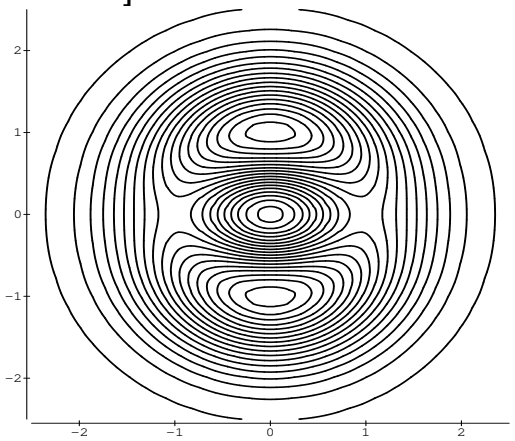
The PlotPoints option specifies the number of points in each direction at which to sample the function. The Lighting option specifies whether to include simulated illumination. The ViewPoint option allows you to select the x, y, z coordinates of your view of the graphic relative to the center of the box. Note that this option is scaled so that the longest side of the box is 1 unit in length. Once you get your bearings straight, it is trivial to confirm that we are viewing the graphic from a vantage point which is 1/2 unit in the plus x direction, 1 unit in the plus y direction and 1 unit in the plus z direction.

Mathematica also supports several other types of plotting that allow you to look at the same data in a whole new perspective. The next example is a Plot3D[] graphic and it's associated contour plot.

```
In[71]:= Plot3D[ (x^2 + 2*y^2)*E^(1 - x^2 - y^2), x, -2.5, 2.5, y, -2.5,
2.5,
        PlotPoints -> 30,
        Boxed -> False]
```

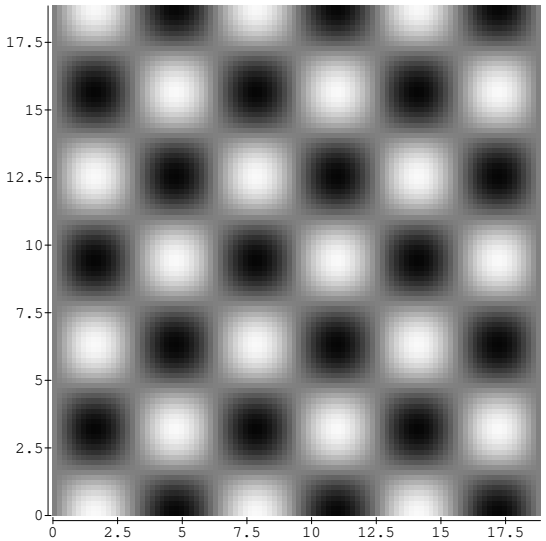



```
In[72]:= ContourPlot[ (x^2 + 2*y^2)*E^(1 - x^2 - y^2),
        x, -2.5, 2.5, y, -2.5, 2.5,
        PlotPoints -> 40,
        ContourLevels -> 20]
```



I have included a density plot as a final example of the different plotting possibilities that Mathematica has to offer. A density plot changes the shade of a datum point depending on the "height" of the function at that point.

```
In[73]:= DensityPlot[ Sin[x] * Cos[y], x, 0, 6Pi, y, 0, 6Pi,
    PlotPoints -> 100,
    Mesh -> False]
```



The limited examples that I have prepared here in no way cover the incredible graphic possibilities that Mathematica has to offer. Mathematica also has a complete set of low level graphic primitives that allow you to graph much more complicated entities than these.

Any graphic can be saved to a (Postscript) file by one of the following methods: using the pasteboard or the Display[] function. The pasteboard is the quickest and most convenient method for moving the (Postscript) graphics between NeXT software packages. Just select the graphic with your cursor and press **CMD**-**Copy** to copy the graphic to the pasteboard. Select the "Convert Pasteboard" option from the "Edit" menu and save the graphic as an EPS file. For a program like WriteNow, you can use the "Place Graphic" option from the "Edit" menu to read in a EPS graphic (this is how I created my original draft of this document).

If you have any questions on this (or any other menu options) use the incredible help facility that is provided. For instance, if you need help on the "Evaluate Notebook" option from the "Action" menu, press **CMD**

/ and the cursor will turn into a question mark (**?**). Now just use the question mark cursor to select any menu item and a detailed help panel will appear. This help facility is truly one of the finest I have seen.

Utilizing the Display[] function is a two step process.

```
In[74]:= PlotZ = Plot[Sin[x] * Cos[x], x, 0, Pi]
In[75]:= Display[ "plotZfile", PlotZ]
```

This saves the raw Postscript image to a file named "plotZfile" in your home directory. Now issue the following command from a terminal/ shell window:

```
/NextApps/Mathematica.app/Kernel/Utilities/psfix plotZfile > plotZfile.eps.
```

This shell script converts the raw Postscript file to the standard NeXT EPS format. Now the new file plotZfile.eps file is ready to be used by NeXT software packages like WriteNow and Scene.

Special Topic: Notebooks

A Notebook is a window that is comprised of a hierarchy of cells such that the user can build up a dynamic workbook of text, expressions and real output. The notebook/cell paradigm is extremely useful and provides you with the capability of giving your work incredible structure and organization. You can open and close cells (and entire hierarchies of cells) by double clicking on any cell that encompasses one or more cells.

Mathematica includes several notebooks (in the Mathematica notebook directory: /NextLibrary/Mathematica/Notebooks) that are worth investigating. Once you have Mathematica up and running, just use the "Window/Open" command in conjunction with the directory browser. These notebooks can provide great demos and tutorials. Some of the more interesting ones include the HyperbolicIcosahedron.ma notebook. This notebook is where the 3-D graphic in the very beginning of this article came from. The TourOfMathematica.ma notebook provides a quick and complete coverage of most of the capabilities of Mathematica and is very useful when you want to give someone a 15 minute tour of Mathematica.

Special Topic: Packages

Packages are ASCII files of related functions and definitions that are not intrinsically built into the kernel. Mathematica provides numerous

packages for your usage that also provide good examples when building your own. Some of the different categories are: Discrete Math, Geometry, Graphics, Number Theory, Numerical Analysis and much more. All Mathematica packages are stored in sub directories in /NextLibrary/Mathematica/Packages. If you don't have the book (which details many of the packages), use the Directory Browser and your favorite editor to view the packages. To instantiate these definitions use the '<<PackageDirectory/Package' syntax (Mathematica starts searching in the main package directory).

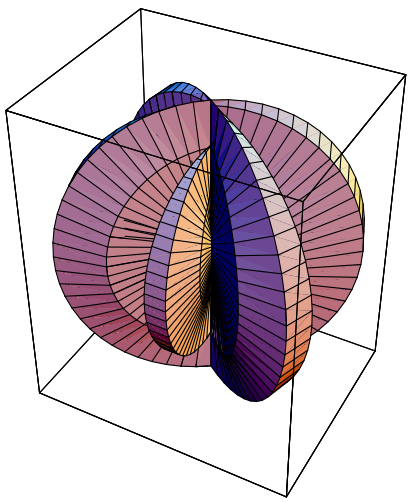
The graphics package "Graphics/ParametricPlot3D.m" allows you to create an infinite variety of 3-D parametric plots. One of the most common 3-D parametric plots is the Spherical Harmonic which is used in quantum mechanics to describe the symmetry of the hydrogen atom.

```
SphericalPlot3D[Abs[SphericalHarmonicY[3, 1, theta, phi]],
    theta, 0, Pi, Pi/30, phi, 0, 2Pi, Pi/15]
```

If you have problems getting the graphic output, either quit the current kernel or start a fresh session of Mathematica.

I decided to let you plot that one for yourself so that I could show you the monster I created by switching the parameters.

```
In[76]:= <<Graphics/ParametricPlot3D.m
In[77]:= SphericalPlot3D[Abs[SphericalHarmonicY[3, 1, phi, theta]],
              theta, 0, Pi, Pi/30, phi, 0, 2Pi, Pi/15]
```

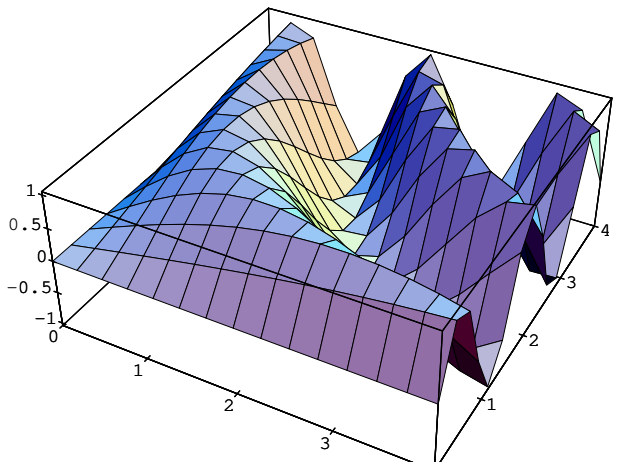


Advanced Topic: Animation

Animation is certainly the most visually impressive aspects of Mathe-

matica. Of the two types of animation found in the animation notebook, I'm going to discuss the type which shows a single graphic from a series of different viewpoints. The animation is accomplished by creating several images of a graphic and then displaying these images in rapid succession.

Start out by opening /NextLibrary/Mathematica/Notebooks/Animation.ma and answer yes to the "Evaluate Initialization Cells" question (these initialization cells contain definitions that the notebook needs). After the initialization is complete open the "Examples" cell followed by the "SpinShow" cell. While making sure the "SpinShow" cell is highlighted, select the "Evaluate Selection" option from the "Action" menu (this will take a few minutes). Once it's done, highlight the "SpinShow[thePlot]" cell that contains the series of images and then select the "Animate Selected Graphics" option from the "Graph" menu. The default graphic for the SpinShow is:

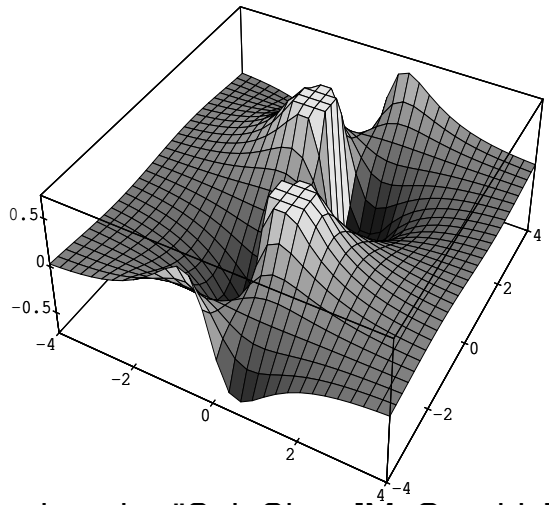


Now let's animate our own graphic by opening the "Implementation" cell (you can also close the "SpinShow" and "Examples" cells to get them out of your way). Open the "Remove any previous..." cell and select the "Evaluate Selection" option from the "Action" menu. Next open the "Define SpinShow" cell and evaluate this cell in the same manner. Click after the "Define SpinShow" cell and type in your graphic definition. Here is the graphic that I used, but feel free to use your own imagination.

```
In[78]:= MyGraphic = Plot3D[ Im[ Sech[x + I y]], x, -4, 4, y, -4, 4,
                             PlotPoints -> 30]
```

Now all you have to do to animate this graphic is start a new cell with the following SpinShow[] definition:

```
In[79]:= SpinShow[MyGraphic]
```



and when it's done select the "SpinShow[MyGraphic]" cell and then select the "Animate Selected Graphics" option from the "Graph" menu. Note that you can override the default options for SpinShow with the following syntax "SpinShow[graphic, options]". Check the SpinShow[] definition for the default option list.

Book Review

If by this point you are really excited about Mathematica and want to have an excellent reference, purchasing the book "Mathematica, A System for Doing Mathematics by Computer" would not be a bad idea. The cost of the book is ~$32.00 and should be available in most computer bookstores (it's a big black book with a color 3-D plot, Published by Addison-Wesley).

The book is the epitome of excellent book-writing. It flows well, addresses a wide variety of users, is correct in syntax, provides a great tutorial and is full of meaningful examples, appendices and summary tables. The book starts real slow and easy and smoothly progresses to a finale of extreme depth and insight. It is recommended that the reader should use the book in conjunction with Mathematica so that examples and concepts can be immediately verified.

David R. Hieb
davehieb@boulder.Colorado.EDU

Installing FUJITSU M2263S SCSI disk (660MB) as NeXT boot disk

Izumi Ohzawa

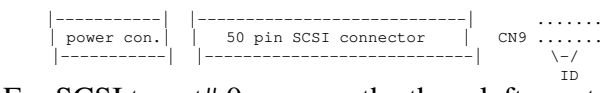
My 660MB Fujitsu SCSI drive now boots fine, thanks to help from Helene Young-Myers and Eugene D. Myers. The drive has about the same capacity as Maxtor XT-8760S, and can be had for a little more than $2000. The drive is, however, substantially louder during seeks than Maxtor (see below).

In any case, since some of you sent me mail asking for the summary report, here it is:

[1] Jumper settings on the drive

(a) SCSI target# can be set by jumber block "CN9" located right next to the SCSI bus connector of the drive. Three jumpers closest to the SCSI connector encode the target number in binary.

View from the back of the drive, with the drive UP-SIDE DOWN.



For SCSI target# 0, remove the three left-most jumpers on CN9.

For SCSI target# 1, keep the left-most one, and remove the second and third jumper from left.

(b) Remove two jumpers at both ends of the jumper block "CNH2".

Locate the jumper block "CNH2" betweeen two surface-mount VSLI's whose labels are obscured by capacitors (1cm square blocks colored blue on my drive, with two leads). There are 8 jumpers in this block.On Jumper Block "CNH2" remove:

[1-2] INQUIRY DATA

[15-16] Synchronous Mode Transfer

If you can't figure out the numbers on the block,

[1-2] is at one end and indicated by a little triangle mark, and

[15-16] is at the other end (8-th position from the triangle).

[This information was kindly provided by Helene Young-Myers, helene@secd.cs.umd.edu, and her husband Eugene D. Myers.Thank you!]

[2] Remove the 40MB swap disk and install M2263S.

Connect the power cable to the drive, and SCSI flat cable.

[3] Boot from OD, and login as root.

Use command "scsimodes" as below to get basic formatting information, e.g., bytes per sector, etc. If "scsimodes" does not work, the most likely cause is improper cable connections.

localhost# scsimodes /dev/rsd0a

SCSI information for /dev/rsd0a

Drive type: FUJITSU M2263S-512

512 bytes per sector

53 sectors per track

15 tracks per cylinder

1658 cylinder per volume (including spare cylinders)

3 spare sectors per cylinder

15 alternate tracks per volume

1312343 usable sectors on volume

[4] Add the following entry to the end of "/etc/disktab".

If the parameters shown by "scsimodes" are different from those above, you cannot use the entry below. If you want the modified disktab to appear also in the SCSI disk's /etc/disktab after BuildDisk, copy /etc/disktab to /usr/template/client/etc/disktab, too.

(BuildDisk will copy /usr/template/client/etc/* onto a newly built disk as /private/etc/*)

Be sure to make backup copies of the unmodified disktab in each directory. Write permission may have to be set to edit these files.

-- cut here and append to /etc/disktab --

FUJITSU M2263S-512 with 3 spare sectors/

cylinder and 512 byte sectors

localhost# scsimodes /dev/rsd0a

SCSI information for /dev/rsd0a

Drive type: FUJITSU M2263S-512

512 bytes per sector

53 sectors per track

15 tracks per cylinder

1658 cylinder per volume (including spare cylinders)

3 spare sectors per cylinder

15 alternate tracks per volume

1312343 usable sectors on volume

#

M2263S-512|FUJITSU M2263S-512|Fujitsu 760MB w/512 byte sectors as 1 part

ition:\

:ty=fixed_rw_scsi:nc#1658:nt#15:ns#26:-ss#1024:rm#3600:\

:fp#160:bp#0:ng#0:gs#0:ga#0:ao#0:\

:os=sdmach:z0#32:z1#96:ro=a:\

:pa#0:sa#646230:ba#8192:fa#1024:-ca#32:da#4096:ra#10:oa=time:\

:ia:ta=4.3BSD:

#

Fujitsu M2263-512, divided into two partitions, sd0a and sd0b.

This entry has not been tested. It may not work.

(To use this entry, uncomment lines below, and comment out lines above

)

#

M2263S-512|FUJITSU M2263S-512|Fujitsu 760MB w/512 byte sectors, 2 part

itions:\

:ty=fixed_rw_scsi:nc#1658:nt#15:ns#26:-ss#1024:rm#3600:\

:fp#160:bp#0:ng#0:gs#0:ga#0:ao#0:\

:os=sdmach:z0#32:z1#96:ro=a:\

:pa#0:sa#426230:ba#8192:fa#1024:-ca#32:da#4096:ra#10:oa=time:\

:ia:ta=4.3BSD:

:pb#426230:sb#220000:bb#8192:f-b#1024:cb#32:db#4096:rb#10:ob=time

:\

:ib:tb=4.3BSD:

#

-- cut here --- end of disktab for M2263S --

---

[5] Check if the modified /etc/disktab is valid by;

localhost# disk /dev/rsd0a

disk name: FUJITSU M2263S-512

disk type: fixed_rw_scsi

Disk utility

disk> quit (quit this program)

localhost#

If the "disk" command does not recognize the drive, you have made a mistake in adding the disktab entry in step [4].

Check /etc/disktab again.

---

[6] Launch "BuildDisk" and run it.

---

[7] If you are using the first disktab entry (the whole disk as one large partition), you must edit "/etc/fstab" on the SCSI disk before you attempt to boot from it. If you are using the second entry (for two partions), skip this step.

mount /dev/sd0a /mnt

chmod 644 /mnt/etc/fstab

... Edit /mnt/etc/fstab, and comment out or remove the line:

/dev/sd0b /clients 4.3 rw,noquota 0 2

umount /dev/sd0a

[8] Shutdown and boot from the SCSI disk.
If the ROM monitor complains:

SCSI unexpeced msg:1

sc: Unexpected msg

.... repeated many times...

Then, you must have not set the jumpers correctly in step [1](b)

Additional information of interest

[9] Transfer Rate
localhost# disk /dev/rsd0a

disk name: FUJITSU M2263S-512

disk type: fixed_rw_scsi

Disk utility

disk> read

starting block? 0

sectors per transfer? 16

number of transfers? 1000

sector increment? 16

16384000 bytes in 21243 ms = 771374 bytes/s

disk> read

starting block? 0

sectors per transfer? 16

number of transfers? 1000

sector increment? 16

16384000 bytes in 21242 ms = 771374 bytes/s

For comparison, according to Ronald Antony & Axel Merk (the authors of "TheFormatter" for Wren drives), CDC Wren VI (CDC 94191-15) gives

520000 - 560000 bytes/sec for 512 bytes/sector drive without cache, and

900000 - 980000 bytes/sec with on-drive cache.

[10] Seek Noise
The drive is quite noisy during seek. My subjective impression is that the drive is as noisy as the current opical drive, although the nature of the noises is quite different. You will certainly hear seek noise every time a Browser opens a directory or when someone does an rlogin, etc. By comparison, Maxtor XT-8760S, the NeXT's official 660MB drive, is dead quiet. Perhaps, the drive may not be suitable if you need a quiet environment. But the 5-year manufacturer warranty is, for me, quite soothing :-)

I don't know how Fujitsu compares with CDC Wren drives with respect to seek noise.

[11] As always, use the information above at your own risk. I am not responsible for any damage to the SCSI drive or to the NeXT cube itself (voiding the warranty) caused directly or indirectly by the use of the information provided above.
Izumi Ohzawa, izumi@violet.berkeley.edu

Market View

New This Month

THIS SECTION IS PROVIDED AS A COURTESY TO OUR READERS AND THIRD PARTY FRIENDS. BuzzNUG IS NOT RESPONSIBLE FOR THE CONTENT OF THESE OR ACCURACY OF INFORMATION PRESENTED IN THIS SECTION

•[OT Palette:1.0] - Objective Technologies Inc. 1-212-227-6767 - The first collection of commercially available custom object palettes for the NeXT Interface Builder. Each palette includes at least one custom Inspector Window. Palettes include:

- SmartFields --a forms package in a field, includes floating point, integer, money, date, alphanumeric and regular expression fields that provide keystroke-by-keystroke validation, plus visual and audio alerts.
- Chooser -- a scrolling selection list. Each entry in Chooser may contain a description, and associated mini-icon, and an associated integer tag.
- MathPalette -- connects data objects in your application to Mathematica. The **MathAccess** object directs input and output, and maintains the connection with the Mathematica kernel. The **MathView** object may be resized at compile time; it may be as small as one square inch or as large as the physical screen. Objects in MathPalette allow programmers to construct sophistated "no-code" applications that use Mathematica as a compute engine.

Each palette is reasonably priced, and university pricing is available.

• Distributed File Editor - Objective Technologies, Inc. 1-212-227-6767 - The Distributed File Editor executes a very simple set of commands which perform sophistated batch-style modification and maintenance of user files; administrators often use DFE when a department gets moved from one server or disk to another, requiring path changes to the groups' .login files. DFE is intended for use by administrators responsible for large central systems or workstation networks; it works just as well, however, for smaller systems.

DFE may only be purchased as source code, and supports NeXT, Sun and most BSD derivitives.

Product Listings

I've received literature from NeXT or Third Party Developers on the following products. No warranty, express or implied, is given. The quotes are mostly the developers' and may not reflect reality. Please send up-to-date info and review copies for new products.

•QuintProcessor - 5 27-MHz DSP56001s on a Board, Ariel Corp, 201-249-2900.

•Objective DB Toolket - 30 classes to links NextStep to Sybase. Professional Software, Inc./Lakeside Office Park,/599 North Avenue - Door 7/Wakefield, MA 01880/(617)246-2425

•uni-REXX and uni-XEDIT - Unix version of IBM's mainframe/procedural language and general purpose editor./The Workstation Group, wrk/grp/6300 N. River Road/Rosemont, IL 60018/(708) 696-4800

•Digital Instrumentation Technology Inc (505)662-1459 is shipping beta copies of **Cube Floppy 1.4**, a 3.5" floppy disk drive connecting to the SCSI port, reading and writing MS-DOS(720K and 1.44MB disks), UNIX and Macintosh (1.4MB disks) file formats.

• WeDesign Inc (415)-479-1105 sent a flyer on **TheLibrary**., an on-line information system with Objective C references and authoring tools.

• Pacific Micro (415)948-6200 is shipping the **PM1.44** and **PM HDE** external 3.5" floppy drive and hard disk enclosure. Special order line is 1-800-628-DISK.

• BYTE's **BIX** service has a special NeXT interest group. Contact 1-800-227-2983 (BIX Customer Service) if interested. Please mention both BuzzNUG and Dave Andrews (of Byte) as contact names.

•Communicae - Active Systems 1-617-576-2000 "high performance communications package. VT240 emulation"

•Wingz Informix Software, Inc. 1-913-599-7100 "graphic spreadsheet featuring advanced charting, desktop presentation capabilities, and HyperScript"

•Scan 300/GS Abaton 1-415-683-2226 "300 dpi flatbed scanner with TIFF compatility"

•DM-N Digital Microphone Ariel Corporation 1-201-249-2900 "software-selectable sample rates from 88.2 kHz to 5.5 kHz per channel"

• BUG-56 : DSP debugger, Ariel Corporation 1-408-982-0400/1-201-249-2900,

•DaynaFILE Dayna Communications, Inc. 1-801-531-0600 "external, SCSI floppy disk drive to write to standard UNIX-formatted diskettes, as well as MS-DOS formats"

•Smart Art Emerald City Software, Inc. 1-800-223-0417 "50 text and graphics effects and easily customized in any NeXT word processor, desktop presentation, or page layout program *NOW OFFERED BY ADOBE! (415)962-2045*

•FrameMaker 2.0 Frame Technology Corporation 1-408-433-3311 "powerful, cost-effective workstation publishing software"

•Artisan Media Logic Incorporated 1-213-453-7744

"high-resolution paint and image processing system"

•TopDraw Media Logic Incorporated 1-213-453-7744 "complete and advanced page-based graphics software"

• HSD incorporated US Scan-X 1600/600 415-964-1400. Scanners for “line art and grayscale”

•TextArt Stone Design Corporation 1-505-345-4800 "array of tools that allow immediate creation of outstanding PostScript images" .

•Encapsulated PostScript ClickArt T/Maker Company 1-415-962-0195 "combines ClickArt EPS portfolios into a collection of high-quality Encapsulated PostScript (EPS) artwork"

•Public Domain Disk #1 - Lighthouse Design 1-800-FOOBAR9 "Public Domain Software & More"

•Scematic Entry - Lighthouse Design 1-800-FOOBAR9 "CAD Tool for designing electrical circuit schematics"

•Media Station - Imagine Inc 1-313-434-1970 "archival, retrieval and processing of multi-media information"

•Fortran 77 -- Absoft 1-313-853-0050 "Objective Fortran-77"

•DisplayTalk - Emerald City Software - 1-800-223-0417 "Complete development environment for Display PostScript programming" *THIS IS NOW OFFERED BY ADOBE! (415)962-2045*

• *SmartArt*, Adobe Systems Inc. 415-962-2045 “Graphics and headline type effects using Display PostScript”.

•Video Monitor and Projector Interfaces Extron Electronics 1-800-633-9876 "offers three video monitor and projector interfaces"

•Digital Ears Metaresearch, Inc. 1-503-238-5728 ""allows entering and recording compact disc-quality sounds"

•Digital Eye Metaresearch, Incorporated 1-503-238-5728 "allows entering and recording NTSC video images"

•NVT High Density Video Drive New Vision Technologies, Inc. 1-415-285-8744 ""video playback device for interactive multi-media applications"

•JETSTREAM Tape Backup System Personal Computer Peripherals Corporation 1-813-884-3092 "high performance tape backup system"

•A/D64x Analog/Digital Interface Singular Solutions 1-818-792-9567 "a low-cost platform for sound recording, experimentation, and analysis"

•Who's Calling Adamation, Inc. 1-415-452-5252 "lets sales & business professionals keep track of phone calls and other client information".

•GEMS (Generalized Equilibrium Modeling System) Data Transforms, Inc. 1-303-832-1501 "a flexible way to model economic systems"

•InDia (Influence Diagram Processor) Data Transforms, Inc. 1-303-832-1501 "graphical application for representing complex decision-making"

•Knowledge Retrieval System (KRS) KnowledgeSet, Corporation 1-415-968-9888 "rapidly searches and retrieves information from large databases of text and graphics"

•OMEN III Microstat Development Corporation 1-604-228-1612 "stock quotation and financial system"

•TACTICIAN Plus SouthWind Software, Inc. 1-316-636-5100 "multi-user spreadsheet that supports high-level functions and adds built-in presentation graphics" .

•Adobe Illustrator Adobe Systems Incorporated 1-415-961-4400 "graphic design and illustration program for generating high-quality artwork"

•Adobe Type Library Adobe Systems Incorporated 1-415-961-4400 "offers more than 500 different typefaces"

•Flash Graphics Flash Graphics 1-415-331-7700 "extensive charting, illustration, and text functions in a graphics package for screen, slide and paper presentations"

•InterFax 24/96N Abaton 1-415-683-2226 "combines a 9600 bps Group 3 fax modem with a 2400 bps MNP 5, Hayes-compatible data modem"

•GatorBox Cayman Systems, Inc. 1-617-494-1999 "LocalTalk to Ethernet gateway that translates the Network File System (NFS) protocol into Apple Filing Protocol (AFP)"

•MacLinkPlus/PC DataViz Inc. 1-203-268-0030 "kit for transferring and translating files between NeXT and Macintosh environments"

•Ethernet PhoneNET, Sound and Interpersonal Communications Farallon Computing, Inc. 1-415-849-2331 "used to build LANs over standard telephone cables"

•Etherport NL Kinetics 1-415-947-0998 "allows the NeXT computer to connect directly to standard twisted-pair Ethernet networks"

•INFORMIX-TURBO Informix Software, Inc. 1-415-926-6300 "database engine for on-line transaction processing (OLTP)"

•INGRES Relational Database Management System Relational Technology, Inc. 1-800-4-INGRES "SQL database engine provides on-line transaction processing (OLTP) in single- or multi- CPU and distributed environments"

•DAN - The Data Analyzer Triakis Inc 1-505-672-3180 "data analysis package for reducing data and generating presentation-quality plots"

•Math++ - Triakis Inc 1-505-672-3180 "C-language math library. Approx 100 math functions"

•Dreams - Innovated Data Design 1-415-680-6818 "Frm the makers of MacDraft, drawing and drafting tools"

•Cross Assember/Simulator Programs - Motorola 1-512-891-2030 "for the 56000 and 96000"

•Fortran, C and Pascal Compilers - OASYS 1-617-890-7889

SIMMS

The SIMM information below was posted by William Smith on the public Comp.Sys.Next newsgroup. It is reproduced without direct permission.

PSI 2005 Hamilton Ave., #220, San Jose, CA,95125 Ph. 408-559-8544,Price: $59.50 per simm ,800-622-1722, Lifetime Guarantee

Third Wave Computing, Inc. 1826-B Kramer Lane Austin, TX 78758, ,Ph. 800-284-0486. They take PO's,Price: ?, 512-832-8282

Memory international,Ph. 714-588-0488. Price: $68.00

Turbo Technologies, Inc.,Ph. 800-542-7466 Lifetime warranty,They take PO's.Price: ?

South Coast Electronics. Ph. 800-289-8801,They take PO's 10920 Wilshire Blvd.Suite 110, LA,calif. 90024,Lifetime warranty,Price: ?

Newer Technology 1117 S. Rock Rd. Suite 4 Wichita, KS 67207,Price: ? Ph. 800-678-3726, 316-685-4904

Micro Electronic Technologies, inc. 35 South St., Hopkinton, MA 01748,Price: ?Ph. 800-766-SIMM, 508-435-9057

Peripheral Outlet 314 S. Broadway Ada, OK 74820, Lifetime guarantee, Ph. 800-332-6581, they take PO's, Price: $62.00, 405-332-6581

ETC 5426 Beaumont Center Blvd. Suite 340 Tampa,Fla 33634, They take PO's, Ph. 800-882-2863, Price: ?, 813-884-2863

Shecom computers 22755 Savi Ranch Pkwy G, Yorba Linda, CA 92686, They take PO's Ph. 800-366-4433, Price: $59.00,3 year warranty, 714-637-4800

Chip Merchant 9285 Chesapeake Dr., Suite L, San Diego, CA 92123, 5 year guarantee Ph. 800-426-6375, no credit cards,Price: $60.00,619-268-4774

Memory Plus, Ph. 800-388-PLUS, Price: $62.00

Megabyte Memory Products 737 Pearl St., Suite #208B La Jolla,CA 92037, 5 year warranty, Ph. 800-748-5766, Price: $62.50

Technology Works 4030 Braker Lane West, Austin, TX 78759,They take PO's, Ph. 800-688-7466, Price: ?, 512-794-8533

Stratum Technologies inc. 12191 Technology Blvd., Austin, TX 78727,Lifetime warranty, Ph. 800-533-1744, 30 day trial period (?),they take PO's, Price: ?

Delta Research Labs 26072 Merit Circle, Suite 119 Laguna Hills, CA 92653,warranty 60ns,70ns,80n-s,100ns,Ph. 800-999-1593, Price:?, 714-367-0344

Computer Care, Ph 800-950-2273/612-371-0061, Price : $99, Ford Center Suite 1180, 420 N. Fifth St, Minneapolis, MN 55401.

Impediment, Ph. 617-837-8877 warranty 6years

Price: (they will generally beat the best price you can come up with), I think they take PO's

ClearPoint Computers, Ph. 800-253-2778,Price:?

Associated Technologies Marketing Ph. 214-248-0700

Memory Plus sells *4 MB* SIMMs (low profile, 80ns) for $299.00,**Memory Internationa**l sells them for $298.00.

User Groups

compiled by
Conrad Geiger
Conrad_Geiger@Next.Com

CANADA

British Columbia

Vancouver NeXT Group, Lionel Tolan, chairman, Computing Services, Simon Fraser University, Vancouver, B.C., Canada, V5A 1S6, phone - (604) 291-4702, email - lionel_tolan@cc.sfu.ca

"We meet every last Wednesday of the month, 6:30 pm at the Vancouver NeXT office: 230-1333 Johnston St., Granville, Island, Vancouver, B.C."

Newsletter: NeXTVieW, quarterly, editors Tom Poiker and Shirley Chan, Address: Dr. T.K.Poiker, Department of Geography, Simon FraserUniversity, Burnaby, Canada V5A 1S6, phone - (604)291-4515, email - USERTONI@cc.sfu.ca or poiker@whistler.sfu.ca

NeXT Tabloid: "Tao" (published monthly), editor is Robert Lin, Objective Software, 1701W 64th Ave, Vancouver, B.C., Canada V6P 2P3, phone - (604 261-0186, email - rlin@cs.ubc.ca

Quebec

Montreal Next Section of Club MacIntosh, Robert Paulhus - president, Club MacIntosh of Montreal - NeXT section, 2250 Guy Street room 303, Montreal, Quebec H3H2N2, phone - (514)939-0382, email - paulhus@calvin.cs.mcgill.ca

Ottawa NeXT User Group, Hugo DeRosier - president, 19 du Muguet, Hull, Quebec, Canada J9A 9Z7, email - hugo@csi2.uofo.edu

Toronto

Toronto NeXT User Group (more next issue)

JAPAN

Tokyo

Japan NeXT User Society - Katsuhiro Ohashi, 4-4-2, Ebara, Shinagawa-ku, Tokyo, 142 JAPAN,NeXus-office@etl.go.jp, NIFTY : NAC01417(Kishimoto) or NCB00436(Shioya)

UNITED STATES

National Groups:

Medical: NeXTMed - National NeXT Medical Users Group, Bill Barker and Jim Brinkley, Biological Structure, SM - 20, University of Washington, Seattle, WA 98195, email - NeXTMed-request@ulnar.biostr.washington.edu

Also : Conrad Geiger, 4130 Bagley Ave. North, Seattle, Washington 98103, phone - (415)780-2771

Music: (more info next issue)

Programmers: NeXT Programmers SIG

To join: email to "next-prog-request@cpac.washington.edu"

United States

States:

Arizona

Phoenix NeXT User Group, Gary Frederick - president, 20826 N 16th Ave, Phoenix, Arizona 85027-3531, phone - (602)869-0316, email - frederic@cimnext.cim.eas.asu.edu

Also : Chet Kapoor, 7150 E. Camelback Road, Suite 300, Scottsdale, Arizona 85251, phone - (602)423-7080

Also: Jim Ames, Arizona State University, CIM Systems Research Center, Tempe, AZ 85287-5106

Tucson NeXT User Group, Robert W. Layhe, CCIT/User Support, University of Arizona, Tucson, AZ 85721, email address - layhe@rcnext1.rc.arizona.edu

California

BaNG (Bay Area NeXT User Group), P.O. Box 8858, Stanford, CA 94309, BaNG-request@meta-x.stanford.edu, Eric Ly, P.O. Box 12318, Stanford, CA, 94309, phone - (415)780-2877, email - dayglow@-portia.stanford.edu.

Also : Joe Barello, 1505 Grand Avenue Piedmont, Cali-

fornia 94611, phone - (415)652-0769, email - joeba@lll-lcc.llnl.gov,

Also : Chris Overton, P.O. Box 2628, Stanford, CA 94309, email - louiex2@portia.stanford.edu

"Meetings are every third Wednesday of the month at 7 PM on the Stanford University campus."

Santa Barbara NeXT User Group, Amir Gharaat - president, email - erone%pumpkin@hub.ucsb.edu

Nuggets (Cal State - Los Angeles), Gary Novak - president, Department of Geology, Cal State, 5151 State University Drive, Los Angeles, Ca. 90032, phone - (213)343-2400

JPL/Caltech NeXT User Group, Leo Blume - president, adr: JPL, MS-510-202, 4800 Oak Grove Dr., Pasadena, Ca. 91109, phone - (818)397-9521, email - leo@emerald.jpl.nasa.gov

UC Riverside (California) NeXT User Group, Paul Lowe, president, phone - (714)787-3883, email - plowe@ucrac1.ucr.edu

SNuG - San Diego NeXT User Group, Nicholas MacConnell, 1135 Stratford Ct., Del Mar, Ca. 92014, phone (619)481-7535 or (619)565-9738

Also: David Rivas, email - david@ece.ucsd.edu

Colorado

rmNUG (Rocky Mountain NeXT Users Group), Dave Hieb, Chairman, 4521 Wellington Rd, Boulder, Colorado 80301, phone - (303)530-2560, email - davehieb@boulder.colorado.edu

Also: Brad Green, Co-chairman, 4600 South Ulster Street, Suite 700, Denver, CO 80237, email - green_bk@cubldr.colorado.edu

District of Columbia (Northern Virginia, Southern Maryland)

NeXT Special Interest Group (SIG of BCS) - Washington, D.C., Hugh O'Neill, President, P.O. Box 39036, Washington, D.C. 20016,

Also :Joel McClung, email: joel@next.com, (703)938-NeXT, Mailing address for newsletter, NeXT, Inc., 8300 Boone Blvd., Suite 558, Vienna, VA 22182, Attn: Fillipe Fuster

"Meet 2nd Wednesday of month at Building 12A, Room B-51 of NIH Complex in Bethesda, Md. Meetings are at 7:30 and run anywhere from 1 to 3 hours."

Georgia

BUZZNUG (Georgia Tech NeXT User Group), Erica Liebman - editor of NeXT Users Journal, 1150 Collier Road NW Apt L-12, Atlanta, GA 30318, email - erica%kong@gatech.edu, phone -(404)352-5551

Illinois

Chicago NeXT User Group, Mark Henderson, Building 203 Room C-246, Advanced Computing Research Facility, 9700 South Cass Avenue, Argonne National Laboratory , Argonne, Illinois 60439, email - henderson@mcs.anl.gov, phone - (708) 972-5963

NU NeXT Users Group - (Northwestern University), Bill Parod - president, Northwestern University, 627 Dartmouth Place, Evanston, IL 60208, email - parod@baris.acns.nwu.edu, phone - (708)491-5368

Massachusetts

BCS - Boston Computer Society, Dan Lavin - president, NeXT User Group, Boston Computer Society, One Center Plaza, Boston, Ma. 02108, phone - (617)969-6555

Minnesota

Minnesota NeXT User Group, Mike Tie - President/Treasurer, Math/CS Department, Carleton College, One North College Street, Northfield, MN 55057, phone -(507)663-4067, email - mtie@carleton.edu

"We meet on the second Tuesday of every month. We hold the meeting at a different location every month, so I send a mailing to all our members with the information about location and the agenda. People who are interesting in attending our meetings should get in touch with me."

Missouri

St. Louis NeXT User Group, John Bartley, Deloitte & Touche, One City Centre, St. Louis, Missouri 63101, phone - (314)343-4996

New Mexico

Albuquerque NeXT User Group, Jeff Jortner, Sandia National Laboratories, Division 1424, PO Box 5800, Albuquerque, NM 87185, email - jnjortn@cs.sandia.gov

Los Alamos NeXT Users Group, Dwight Barrus, Chairman, Group C-6, MS-B272, Los Alamos National Laboratory, P.O. Box 1663, Los Alamos, NM 87545, phone (505)667-8870, email - dmb@lanl.gov

Also: Joe Kleczka, Co-Chairman, email - jhk@lanl.gov, phone -(505)667-4584 (page number)

New York

New York City NeXT User Group, Tim Reed, 7 Dey St. Suite 711, New York, New York 10007, phone (212)227-6767

Ohio

Columbus NeXT User Group (Ohio), Chuck Dyer, Sr. Systems Analyst, The Ohio State University, 1971 Neil Ave-IRCC, Columbus, OH 43210, phone - (614)292-4843, email - dyer-c@osu-20.ircc.ohio-state.edu

Oregon

OSU - Oregon State University NeXT User Group, Tom Leach - president, Ocean Admin 104, College of Oceanography, Oregon State University, Corvallis, Oregon, email - leach@satchmo.oce.orst.edu

Meet 2nd Thursday of month at 2:30pm in Milne Computer Center - Faculty Development Lab.

Portland NeXT User Group (more next issue)

Pennsylvania

Allegheny College - IMUG (Interface Builder User Group), Dr. Joel Smith, Allegheny College, Allegheny, Pa., email - js01@music.alleg.edu

Texas

DFW NeXT User Group (Dallas/ Ft. Worth), Dirk Hardy, president, Hofbauer Information Systems, 5080 Spectrum Drive, Suite 912W (Lock Box 21), Dallas, TX 75248, phone - (214) 385-2991

Also : Charlie Lindahl, Automation and Robotics Research Institute, University of Texas at Arlington, 7300 Jack Newell Blvd. S., Ft. Worth, TX 76118, phone - (214)284-6122, email - lindahl@evax.arl.utexas.edu

"We meet the third Thursday of every month at theNeXT office in Las Colinas. Our meetings last from 7 until 9 (official hours), and there are usually informal discussions until 10PM or so."

hAng - Houston Area Next Group, John R. Glover - president, E.E. Department, University of Houston, Houston, TX 77204-4793, email - glover@uh.edu, phone - (713)749-1820

"We meet first Wednesday of each month at the Engieering College, UH, from 6 PM until around 8:30 PM."

Washington

University of Washington NeXT User Group,Corey Satten - president, University of Washington, 156D Academic Computer Center, HG-45, 3737 Brooklyn Avenue, NE, Seattle, WA 98105, email - corey@cac.washington.edu, phone - (206)543-5611

"We meet the third Wednesday of each month at 3:30PM in Parrington Hall, room 306 on the University of Washington campus."

Washington State University NeXT User Group, Joe Gerkman - president, N.E. 545 Kamiaken St. #8, Pullman, WA 99163, phone- (509) 334-9594, email - gerkman@pcc-next.csc.wsu.edu -OR- gerkman@wsuvm1.bitnet

NEW NeXT USERS' GROUP AT TEXAS A&M

Walter Daugherity

After a demonstration of the NeXT to about 20 students and faculty September 10 by Jeff Martin and Chuck Herrick, our NeXT Campus Consultants, we organized a NeXT Users' Group.

A number of clever names were suggested (along with a few not so clever ones, of course), but the vote was for "TexNeXT." [That's an inside joke: besides Texas and NeXT, the obvious referents, there's also "Tex-Mex," Mexican food as served in Texas.]

The officers chosen were Keith Perkins (n245bq@tamuts.tamu.edu), an undergraduate student, President; Dr. Walter Daugherity (daugher@cs.tamu.edu) [*Ed : Atta-boy!*], faculty, Vice President; and David Cheslow (dcheslow@cs.tamu.edu), a grad student, Secretary-Treasurer.

Regular meetings will be held at 5:00 P.M. on the first Monday of each month, with special meetings when appropriate (e.g., when NeXT will bring a color machine to demo--hint, hint).

Announcements will be posted to the tamu.unix.next newsgroup. For further information please contact one of the officers.

--Walter Daugherity

New York User Group Organizing

Timothy Reed

The New York NeXT Users Group - Tim Reed 1-212-227-6767 and Rob Allen 1-212-980-3408- The New York NeXT Users Group is having it's initial meeting during the first week of October at a "new music" center in Manhattan. The meeting will feature Mara Helmuth

and the NeXT Music Kit. The meeting will also include assorted announcements and product demonstrations, and organizational discussion to help schedule additional meetings and events. There is a strict 75 person limit to the number of attendees; those interested should send electronic mail to RSVP@object.com or call Rob at the above number.

BANG

SUMMARY OF BaNG MEETING #4, July 18, 1990

By Declan McCullagh,

declan@portia.stanford.edu

GETTING THE MOST OUT OF DISPLAY POSTSCRIPT

Ken Anderson, Manager, Developer Support Group, Adobe Systems.

The first part of the BaNG meeting was a one-hour talk about Display PostScript given by Ken Anderson of Adobe's Developer Support Group.Targeted at the entire audience, not just NeXT programmers, it covered a wide range of topics regarding efficient PostScript programming.

During the course of the talk, Ken demonstrated a variety of sample programs which illustrated various programming techniques needed under NextStep. The first PostScript programming technique mentioned was the use of NextStep's "pswrap" procedure, which allows programmers to "wrap" a standard C function around a body of PostScript code. Ken showed the audience Adobe's DrawLine demo, which provided timing information for pswrap code execution compared to execution of the same PostScript code not in a pswrap; the pswrap code consistently executed about 25% faster. In Adobe's port of Adobe Illustrator to the NeXT computer, over 80% of their PostScript code was placed in pswraps.

Next, Ken demonstrated the use of the User Path cache. This 300K cache remembers the most frequently-used images and stores them as lists of trapezoids. Using two sample PostScript drawings, he compared traditional "Red Book" drawing to drawing with the cache enabled; with the cache turned on, redrawing was up to three times as fast.

Ken also talked about the new PostScript language definition known as PostScript Level II. This new standard will replace Adobe's "Red Book" reference, and include a number of performance optimizations such as improved object caching, faster pattern rendering, automatic stroke adjustment, and image compression. In addition, multiple compression algorithms will be available, each performing better on a different kind of bitmapped data. He suggested that Level II will be available in printers in early 1991 and will also be included in future NeXT system software.

To receive information about the new PostScript Level II standard and other technical information, Ken recommended that programmers apply as Adobe Developers. It costs $195 per year to be a developer, and Adobe supplies training through developer seminars, distributes technical notes, and provides 40% discounts on OEM PostScript printers and Adobe software. Also at the meeting, Adobe distributed seven technical papers on Display PostScript, similar to what a registered Adobe Developer would receive. These technical notes can be accessed through electronic mail by using Adobe's mail server.

TOUCHTYPE

Glenn Reid, Independent NeXT Developer

The next talk was given by Glenn Reid, who previously worked at both NeXT and Adobe. He demonstrated the use of his TouchType application, which should prove to be an enormous boon to people with serious typesetting needs.

TouchType is unlike any other text-manipulation program to date. It takes the traditional "draw program" metaphor used by programs like TopDraw and Adobe Illustrator and extends it to encompass selective editing of individual characters of a text object. To TouchType, text objects are not grouped as sequences of characters, but as individually movable letters. For instance, the "a" in "BaNG" can be moved independently of the rest of the word, yet TouchType still remembers that the "a" is associated with the other three letters.

Perhaps the best feature of this program is the ability to do very accurate and precise kerning (the ability to place characters closer together to create a more natural effect). TouchType supports intelligent automatic kerning and very intuitive, manual kerning done with a horizontal slider or by direct character manipulation. It also incorporates useful features such as sliders to change font sizes, character leading, and character widths, and an option which returns characters to a single base line.

TouchType, only six weeks in development, should be available in early August, with a tentative price of $249. BaNG members were given the opportunity to purchase the software for $150.

DIAGRAM!

Jonathan Schwartz and Roger Rosner, Lighthouse Design

The last demonstration was by Lighthouse Design, which showed the BaNG audience their Diagram! application. Like TouchType, Diagram! is revolutionary in the way it takes a traditional object-manipulation metaphor and extends it with a new action paradigm.

When launched, Diagram! restores the user's last used workspace, recovers any documents being edited, and sets up the object palette and object inspector. Then, the user has the option to drag objects from the object palette, place them in the drawing space, and draw connec-

tions between these objects with either straight lines or splines. The novel concept behind Diagram! is that after a connection is made, the object's position on the page can be changed, and the connection lines follow the object, rubberbanding to accommodate the new position. This single feature lends itself to a multitude of "white board" applications, such as organizational charts, project diagrams, business graphics and network diagrams.

Diagram! also allows users to import TIFF and EPS files into documents, lock items to the background, annotate images with text and graphics, and link objects to documents so that opening a link in Diagram! will launch the application needed to open the linked file. In addition, a possible future Lighthouse Design product is "The Exploder," a object-oriented database for the NeXT.

ANNOUNCEMENTS

The LaST Lock, a high-security device for the NeXT Computer, is now available from Prevail (408) 296-6550 for $195. It is comprised of industrial strength steel security cables and cold rolled black steel plates which attach the monitor to the cube and both to a plate which can be bolted to any fixed surface.

Registered BaNG members will receive official BaNG literature, a subscription to BaNG's quarterly newsletter, a BaNG T-Shirt, and notification of pending BaNG events. For more information, please contact BaNG at one of the addresses below.

CONTACTS

ADOBE SYSTEMS INCORPORATED
1585 Charleston Road, P.O. Box 7900
Mountain View, CA 94039-7900
(415) 961-4400
To access the mail server, send a message with a body of "help" to ps-file-server@adobe.com.

TOUCHTYPE

Right Brain Software
20 Medway Road
Woodside, CA 94062
(415) 851-1785
heaven!glenn@next.com

DIAGRAM!

Lighthouse Design, Ltd.
6516 Western Avenue
Chevy Chase, MD 20815-3212
(800) 366-2279, (301) 907-4621
lighthouse!diagram@uunet.uu.net

BaNG INFORMATION

P.O. Box 8858
Stanford, CA 94309
(415) 780-2877 (Voicemail)
BaNG-request@meta-x.stanford.edu

Buzz's Hint Corner

Coordinated by EJ Liebman

I've an application where I wanted to use 'printf' to a Text object in a ScrollView that I created with Interface builder. I wanted to do so from numerous methods (it was a debugging log) so I didn't want to have to code 'buffered printf' everywhere I needed it.

I came up with the following solution that others may find useful. In my code it is a method of my Application and messages an outlet, but I've recast it here as a method of a specialization of the Text object, for simplicity:

```
#import <appkit/Text.h>

#define BUFFERSIZE 512

@interface MyText:Text { }

- (int) printf:(const char *) format, ...;

@end

@implementation MyText

- (int) printf:(const char *) format, ...
{
    int result, length = [self textLength];
    char buffer[BUFFERSIZE];
    va_list ap;

    va_start(ap, format);
    result = vsprintf(buffer, format, ap);
    va_end(ap);

    [[self setSel:length :length] replaceSel:buffer];
    [[self scrollSelToVisible] display];

    return result;
}

@end
```

Using this, you can do arbitrary printf style output to a Text object:

```
[text printf:"The date is %s.\n", ctime(time(0))];
```

The 'setSel::' and 'textLength' calls can probably be optimized out (just done on Text initialization) if you're always writing to the end of the Text object from your other routines that access it. Comments/suggestions welcome, - *Christopher Lane*

p2c runs on the NeXT! David Gillespie's Pascal to C translator, p2c, compiles onthe NeXT if one change is made to one file. In src/trans.h (of version 1.18), change line 80 from

```
#if defined(BSD)
```

to

```
#if defined(BSD) && !defined(__STDC__)
```

Then lines 80-87 become

```
#if defined(BSD) && !defined(__STDC__)
# include <strings.h>
# define memcpy(a,b,n) bcopy(b,a,n)
# define memcmp(a,b,n) bcmp(a,b,n)
char *malloc(), *realloc();
#else
# include <string.h>
#endif
```

This change was suggested by Gillespie after seeing ScottHess's modification:

```
#ifdef BSD
# include <strings.h>
/* Next already has perfectly good definitions - scott */
#ifndef NeXT
# define memcpy(a,b,n) bcopy(b,a,n)
# define memcmp(a,b,n) bcmp(a,b,n)
char *malloc(), *realloc();
#endif
#else
# include <string.h>
#endif
```

Version 1.18 can be ftp'd from csvax.cs.caltech.edu. It isin pub/p2c-1.18.tar.Z.

We will find p2c to be very useful at Carleton because weuse Pascal in our CS I. P2c can translate the followingPascals into C: HP, Turbo 5.0, UCSD, DEC VAX, OregonSoftware Pascal/2, Macintosh Programmer's Workshop and Sun/Berkeley. It can also translate Modula-2. One ofthe example programs is a nice little BASIC called ChipmunkBASIC.- *Roger Kirchner*, rkirchne@mathcs.carleton.edu

NeXT Support Answers

Care of Doug Keislar

print alpha postscript (QA312)

Q: How do I get an image with alpha (transparency) to print?

A: There is no device-independent way to print images that are represented with an alpha channel. And to do it at all requires a fair amount of programming. So the best solution is to not write programs which use alpha

on screen in a way that it would need to be printed.

If you really need to print something with alpha channel, you need to render it into an offscreen (printing store) bitmap at the dpi that it will be printed, then read those bits with the image operator to be printed. That way the calculations involving alpha will be performed when you render into the printing store bitmap.

Usually this will involve creating a virtual 400 dpi window, do a scale, and then drawSelf:: into that. Send the resulting bitmap to the printer with the PostScript image operator, after doing a scale back. There are some issues involved; the 400dpi window will occupy lots of memory for one thing, and it will have 4 gray values which will hopefully correctly print out on a 2 gray printer.

For good results you must use a bitmap at the dpi of the device you are printing on. The Adobe/NeXT licensing agreement says that you can only print images at resolutions greater than 150dpi on (ONLY on) the NeXT printer. This allows us to create 92dpi bitmaps in our window server and print them, but not 300 or 400dpi bitmaps (This is on page 6 of the license shipped with 1.0). Thus, if you want to print with alpha to another printer, the only way to do it is at 92dpi.

When structuring the print methods of your application it would be best to create parallel imaging methods in your drawSelf:: for printing and drawing. Image the source bitmaps for composite and the basic printing from the same PostScript code and add alpha and composite for on-screen in an intermediate bitmap.

TIFF compatibility Draw rasterfile format Icon (QA426)

Q: I'm trying to convert a TIFF file made from Draw into a rasterfile format (for, e.g., an SGI IRIS). All the TIFF converters I know of complain about an invalid TIFF format. What's going on?

A: NXWriteTIFF() writes out data and alpha as one strip, not two separate strips; alpha is assumed to follow the data. But, the TIFF standard says that in this case (samples per pixel of 2) there should be 2 StripOffsets, not 1. So the file is not considered valid.

A workaround is to load the file into Icon and save it, ideally without Alpha Channel. (Saving with Alpha will lead to many display devices getting confused, because of the photometric interpretation value of 5.

IOT Trap mount (QA467)

Q: Why do I get the message "IOT Trap" when I try to mount (or unmount) a filesystem?

A: This is generated when the account invoking mount (or unmount) is a member of "too many" groups. It's actually generated in the authunix_create_default() RPC library routine.

Use the "groups" command to determine the number of groups to which the user invoking mount belongs. Also, check to see if there might be DUPLICATE group entries listed, each of which in fact takes up a "slot."

How many groups is too many? Eight is okay, nine will cause errors.

copy HashTable (QA338)

Q: The copy method doesn't work on a subclass of HashTable.

A: This is a HashTable bug; it does a [HashTable new] in doing the copy. The solution is to override the copy method and roll your own -- not a clean solution but probably the only one.

shell script csh (QA448)

Q: I have a shell script with csh commands (such as "set head = $path:h"), but the csh commands don't seem to work. What could be wrong?

A: Did you put `#! /bin/csh` at the start of your script? You probably also want the -f flag (as in "`#! /bin/csh -f`") so that it doesn't read the invoker's .cshrc file.

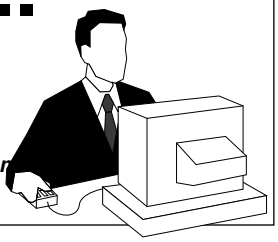
The background on this #! /bin/csh stuff has to do with a modification to the kernel loader (the thing that reads in executable programs). As we recall, this modification came from Berkeley several years ago.

When the loader reads a file that's marked "executable" according to the file's modes (permissions), it examines the first few bytes of the file, looking for a "magic number" -- an "integer" which identifies the type of the file. A Mach-O file, whether an executable or an object file (or a shared library, core file, or preload) will have a four-byte magic number of 0xfeedface. In old-style UNIX a.out files, the first two bytes specified the machine type, and the second two bytes specified the type of the executable file (OMAGIC - 0407 - for old impure, NMAGIC - 0410 - for read-only text, or ZMAGIC - 0413 - for demand-load files).

If the kernel loader sees a "magic number" of 0x2321, it treats the file as an "interpreter file". (0x2321 is the hex value of the characters #! in case you were wondering.) When an interpreter file is found, the remainder of the line following the magic number is examined; the file specified on the remainder of the line is executed, and given the interpreter file as input. Arguments to the interpreter may be included on that first line.

The way I see it...

by
Bruce Henderson
atncpc!jupiter!bruce@NeXT.com



For those of you who have had too much NeXT bashing in the last couple of weeks, read no more. NeXT is faltering. That's the truth. They are far from requiring "a miracle", as the August 29th article in the Wall Street Journal would have us believe. Instead, I would say that NeXT has a few weakness that they must address, before some of the "less easily convinced" out there will consider his vision of computing's future.

A rather crippling weakness is a lack of application software. Although the industry know it alls would disagree, I would consider this NeXT's biggest blockade. We have been hearing now for some time that "Truly revolutionary" and "insanely great" applications are on the way. Yeah, and the check is in the mail, too.

NeXT need to attract more developers, too. And their current agenda will do any thing but. It must first be stated that NeXT's development environment is second to none. The tools available to create the best looking and easiest to use applications are all there, as is some truly inspiring hardware. So NeXT Engineering should take a bow. But NeXT marketing behaves as if they had never operated in the computer industry before.

Recently, those of us who were luck enough, received beautiful invitations to NeXT's upcoming product announcement at Davies hall in San Francisco. To my surprise (an probably the surprise of all the people who have been sweating over the world's most annoying keyboard for the last year or more) the announcement invitation prominently features Lotus Development Inc. To be specific, the invitation reads "NeXT Computer, Inc. invites you to join Lotus (in BIG gray letters) and other industry partners at Davies Symphony Hall 18 September 1990".

This is the most direct and effective way to infuriate every one else who has been writing NeXT software. To be fair, I should point out that there is a whole slew of companies listed when you open the invitation open further. But to prominently feature Lotus in this way can only be taken as a clear slap in the face to all of the hardworking folks at places like WordPerfect, Ashton-Tate, Software Ventures, Stone Design and Lighthouse Design. But I reserve the bulk of my pity for Frame Technology and Wolfram Research. In the early days these two companies were the only real friends in the software industry that NeXT had. But apparently they were only friends as long as NeXT found them useful.

It makes one wonder what sort of secret deals Steve Jobs and Jim Manzi must have worked out.

How senseless, too. There is no reason to feature any one developer over another. All of them have laboured like titans to produce the one thing that NeXT must have to survive: quality applications. The paying field must be as close to level as possible, otherwise, the small developers (who usually have the best ideas and the most innovative ideas) will be frightened away. Because in the end, you may have the coolest, fastest color machine in the whole world, and you could sell it for next to nothing, but with no software, It's just a attractive, black doorstop. And that's the way I see it.