Computational Tools for Cyclotron Design, Commissioning, and Operation

C.J. Kost

TRIUMF, 4004 Wesbrook Mall, Vancouver, B.C., Canada V6T 2A3

ABSTRACT

Many support systems are required in the design, commissioning, and normal operation of a modern cyclotron. Presented is an overview of the computing environment developed during these various stages at TRIUMF. The current computing environment is also discussed, with emphasis on how one can provide an integrated system which is user-friendly.

INTRODUCTION

Our Computing Services Group has been involved in many of the computing aspects in the design, commissioning, operation, and application phases of the TRIUMF cyclotron. I will not discuss any details concerning the many computer codes[1)] that are now used in particle accelerator design. I will review how our computing has evolved over the years but due to my own limited knowledge of all the computing aspects involved I will confine the scope of this paper to those areas in which our group was personally involved. Some emphasis will also be placed on the general application software tools OPDATA,[2)] EDGR,[3)] and PLOTDATA[4)] developed by our group.

THE PAST

In any historical discussion of the development of the computing environment at a cyclotron during its design, commissioning, and operational phases, one must take into account the dramatic changes in computing *hardware* technology which take place during these stages. In just the aspect of computing capacity per constant dollar value, there has been, and there continues to be a rough doubling every 2-3 years! Even taking this aspect into account, the computing environment currently supported by the 5 full-time members of the "Computing Services Group"is the result of a unique set of circumstances. Back in 1969, just before starting the construction of TRIUMF, the main campus computer was one of the earliest computers to offer a time-sharing operating system, called the Michigan Terminal System (MTS). During TRIUMF's initial design and commissioning stages (1969-1973) the computing tools used were mainly a series of 'batch' oriented codes. Dominantly these were orbit codes such as GOBLIN[1], CYCLOPS,[5)] COMA,[6)] and TRIWHEEL[7)] and beam line transport codes such as TRANSPORT[8)] and REVMOC[9,10)] used to design the primary and secondary beam lines. However, it was during this time that the seeds were sown for the development of 'interactive' codes. Small interactive codes were written, which allowed for better feedback than submitting jobs on a card reader, and run from teletypes in 1971 and raster scan video display terminals the following year. Batch style programming peaked in 1973 with the monumental effort required to 'shim' the main magnet using a computational loop involving at least 6 batch programs. During 1974 many small interactive codes were written in support of beam development. These codes had a lot in common, differing mainly in some details. It was at this time that, as an experiment in programming, we developed RELAX3D,[11)] a pure FORTRAN version of a program partly written in assembly language, which calculated 3 dimensional electrostatic fields for the central region of a cyclotron. Due to memory constraint of the then current computers, this program was limited to solving only 'test' cases . It was to take another 5-10 years before this code[12)] became a valuable resource for TRIUMF and other laboratories. However, it illustrates the value of long range planning and the need for continuity in program development. Without such commitments, many prematurely developed programming projects would be lost.

Using some more discretionary time, fortunately provided by management, our computing group began to develop a library of common interactive support subroutines.

[1] General OrBit LINear code from MSU

Our interactive programming style at first used the menu driven approach (eg. SPEAM,[13] written in 1973 to calculate beam line envelopes taking account of space charge). By the summer of 1975 we had developed our first full blown interactive code — INTRAN,[14] which was essentially a human interface to TRANSPORT but which more than doubled the total length of this program! We quickly found the menu driven approach awkward to use, as too many options became available, and settled instead on the command driven interactive style. By the following summer INTRAN had been completely re-written as a command driven program, now also running on graphics terminals. The next two years, with the help of some inspired work done by summer students, a new program called OPDATA was developed. OPDATA was an attempt to integrate into a single program the myriad of tasks previously done by many separate programs. Details of this versatile program are discussed below.

From a computing support point of view, the year 1980 was a critical year for TRIUMF. It was at this time that I was able to spend a year at SIN (now PSI) and work on implementing the MTS interactive graphics routines on a VAX. Upon my return TRIUMF had just purchased its own VAX and our computing services group was immediately able to provide centralized support, rather than the usual 'free-for-all' approach taken at many other laboratories. This head-start allowed our group to provide an integrated package of tools capable of solving a wide variety of computing problems.

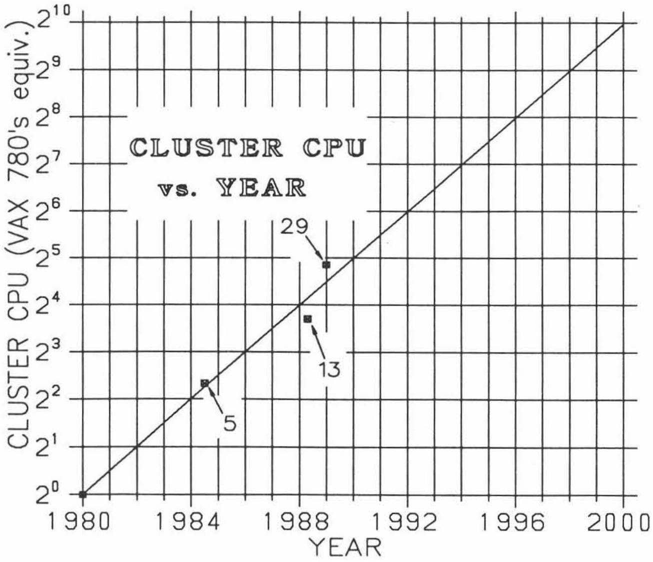


Figure 1: VAX Cluster CPU vs. Year

THE PRESENT

The long range goal of TRIUMF's Computing Services Group has been to provide an integrated computing environment which can deal with all scientific (as opposed to administrative) computing aspects of a modern cyclotron laboratory. This means dealing with (but by no means limited to) such aspects as:

- electronic networking.
- computing data security (backup, archive).
- computing hardware and software.
- cyclotron modelling and simulation.
- cyclotron controls and operation.
- data analysis.
- documentation preparation systems.

Remarkably, after some twenty years, the operating system for the main campus computer is still MTS. However its use at TRIUMF has been steadily declining with the growing popularity of VMS on VAXes. As can be seen from Figure 1, the CPU power of TRIUMF's VAX cluster is doubling approximately every 2 years. Figure 2 shows our current cluster configuration and network links in some detail.

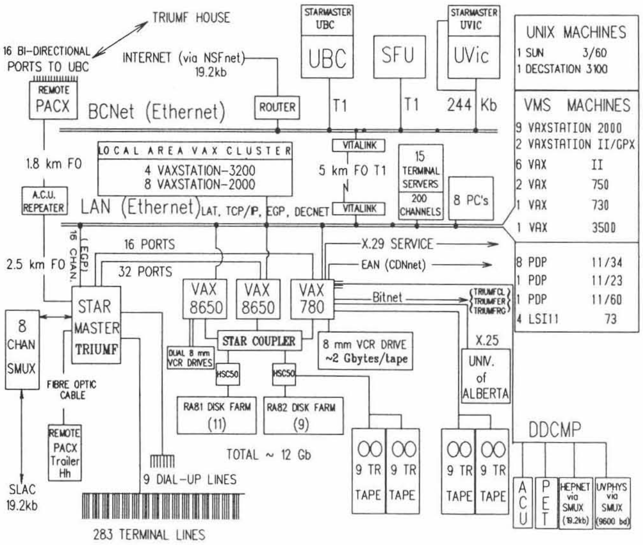


Figure 2: Current TRIUMF VAX Cluster

Electronic Networks

The evolution of our electronic networks has been just as dramatic as the increase in raw CPU power. As can be seen from Figure 2, many of TRIUMF's computers are linked to the site-wide Ethernet. As well, the three provincial Universities' Computing Centres and TRIUMF are interconnected by Ethernet (the slowest segment being a 244 Kbaud link to the University of Victoria, located on Vancouver Island some 80km away). This allows transparent access to our computing resources for these founding members of TRIUMF. The HEPNET and NSFNET links allow TRIUMF staff to keep in touch with the worldwide community involved in high energy physics. These links not only support electronic mail but also remote login access to thousands of DEC nodes around the world, many located at accelerator laboratories. This allows for easy exchange of software and access to data bases such as SPIRES at SLAC. Global electronic conferencing can readily be done using for example the VAX PHONE utility. Although most external mail is exchanged via BITNET, for those sites not on BITNET or HEPNET we use the utility GMAIL[2]. This relieves the users from having to learn about gateways and having to address them properly. Connections to our VAX cluster can also be made via X.29 using the public packet switching networks. The value of these electronic networks is hard to over-estimate — even such mundane tasks as arranging accommodations at TRIUMF House can be made by just sending an electronic mail message to `HOUSING@TRIUMFCL`.

As part of the overall strategy of providing network access to TRIUMF's computers we have installed a number of 2400 bd dial-up ports. With the popularity of the Atari-ST as terminals (discussed below) this has proven to be beneficial for home use, by experimentalists who want to monitor their experiment, programmers who want to develop and debug programs, and system managers who perform remote diagnostics on the site computers.

Hardware Standards

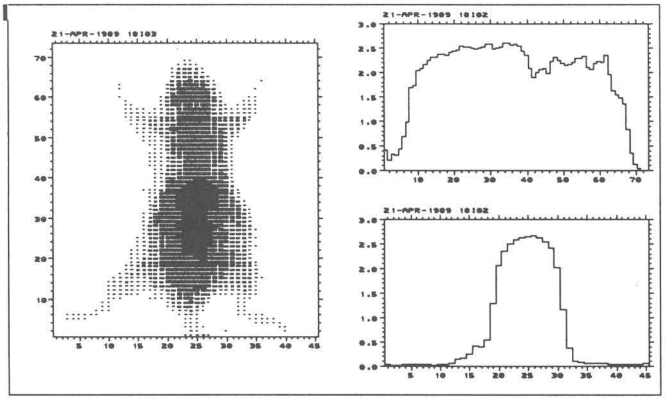


Figure 3: Atari high speed display mode.

CPU's and Workstations

Although in the past 2 years a considerable effort has been expended by our group doing evaluations of various computer hardware alternatives, we have repeatedly stayed with DEC computers. This is because of their user friendly operating system, wide range of integrated product line, network transparency, their extensive use at other cyclotron laboratories and the fact that it is much easier to maintain and develop software as well as manage systems for a single vendor environment than for a heterogeneous one. The popularity of VAXes is an extremely important factor for an international laboratory like TRIUMF. Many visitors coming to TRIUMF just bring a 'backup' tape, load it into a VAX and Voila!, they have the same environment as they had on a VAX back home... no learning curve. No matter how friendly our own utilities, some visitors still prefer to use their own (especially if they require non-English utilities!)

From Figure 2 one can readily see the significant impact made by the recent developments in workstations. They now perform more calculations than our mainframe machines. This is because they are currently the most cost effective way to add CPU cycles to our cluster. The current pace of workstation evolution is in such a frantic state that we can soon expect to see CRAY-I like performance on our desk for $10k! This decentralization of computing is expected to continue into the foreseeable future. Providing computing support in such an environment is considerably simplified by the VAX cluster environment which allows all CPU's transparent access to all cluster application programs.

Terminals

For terminals we have exclusively adopted VT100 compatible terminals which simultaneously support the Tektronix 4010/4014 graphics mode (mono, or colour). In line with maintaining an integrated computing environment our group continually evaluated and recommended the most suitable terminals. More than two years ago it became clear that PC's, acting as terminal emulators, could provide a more cost effective and flexible solution. The Atari-ST's with a locally developed terminal emulation package called ST640[15)] now form an integral part of our overall computing environment. This is because, unlike ordinary terminals they offer the following features:

- Low unit cost compared to terminals.
- Good quality 600×400 non-interlaced monochrome display.
- VT100 style keyboard.
- Mouse and/or keyboard control of graphics cursor.
- True zoom (4096×3120 resolution) and pan functions.
- Local storage/replay of text and graphics.
- Multi-page text recall buffer.
- Local graphics processing for fast 1/2d histograms (see Figure 3.
- Screen image dumps (under 6 seconds) to laser printers.
- Remapping of stored vector list (12 bit precision for 4014 mode) to the full resolution of an attached (laser) printer.
- Ability to add new terminal features!
- Can run LaTeX /TeX , word-processing, spreadsheet and other applications.

In addition, by attaching an IEEE-488 interface module

[2] Provided by Ed Miller—SLAC

the Atari can be used as a low cost programmable controller and/or data logger for HP-IB bus instruments. Overall the Atari's have been enthusiastically accepted by our user community, especially by the experimentalists who use its many tailor made features.

PC's

PC's can now run many beam dynamics codes but do not integrate well into the overall computing environment due to the limited availability of local tools and network support as has been provided on the VAX machines. DEC's Personal Computing Systems Architecture (PCSA) does provide virtual disk services, file services and printer services in addition to DECnet-DOS, and this has resulted in a better merging of the VMS and DOS environments. However, PC memory limitations, system management, a non-VT100 style keyboard, etc. have all impeded the smooth transition of conventional VAX users to the PC's. Nevertheless PC's are in extensive use at TRIUMF, solving many individual and usually dedicated tasks. Generally the software used on them are commercially developed products such as Autocad.

Digitizers

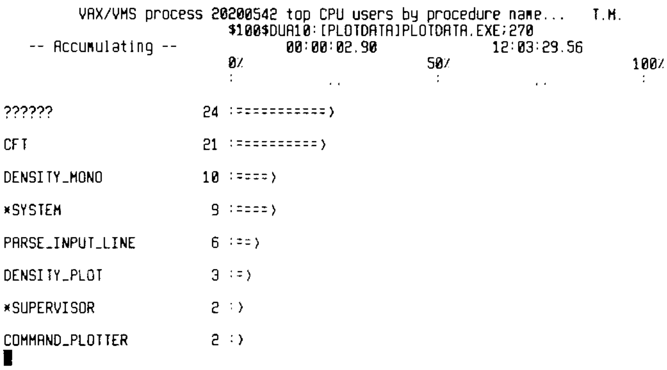
There is often a need to digitize data (eg. from other graphics publications). To meet this need we have interfaced our graphics package to two types of digitizers. One type is connected to a VAXstation, substituting for the usual mouse, while another type is more flexible in that it is simply juxtaposed between any graphics terminal and a computer and can thus send digitized co-ordinates directly to a file without the need of an application program.

Software Standards

For the development of application software we have standardized on using VAX FORTRAN-77. It was chosen because FORTRAN is the language understood by most scientists and engineers. In support of this programming environment we also provide FORTRAN versions of the IMSL, NAG, SLATEC, CERN, and TRIUMF libraries. In general we discourage the use of the *commercial* libraries for applications which may need to be ported to other systems since they may not be available there.

The symbolic mathematical computation systems SMP, MACSYMA, and REDUCE have also been made available. These systems tend to emphasize purely symbolic capabilities with some provision to translate lengthy algebraic expressions into FORTRAN source code. SMP (as are some other such systems) is also fairly advanced in being able to formulate large models or calculations symbolically *and then run them numerically.* Of the three systems SMP is supported in some depth by our group[16,17] and is now used on a daily basis. Since the learning curve is steep (but with commensurate rewards) it is essential that users have access to a local 'guru'.

INFO



```
          VAX/VMS process 20200542 top CPU users by procedure name...   T.M.
                              $100$DUA10:[PLOTDATA]PLOTDATA.EXE;270
   -- Accumulating --                00:00:02.90              12:03:29.56
                              0%                       50%                       100%
                              :           ..           :           ..            :

??????                     24 :===========)

CFT                        21 :==========)

DENSITY_MONO               10 :====)

*SYSTEM                     9 :====)

PARSE_INPUT_LINE            6 :==)

DENSITY_PLOT                3 :=)

*SUPERVISOR                 2 :)

COMMAND_PLOTTER             2 :)
█
```

Figure 4: INFO utility display.

This locally developed utility allows *anyone to monitor* the CPU usage *of any user process* running on a VAX and display a real-time histogram (optionally cumulative) of the % CPU time spent in each subroutine. See Figure 4 for a sample display. This tool has been of great value to easily and quickly isolate tardy subroutines. Commercially available tools such as DEC's PCA (Performance and Coverage Analyzer), which requires one to recompile the program with the debugger enabled and link with this utility, can yield more detailed information but are more difficult to use. In general these tools are not used as a matter of course but only when we find program performance to be unsatisfactory.

The Dynamic Trio

One of the main tasks of our group has been the pre- and post-processing of information used by orbit/beam dynamics codes. The philosophy was to write a series of programs (each fully integrated with presentation graphics) which, using a set of simple commands, can manipulate data files (on disk or in computer memory) in the manner required by a diverse set of needs — thus relieving the scientists and engineers from the rather mundane task of programming and allowing them to do those jobs for which they are particularly suited. To satisfy this ambitious goal one must always obey the traditional trade-off shown below:

GOOD – FAST – CHEAP
(PICK ANY TWO)

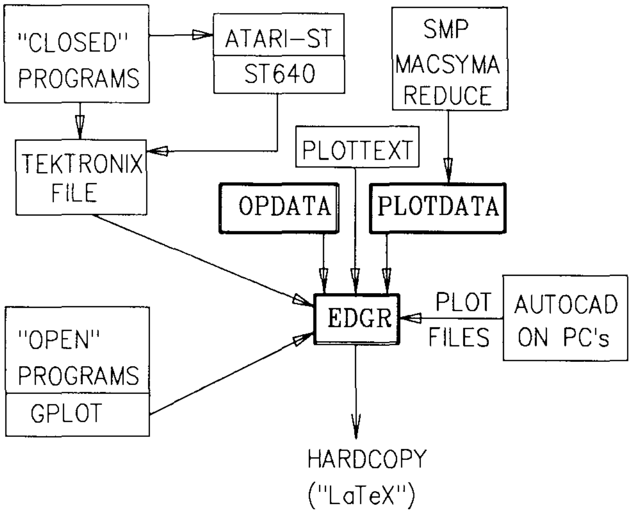


Figure 5: Trio overview.

We have chosen the first two – for the development costs have been high – more than 10 man-years.

As a product of long range goals our group has provided the VAX users with an integrated package of tools written in FORTRAN. The three most commonly used being OPDATA, PLOTDATA, and EDGR. The foundation on which these programs are built is a comprehensive library of graphics routines. This trio has proven to be so popular that its use now extends far beyond the bounds of cyclotron beam dynamics and has been freely distributed to over 100 research and educational institutions worldwide.

These programs are in a continual state of evolution, with the addition of features driven largely by user demands and changes in hardware technology. Their capabilities thus largely reflect the needs of cyclotron laboratories like TRIUMF to solve a large class of computing tasks without the need for the user to do any computer programming. Since they are such an integral part of computing at TRIUMF I feel that an outline of their capabilities is warranted.

Figure 5 shows an overview of how these 3 application tools relate to the user community. Note how virtually any program which produces graphics can benefit from our overall environment.

OPDATA (•) and PLOTDATA (▷)

OPDATA is an interactive command driven language designed mainly for the manipulation of dynamic data arrays, while PLOTDATA is an interactive command driven language whose central function is to produce graphical presentations of numeric data. The features provided by each is given below.

•▷ Curve fitting, smoothing and plotting.
•▷ Integration, Differentiation, Interpolation.
•▷ Digitizing data from screen plots.
•▷ Over 100 built-in math functions.
•▷ Dynamic allocation of arrays.
•▷ Built-in EDT and graphical editor EDGR.
•▷ Dual level built in calculator.
•▷ Generate arrays which are functions of others.
•▷ Expression evaluation including Boolean logic
•▷ Non-linear fitting of data with any expression
•▷ Flexible input/output of user data arrays.
•▷ Forgiving of input errors.
•▷ Allows scalar, string, and vector variables.
•▷ Display status and history of selected variables.
•▷ Asynchronous trap handler to allow command abort.
•▷ Dynamic and static 20 line input recall buffer.
•▷ Temporary access to operating system (DCL).
•▷ Macro capability (to 9 levels) with parameter passing.
•▷ Macro generation by 'prototyping'.
•▷ Extensive on-line help facility, instant command help.
•▷ Reference manual.
•▷ Customized X-Y line graphs.
•▷ Unlimited text titling, labelling and annotating.
•▷ Multiple axes with flexible positioning.
•▷ Log scales to any base, including e.
•▷ Underlying graphics support allows:
□ Output to many graphics devices.
□ More than 40 fonts available.
□ Zoom and pan functions via EDGR.
□ Windowing for multiple plots/image.
□ Merge current plot with previous ones.
□ Device-independent graphics metafile.
□ Generate TeX compatible graphics output.
□ Labelling (optionally) follow rescaled plots.
▷ Allow matrix variables.
▷ Two-dimensional interpolation for sparse data.
▷ Density, contour, surface plotting.
▷ Multiple viewports and windows.
▷ Contour plots with following features:
□ Use of gridded or sparse data.
□ Labelled or unlabelled contours.
□ Full or partial map scaling.
□ Rectangular or polar map.
□ Area/volume tabulations.
▷ Bar, pie and tile graphs.
▷ Density plots with following features:
□ Rectangular or polar map.
□ Optional profile projections.
□ Random point type w. optional colour.
□ Boxes w. size related to density.
□ Colour region fill.
□ Monochrome dithering patterns.
▷ 2-D surface plots with hidden lines.

▷ Filling between curves.
▷ Unlimited mixing of graph types on a single picture.
▷ Ellipse fitting to scattered data.
▷ Automatic prompting of command parameters.
▷ Looping, branching and conditional statements in macros.

▷ Create your own commands by 'aliasing'.
• Data filtering, Sorting, Binning, (de)Convolution.
• (Inverse) Fast Fourier Transform of data.
• Recursive and non-recursive digital filtering.
• Peak finding (manual or automatic).
• Minimum/Multiple real roots of an expression.
• Dynamic loading of user written modules.
• Up to 8 user functions from shareable image.
• Session conversation logging.
• Save/restore of session (including graphics environment).

EDGR

An interactive facility for creating, editing, manipulating, and storing graphical data. It has been developed primarily for scientific applications, as an aid to preparing high quality graphs and diagrams for presentation or publication. It also provides a device–independent medium of storage for graphical output produced by application programs. It is extensively used in conjunction with the above utilities. Its capabilities are comparable to those of a text editor operating on text files.
• All graphics attributes stored in 32 bit precision.
• Box, circle, ellipse, arc, (poly)line, with fill patterns.
• Text in over 40 different fonts, with easy entry of math symbols following the TeX syntax.
• Graphical elements or images can be cut, pasted, scaled, rotated, translated and included in larger images.
• Zoom/pan functions (multiple windows on VAXstations).

• Layer facility allows selective editing and overlaying.
• Magnified views are available on display and hardcopy.
• Entry or modification of drawing elements attributes such as position, size, colour, orientation, justification, font and line type.
• Attributes such as colour, line type, or text font, angle and justification can be modified (sub)globally or on an element-by-element basis.
• Merging of one image into another, by overlaying or by scaling into a specified area.
• Composition of several images onto a single page.
• Multi-frame facility for archiving of images.
• Translation, scaling, rotation and duplication of elements.
• Various lines types (user definable) and thicknesses.
• Standard function keys for distance and angle measurements, reference points, markers, numerical coordinate entry.
• Grid overlay with variable origin and spacing with 'snap'.
• Read as input images generated in Tektronix format from other packages (eg. UGS).
• Read in Autocad drawings saved in plotter format.
• Hardcopy support to many devices in various formats.
□ TeX compatible output for HPLaserjets.
□ HPLaserjet, QMS, $LN03^+$, Imagen laser printers.
□ Zeta, HP, and Houston Instruments plotters.
□ HP Thinkjet, HP Paintjet, Printronix printers.
□ Postscript printers (e.g. Apple Laserwriter).
• Survey function for export of non-text numerical coordinates.
• Comprehensive on-line documentation.
• Key driven help menus for active graphics cursor.
• User's Guide.

The graphics editor EDGR can be run as a stand-alone program or called as a procedure by user programs.

Underlying the above routines are the TRIUMF graphics subroutines. They consist of various drivers, viewporting, windowing, and other low level routines such as AXIS and NUMBER. At an intermediate level we have the GPLOT routines, which are a set of user friendly subroutines for X/Y plotting. For example, to plot X vs Y complete with axes and bring up a menu to allow the current plot to be edited, copied to some hardcopy device, etc.

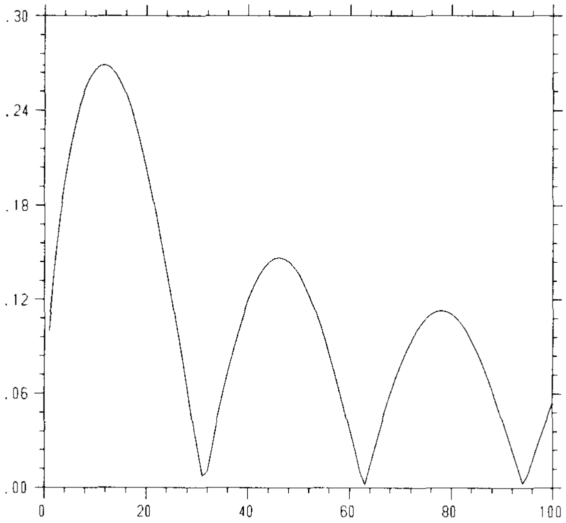


Figure 6: Original sample program output.

```
C
C  Complete test of GPLOT
C
      REAL*4 X(100),Y(100)
      NPT=100
      IAXIS=3 ! plot data first
C
C Define your own data.....
C
      DO 10 I=1,NPT
```

```
        X(I)=I
       Y(I)=ABS(SIN(I/10.)/SQRT(I*1.))
   10  CONTINUE
C
   90  CALL GPLOT(X,Y,NPT,IAXIS)
C
C  Now change what you want !
C
       CALL GPLOT_CONTROL(' GPL>',&90)
       END
```

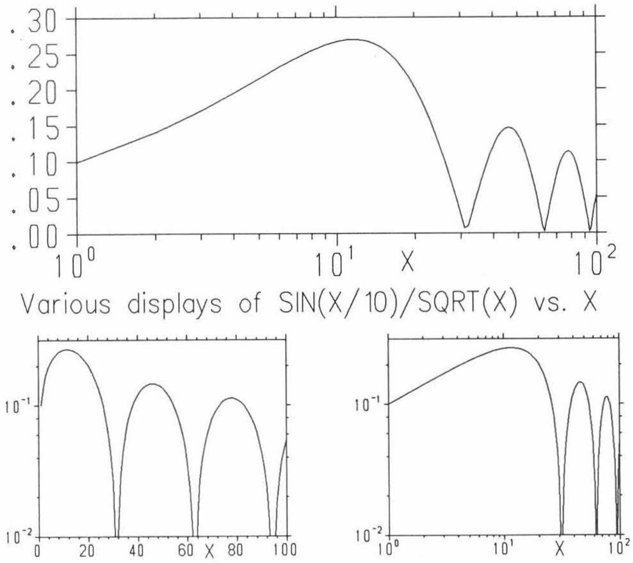


Figure 7: Modified sample program output

Figures 6 and 7 illustrate the initial and possible final output produced by this sample program.

Sample Applications

The application of this trio of programs is so wide spread that it is used by nearly all computer users at TRIUMF. To illustrate just a few:

Cyclotron Design

An important part of cyclotron design is the graphical visualization of numerical data. As an example Figure 8 show the output of OPDATA in plotting a vector field. Note how each point has an associated co-ordinate, vector length, angle and colour (not shown).

The harmonic field profiles, polar 'density' and field gradient plots of a 4-fold symmetric magnetic field are shown in Figure 9. A listing of the sequence of PLOTDATA commands generating this follows.

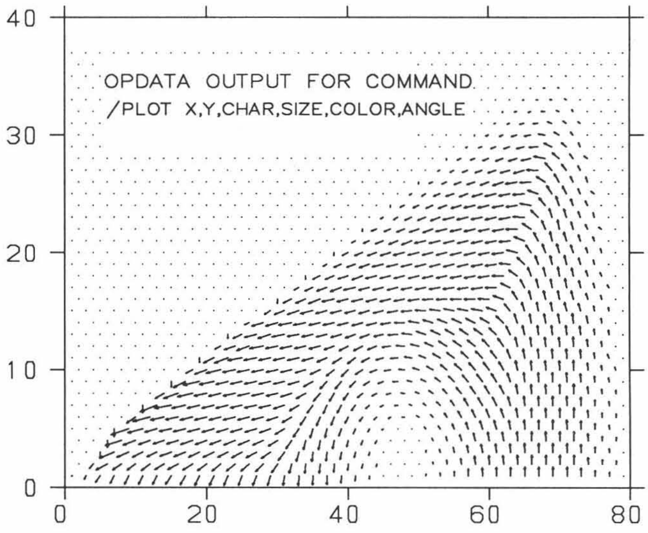


Figure 8: Electric field map

```
READ2/MATRIX LAX2.DAT/(97) M 90            |  SET  %YLOC   53.
GENERATE RADIUS 1,,99 99                   |  TEXT "Radius (cm)"
GENERATE THETA 1,,90 90                    |  SET  %XLOC   26.
TRANSPOSE M                                |  TEXT "Radius (cm)"
FFT M A P                                  |   SET
BITMAP HPLASER 300                         |    %XLOC    55.
DEVICE OFF                                 |    %YLOC    1.
PORTRAIT                                   |
AUTOSCALE COMMENSURATE                     |  TEXT "Radius (cm)"
WINDOW 5                                   |   SET
DENSITY/POINTS/POLAR RADIUS THETA M        |    %XLOC    5.
WINDOW 7                                   |    %YLOC    40.
DENSITY/POINTS/POLAR/DERIV RADIUS THETA M  |
WINDOW 4                                   |  TEXT "Kg"
DEFAULTS                                   |   SET
AUTOSCALE ON                               |    %XLOC    39.
GRAPH RADIUS A[1,1:99]                     |    %YLOC    40.
CURVE RADIUS A[2,1:99]                     |
CURVE RADIUS A[4,1:99]                     |  TEXT "Average field"
WINDOW 0                                   |   SET
 SET                                       |    %XLOC    55.
  %XLOC    25.                             |    %YLOC    26.
  %YLOC    92.                             |
  %TXTHIT  2                               |  TEXT "4<^>th<_> harmonic"
   TXTANG  0                               |   SET
   CURSOR  -2                              |    %XLOC    41.
                                           |    %YLOC    13.
 TEXT "Magnetic Field"                     |
 SET  %XLOC    76.                         |  TEXT "8<^>th<_> harmonic"
 TEXT "Magnetic Field Gradient"            |  HARDCOPY
```

Cyclotron diagnostics

Figure 10 shows the result of using OPDATA to obtain $V* \cos(\phi)$ vs. energy from a radial probe scan of the time-of-flight vs. radius. The OPDATA procedure used was:

```
/READ TMON.DAT (X,Y) (2,5)          ! Read data from columns 2 & 5
TUSEC=(Y*327.4/1023+95.)            ! Convert time to micro-seconds
TURNS=TUSEC/.2168                   ! Calculate turn number
/READ HE1TABLE.DAT (RHE1,RAV)       ! Read probe conversion table
/INTER (RHE1,X) (RAV) (R)           ! Convert probe radii to averages
E=939.3*(1/SQRT(1-(R/407.4)**2)-1)! Calculate energy from average radii
/SMOOTH .01 (E) (TURNS) (TI)        ! Smooth turns vs. energy
DT=TURNS-TI                         ! Difference between raw and smoothed
/SMOOTH .2 (E) (DT) (DTS)           ! Smooth difference only
TS=TI+DTS                           ! Add smoothed difference to baseline
/DERIV (TS) (E) (DEDT)              ! Calculate derivative DE/DT
VCOS=DEDT*1000/4                    ! Energy gain is 4*dee voltage
/PSCALE 0 600 45 85                 ! Set plot scales
/PLOT (E,VCOS)                      ! Plot V*cos(phase) vs. Energy
/PSCALE 0 600 0 0                   ! Autoscale Y for overlay
/PLOT (E,TI)                        ! Overlay Time-of-flight vs. Energy
```

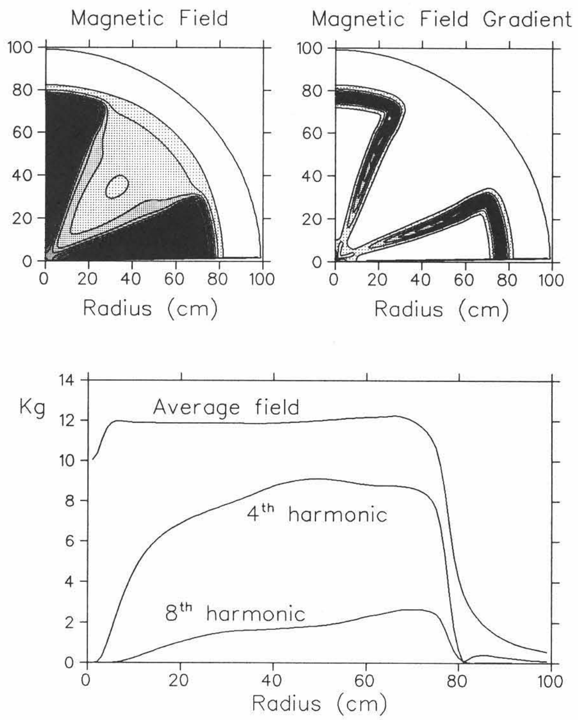


Figure 9: CP30 magnetic field plots.

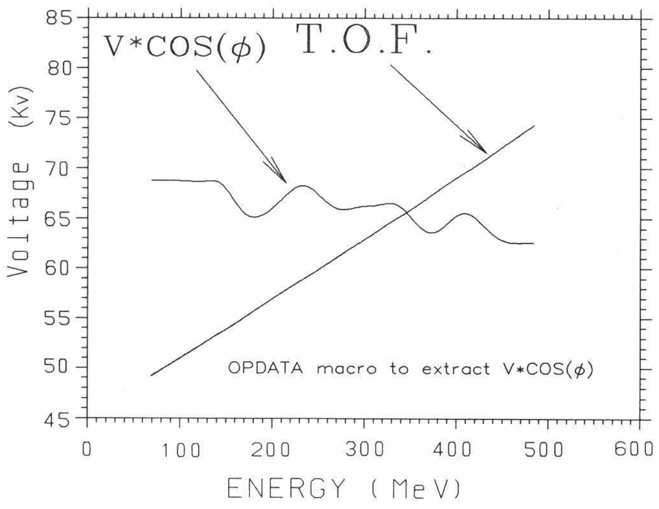


Figure 10: Determine V*COS(ϕ)

Figure 11 shows the application of OPDATA in smoothing data using Fast Fourier Transforms. For illustrative purposes a noisy Gaussian was used. The OPDATA procedure used was:

```
X=I                                     ! Set an array to the integers
VECTOR (X) 200                          ! Set length of X to 200 elements
Y=EXP(-(X-100)**2/500)*(.8+.4*RAN(X))! A noisy gaussian
FFT Y YA,YP                             ! FFT to get amplitudes and phases
YA=YA*(I<8)                             ! Zero amplitudes below 'cut-off'
IFFT YA,YP YS                           ! Reconstruct smoothed array YS
```

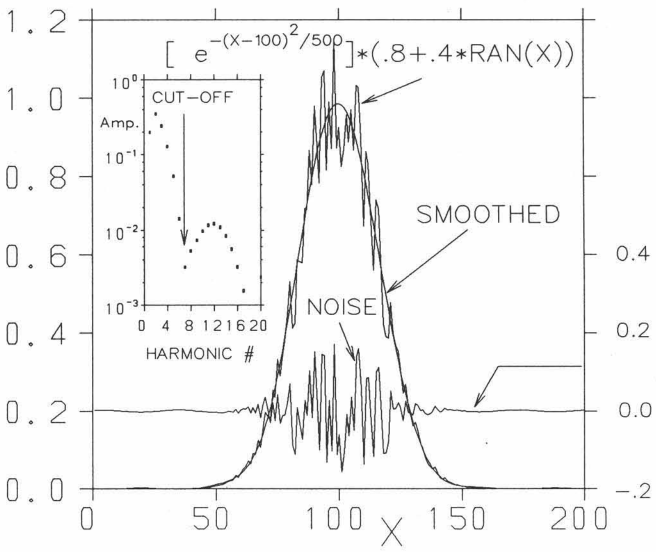


Figure 11: Data smoothing using Fast Fourier Transforms

```
PLOT X,Y                              ! Make plot ....
  ...
  ...
```

Cyclotron controls

Figure 12 shows a snapshot of a real time display of the trim coil monitor program[18] running in 'overlay' mode. This program is wholly based on the low level graphics package GPLOT.[19]

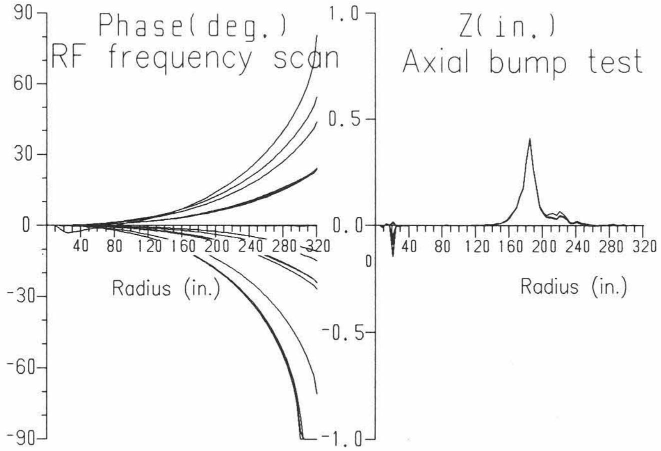


Figure 12: Real-time trim coil monitor

Cyclotron operations

A set of macros was used, together with OPDATA and DCL procedures to prepare weekly beam status reports as shown in Figure 13.

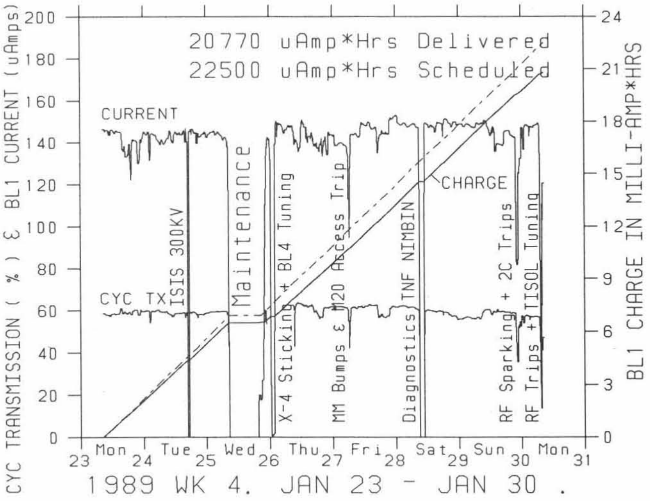


Figure 13: Weekly beam status reporting

Text Processing

It is interesting to note how modern technology has once again returned the generation of publications back into the hands of many scientists and engineers! This does not mean that secretaries are out of a job. In fact, the secretarial staff has been just as enthusiastic with this new technology and allows for an even more harmonious relationship since we now 'speak the same language'.

Almost all work involving scientific formulae is printed using the typesetting programs TEX and LATEX, on the HP Laserjet printers located around the site. Rather than provide a high volume centralized printing facility the users prefer to have the printing device close at hand. This is largely the result of the fact that these programs are not **wysiwyg** (*what you see is what you get*). There is still a need for a wysiwyg system (as currently provided by our somewhat dilapidated AES system) and we are currently planning to use WordPerfect on VAX workstations, networked to the site Ethernet, for this support.

The tools we have developed at TRIUMF allow almost all graphics programs in use at TRIUMF (whether incorporating our graphics packages or those from other laboratories) to generate presentation quality plots which can be merged with the TEX documents, thus doing away with the traditional 'cut-and-paste' method.

Misc. Utilities

To aid the user community in other associated (perhaps non-computing) tasks at TRIUMF, the following is a partial list of some readily available miscellaneous utilities.

- TELBOOK: Computer search of TRIUMF employees and visitors to find their phone numbers or visa-versa.
- BOOK: To maintain your own personal directory of telephone numbers.
- BROADCAST: Broadcast a message to another user.
- CALCULATE: A calculator utility.
- CLUSER: Everything you want to know about current cluster usage!
- CYSTATUS: Prints current Cyclotron status information.
- DSM: Allows group managers to control disk space quotas within a group.
- FREE: Shows disk space usage for all clustered machines.
- KERMIT: Communications protocol to transfer files to PC's etc.
- MATHDOC: Command driven information retrieval system for the SLATEC, IMSL, NAG, and TRIUMF libraries.
- PLOTTEXT: Another popular command driven program used to produce plots of formatted text. It is extensively used by those wanting to prepare presentation viewgraphs or poster session material in large formats.
- S38: Allow access to budget/accounting information.
- SPELL: A spell checker which works well with LATEX.
- STORES: Keyword search of items available in stores.
- TRIP: Lists site wide printer queue names, logical names, physical location, and printer type.
- WHOIS: List all registered usernames and accounts by knowing only part of this information.

Documentation

Most of our integration efforts would be wasted if we did not properly document our work and provide easy access to this documentation. To this end we have documented all our application programs in the machine readable form of LATEX documents, provided for a good way to bind them so that pages do not get lost and finally distribute them from an off-the-shelf central location. We rely heavily on feedback from our user community to improve both the quality of the application programs and the documentation.

THE NEAR FUTURE

- With the dispersal of computing to many (possibly heterogeneous) workstations an important task will be to provide software support for this distributed computing environment.
- As users migrate away from using conventional graphics terminals, much of our efforts will be devoted to software support of X-window based workstations, personal computers, and terminals.
- The functionality of the core utilities OPDATA, EDGR, and PLOTDATA will continue to expand rapidly. Emphasis will be placed on merging OPDATA and PLOTDATA into a single utility as well as providing them with a better

interface to the experiments being done at TRIUMF.

CONCLUSIONS

Our group has endeavored to provide an integrated computing environment at TRIUMF, the heart of which is a good graphics system and user friendly hardware. With the tools provided, data, in all its various forms can be readily manipulated, displayed, and the desired results placed into publications for submission to journals. Improving these tasks as advances in computing hardware take place will be our main future goal. It is interesting to note that our software efforts to solve many aspects of the design, commissioning, and operating of a cyclotron have become important applications at many other sites and in many ways provided superior alternatives to scientific data analysis and visualization compared to the few costly commercial products available for these tasks.

ACKNOWLEDGEMENTS

The author is indebted to contributions from the members of the Computing Services Group – Ron Balden, Joe Chuma, Fred Jones, and Richard Lee, as well as the many who have contributed over the past 15 years– especially Arthur Haynes and Phil Bennett.

REFERENCES

1)Los Alamos Accelerator Code Group, "Computer Codes Used in Particle Accelerator Design", LAUR-86-3320.

2)Bennett P. and Kost C.J., "OPDATA Reference Manual", TRIUMF internal report, TRI-CD-87-04.

3)Jones F.W., "EDGR: Graphics Editor User's Guide", TRIUMF internal report, TRI-CD-87-01.

4)Chuma J., "PLOTDATA Reference Manual", TRIUMF internal report, TRI-CD-87-03.

5)Gordon M.M., "Computation of Closed Orbits and Basic Focusing Properties for Sector-Focused Cyclotrons and the Design of 'Cyclops'", Particle Accelerators, 1984 Vol.16 pp.39-62

6)Kost C.J. and Mackenzie G.H., COMA - A linear motion code for cyclotrons., IEEE Trans., NS-22(3), 1922 (1975)

7)Kost C.J., "Users Guide to TRIWHEEL", TRIUMF internal report, TRI-DN-71-48.

8)Brown K.L., Carey D.C., Iselin Ch., and Rothacker R., "TRANSPORT, A Computer Program for Designing Charged Particle Beam Transport Systems", CERN manual 80-04. (1980)

9)Kost C.J. and Reeve P.A., A Monte Carlo Beam Transport Program, REVMOC, Proc. EPS Conference on "Computing in Accelerator Design and Operation" Berlin, September 1983, p. 158

10)Kost C.J. and Reeve P.A., TRIUMF internal report, TRI-DN-82-28.

11)Houtman H. and Kost C.J., A Fortran Program (RELAX3D) to Solve the Three Dimensional Poisson (Laplace) Equation., Proc. EPS Conference on "Computing in Accelerator Design and Operation" Berlin, September 1983, p. 98

12)Kost C.J. and Jones F.W., "RELAX3D User's Guide and Reference Manual", TRIUMF internal report TRI-CD-88-01

13)Joho W. and Kost C.J., " SPEAM, A Computer Program for Space Charge Beam Envelopes", TRIUMF internal report,TRI-DN-73-11.

14)Kost C.J., "INTRAN, A User Friendly Interface to TRANSPORT", TRIUMF internal report.

15)Kost C.J., "ST640, a Terminal Emulator for the Atari ST", TRIUMF internal report.

16)Balden R., "Symbolic Computation and Very High Level Programming Languages, Part I: A Potpoutti of SMP", TRIUMF internal report TRI-DN-88-13.

17)Balden R., "Symbolic Mathematical Computation using SMP: Worked Examples I", TRIUMF internal report TRI-DN-89-20.

18)Kost C.J., "Application of High Level Programs in a Controls Environment", Proc. 1987 Particle Accelerator Conference, IEEE, 719

19)Chuma J.L., "Low Level Graphics and Graph Plotting", TRIUMF internal report TRI-CD-87-02