the I.P. Sharp newsletter

JANUARY/FEBRUARY 1978
Volume 6 Number 1

One of the characteristics of time-sharing users is that they sooner or later feel that they must go in-house. As a result of this pressure, which is brought about for several reasons, some to do with economics, some with politics, we are making a fairly significant change in our overall strategy with respect to APL.

We intend to encourage such aspirations and we will be producing versions of our APL system which will run on customers' IBM (or equivalent) computers under a variety of IBM-provided operating system environments. We have, in the past, provided our software to run on a dedicated computer, but the cost to a customer is quite high. Now, we are talking about a much lower threshold of economic viability for the move in-house by making the SHARP APL software operate under VM, MVT, SVS and MVS in addition to DOS.

Another inhibiting factor for the APL in-house user has been the high cost of duplicating those parts of our communications network in which he is interested. We are proposing to provide the communications software as part of the overall package, and hence require that the customer provide a small, dedicated IBM 3705, which we will in turn connect to our network. The customer's computer would then become a host processor on our network, and could be selected as part of the normal sign-on sequence from any network location.

Unfortunately, in certain countries the common carriers do not permit this degree of efficiency in the use of facilities they provide, but it is clear that the rules will have to be relaxed in the future to permit line-sharing arrangements, even though we do not expect to see them encouraged. Thus, complete network access may not be possible from all countries initially, but we feel that it should constitute part of our customers' long term planning for APL usage.

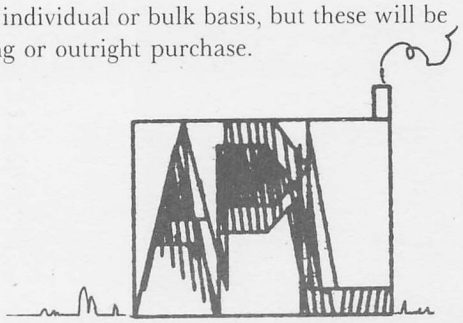
The charges we propose for a SHARP APL in-house system will be published soon, and they will vary a little bit with the degree of difficulty in installing, maintaining and updating the system in a particular environment.

Charges will be made in three areas:

1. **Installation:** This will be a fixed one-time payment covering training of customer personnel in operations and system programming, together with conversion of customer APL application programs.
2. **License fee:** This will be a fixed monthly fee.
3. **Usage charges:** This will be a variable charge proportional to the number of SHARP APL connect hours logged on the customer's machine.

These charges will include a complete maintenance and updating service. The company currently employs about a dozen system programmers who are engaged full time in the addition of enhancements and new facilities. All customer installations will be kept in step with the software which we ourselves run as a time-sharing service by software distributed through the network.

Application library programs will also be available for distribution on an individual or bulk basis, but these will be subject to a different licensing arrangement which will provide for leasing or outright purchase.



RATES: The standard rates for APL service have been revised. The most significant change is occurring in the connect charge which is being dramatically reduced. We have, in effect, replaced the connect charge with a character charge. Thus the cost to a customer of printing a fixed report is independent of the speed at which his terminal prints. Higher speed terminals do not materially affect the cost of doing a specific job, they are merely a convenient means of getting the job done quicker. The full rate structure is as follows:

$ 1.00 per hour of connect time for any speed of terminal
0.35 per cpu unit (reducing to .25 and .20 for immediate and deferred batch tasks)
0.60 per 1000 characters of input or output
0.55 per workspace per day (maximum size is 100K bytes)
0.55 per 100,000 file bytes per day.

Similar rates apply in all European countries and Australia. These revised rates are applicable to all new customers and we expect the majority of our existing customers will wish to switch over to them as soon as possible.

SHARP APL USER'S CONFERENCE

Next September 18th, 19th and 20th we will be holding a 3-day APL users meeting in Toronto. We want to let you know clearly the directions in which we as a company are moving, to tell you about the enhancements and additions to the system that we have planned, and to get your input on what you would like to see added to the system. That will occupy the first day.

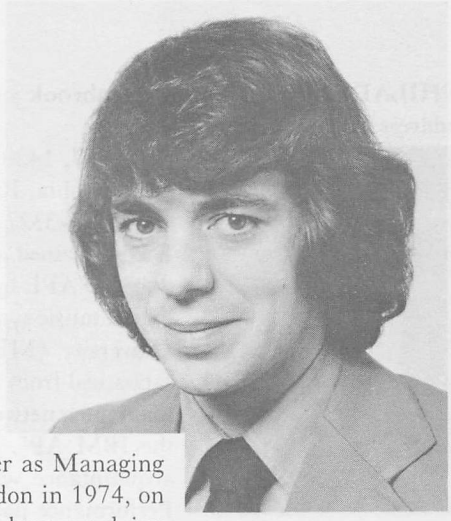
The next two days will be divided into half-day industry oriented seminars with emphasis on the way in which APL is being used. Each seminar will be led by two or three speakers from a particular user group who will describe successful and unsuccessful applications of APL that they have actually experienced. The industry groups and application areas to be covered will include:

1. Budget and financial consolidation
2. Applications in the petroleum industry
3. Applications in international banking
4. Quantitative analysis and statistical techniques
5. Actuarial and insurance industry applications
6. Logistic systems in multi-national manufacturing industries
7. Successes and failures in migration to in-house systems
8. APL programming standards
9. Applications for graphics
10. The use of minicomputers as special purpose terminal controllers.

The three-day meeting will be held immediately following our own Branch Managers' meeting, so you will find that your local Branch Manager will be in attendance and will be able to introduce you to whoever you feel could contribute most to the solution of your particular problems. There will be a registration fee and our aim will be to run the meeting on a break-even basis. We are of course anxious to get an early feel for the number of people who would like to come. A note or call to Rosanne Wild in Toronto or your local Branch Manager would be much appreciated.

SHARP NEWS

LONDON, ENGLAND:



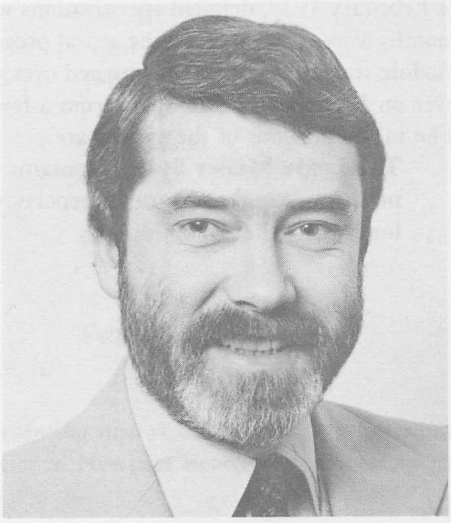
Fred Perkins

As mentioned briefly in the last newsletter, **Fred Perkins** has taken over as Managing Director of the U.K. company. Fred joined I.P. Sharp Associates in London in 1974, on completion of a research project at Dundee University where he had been applying computers to the investigation of predictive monitoring of the human electrocardiogram.

His basic training was in electrical engineering, with a BSc from Strathclyde University. Subsequently he gained an MSc in Control Engineering from the Universities of Warwick, Sussex and Bangor. It was a short research project on biological control systems that led him into biomedical research, and stimulated his interest in computers.

In London, Fred saw Sharp's Communications through its early days, before moving to Amsterdam in 1975. There he took control of the company's European network through its transition to the Alpha concentrator packet switched system. Having spent the last year as Managing Director of the Dutch I.P. Sharp Associates subsidiary, Intersystems BV, Fred has now returned to London.

He will maintain his commitments to IPSA communications, while having overall responsibility for the U.K. Company and its branches. He will be assisted in his task by **Hazel O'Hare**, the London Branch Manager (see October 1977 newsletter). On John Bassingthwaighte's return to Vancouver (see below), Hazel becomes Senior U.K. Branch Manager.

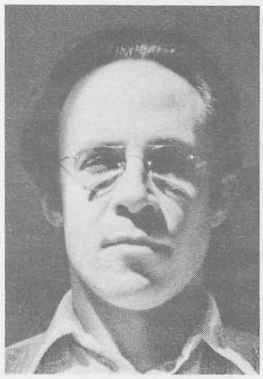


John Bassingthwaighte

John Bassingthwaighte was educated at the Universities of Toronto and Western Ontario. After joining Sharp as a summer student in 1967, he worked on data gathering and process control systems on PDP-8's, and a data management system on a SIGMA 7. He joined the APL group in early 1969 and organized the installation of the 360/50, thus becoming Operations Manager by default. After 18 months of operating experience, John moved over to the Toronto Branch as Manager in 1971, and established the branch in Vancouver a couple of years later. For the past year, John has been Managing Director of the U.K. Company, assisting the growth of branch operations there. Until he leaves in February to resume management of the Vancouver branch, he will be Senior Branch Manager for the U.K.

Many thanks to **Carol Aitchison** who looked after Vancouver so ably in John's absence.

PHILADELPHIA: **David Steinbrook** joined Sharp in January to open up the new Branch in Philadelphia. The address:



David Steinbrook

Suite 407, 1420 Walnut St.
Philadelphia, Pa 19102
(215)735-3327

David, trained as a pianist, composer, music-theorist, conductor and teacher, was introduced to APL by Dick Lathwell and Ken Iverson in May, 1974. Although familiar with digital music synthesis techniques through graduate study and instructorship at Princeton University (MFA 1965), he did not begin serious work in APL with music until a sabbatical from Swarthmore College starting in July, 1975. Investigation of an extended tonal pitch network first glimpsed in August 1975, was aided by a visitor's agreement with the IBM APL Design Group from November 1975 to June 1976. During this time, acquaintance with the work of Don Orth led to a prototype Composition/Analysis/Performance package in APL. Dick Lathwell's interface enabled experiments in sound synthesis starting in October, 1976, using an IBM 5100.

Business systems designed for the 5100 from March to October, 1977, include packages for Sales Contribution Analysis and Press Productivity. Extension of work on General Arrays in musical contexts to other time-process-oriented control systems may eventually lead to a fully general music system.

THE IBM 5100

Clive Edwards, Toronto

Hammerson Property Corporation is a major commercial development company headquartered in Toronto, which concentrates its business in Western Canada. In October of 1976 Johan Carnel, Controller of Hammerson contacted us to discuss the possibility of our programming their IBM 5100. Hammerson's requirement was to develop a full scale accounting system for six office buildings housing some 400 tenants in close to one million square feet of space.

After reviewing the initial specifications, we agreed to tackle the problem. Between mid-October and the beginning of February 1977, detailed specifications were laid out and the overall system design was finalized. The next seven months were spent in doing the actual programming, testing and rewriting, more testing and documentation. As each module was completed it was turned over to Hammerson to begin live usage. The entire system was finally turned over on September 1 and apart from a few minor changes has been running perfectly ever since.
The major portions of the system are:

The Lease Master System contains information about all building leases (approximately 400). The system produces leased and vacancy reports, expiry reports and detailed occupancy and lease income reports, by floor, for each of the six buildings.

- Each month, the Lease Master System produces monthly **Rental Invoices** for each of the tenants. The invoice amounts are automatically posted to a transaction tape that will later update the General Ledger and Accounts Receivable files.
- **Service Invoice Amounts** are entered manually each month and the system creates typed invoices and posts the amounts to the A/R and G/L files.
- **The Accounts Receivable System** which is linked with the G/L, provides an A/R listing plus an aged analysis of all open items.
- **The Accounts Payable System** provides the facility for automatic production of checks for payable items as well as a manual cash disbursements journal.
- **The General Ledger System** provides a complete set of G/L reports plus explanations of all entries.
- **Financial Statements** including Balance Sheets, Income and Expense statements and Budget comparisons. The reporting format for each of these reports is user definable at the time of execution of the programs.
- **The Budget Reporting System** provides plan versus actual comparisons for up to two years of data.

The hardware for the system consists of a 32K cpu, with an auxiliary tape drive and printer. Plans are underway to increase memory size to 48K to handle an expected increase in volume. Programs and data for the system are stored on some 45 200K tape cartridges. The entire system is written in APL and is completely interactive, so that operator training is extremely easy.

Now that the system is fully operational, the invoicing and reporting which once took up to six months to do manually, is finished in 3-4 days. In addition, management is alerted to out-of-line situations through a number of checks built in to the applications software.

FORECASTING IN SHARP APL

Frank Arthur, Calgary.

Introduction

Planning is one of the most important activities in any organization, private or public, and the cornerstone of this activity is forecasting - the process of predicting the future state of affairs, based on certain assumptions. This article reviews some of the quantitative forecasting methods currently in use, and relates them to programs available in the SHARP APL public libraries.

Quantitative forecasts are generated from either deterministic or non-deterministic models. A deterministic model assumes that factors affecting the object of the forecast and their associated parameters are known with certainty. Non-deterministic models, on the other hand, make no such assumption, and thus necessitate the use of statistical techniques. If yours is a world of certainty do not read any further, since this discussion deals exclusively with the latter class of models.

>PET< PA03630-Y660

FORECASTING (continued)

A fundamental assumption in statistical forecasting is that the factors that have affected the object of the forecast in the past are the ones that will determine its variations in the future. If this assumption does not hold, any figures generated from such models have to be adjusted appropriately. Statistical forecasts are generated using one of two possible approaches, namely the structural model approach and the time series analysis approach.

Structural Models

In a structural model, one attempts to establish a relationship between the object of the forecast and a number of explanatory variables. For example, the *SALES* of a product may be expressed as a function of the *GNP*, the *PRICE* of the product and *ADVERTISING* expenditures. Functions in 32 *REGRESSION* can be quite helpful in this process:

- the function *REGR* may be used to estimate the coefficients of the equation relating *SALES* (dependent variable) to the explanatory variables.

- the output from *REGR* also helps in the testing of the statistical significance of these coefficients, and the extent to which the independent variables explain variations in the dependent variable.

- given future values of the explanatory variables, point and interval estimates of the dependent variable may be generated with the function *PREDICT*.

A basic requirement in the estimation process is historical data for all the variables in the model. In the case of micro-economic variables like *SALES*, *PRICE* and *ADVERTISING*, raw SHARP APL or functions in 39 *MAGIC* may be used to handle the data management aspects. In the case of macro-economic variables like *GNP*, the user can retrieve the data from any one of the many public data bases available on the SHARP APL system.

Future values of the explanatory variables are needed for the generation of forecasts for the dependent variable. Forecasts of macro-economic variables may be obtained from outside forecasting agencies, such as the National Forecast Service, whose forecasts for the Canadian economy are on the Sharp system (accessible through 582 *NFS* or 39 *MAGIC*).

One of two approaches can be taken with regard to the other explanatory variables. If they are considered to be decision variables, then future values may be obtained from management plans. However if they are subject to uncontrollable factors, then ideally one has to develop additional equations for generating forecasts for these variables, if possible. In our example, we may have another equation relating *PRICE* to a set of other explanatory variables. The forecasting model is thus transformed from a single equation to a system of equations. Functions in 84 *EPLAN1* and 84 *EPLAN2* can be used to generate forecasts from such multiple equation models.

Although the structural model approach to forecasting makes sense intuitively, the implementation of an effective system is not an easy undertaking. The specification of the model alone can be a problem; added to this is the fact that future values of variables not forecasted within the model have to be provided. Consequently most users turn to the relatively straightforward approach of time series analysis.

>PET< PA03630-Y660

Time Series Analysis

The basic assumption behind time series forecasting models is that, if the factors that have affected the object of the forecast in the past are the ones that will affect it in future, then historical data on the variable contains **all** the information one needs for generating future values. The data requirements are therefore reduced, since only past values of the variable are needed.

Time series models range from the naive all the way to the very sophisticated. The simplest case involves the successive application of a growth rate to the current value of the variable. The growth rate can be either determined subjectively, or estimated from past data. Functions in 56 *GROWTH* may be used to estimate the growth rate implied in the past observations of a variable.

Another simple method is the moving average model, whereby future values are determined by computing the average of a number of past observations. A slightly improved version is the exponentially weighted moving average (EWMA) method, which assigns higher weights to the most recent observations (available in 39 *MAGIC*).

The above methods are simple and easy to apply, however the underlying theory is rather weak, and one cannot make any statistical inferences from the results. A small improvement over these methods is the method of fitting a curve (or a straight line) to past data, using functions in 32 *REGRESSION*. Where seasonality is present, some users apply a hybrid of techniques: 39 *X11* is used to deseasonalize the data, then a straight line is fitted to the result; forecasts from the straight line model are then adjusted by the appropriate seasonal factors.

The ultimate in time series forecasting however is the use of autoregressive, moving average (ARIMA) models, made popular by Box and Jenkins. The procedure is a three-stage process involving: the identification of the class of ARIMA models that fits the data (a process much closer to an art than an exact science); estimation of the parameters in the model and the performance of diagnostic checks; generation of forecasts using the selected model. Functions for doing all this can be found in 39 *BOXJENKINS*. Among the strong points of this technique are the fact that past forecast errors are explicitly taken into account in the estimation and forecasting stages, and the confidence intervals that are provided with the point forecasts. These models may even be used to generate forecasts for the explanatory variables in structural models. Used by themselves, ARIMA models have been shown to perform just as well as sophisticated multiple-equation structural models.

Conclusion

For those who want to venture into the business of forecasting (or are being forced into it), there is a wide spectrum of techniques available - from the very simple to the more complex. The method selected should be based on a number of factors, including

- the time available for doing the forecasting
- the availability of trained personnel and financial resources
- most important of all, the expected opportunity cost associated with being wrong.

>PET< PA03630-Y660

THE ART OF GOOD CONVERSATION

Stephen Taylor, Copenhagen.

My favourite systems are "open" systems. That is, a workspace full of functions, and a manual that says what arguments you can give them. No conversations, just good flexible functions that combine to do all sorts of useful things. (*MAGIC* is a good example of an open system.) Use the wrong arguments, kazowie! the function blows up. (But I **told** you not to do that. . .)

Which is fine, but some systems are used by people who can't tell the difference between a *VALUE ERROR* and a vallium pill; and find that they frequently go together. For them we write conversational programs. With these users, it is especially important that the conversation supports their work. Printing a text string and sitting around in ⎕ input may work a treat for you, but if someone panics at this point, they are likely to type in any response they know. One such response is always the name of the function they just typed in. . . Sort of chills you, doesn't it?

"Sometimes I wonder why I ask. . ."

Conversational programs have traditionally been the meat and drink of interactive programming, and SHARP APL with its ⎕ and ⍞ makes them particularly easy to write. But have they ever given you indigestion? If they have, the chances are that you'll recognise these symptoms.

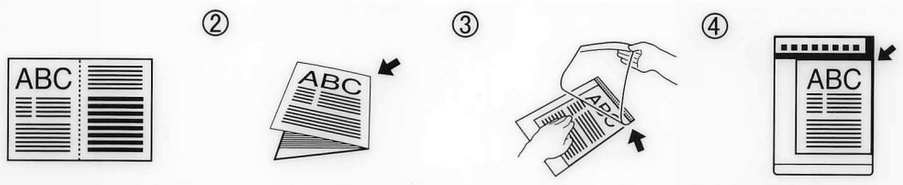
1. You got bored. Your program is a hundred and god-knows-how-many lines long, crammed with text strings you don't want to give workspace room to; and it keeps repeating tedious checks which only catch some of the errors anyway. Either the code is full of ⎕*ARBOUT* '' and assignments to ⍞, or the terminal paper from the input sessions looks more like wallpaper than an input record.
2. Your typist quit. That wonderful personality who knew what all the prompts meant, **and** how to restart the program just walked out of your life. No-one else has managed to get all the data in without typing a right-arrow, or re-invoking the function from one of its ⎕ prompts.
3. Worse, the only other person in your company who has *APL* experience also has psychotic fits during which he types garbage selected to crash your program and put all the files out of step.
4. Then your system expanded. Now that **handy** little one-off function for file updates has to be run fifty times in a row. You could get the input data into a file and run the whole thing overnight if someone didn't have to sit and answer all the prompts. Of course **some** day you'll get the time to write a new, batchable version.

". . . what the people need. . ."

The recurring problem of writing good conversation is that you have to be ready to respond intelligently to just about anything.

```
WHAT DID YOU SEE? CAMEL
** TOO MANY LEGS: 4
** TOO MANY HUMPS: 2
** NOT ENOUGH LEAVES: 0
NO JANE, NOT A CAMEL. . .
```


>PET< PA03630-Y660



```
WHAT DID YOU SEE? TRUCK
** TOO MANY WHEELS: 4
** TOO MANY FENDERS: 2
** NOT ENOUGH LEAVES: 0
NO JANE, NOT A TRUCK. . .
WHAT DID YOU SEE? HELP
ON THE WALK THIS AFTERNOON
WHAT DID YOU SEE? HELP
A SORT OF TREE.  NOW. . .
WHAT DID YOU SEE? LARCH
RIGHT.  NOW AFTER THE LARCH
WHAT DID YOU SEE? STOP NOW IT'S TIME FOR MY TEA
BYE BYE JANE
```

Now that was a supportive conversation. It's not hard to pick out some characteristics of good conversation, nor to add a few more:

* Explain yourself. If you don't like something that's said, say what you don't like and why.
* Be helpful when, and only when, you are asked for help. Go into some detail, and when you're asked for more detail, give it.
* Let people interrupt you. Don't be caught by surprise if they decide that they have something better to do.
* Don't be longwinded. The person you're talking to may spend all day answering these questions, so make them short to start with.
* Don't be pedantic. If you are offering a choice between CAMEL, CAT, and MONGOLIAN DESERT RAT, then M will do for the desert rat.
* Don't be fooled. If you let someone get away with 29th Feb 1975 once, they will forever be trying to trip you up again. Who needs it?
* Be slow to take offense. Some people may try to upset your program by making unexpected responses. Don't give them the satisfaction. These people get hot flushes from seeing

```
WHAT WOULD YOU LIKE FOR CHRISTMAS? HELP
CHOOSE CAMEL, CAT, OR MONGOLIAN DESERT RAT
WHAT WOULD YOU LIKE FOR CHRISTMAS? RAQUEL WELCH
INDEX ERROR
CHRISTMAS[103]
```

* Don't make people repeat themselves. If they really need to say something three times, let them say "ditto" twice.
* Accept stand-ins. People can't always spare the time to talk to you direct, and may wish to leave you with a file on which they have recorded their answers.
* Be reliable. People who type data in all day are often surprisingly nervous about things going wrong. Reassure them by not crashing, by anticipating their problems, and by behaving consistently.

>PET< PA03630-Y660

Now that is a lot to program around each prompt. Is there a general approach to the problem?
Yes. *TASK*. (*T*erminal *ASK*ing.) *TASK* is a prompting subsystem designed to follow the principles outlined above, and also to be readily integrable with other systems. Moreover, it cuts down the length of your own programs by keeping all the conversational programming out of sight in functions that you know will be returning a date, a vector, etc. You can find it in 999 *TASK*. It is intended for free use and adaptation, and there's documentation for it in file 999 *TASKMANUAL*. All the functions are unlocked and heavily commented. The *TASK* subsystem is supported by me (*SJT*). I'd welcome your comments, complaints and requests for help or extensions.

MAGIC NEWS

Clive Edwards

Additional features have been added to MAGIC to enhance its power and convenience. The following examples are used to demonstrate the changes:

1. The use of α embedded in the right argument of *LABEL* followed by the use of *DISPLAY* will cause a skip or blank line to be inserted for each occurrence of α.

2. For users of the SHARP APL CANSIM* Mini Base Supplement, concurrent series may now be retrieved using the *TO* option. When *TO* is employed, MAGIC will display an error message for those series that are not available, but will still retrieve any available series.

```
      )LOAD 39 MAGIC
SAVED  11.43.50 11/29/77

      CLEAR
      NOTOTALS
      DECIMALS 0
      SCALE 1000
      QUARTERLY,DATED 1 76 TO 4 76
      LABEL 'POPULATIONαCANADA,αNEWFOUNDLAND,P.E.I.,NOVA SCOTIA'
      DISPLAY CANSIM 'D1 TO D4'
```

	1ST/76	2ND/76	3RD/76	4TH/76
POPULATION				
CANADA	22,884	22,946	23,025	23,111
NEWFOUNDLAND	555	556	559	560
P.E.I.	118	118	118	119
NOVA SCOTIA	827	828	829	833

*CANSIM is the registered Trade Mark for Statistics Canada's machine-readable data base. When publishing any data retrieved from CANSIM, the following must be used as the source:

"These data originate from CANSIM which is a registered Trade Mark for Statistics Canada's machine-readable data base."

>PET< PA03630-Y660

Technical Supplement -14

LETTER FROM LONDON

APL USER GROUP - A CRABBY LOT Valerie Lusmore, London.

The APL user group is now a year old. Formed by a few APL enthusiasts, it has now grown to nearly 100 members and is affiliated with the British Computer Society as a special interest group. However, membership is not confined to "computer people" - APL users come from a variety of disciplines and the group provides a forum for the exchange of ideas. The increasing popularity of APL in the United Kingdom has unfortunately had side effects on the user group committee - they have all been too busy with work to arrange many meetings this year. This situation should improve soon. At the recent AGM, a new committee was elected, including two members from I.P. Sharp Associates, Dave Saunders and myself. If you would like to join and receive the newsletter CRAB-APL, contact us at our London office.

During a most interesting afternoon meeting, debugging techniques and *⎕ARBIN/⎕ARBOUT* were discussed. Mel Chapman, senior lecturer in Computing Science at North Staffordshire Polytechnic, who is at present spending six months sabbatical leave with Sharp in London, summed up:

Debugging Techniques. Rather than attempting the impossible - creating a set of ground rules representing a panacea for all system ailments, Romilly Cocking (who, incidentally, had just been re-elected chairman of the group) highlighted various pitfalls he had encountered recently. Contributions from the floor generated interesting and valuable examples.

At the start, the "ideal system" was identified as one that could grow, was flexible and that could be maintained with the minimum of heartache. It was generally agreed that as much as possible, this work should be done using the computer. Some arguments were generated over how and where functions should be commented. Meaningful variable names, and the problems caused by symbol table full errors were discussed. Mention was made of diagnostic aids available on various APL systems such as cross-references and workspace documentation packages. One common rule agreed upon is that you should always check the validity of your assumptions when faced with inexplicable bugs. "Efficient" but often less readable code was sometimes economically necessary, but it was concluded that compression should generally be tackled as a second stage of the problem, once the system works.

⎕ARBIN and *⎕ARBOUT*. The second talk consisted of a resume of the new SHARP APL goodies - *⎕ARBIN* and *⎕ARBOUT*. Dave Saunders reminded us of the good old days when only 15 character per second 2741-type terminals were available. These have given way to more versatile terminals with potentials which can only be realised by extending the range of instructions to the terminal. After a few examples of graphics and certain nice plotting facilities, he concluded with a strong recommendation that these tools should be used judiciously by those with the knowledge and shouldn't need to occur much in real life.

GUIDELINES FOR THE DOCUMENTATION OF APL SYSTEMS

Valerie Lusmore

Systems, and problems, generally come in three flavours.
(a) today's (one's own)
(b) this week's (which might well become someone else's)
(c) next month's (definitely someone else's)

These notes are intended as a guide for documenting a system with the minimum of trouble but sufficiently well so that the author can leave it alone and move on.

In a recent seminar on this thorny topic, the only conclusion reached was that, in order to document easily and well, the system should have been well-designed in the first place. These notes can be used to apply to systems that are perhaps not exactly well-designed, but it may not work as well in that case.

The required documentation can be split into 2 well-defined parts:

1. The user manual
2. The technical documentation.

Each part should also contain the general description of the system. This should be about a page long and should cover what the system does, for whom, how often, and so on.

USER MANUAL

This should contain:

1. How to sign on.
2. How to load any of the workspaces. What functions to use.
3. Security measures - how to be assigned access if you dont have it.
4. Options - the level of this depends on the system itself - how much on-line assistance there is. If state setting functions are used, how to tell the state. General responses to prompts, etc.
5. How to withdraw requests for highspeed prints or BTASKS if used. Reference numbers.
6. What to do when things go wrong - APL system crash, line-drop, error messages, restart procedures. How to tell if your system has **really** gone wrong - who to contact.

Items 1,5 and 6 are fairly standard - the others should be more (or less) detailed depending on the system and the assumed state of knowledge of the user, and how much on-line assistance is available.

TECHNICAL DOCUMENTATION

This should contain sufficient data so that someone else previously unacquainted with the system (but with some APL knowledge), should be able to fix bugs if any, or make changes. A reasonable requirement would be such that a person with one month's experience could, for a medium system (couple of workspaces, data base, limited multi-user, say) be able to modify an existing report or change a calculation line within, at most, 1-2 days. Modifications to such a system to add another country or data name should not take more than 2-4 days.

So that's what we're aiming at!
What do we need?
Basically, lists of the components of the system, workspaces and files, copies of input forms and general output examples. Also, notes on certain topics (see list). We should try to make the computer do as much as possible, but remember, quantity is no substitute for quality! Ideally, the text should remain on-line for ease of modification.

Notes should be concise and clear. Together, they should be a guide to the elusive factor - style. If the style of the author can be captured within these short notes, then the person using this documentation should be well on the road to success in understanding what to do. When there is no discernible style and consistency, perhaps this set of APL functions needs a clean-up before it is called a system.

Certain standards in function-writing will assist in making the whole system easy to understand and modify. If these have been used, this should be noted. For example, some good practices are:

1. Always have one exit line from all functions labelled *EXIT*. This makes the addition of monitoring simple.
2. Use meaningful variable names.
3. Explain global variables.
4. Be consistent.
5. For every function Line [1] should be a comment, stating briefly parameters, result and what it does. Put in enough to recreate the function without looking at the APL code.
   Example:

```
    ∇R←DUPSOUT X
[1]  ⍝R IS VECTOR X WITH DUPLICATES REMOVED
[2]  R←((⍳⍴X)=X⍳X)/X ∇
```

The comment is sufficient to produce another DUPSOUT function.

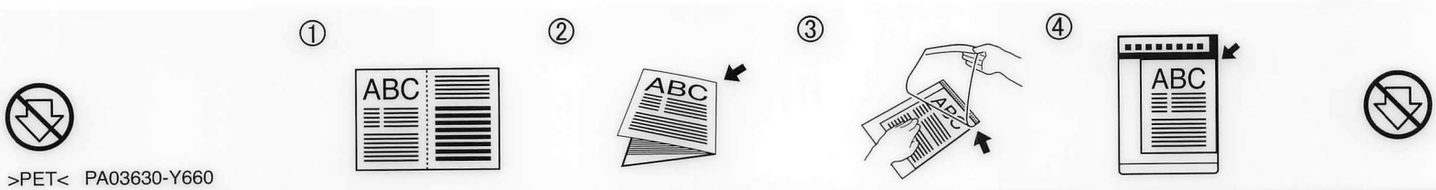
If you can't do this you probably haven't modularised correctly - especially at the bottom. Remember, each function should do **one** thing, however complex.

DOCUMENTS

1. *WSDOCS* for every workspace
2. *WSXREF* for every workspace
   Also for functions on a program file where possible.

The following items should then have one paragraph or page per topic. Some topics are optional because the system doesn't contain them but, if they have been used at all they should be covered, briefly and clearly.

1. SYSTEM ORGANISATION: An overall view of the workspaces and how they join together. Overall view of the files and how they "mash" together. If workspaces are organised in groups or packages from a program file, describe this in overview.



>PET< PA03630-Y660

2. FILES: For each file, a one-page description. Basic structure, layout. If a system standard is to keep this in the first component of the file, note it.
3. ⎕*LX*: For each workspace explain the ⎕*LX* and what it is intended to do. If changed, how, why and where. Restart procedures and functions.
4. FILE ACCESS ROUTINES: File-tie and read and write functions. File names, variable names for file access.
5. NAMING CONVENTIONS: Any major conventions, either for files, variables, or functions. Line-labels and branching conventions (e.g. use of if, or skipif)
6. USER INTERFACE: Any "ask" functions plus their syntax. Any limited vocabulary that produces special reactions *HELP*, ⎕ or →.
7. SYSTEM SECURITY: Show this is achieved. Any checking functions. How changed.
8. BATCH PROCEDURES: *HSP* or N/BTASKS. Reference nos. Description of special procedures. Any options.
9. SPECIAL MODULES: Example of report module or usage monitoring. Associated functions (see tree in *XREF*). Brief overview.
10. DEBUGGING TOOLS: An example of a "clean" function before)*SAVE*... or ⎕ to crash out the "ask" functions. How to stop driver functions.
11. LIBRARY INTERFACES: Examples with *HSP* or *PLOT*. Standard state-settings for these, if any.
12. ENVIRONMENTAL ASSUMPTIONS: Examples of terminal-type for tabs or line-throws.
13. LINKS: Any cross-references to other systems or data bases.
14. PROMPT CROSS-REFERENCE: Cross-referencing and listing of message file, if used.
15. CROSS-REFERENCE FOR ADDITIONS: If necessary, a brief description of all parts of the system which would be affected by the addition of, for example, a new product. Rough pointer to analysis needed.
16. FUNNIES: Anything that is likely to cause confusion to the uninitiated should be noted here. For example, the use of local variables global to functions further down the tree should be noted.

CONTEST We have a promise from Jerry Cudeck that there **will** be another contest. Entries for Contest No. 5 (W-Wre-Wreck-Wreckursion) should be submitted before March 1st, 1978.

Please accept our apologies for the error in recurrence relation (I) as printed in the December 1977 issue. It should read:

(I) `R[T]←A[T]+M[T]×R[T-1]`

NEW SHARP PUBLICATIONS
MAGIC for Time Series Analysis: Brochure
Applications Library Catalogue.
Chartered Banks Data Base, Annual: Application Brief.
Currency Exchange Rates Data Base: Manual
Financial Analysis: Brochure.

>PET< PA03630-Y660

3. Again for CANSIM users, a new feature *AUTOLABEL* (*NOAUTOLABEL*) will automatically retrieve the labels for series as they are accessed. It should be noted, however, that when *AUTOLABEL* is being used in conjunciton with *PUT*, only the labels for the last set of series which were *PUT* will be retained.

4. The *DECIMALS* command has been extended to allow individual specifications for each series to be printed.

```
DECIMALS 0 1 2 1 2 0
HIGHLIGHT
AUTOLABEL

TABLE CANSIM 'D1,D3,D6 TO D8,D11'
```

	CANADA	PRINCE EDWARD ISLAND	QUEBEC	ONTARIO	MANITOBA	BRITISH COLUMBIA
1ST/76	22,884	118.0	6211.50	8228.8	1018.20	2,457
2ND/76	22,946	118.0	6224.60	8250.0	1020.30	2,462
3RD/76	23,025	118.4	6240.00	8278.1	1022.10	2,470
4TH/76	23,111	119.1	6255.40	8306.1	1024.40	2,478

For a full list of recent MAGIC enhancements
)*LOAD* 702 *MAGICNEWS*
or contact your Sharp representative.

OTHER CANSIM NEWS (see 81 *CSNEWS*)

Supply and Disposal of Energy Data (Annual) Now Available
All series in Matrices **7140** through **7436** are now available as part of the I.P. Sharp CANSIM Mini Base Supplement.

Estimated Population of Canada by Province (Matrix 1)

The data contained in Matrix 1 is being revised from July 1971 to April 1976 to take into consideration the 1976 census counts. **All** data in the matrix is being reloaded to include a decimal point. The scheduled release date on the CANSIM Main Base for these data was November 30, 1977. The data has been available on the Sharp system from December 1, 1977.

FINANCIAL DATA BASE UPDATE

Mark Odho, Toronto.

F.P. Canadian Data Base: The Toronto Stock Exchange 300 Index data (High, Low, Close and Volume) for major and minor indices, is available from January 1977 to the present. Historical TSE 300 Close data as available via the workspace 51 *NEWTSEINDEX* has also been incorporated into the Financial Post data base (1956-1976).

TSE 300 Indices Daily Dividend Yield and P/E ratios can be accessed via the functions *FPDIVIDENDS* and *FPEARNINGS* respectively as in *LIB* 51 *FP*. These values are for the most recent trading day and change daily.

Bank of Canada Weekly Statistics: The Bank of Canada's weekly financial statistics are now available on the SHARP APL system. This data base contains the Bank of Canada's weekly financial statistics as released every Thursday afternoon. The data will be available in the data base by Friday morning. Historical data is available from August 31, 1977. All revisions to the previous week's data are made as they become available. Access is through workspace 54 *WBANKS*. There is no surcharge for the use of this data.

Chartered Banks Data Base (Monthly): The function *BREPORT* has been added into the workspace 54 *MBANKS*.

BREPORT produces a report to be printed on the highspeed printer, of the Chartered Banks assets and liabilities. The report includes fact and bank names as well as their numeric identifiers. One month of data appears per page. The function prompts the user for month numbers to be listed, and for delivery instructions.

Currency Exchange Data (UK) is updated **daily**. Please accept our apologies for incorrectly specifying weekly updates.

APPLICATIONS LIBRARY UPDATE

New workspaces

13 *NATIONS*	A system designed for demonstrating the capabilities of APL.
54 *WBANKS*	New weekly data for Canadian Banks (see above).

Workspaces Moved

54 *MBANKS*	The new function *BREPORT* provides enhanced error-checking (see above).

57 *LOAN*, 57 *LEASE*, 58 *BONDS*, 58 *DEPRECIATE*
all moved to *LIBRARY* 56, which now contains all financial analysis packages.

SYSTEM NEWS

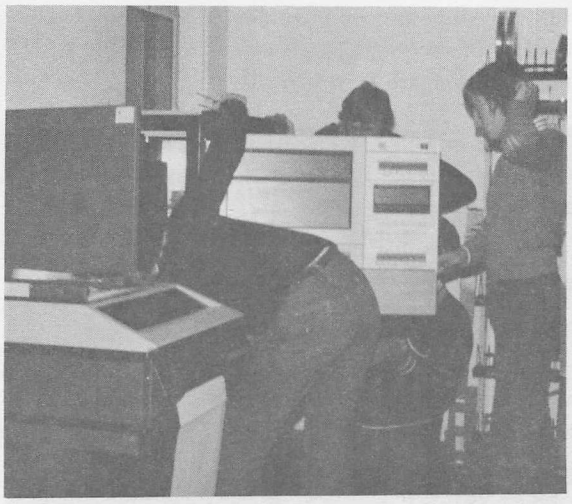
RELIABILITY

Bill Apsit, Toronto.

	Number Downs	Minutes Down	Scheduled Hours	% Up
1977 First Quarter:	27	220	1581	99.8
1977 Second Quarter:	16	286	1612	99.7
1977 Third Quarter:				
July	12	120	532	99.6
August	8	66	562	99.8
September	13	207	540	99.4
Total:	33	393	1643	99.6
1977 Fourth Quarter:				
October	9	92	532	99.7
November	7	118	540	99.6

COMMUNICATIONS NETWORK

The U.K. communications Group moved away from Piccadilly to Buckingham Palace Road. Moving day was recorded for us by Tim Cureton.



Dismantling the tape drive, processor, and line printer in the Piccadilly Communications area.

MDS 2408 Processor leaving 118 Piccadilly.

LONDON MOVE (continued)

Buckingham Palace Road office with scaffolding erected to drop MDS gear down into the basement by block and tackle.

Communications staff looking on as processors are being prepared to be slung into the basement.

Processor being lowered into the basement.

As a result of the move, the numbers for local dial access in London are now:

Nodes 32 and 43	01-581-1411
Node 20 (Piccadilly)	01-499-0422

DYNAMIC LOT SIZE Andreas Buch, Curator AG, Zurich.

The DLS programs to handle production scheduling and purchasing (March-April 1977 newsletter) have now been complemented by a versatile data management system. The system is extremely easy to use, conversational prompts in both English and German are now available and error messages have been included. Using the DLS programs in production scheduling does not necessarily imply that the present organisation of a production plant must be changed. The main use of the DLS programs lies in master production scheduling which is only done once a week or month or whatever the planning period may be. The master schedule delivered by the DLS programs becomes input to the lower level production planning system in the production plant. The SHARP system seems ideal for this kind of application.

The DLS programs can be implemented by using a simple rolling system working in the following way:

1. Determine the length of the planning horizon, say January through June.
2. Run the respective DLS program for the six time periods January through June, entering the data as they appear in the beginning of January.
3. Take the action indicated for January.
4. Let the system roll to the beginning of February.
5. Run the respective DLS program for the six time periods February through July, entering the data as they appear in the beginning of February.
6. Take the action indicated for February.

   and so on.

The user can determine the level of stock needed in order to secure delivery even when demands are unpredictable. A tailor-made formula that will calculate the safety stock level as a function of past errors in demand forecasting can be built into the system.

Another important application of the DLS programs is its use in sensitivity analysis. By varying the input data, any relevant what-if question can be answered. For example, cost sensitivity and the associated production decisions that have to be taken can be determined by the multi-item production planning programs:

1. Reduction or expansion of capacities. How should one evaluate capacity?
2. Removal or addition of items on a production line. Which items block an assembly line?
3. How does
   a) another delivery plan
   b) another cost development
   c) other production schedules
   d) another planning horizon

   influence the production decisions of today?

By being able to trace out future shortages and/or surpluses, the DLS programs can play an important role in capacity planning. A new version of DLS2 to be installed in the near future will be able to consider any number of capacity bottle-necks. The DLS programs can be tailored according to a user's specific needs. Only if he has an idea of a specific cost development, should he let the respective cost elements vary with time. In this case the DLS programs allow for sophisticated planning. Rough planning is possible if the costs are only vaguely known.

Finally, just to give an idea of the efficiency of the DLS methods, the cpu-cost of a calculation involving a facility with 100 items over a planning horizon of 6 time periods amounts to no more than $25.00. This low cost will hopefully encourage users to do sensitivity analysis.

For further information please contact Sharp or: Andreas Buch,
Curator AG, Auditing & Trust Co.,
Freigutstr. 27,
8039 Zurich.

GRAFPLOT DEMONSTRATIONS

Dave King, Toronto

In the last newsletter, I neglected to mention how to use the demonstrations in workspace 3 *GRAFPLOT*. Here we go...

Since graphics output is terminal dependent, you must define the function *OUTTERM* before attempting to use *GRAFPLOT*.

Type:

TERM 'code' where "code" is the code-name of the terminal you are using.

For example,

TERM 'AJ832' will define the *OUTTERM* needed for plotting on the Anderson Jacobsen 832.

Typing:

TERM '' will give you a list of codes for all terminals currently supported by Sharp graphics software.

There are two "demo" functions for GRAFPLOT, *DEMO* and *SEEDEMO*. *DEMO* is for use only on the Tektronix 4010 series terminals. Simply type:

DEMO and select the desired demos from the menu with cross-hairs.

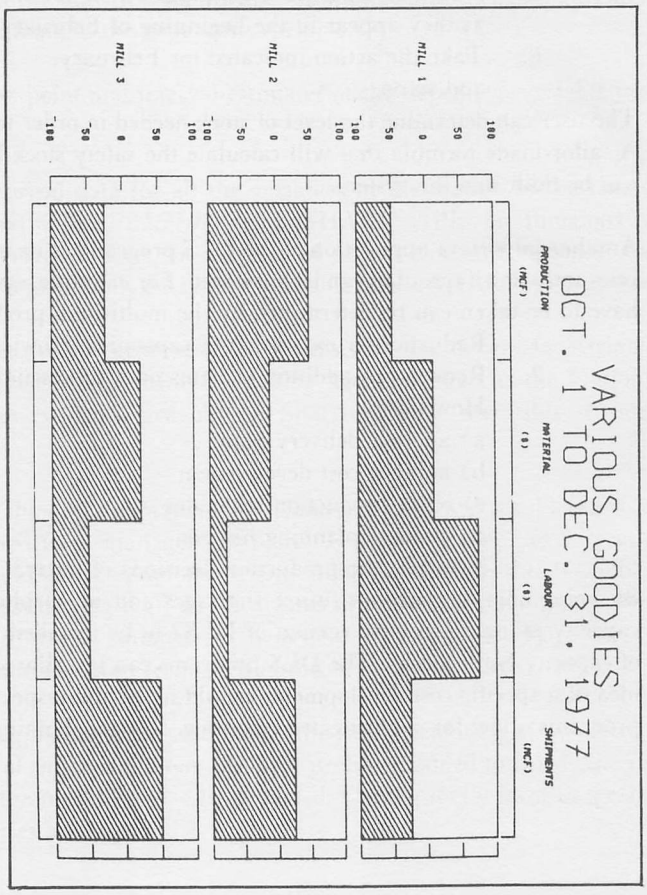
Users of other terminals should type:

SEEDEMO 'demo' where "demo" is one of the demonstration-names listed in the workspace. Typing *SEEDEMO* '' will print the list.

Type:

ERASE before each plot to avoid apparently random carriage movements.

TERM 'TEK'
SEEDEMO 'GOODIES'



SHARP APL COURSES

Introduction to APL:	DEC	JAN	FEB	MAR	APR	MAY
CALGARY (5-day)		10,11,18,25→Feb. 1				
MINNEAPOLIS	06-08		07-09	14-16		
MINNEAPOLIS	13-15		14-16	21-23		
OTTAWA	05-09	09-13	06-10	06-10	03-07	
ROCHESTER		23-27	20-24	20-24	17-21	
SEATTLE			14-16			
TORONTO	12-14	03-05	13-15	06-08	17-19	08-10
TORONTO		23-25		27-29		
U.K. (LONDON)		09-11	13-15	13-15	10-12	15-17
VANCOUVER		11-13	08-10			
WINNIPEG		23-26	20-23			

Intermediate

MINNEAPOLIS (6 half-days over 2 wks)	Jan. 10-12,17-19
SEATTLE	On demand
TORONTO	March 13,14,15
WINNIPEG	On demand

Advanced U.K.	'APL System Design'	Jan. 12-13	Mar. 9-10

Seminars

CALGARY:	'Designing SHARP APL Data Bases'	Jan. 17
EDMONTON:	'Good APL Programming Techniques'	Jan. 19
	'Regression Analysis in APL'	Feb. 17
MINNEAPOLIS:	'APL in Financial Modelling'	On demand
	'What is APL?'	On demand
SEATTLE:	'MAGIC'	Jan. 16-17
TORONTO:	'MAGIC for Time Series Analysis'	Feb. 7
	'Compound Interest Theory'	Jan. 10
	'Mathematics of Mortgages and Bonds'	April 3
	'Saving Money with NTASKS & BTASKS'	Jan. 30
	'AIDS' (3-day)	Feb. 21
U.K. (LONDON):	'MAGIC for Time Series Analysis'	On demand
	'Appreciaton of APL'	Feb. 2
	'Introduction to the SHARP APL system'	On demand
VANCOUVER:	'APL User Seminars'	Monthly.

Special User Seminar for the Aviation Industry

LOS ANGELES (one day) February 7, 1978

The theme for the one-day seminar, hosted by the Newport Beach office, is:

Application of the Sharp Aviation Data Bases.

David Keith, the author of the Sharp Aviation Newsletter, will be presenting one of the papers. We hope that interested users will come. Further information is available from the office at Newport Beach, or from your local Sharp representative.

>PET< PA03630-Y660

READERS' CORNER

The texts published by APL PRESS are now available from our Toronto office, for those users who wish to order them from us together with Sharp literature. They are also available directly from

APL PRESS,
Box 378, Pleasantville, N.Y. 10570.

The order form can be obtained from APL PRESS - as well as the APL VADE MECUM, APLSV summarized on a wallet-sized plastic card. The summaries below have been reprinted (by permission) from the APL PRESS order form.

Mathematics

A coherent three-volume sequence in the central topics of elementary mathematics, bridging the years from high school through the first year of college, is provided by:

ALGEBRA, AN ALGORITHMIC TREATMENT	K.E. Iverson	361pp	$9.35
ELEMENTARY ANALYSIS	K.E. Iverson	286pp	$6.25
CALCULUS IN A NEW KEY	D.L. Orth	286pp	$8.00
SOLUTIONS (for the ALGEBRA text)		42pp	$1.50

APL is used as the exclusive mathematical notation throughout - in exposition, proofs, and in exercises. This notation is very close to conventional algebra but is simpler and more general, and is directly executable on an APL computer exactly as written. Although an APL computer is in no way essential, it is a useful adjunct, removing the tedium from some of the exercises and encouraging an experimental approach to the exploration of mathematical topics.

The simplicity and precision of the notation makes possible an algorithmic treatment - every function introduced is defined in terms of at least one explicit construction. Even if a computer is not available, the algorithmic treatment presents the essentials of computer programming in a mathematical light, i.e., as the precise definition and application of functions.

The simplicity and precision of the notation have also made it possible to place a greater burden than normal on the exercises. This is reflected in their number and variety, and in the fact that interesting and non-trivial results are developed in them. The extensive exercises make it possible to foster a great deal of independent work on the part of the students. Finally, the exercises can be used effectively in courses devoted to teaching the APL language itself.

Because the notation applies to vectors and matrices (or **lists** and **tables**, as they are called in the algebra text) in a simple and uniform manner, tables are used extensively to give a graphic view of functions by displaying the patterns produced by applying them to vectors. They are also used to clarify topics which use vectors directly, such as linear functions and polynomials.

The notation provides simple explicit means for function definition, and the notation of **function** is emphasised in all three texts. The **Elementary Analysis** and **Calculus** texts employ a particularly simple and effective form (that covers recursive definition) which is discussed in the first issue of **APL News**, available from APL PRESS on request.

A 42-page solutions book for the algebra text is available; others are in preparation.

>PET< PA03630-Y660

Introductions to APL

INTRODUCING APL TO TEACHERS K.E. Iverson 25pp $0.75

In introducing the use of a computer to teachers it is desirable to start as soon as possible with material they can see is relevant to their topic and their students, and to avoid digressions concerning the computer and computer language. This book presents such an introduction to APL for teachers of high school mathematics. Much of this material should also be suitable for teachers of other topics at other levels, although they would also benefit from auxiliary material specifically addressed to the topic of interest.

AN INTRODUCTION TO APL FOR SCIENTISTS AND ENGINEERS K.E. Iverson 26pp $0.75

This is an introduction to APL addressed to the scientist or engineer and designed to exploit any previous acquaintance with the very similar notation of vector algebra. A careful study of these pages should bring the reader to the point where he can begin to make serious use of APL in some topic of interest to him. The use of an APL terminal in this study, while not absolutely essential, adds greatly to the depth and interest of the work. The pleasure and efficiency of learning by experimentation is not sufficiently appreciated, and the first six pages are designed to encourage this type of use of a terminal in learning APL. Two pages are devoted to a variety of identities and proofs expressed in APL.

APL IN EXPOSITION K.E. Iverson 61pp $1.00

This book illustrates the use of APL for exposition in the teaching of various topics. The first section presents the characteristics of the language, and each of the other sections illustrates its use in some discipline. The topics treated are: Elementary Algebra; Coordinate Geometry and Statics; Finite Differences and the Calculus; Logic; Sets; Electric Circuits; The Computer.

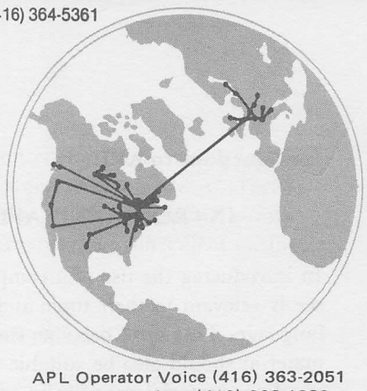
A new book (McGraw Hill) that follows the example of Iverson's **An Introduction to APL for Scientists and Engineers**, is specifically designed for teaching circuit theory at the freshman level:

RESISTIVE CIRCUIT THEORY Robert Spence 279pp $12.00

"One significant aspect of the text which derives from the use of APL, is its algorithmic and experimental flavour. At each stage in his understanding of the theory, the student is able to obtain experience of the pertinent relationships, both by executing the expressions appropriate to specific circuit examples, and by incorporating the expressions in defined functions which are then available as building blocks in further analyses. Moreover, the function definitions serve to summarize, in a precise and readable symbolic form, important relations which the student has just learned and which he will later wish to use as tools."

>PET< PA03630-Y660

I.P. Sharp Associates **Head Office:** 145 King Street West, Toronto, Canada M5H 1J8 (416) 364-5361



APL Operator Voice (416) 363-2051
Communications (416) 363-1832

SHARP APL Local Access In:

Canada
Calgary
Edmonton
Halifax
Hamilton
Kitchener
London
Montreal
Ottawa
Quebec City
Regina
Saskatoon
Toronto
Vancouver
Victoria
Winnipeg

U.S.A.
Ann Arbor
Boston
Buffalo
Chicago
Dallas
Dayton
Des Moines
Greene, N.Y.
Houston
Los Angeles
Minneapolis
Newport Beach
New York City
Raleigh, N.C.
Rochester, N.Y.
San Francisco
Seattle
Stamford, Ct.
Syracuse, N.Y.
Washington, D.C.
White Plains

Europe
Amsterdam
Birmingham
Brussels
Copenhagen
Coventry
Düsseldorf
Gloucester
Liverpool
London
Manchester
Milan
Paris
Stockholm
Vienna
Zurich

INTERNATIONAL BRANCH OFFICES

Amsterdam
Intersystems B.V.
Herengracht 244
1016 BT Amsterdam
The Netherlands
(020) 24 40 50

Birmingham
I.P. Sharp Associates Limited
2nd Floor, Radio House
79/81 Aston Rd. North
Birmingham B6 4BX
England
021-359-6964

Boston
I.P. Sharp Associates, Inc.
Suite 812
148 State St.
Boston, Mass. 02109
(617) 523-2506

Brussels
I.P. Sharp Europe S.A.
Ave. General de Gaulle, 39
1050 Bruxelles, Belgium
(02) 649 99 77

Calgary
I.P. Sharp Associates Limited
Suite 2660, Scotia Centre
700 - 2nd St. S.W.
Calgary, Alberta T2P 2W2
(403) 265-7730

Chicago
I.P. Sharp Associates, Inc.
2 North Riverside Plaza
Room 1746
Chicago, Illinois 60606
(312) 648-1730

Copenhagen
I.P. Sharp ApS
Østergade 24B
1100 Copenhagen K
Denmark
(01) 112 434

Dallas
I.P. Sharp Associates, Inc.
Suite 1148, Campbell Center
8350 Northcentral Expressway
Dallas, Texas 75206
(214) 369-1131

Düsseldorf
I.P. Sharp GmbH
Leostrasse 62A
4000 Düsseldorf 11
West Germany
(0211) 57 50 16

Edmonton
I.P. Sharp Associates Limited
Suite 505, 10065 Jasper Ave.
Edmonton, Alberta T5J 3B1
(403) 428-6744

Gloucester
I.P. Sharp Associates Limited
29 Northgate St.
Gloucester, England
0452 28106

Houston
I.P. Sharp Associates, Inc.
Suite 405, One Corporate Square
2600 Southwest Freeway
Houston, Texas 77098
(713) 526-5275

London, Canada
I.P. Sharp Associates Limited
Suite 510, 220 Dundas St.
London, Ontario N6A 1H3
(519) 434-2426

London, England
I.P. Sharp Associates Limited
132 Buckingham Palace Rd.
London SW1W 9SA
England
(01) 730-0361

Miami Lakes
I.P. Sharp Associates, Inc.
Suite D, Kennedy Bldg.
14560 N.W. 60th Avenue
Miami Lakes, Florida 33014
(305) 556-0577

Minneapolis
I.P. Sharp Associates, Inc.
Suite 1371, 1 Appletree Square
Bloomington, Minn. 55420
(612) 854-3405

Montreal
I.P. Sharp Associates Limited
Suite 1610, 555 Dorchester Blvd. W.
Montreal, Quebec H2Z 1B1
(514) 866-4981

New York City
I.P. Sharp Associates, Inc.
Suite 250, East Mezz.
Pan Am Bldg.
New York, N.Y. 10017
(212) 986-3366

Newport Beach
I.P. Sharp Associates, Inc.
Suite 1135, 610 Newport Center Dr.
Newport Beach, Ca. 92660
(714) 644-5112

Ottawa
I.P. Sharp Associates Limited
Suite 600, 265 Carling Ave.
Ottawa, Ontario K1S 2E1
(613) 236-9942

Palo Alto
I.P. Sharp Associates, Inc.
Suite 201,220 California Ave.
Palo Alto, Ca. 94306
(415) 327-1700

Philadelphia
I.P. Sharp Associates, Inc.
Suite 407, 1420 Walnut Street
Philadelphia, PA. 19102
(215) 735-3327

Rochester
I.P. Sharp Associates, Inc.
Suite 1150, 183 Main St. E.
Rochester, N.Y. 14604
(716) 546-7270

San Francisco
I.P. Sharp Associates, Inc.
Suite C409, 900 North Point St.
San Francisco, Ca. 94109
(415) 673-4930

Seattle
I.P. Sharp Associates, Inc.
Suite 217, Executive Plaza East
12835 Bellevue-Redmond Rd.
Bellevue, Wa. 98005
(206) 453-1661

Stockholm
I.P. Sharp AB
Kungsgatan 65
S111 22 Stockholm, Sweden
(08) 21 10 19

Sydney
I.P. Sharp Associates (Pty.) Ltd.
c/o Barry Ferranti Associates Pty. Ltd.
16 Hunter St., P.O. Box H101
Australia Square
Sydney N.S.W., Australia 2000
960-2670

Toronto
I.P. Sharp Associates Limited
145 King Street West
Toronto, Ontario M5H 1J8
(416) 364-5361

Vancouver
I.P. Sharp Associates Limited
Suite 604, 1112 West Pender St.
Vancouver, B.C. V6E 2S1
(604) 682-7158

Vienna
I.P. Sharp GmbH
Rechte Wienzeile 5/II
3 Wien, Austria

Warrington
I.P. Sharp Associates Limited
48A Horsemarket Street
Warrington, Cheshire
England
(020) 55342

Washington
I.P. Sharp Associates, Inc.
Suite 307, 1730 K Street N.W.
Washington, D.C. 20006
(202) 293-2915

Winnipeg
I.P. Sharp Associates Limited
Suite 909, 213 Notre Dame Ave.
Winnipeg, Manitoba R3B 1N3
(204) 947-1241

Zürich
I.P. sharp A.G.
Badenerstrasse 141
8004 Zürich
Switzerland
241 52 42

TELENET AND TYMNET

Alabama
Birmingham

Arizona
Phoenix

Arkansas
Little Rock

California
Alhambra
El Segundo
L.A. Central & East
L.A. West & Valley
L.A. South & Long Beach
Mountain View
Oakland
Orange County
Oxnard
Riverside
Sacramento
San Clemente
San Carlos
San Diego
S.F. Peninsula
San Jose
Santa Ana
Santa Barbara
Santa Rosa

Colorado
Denver
Colorado Springs

Connecticut
Bridgeport
Danbury
Darien
Hartford
New Haven/Bridgeport
Waterbury

Delaware
Wilmington

Florida
Fort Lauderdale
Jacksonville
Miami
Orlando
Pensacola
St. Petersburg
Tampa

Georgia
Atlanta

Idaho
Boise

Illinois
Freeport
Rockford

Indiana
Fort Wayne
Indianapolis
South Bend

Iowa
Cedar Rapids
Iowa City

Kansas
Topeka
Wichita

Kentucky
Louisville

Louisiana
Baton Rouge
Lafayette
New Orleans

Maryland
Baltimore

Massachusetts
Cambridge

Michigan
Detroit
Jackson
Kalamazoo
Southfield

Missouri
Kansas City
St. Louis

Nevada
Carson City
Las Vegas

New Hampshire
Nashua

New Jersey
Englewood Cliffs
Moorestown
Princeton
Piscataway
Newark
Union
Wayne

New York
Albany
Corning
Hempstead
Long Island
Niagara Falls

North Carolina
Durham
Raleigh/Durham

Ohio
Akron
Cincinnati
Cleveland
Columbus
Toledo

Oklahoma
Oklahoma City
Tulsa

Oregon
Portland

Pennsylvania
Allentown
Erie
Harrisburgh
Philadelphia
Pittsburgh
Valley Forge
York

Rhode Island
Providence

South Carolina
Greenville

Tennessee
Memphis

Texas
Austin
Baytown
Beaumont
El Paso
Ft. Worth
Midland
Odessa
San Antonio

Utah
Salt Lake City

Virginia
Norfolk
Richmond

Wisconsin
Madison
Milwaukee
Oshkosh

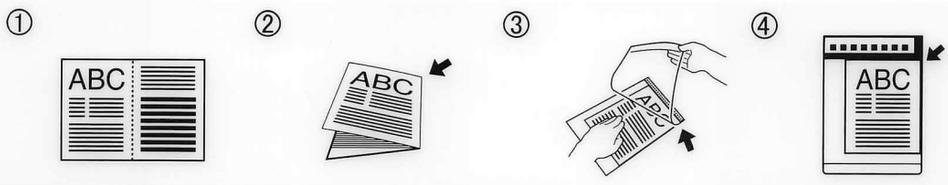
☐ Please amend my mailing address as indicated.
☐ Add to your mailing list the following name(s).
☐ Send me a SHARP APL publications order form.

Name: _____

Co.: _____

Address: _____

The Newsletter is a regular publication of I.P. Sharp Associates. Contributions and comments are welcome and should be addressed to: Jeanne Gershater, I.P. Sharp Newsletter, 145 King Street West, Toronto, Canada M5H 1J8.



>PET< PA03630-Y660