

VECTOR

APL NEWS, REVIEWS AND ARTICLES

- Switch onto external databases.
- Graphics – what's wrong with G.K.S.
- Learn how to operate with APL2.
- More steps to a better BASIC.
- APL86 in Manchester – preview.
- £50 competition.



*The Journal of the
British APL Association*

A Specialist Group of the British Computer Society

Vol.2 No.3 January 1986

Contributions

All contributions to VECTOR should be sent to the Editor at the address given on the inside back cover. Letters and articles are welcomed on any topic of interest to the APL community. These do not need to be limited to APL themes nor must they be supportive of the language. Articles should be submitted in duplicate and accompanied by as much visual material as possible, including a photograph of the author. Unless otherwise specified each item will be considered for publication as a personal statement by its author, who accepts legal responsibility that its publication is not restricted by copyright. Authors are requested wherever possible to supply copy in machine-readable form ideally text files on a 5¼" IBM-PC compatible diskette. For other standards, please contact the Editor beforehand. Program listings should indicate the computer system on which they have been run. APL symbols should be displayed on a separate line and not embedded in narrative. Except where indicated, items published in VECTOR may be freely reprinted with appropriate acknowledgement.

Membership Rates 1986-87

Category	Fee p.a.	VECTOR copies	Passes
Nonvoting student membership	£5	1	1
UK Private membership	£10	1	1
Overseas private membership	£18	1	1
Supplement for airmail (not needed for Europe)	£8		
Corporate membership	£85	10	5
Sustaining membership	£360	neg	5

The membership year runs from 1st May to 30th April. Applications for membership should be made on the form at the end of the journal. Passes are required for entry to some Association events and for voting at Annual General Meetings. Applications for student membership will be accepted on a recommendation from a course supervisor. Overseas membership rates cover VECTOR surface postage and must be paid in £UK.

Corporate membership is offered to organisations where APL is in professional use. Corporate members receive multiple copies of VECTOR and are offered group attendance of Association meetings. Partaking individuals need not be identified but a contact person should be nominated for all communications.

Sustaining membership is offered to companies trading in APL products; this is seen as a method of promoting the growth of APL interest and activity. As well as receiving public acknowledgement for their sponsorship, sustaining members receive bulk copies of VECTOR, and are offered news listings in the editorial section of the journal and opportunities to inform APL users of their products via seminars and articles.

Advertising

Advertisements in VECTOR should be submitted in typeset camera-ready A5 portrait format with a 20 mm blank border. Illustrations should be black-and-white photographs or line drawings. Rates are £250 per page. A6 and A7 sizes are offered subject to layout constraints.

Deadlines are:

Advertisement booking: Beginning of June, September, December & March
Camera-ready copy: 1 week later. Distribution: Following month later.

Advertisements should be booked with and sent to Steve Lyus, whose address is given beneath the Index of Advertisers.

CONTENTS

		Page
EDITORIAL: APL – USP or RIP?	David Preedy	3
APL NEWS		
Quick-reference diary	David Preedy	5
General Correspondence		7
British APL Association news	David Preedy	9
Education Officer's report	Dick Gray	11
Portsmouth Grammar School	Romilly Cocking	12
News from sustaining members	Steve Lyus	13
British Computer Society news	Dick Bowman	19
APL Press Review	Steve Lyus	21
APL Product Guide	Steve Lyus	23
Book reviews	David Preedy	33
APL86 PREVIEW		38
RECENT MEETINGS		
APL systems on Micros	Eileen Dyson	50
Developments in PEFAC	Chris Staffurth	51
Corporate Planning at Laura Ashley	Adam Dakin	54
An Integrated Manufacturing System on a Local Area Network	Ed Shaw junior	56
Migration from Mainframe to Micro	John Birch	57
APL in Practice		
– Commercial Applications	Bern King	60
A Sales Forecasting System	Patrick Rushton	60
APL in Use with a Large Commerical Database	Christine Brewster	61
APL Statistics Users Group	Jake Ansell	63
Operational Research Conference – 1985	Adrian Smith	65
GENERAL ARTICLES		
External Databases	Katie Williamson & Geoff Wells	79
Why GKS is unsuitable for business graphics	Martyn Adams	83
Steps to a better BASIC	Anthony Camacho	89
TECHNICAL SECTION		
Technical Editorial	Jonathan Barman & David Ziemann	93
Technical Correspondence	Sullivan, Buckland, Horton & Last	95
Competitions:		
“Wrap up” result	David Ziemann	101
“Test your skill” update	David Ziemann	104
“My other car's a Porsche”	David Ziemann	105
Surely there must be a better way	David Ziemann	107
Making effective use of GDDM and AP126	David Piper	109
A guide to using operators in APL2	Norman Thompson	118
Index to Advertisers		127
Membership Application Form		128

APL CONSULTANTS

LONDON & READING

Account Managers	(6 years+)	to	23K
Senior Consultants	(4-6 years)	to	19K
Consultants	(2-4 years)	to	16K
Junior Consultants	(1-2 years)	to	11K

Are your APL skills and potential being recognised and rewarded?

Cocking & Drury consultants have been implementing successful decision support applications for 10 years, with clients who appreciate the productivity benefits of APL.

In our professional team you will experience a range of APL environments - APL*Plus, VSAPL, APL2 and Unix, on both mainframes and micros. You will also be developing systems which, increasingly, need to interface with non-APL Information Centre products.

Of course as the leading APL consultancy, in a rapidly expanding market, we offer a rewarding career with first class benefits - profit sharing, free health insurance, and a non contributory pension.



For further details call Brian Drury on 01 493 6172

COCKING & DRURY LTD.
16 Berkeley St, London W1X 5AE

EDITORIAL: APL – USP or RIP.

by David Preedy

Most readers of this editorial will be APL professionals. You probably came across APL a few years ago, tried it on an application or two and found that you liked what you saw. Some of you may have been so excited by the power that it gave you, that you felt obliged to shout "Eureka" and tell your colleagues about the new panacea. Others may, like myself, have stood up in public places and exclaimed:

"FORTRAN is dead; long live APL!"

Why then, several years later, are people being awarded grants to develop computer-independent FORTRAN subroutines for graphics, when we were swapping APL graphics functions from IBM mainframes to 8-bit micros years ago? Why is APL the only computer language whose users can be (or even want to be) gathered once a year under one roof? And why are we continually being asked to defend our choice of APL by people who haven't taken the time to understand what APL can do?

Over the past year, several articles in VECTOR have discussed factors that have held back APL's progress. It cannot help to use an obscure character set that is only available on a small percentage of relatively expensive equipment. The APL community cannot hope to compete with the mass of publicity on BASIC that bombards computer beginners. APL will never be introduced to schoolchildren when no teachers have heard of it. Computer departments at colleges and universities cannot reasonably claim to offer a broad syllabus when none of their libraries subscribe to VECTOR.

However these explanations are merely symptoms of the major problem facing APL. The underlying failure of APL is a marketing failure, and its solution has to be a marketing solution.

Like any other product, APL has to force itself onto the market. We need to establish APL's Unique Selling Proposition – the key advantage that distinguishes APL from the mass of computer software now available. The strength of that USP will determine whether APL can become established as a mainstream computer language, or whether it will fade away as its successful applications are replaced by packages.

Only the APL community can define the USP and only we can initiate the required publicity campaign based on it. A powerful USP will force the major companies to offer APL as standard on their equipment at a sensible price. Without it the future of APL depends on the few people with the expertise and interest to develop APL interpreters for new systems.

Here then are some ideas that may spark the debate:

- APL is the only tool that allows engineers, mathematicians and scientists to define complex, new calculations precisely and effectively.
- APL is the language that lets you overcome the limits of your software packages.
- APL is the only language that I actually enjoy programming in.
- APL is the language for prototyping, for innovation and for developing new applications.

Other, more experienced APLers will have different and no doubt better ideas. The future availability of APL depends on our successfully pooling those ideas into a clear statement. Without its USP, APL will surely fade away and die.

VECTOR – new look

By now you will probably have noticed the new typeface used in this issue of VECTOR. We hope this will make the journal more legible, and welcome readers comments

Dates for future issues of VECTOR

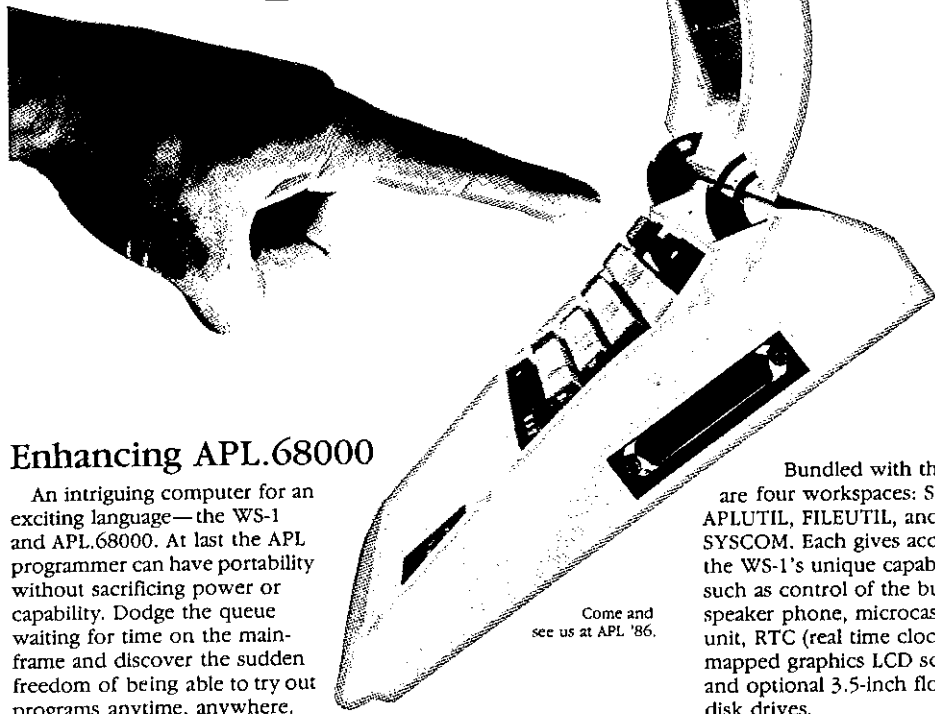
	Vol. 2 No. 4	Vol. 3 No. 1	Vol. 3 No. 2
Copy date	31 Jan 86	18 Apr 86	25 Jul 86
Ad. booking	7 Mar 86	16 May 86	22 Aug 86
Ad. copy	14 Mar 86	23 May 86	29 Aug 86
Distribution	April	July	October

Quick-Reference Diary

compiled by David Preedy

Date	Venue	Event
1986		
21 March	London	British APL Association meeting APL System Design
16 May	London	British APL Association meeting A.G.M. & Relational Database
1-3 July	Wembley	Knowledge Based Systems '86 Organized by Online Conferences
7-11 July	Manchester	APL86 – APL in Action
19 September	London	British APL Association meeting
17 October	London	British APL Association meeting
21 November	London	British APL Association meeting

POWER



Enhancing APL.68000

An intriguing computer for an exciting language—the WS-1 and APL.68000. At last the APL programmer can have portability without sacrificing power or capability. Dodge the queue waiting for time on the mainframe and discover the sudden freedom of being able to try out programs anytime, anywhere.

The APL.68000 interpreter is implemented in 86KB of ROM, running under a multi-user, multi-tasking operating system called BIG. DOS. Speed is the essence of APL programming, and now the WS-1 makes development even faster.

APL.68000 on the WS-1 has attractive enhancements such as a powerful component file system, QUAD. FMT function for alpha report formatting, QUAD. CC function for full-screen control, and extended error trapping facilities.

Come and see us at APL '86.

Bundled with the WS-1 are four workspaces: SYSFNS, APLUTIL, FILEUTIL, and SYSCOM. Each gives access to the WS-1's unique capabilities such as control of the built-in speaker phone, microcassette unit, RTC (real time clock), bit-mapped graphics LCD screen, and optional 3.5-inch floppy disk drives.

Compress these capabilities into a sleek footprint measuring less than 13 inches by 11 inches, and you have the ultimate definition of power.

ampère

FOR DISTRIBUTORSHIP INFORMATION AND PRODUCT DETAILS PLEASE CONTACT:

ampère
INCORPORATED

Asahi Bldg., 7-5-20 Nishi-Shinjuku, Shinjuku-ku, Tokyo, Japan. Phone: 03-365-0825.

Telefax: 03-365-0999. Telex: J33101 AMPERE. IP Sharp Mail Box Code AMP (Group Code APLWS).

General Correspondence

The VECTOR working group welcomes correspondence on any topic affecting the APL community. All such letters should be addressed to the Editor and should indicate whether they are intended for the general or the technical section. Letters containing APL code will normally appear in the Technical Section of VECTOR, and authors are asked to observe the requirements on the inclusion of APL code stated on the inside cover. The Editor reserves the right to edit any letter unless the writer states that it is to be published in full or not at all.

Classical APL

From Mr. Paul Barnetson

5th December 1985

Sir: In the October copy of VECTOR (which I did not actually receive until early December), you discuss some famous quotations relevant to interpreters. I have never used any of these myself, but I do list some of my more favourite quotations concerning apples. As you will see, I have taken certain liberties with the spelling of the word "apple", but the quotations are otherwise accurate. It is in this form that I have used them for various presentations, and your readers are more than welcome to do the same if they wish.

"An APL damned mankind" Thomas Otway, *The Orphan*, Act III, line 594

"He didn't want the APL for APL's sake, he wanted it only because it was forbidden" Mark Twain, *Pudd'nhead Wilson*

"The APL press'd with specious cant" Thomas Hood, *A Reflection*

"A goodly APL, but rotten at the heart" Shakespeare, *The Merchant of Venice*

"Stay me with flagons, comfort me with APLs, for I am sick of love"
Holy Bible, *The Song of Solomon*

Yours sincerely

Paul Barnetson,
Technical Support,
Academic Systems Marketing,
IBM United Kingdom Ltd.,
PO Box 41, Northern Road,
Portsmouth, Hampshire.

Editor: Those of you with timesharing problems may also sympathise with Mark Twain, clearly an APL fanatic, who wrote:

"There's plenty of boys that will come hankering and gruvvelling around when you've got an APL, and beg the core off you; but when they've got one and you beg for the core and remind them how you give them a core one time, they make a mouth at you and say thank you "most to death, but there ain't-a-going to be no core." Tom Sawyer Abroad

Danish APL Group

From Kim Andreasen

16th September 1985

Sir: Responding to a query from APL Quote Quad, the Danish APL group sent a brief description of the group and its activities to that magazine.

As readers of VECTOR might be interested in that information too, I enclose a copy, which I hope you can use in your publication.

Yours sincerely,

Kim S Andreasen,
Beta-plan Business Planning Systems ApS,
Stengade 75,
DK-3000 Helsingør,
Denmark.

Editor: More details of the Danish APL group will appear in the International APL News section in the next issue of VECTOR.

British APL Association News

Membership Rates for 1986/87

The British APL Association committee has fixed the subscription rates for next year. In order to maintain the quality of services, the committee has decided on a small increase in the basic rates.

The rates for 1986/87 will be as follows:

Pounds sterling:	U.K.	Overseas
Student members	5	n.a.
Private members	10	18
Corporate members	85	140
Libraries	55	85
Sustaining members	360	

The air mail supplement for overseas private members outside Europe stays at £8.

Members are reminded that subscriptions are due for payment at the start of the financial year (1st May). In future the Association has decided that it cannot afford the luxury of mailing out any issues of VECTOR to members whose subscription has not been received. So make sure you renew your membership with the Treasurer as early as possible.

Free membership

The Association has traditionally relied on word-of-mouth from existing members to attract new subscribers. In order to continue the recent increase in membership, it has been decided to give a year's free membership to any existing member who recruits three new members.

In order to qualify for this special offer, you must send the Treasurer your own membership renewal form together with application forms and payment from three individuals who have not been members of the Association. So hurry up and find your recruits before the start of May! This offer applies to U.K. private membership only. Overseas members who can generate three new subscribers will be entitled to a reduction of £10 on their subscription.

APL Publications

We regret to announce that APL Publications has ceased to trade. We would like to thank Les Hollingbery for running APL Publications, selling the Association's books over the last few years. He has now handed the stock to Mine of Information Ltd. who will supply all APL books.

The address for orders by post is:

APL Book Service,
P.O. Box 1000,
St. Albans,
Herts. AL3 6NE, U.K.
(Phone 0727-52801)

Telephone orders are accepted and you can quote you Access or Visa number and expiry date.

The Association hopes that this service will be well supported by all APL users, so that it can become an established part of the European APL community.

Education Officer's Report

by Dick Gray

Are you an educator?

Do you use computers as a teaching aid?

Do you wish you had APL available to you?

Now read on

Most users of APL agree that the language has not yet been accepted in the U.K. to anything like the extent of its promise, and this failure to develop is often ascribed to the lack of APL systems in the educational resources of the country. It is our intention to do something about this.

Although not much is heard about APL in education, it is in fact used very selectively and effectively in quite a number of our universities, colleges and schools, most notably the University of Swansea, Imperial College – London, and the Polytechnic College of North Staffs.

We have recently held a meeting of educators, who use APL, to discuss ways and means of extending its use in the classroom. One particularly significant attendance was the welcome presence of Mr. Roy Thornton, Director of Computing at Portsmouth Grammar School, where the fortunate pupils are being introduced to APL at an early stage.

Several useful suggestions were made at the meeting and as a result the British APL Association is hoping to put together a form of 'road-show', the various elements of which could be modified to suit a particular audience.

The idea would be to visit and make a presentation of APL to those persons or bodies who are responsible for influencing the purchase and use of computers in schools, universities or business colleges. This demonstration would illustrate the many advantages of using APL as a teaching medium, and the fact that the cost of doing so is now no greater than that of any other teaching language. The presentation would be timed to run for about 1½ to 2 hours.

If any educator wishes to invite the 'Road-Show' to 'perform' to their local computer advisory body or other influential committee, or is interested in any way in furthering the use of APL in our educational system, please contact – Dick Gray at:

Dick Gray Associates,
Horseshoe House,
Sproxtton,
Melton Mowbray,
Leics. LE14 4QB.
Tel: Grantham (0476) 860483

APL Computer for Portsmouth Grammar School

by Romilly Cocking

Portsmouth Grammar School were the deserving recipients of an APL micro-computer at a brief ceremony on November 15th 1985.

The computer (a TRS 80 running APL*Plus/80) was awarded by the committee of the British APL Association. Romilly Cocking handed it over on behalf of the association to Roy Thornton (Director of Computing at the school). Also present were two younger APL experts – Andy Tonks (currently a student at the School) and Mark Taylor (an old boy).

In a short presentation speech, Cocking emphasised the importance of nurturing educational interest in APL. "If the language is to continue its exponential growth, it is vital that we do everything possible to ensure a steady stream of graduates and school-leavers who are familiar with the language."

Portsmouth Grammar School were selected as recipients because of their prior involvement with APL. We look forward to hearing further news of their achievements.



News From Sustaining Members

Compiled by Steve Lyus

"Sustaining Membership" of the British APL Association is available to any company trading in APL products or services. It provides a tangible way for such companies to express their commitment to APL and to promote increased interest and activity in APL and in the Association.

As well as receiving public acknowledgement for their sponsorship of the Association, Sustaining Members receive bulk copies of VECTOR for distribution amongst their customers and are offered the opportunity to submit news material for this section of the Journal. They are also able to submit announcements for Association meetings (subject to the approval of the Activities Officer) and are invited to inform APL users of their products via seminars and articles.

The Committee of the British APL Association would like to acknowledge the generous support of the following organisations that have become Association Sustaining Members.

APL People	17 Barton Street, Bath, Avon. Tel. 0225-62601
APL Software Technology(UK) Ltd.	14 Rosewood Avenue, Alveston, Bristol, BS12 2PP Tel. 0454-415737
Cocking & Drury Ltd.	16 Berkeley Street, London, W1X 5AE Tel. 01-493 6172
Dyadic Systems Ltd.	Park House, The High Street, Alton, Hampshire. Tel. 0420-87024
Inner Product Ltd.	Eagle House, 73 Clapham Common South Side London, SW4 9DG. Tel. 01-673 3354
MetaTechnics Systems Ltd.	Unit 216, 62 Tritton Road, London, SE21 8DE Tel. 01-670 7959
MicroAPL Ltd.	Unit 1F, Nine Elms Industrial Estate, 87 Kirtling Street, London, SW8 5BP. Tel. 01-622 0395
I.P.Sharp Associates	10 Dean Farrar Street, London, SW1. Tel. 01-222 7033

APL People Limited

As part of our expansion programme, APL People has taken over Walton Technical Systems Limited, known for their computerised estimating system PEFAC. PEFAC is essentially a multifunctional tool for engineers, which runs on the IBM PC and costs around £4000. It allows an engineer to make fast, consistent estimates of the time taken to manufacture any item, with options for all the usual engineering operations – drilling, cutting, grinding, milling etc., and for a vast range of differing sizes and types of engineering tool – in fact many consider it an ‘expert’ system, having more knowledge built into it than any single industrial engineer. Those who have been with the APL Association for many years will recall Herbert Walton, author of PEFAC, who was Chairman of the user group for two years soon after it started and who can take credit for getting the infant user group organised into a coherent Society with a well-planned annual programme.

APL Software Technology (UK) Limited

1985 has been an exciting year, with a healthy expansion of their consultancy services within the Insurance and Banking fields. To cater for this expansion they are now actively looking for new London premises and new staff to resource many interesting projects coming up.

Cocking and Drury are now Dealers for the *Powertools/PC* product, and work is progressing well on the next release. This product is generating a lot of interest with many enquiries about production run time versions, which are available.

Torus Systems have appointed APL Software Technology as dealers for the Icon and Tapestry networking systems and they are pleased to have this product as part of their armoury.

Another exciting development has been the joining of forces with the cream of the Swedish APL community in the forming of a Sales and Marketing organisation in Scandinavia under the name Metalogic AB. This opens up a whole new market place for them, one that has a large APL usage.

Cocking & Drury Ltd

Demand for Cocking & Drury’s new in-house course, ‘APL2 IN DEPTH’, continues to grow rapidly, and at time of writing (December), bookings are being taken for February 1986. As well as being given to IBM, schedules already show demand across Europe. The course is primarily for experienced APLers. A complementary course, ‘INTRODUCTION TO APL2’, is also now available for people getting their first insight of APL through APL2. Both courses focus on commercial use of the language and modular design techniques.

As STSC's U.K. distributor, Cocking & Drury have launched Release 5 of APL*PLUS. PC. New features include a numeric editor, support for IBM's Enhanced Graphics Adapter, fast assembler routines, some APL2 primitives and window support. STATGRAPHICS, the comprehensive statistical and graphical package has been extremely favourably reviewed in the November issue of PC USER, and the rapidly increasing sales reflect its success. A series of half-day technical forums on STATGRAPHICS has been scheduled in London and Manchester over the next three months.

On the mainframe side, SHAREFILE/AP has been commercially launched. It offers STSC's familiar sharefile system under APL2. The first compiler for VS APL code is nearing the end of its beta test period and should be available for in-house purchase shortly.

Dyadic Systems Ltd

Dyadic has announced the appointment of VBB as the exclusive distributor of Dyalog APL in Sweden. VBB will also handle sales and support in Denmark and Norway. Interested parties should contact Christer Ulfhielm at VBB in Stockholm on 8-782-7000.

Dyadic has recently completed the 'port' of Dyalog APL onto Apollo Domain computers. The new implementation includes an APL font and full window support, and runs on the entire range of Apollo workstations including the 68020-based DN330.

Dyadic has begun work on APL versions of Lynwood's new j-Series terminals, and intends to produce a new breed of VDU designed specifically for the APL user. The Lynwood j300-APL VDU will be based on the ANSI 3.64 standard and will include an APL Session Manager and Full-Screen Editor, together with sophisticated graphics capabilities.

MicroAPL Ltd

The past few months have brought a number of developments at MicroAPL. During October, the company received a substantial venture capital injection from MTI, a partnership formed by Morgan Grenfell, Prudential Assurance and PA Management Consultants. In recent months it has become very difficult indeed for computer companies to raise venture capital from organizations with such a pedigree, and the investment therefore represents a significant vote of confidence both in the company and in the future of APL.

The immediate fruit of MTI's participation was the purchase during November of APL.68000 from The Computer Company (TCC). This means that MicroAPL is now the world-wide owner of APL.68000, and the transaction puts MicroAPL in a strong position to exploit the new generation of small, 68000-based systems which are now available. These include the Apple Macintosh and the new Atari ST and Commodore Amiga machines, which are attracting a great deal of interest because of their extraordinary speed, excellent

graphics, and use of modern windowing, icon and pointing techniques. MicroAPL will also be promoting APL software on larger 68000-based systems such as the NCR Tower and the Wicat range. At the same time, MicroAPL has been continuing its tradition of innovation on the hardware side of its business. Its new 32-bit supermicro, based on the immensely powerful Motorola 68020 true 32-bit processor, will have been launched by the time this issue of VECTOR appears. The new machine, which slots in above the existing Spectrum range, is based on a very advanced architecture which allows a multi-processor configuration. The basic system – with a single 68020 processor – offers a performance improvement of about four times over Spectrum, but this is only the beginning: extra processing power can be added not only by plugging in extra 68020 cpu cards each with up to 8 Mb of memory, but also by adding the new Motorola 68881 floating point processors. In this way, the system can build up to a configuration powerful enough to tackle almost any commercial application of APL – even those which until now had to stay on mainframes because of heavy processing demands.

I.P.Sharp Associates

I.P.Sharp has recently introduced an extended Telex link which enables Telex messages to be sent or received directly from the APL system. This has been used to provide the ability to send Telex messages directly from IPSA's MAILBOX. Incoming Telexes can be routed direct into MAILBOX if the appropriate MAILBOX codes are included as part of the Telex message. They are continuing to use SHARP APL on the IBM PC XT/370 as a way of providing customised solutions often involving downloading of significant sections of code from their standard application products. The same implementation of APL available on both PC and mainframe has major advantages for application development.

As a part of I.P. Sharp's increasing involvement in the banking and financial business area, Robert Hayim has been appointed as European Manager for Banking and Investment Sales. Robert was previously Manager of Banking Products. The Stock Exchange is now using I.P.Sharp as the database provider to their new technical analysis service called 'TRADEPLAN'. This is a specialised micro that maintains local databases of price information by auto downloading from IPSA databases as required. It was launched at 'Computers in the City' in November and is attracting great interest among stock and commodity analysts. I.P.Sharp has added the EXSHARE database which offers price and other information on over 80,000 securities and bonds throughout the world. This complements its other databases, many of which take information directly from the markets. For example, a real time price is now available to users of MAGIC from the Toronto Stock Exchange.

SALES EXECUTIVE APL PRODUCTS

Mercia Software markets a range of APL products and services which include:-

- *STSC's APL*PLUS/PC*
- *Mainframe spreadsheets*
- *Financial Modelling software*
- *Statistics and Graphics systems*

We are seeking an individual with a sound knowledge of APL, who is either currently in the sales and marketing field, or who wishes to move into this area.

The position offers a competitive starting salary, with excellent prospects in a small company with growth potential.

This interesting and challenging position offers a company car and a performance related bonus scheme.

Please send your application to Barrie Webster at the address below, or call us on 021-359 5096.

MERCIA SOFTWARE LIMITED

Aston Science Park, Love Lane, Birmingham B7 4BJ.
Telephone: 021-359 5096

ANNOUNCING

dyalog APL

VERSION 3.0

The **HIGH PERFORMANCE APL** for UNIX* systems

Dyalog APL is FAST

The new version of Dyalog APL is at least twice as fast as before. The component file system and the performance of many primitive functions has been substantially enhanced, so that for some applications even greater improvement will be experienced. Our own benchmarks based on real applications indicate that there is now no faster APL available for supermicros.

Dyalog APL is FUNCTIONAL

Features include nested arrays, full-screen editor, full-screen data manager, component file system, error trapping, external variables, large dynamic workspace.

Dyalog APL is POWERFUL

Dyalog APL's interface to the UNIX* Operating System is unrivalled. Not only can you run any UNIX* program from within the workspace, but you can capture its results in an APL variable for subsequent processing. Furthermore, you can write functions in a compiled language such as C or Fortran.

Dyalog APL is RELIABLE and WELL ESTABLISHED

Dyalog APL is now implemented on over 20 different UNIX* systems and has already been installed in more than 50 companies in eight countries.

For further information contact

Sales Department
Dyadic Systems Limited
Park House, The High Street, Alton, Hampshire
Telephone Alton (0420) 87024 Telex 858811

*UNIX is a trademark of
A T & T Bell Laboratories

British Computer Society News

compiled by Dick Bowman

The intention of this section is to bring some of the activities and publications of the British Computer Society to the attention of British APL Association members who may not otherwise receive such information.

Further information about the BCS (including membership application) is available from The British Computer Society; 13 Mansfield Street; London W1M 0BP.

Publications:

Software Engineering Journal – a new journal, published jointly with the IEE, first issue due early in 1986, six issues a year at an annual subscription rate of £17 for BCS members (£47 for non-members, and £15 for students). ‘The Editors welcome papers’. Free sample copies will be available from IEE; PO Box 8; Southgate House; Stevenage; Herts SG1 1HQ.

In every industry, there has to be a leader

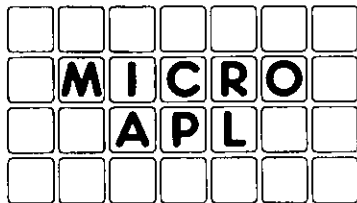
Personal APL – Our version of APL for the **Sinclair QL** brings full-powered APL at the lowest possible price.

PC-APL – MicroAPL supplies STSC's **APL*Plus/PC** with full backup, support, and ancillary software.

Departmental APL – For the ultimate in **performance and power**, our multi-user APL supermicros can handle even the largest applications.

Company-wide APL – With unrivalled experience in networking and communications, MicroAPL can offer a **comprehensive APL facility** linking PCs, supermicros and mainframes.

Consultancy in APL – MicroAPL's commitment to quality doesn't stop at our hardware products. Our uniquely experienced team of APL consultants will undertake any APL software project, from one day's help with a specific problem, to major team developments.



MicroAPL Limited

Unit 1F, Nine Elms Industrial Estate, 87 Kirtling Street, London SW8 5BP.
Telephone: 01-622 0395

APL Press Review

by Steve Lyus

This is the second review of recent articles in the Press that mentioned APL, or might be of interest to the APL community, and covers the period from September to November 1985. Please write and let us know if you find it useful and/or interesting. If anyone sees an interesting article in an unusual publication, the Editor would be pleased to collate the copy.

As we all know, APL does not get a very good coverage in the computer press, let alone publications of a more general nature. I was able to find a mention of APL in 11 articles. On the face of it this might not sound too bad, however over half of them only mention it as an example in a list (usually long, if it includes APL) of computing languages. Viz:

Brian Glassup in *Infomatics* (November) mentions it as one of the 'plethora' in his article 'Educating expert systems'.

PC Week (13th November) considers it a more specific (?) high level development language than, say, Fortran in 'Rivals for their own say'.

Boris Allan in *Datalink* (11th November) uses APL as the example of a 'functional programming language' in 'Setting out a style for languages of the future'.

Systems International (November) has an article on 'Micro Languages' by Tony Hetherington et al that includes it in a 'Babel'. The rest of the article does say a bit more about Fortran, Cobol and Pascal but then suddenly produces a 'suitability for application' table, with no explanation of how it was derived, of 14 languages by 15 so-called applications. Needless to say, APL is one of the also-rans (it does beat Basic and Cobol by one point, though). The most versatile languages, with almost double the number of points that APL scraped, were Ada and Modula-2 (?). I am afraid it greatly irritated me.

In *Datalink's* Main Feature (2nd September) Russell Jones reviewed an Inbucon survey on fourth generation languages ('Cure-alls they're not') and found that APL had been mentioned as one of the three most popular in the mini and mainframe market.

Finally, leaving the best until last, in another Main Feature in *Datalink* (11th November), Christine Guilfoyle wrote an article entitled 'No hope for Esperantists'. Although APL was initially described as 'academic', it was then one of four described in some detail (RPG, PL/1, Assembler being the others!). APL actually got 18 column-inches, one third devoted to the usual comments ('a scientific and mathematical cult language that gobbles up machine resources like nobody's business') produced by writers who have never bothered to actually try out the language. But, to counteract this view, two thirds was devoted to Romilly Cocking riding to the rescue, putting forward a far more constructive point of view (we even got a half page colour picture of him - at night). Well done Romilly!

A more natural mention of APL is made in the *IBM System User* in October where Sue Jelley has an article, 'Push the power back to the people', describing the development of the services provided by Information Centres.

The most heart-warming article in this collection of cuttings is that written by J.D. Matthews in the September *O.R. Newsletter*, entitled 'Keeping Pace with change using APL'. Did you know that knowledge of APL is a prerequisite for the Statistics course at Swansea University?

There are two articles, under the heading of New Product News, mentioning APL. *Computer Weekly* (28th November) announced that MicroAPL had bought the rights to the software code for APL68000 and were not only planning to introduce it on some of the new pcs but were also going to produce an upgrade in the APL2 direction. I have no doubt that the article in *Datalink* (11th November?), describing the APL compiler that Cocking and Drury will be distributing, raised a guffaw around the country! I always thought that the main benefit of compiled code was that it ran faster? It seems that this new compiler is different – it 'doubles execution time'! In the previous Press Review David Preedy mentioned the lack of coverage of APL85. I believe David Ziemann was responsible for providing the picture of APL users singing 'APL Blossom Time' that appeared in the GIGO column of *Datalink* (4th November). A few highlights of the lyrics were also given. Can anyone send us the full epic? We would be happy to publish it, we might even run a competition to expand it.

To end, I would like to draw the readers attention to 'Living C – Personal' written by Dick Pountain in *Byte* (November). My eye was drawn to the last column – 'Perhaps the most interesting application of Living C, though, is the maintenance of existing programs. C does not encourage the writing of readable programs, but now you can always understand how someone else's otherwise obscure code works by animating it and watching the bouncing ball'! It seems that you can display the source code on the screen and the cursor hops around pointing to the statement currently being executed, much in the fashion of those bouncing balls that traced out song lyrics for sing-alongs. Does any enterprising soul out there fancy emulating it for APL?

APL Product Guide

Compiled by Steve Lyus

VECTOR's exclusive APL Product Guide aims to provide readers with useful information about sources of APL hardware, software and services. We welcome any comments readers may have on its usefulness and any suggestions for improvements.

We do depend on the alacrity of suppliers to keep us informed about their products so that we can update the Guide for each issue of VECTOR. Any suppliers who find they are no longer included, have not bothered to reply to my two warnings. If they would like to be reinstated, would they please contact me to confirm their entries. Any suppliers who are not included in the Guide should contact me to get their free entry – see address below.

We reserve the right to edit material supplied for reasons of space or to ensure a fair market coverage.

The listings are not restricted to UK companies and international suppliers are welcome to take advantage of these pages. Where no UK distributor has yet been appointed, the vendor should indicate whether this is imminent or whether approaches for representation by existing companies are welcomed.

For convenience to readers, the product list has been divided into the following groups:

- Complete APL Systems (Hardware & Software)
- APL Timesharing Services
- APL Interpreters
- APL Visual Display Units
- APL character set printers
- APL-based packages
- APL Consultancy
- Other services
- Vendor addresses

Every effort has been made to avoid errors in these listings but no responsibility can be taken by the working group for mistakes or omissions. The Guide has now been entered on our Wordprocessor, so there might be some this time!

Note: 'poa' indicates 'price on application'

All contributions to the APL Product Guide should be sent to:

Steve Lyus
Group Management Services
Imperial Group plc
East Street
Bedminster
Bristol BS99 7JR

COMPLETE APL SYSTEMS

COMPANY	PRODUCT	PRICES £	DETAILS
Analogic	The APL Machine	\$60,000+	AP500 array processor, 4 Mb data memory, 80 Mb disk drive.
APL Software Technology	Ericsson PC	2,000 -5,000	Authorised Ericsson dealer supplying complete APL systems based on Ericsson PC - fully IBM compatible including hardware add-on boards.
Cocking/Drury	MicroAPL SPECTRUM SAGE II SAGE IV	6,000 -35,000	Supplied as part of a turnkey system. See MicroAPL entry.
Gen. Software	Myriade	poa	TI computer + APL & APL operating system
Inner Product	IBM PC	2,000 -6,000	IBM PCs supplied for turnkey applications
M.B.T.	MBT Series 10 TORCH	poa poa	UNIX/68010 based multi-user APL system 68000/Z80 multiprocessor
MetaTechnics	—	3,000+	Details on application
MicroAPL	Sinclair QL with QL/APL SPECTRUM STRIDE 440	300+ 11,000 -15,000 8,500	Fully expandable APL system with colour graphics. Expandable multi-user APL computer using Motorola 68000. Std. configuration 1 Mb RAM, 12/36 Mb disc, 12 ports. Multi-user APL computer, 1 Mb RAM, 12/18 Mb disc.

APL TIMESHARING SERVICES

COMPANY	PRODUCT	PRICES £	DETAILS
Boeing	Mainstream APL	poa	Enhanced IBM VS APL (CMS)
Mercia	APL*PLUS	poa	STSC's Mainframe Service - MAILBOX etc.
I.P. Sharp	SHARP APL	poa	International Network application systems and public databases.

APL INTERPRETERS

COMPANY	PRODUCT	PRICES £	DETAILS
APL Software Technology	APL*PLUS/PC	600 -695	See Cocking & Drury entry. Release 5 now available. Discounts available on multiple copies.
Cocking/Drury	APL*PLUS/PC Release 5	695	STSC's full-feature APL for IBM PC, PC/AT, Compaq, Olivetti, Wang, Apricot, Ericsson etc
	Release 4 update	150	Extension upgrade from release 4
	Release 3 update	110	Extension upgrade from release 3
	Run-time	poa	Closed version of APL*PLUS PC which prevents user exposure to APL.
	APL*PLUS UNX	poa	STSC's 2nd generation APL for IBM PC/AT, DEC, AT&T and other Unix computers.
Dyadic	Dyalog APL	1,000 -8,000	2nd gen. portable APL for UNIX systems, e.g. IBM PC AT, SUN, MC68000, HP9000, NCR, VAX, Gould, Amdahl.
Gen. Software	APL*MYRIADE	poa	Runs on Texas Instruments TI990 range.
IBM UK Product Sales	IBM PC APL	poa	Event-handling & AP's for full-screen I/O disks, diskettes, asynch. comms.
Inner Product	VIZ::APL	250 -350	8-bit Zilog Z-80 CP/M
	APL*PLUS/PC	600	See under Cocking & Drury
M.B.T.	Dyalog APL MBTAPL VIZ::APL	poa poa poa	See Dyadic Systems entry Enhanced Dyalog APL for MBT hardware. Customized for TORCH hardware
Mercia	APL*plus/PC Release 5.0 upgrade to 5.0 APL*PLUS/UNIX	695 150 poa	See under Cocking & Drury Interpreter for UNIX systems: WICAT, CADMUS, CALLAN, FORTUNE 32-16, HP, 9000/500, OLIVETTI 3B2
MetaTechnics	APL*PLUS Release 4	695	Discount on quantity.
MicroAPL	APL.68000 QL/APL (keyword) QL/APL (APL chars)	1,000+ 87 113	Full implementation with component files, error trapping etc. for SPECTRUM, SAGE & other MC68000-based computers. Full keyword APL for QL with many extra features. VSAPL compatible APL for QL with many extra features.
Portable	PortAPL	\$195 \$275 \$2,995	IBM PC Software Mackintosh DEC VAX
I.P. Sharp	Sharp APL/PCX Sharp APL/PC	2,575 1,000+ 325	For IBM XT/AT For IBM mainframes For IBM PC or PC/XT

APL VISUAL DISPLAY UNITS

COMPANY	PRODUCT	PRICES £	DETAILS
Farnell	Tandberg TDV2221	995	Ergonomic design APL terminal, 50-19200 baud, 15" anti-reflex screen, low profile keyboard
Gen. Software	Tandberg TDV2271	1,195	Combined APL/ANSI ergonomic terminal as above.
	Mellordata Elite 3045A	400	Second-hand
M.B.T.	various		Contact MBT for details
MicroAPL	Insight VDT-1	795	Inexpensive APL VDU
	Insight GDT-1	1,450	With monochrome graphics
	Tektronix 4105	4,150	High resolution colour graphics supporting APL character set on MicroAPL hardware.
Shandell	AVT-APL+ GVT-APL+	1,200+	ANSI 3,64; DEC compatible; full overstrike chars; 4/8 pages; 2/3 comms. ports; 80/132 cols; windowing; 12" screen; 46 PF keys; Tek 4013 graphics compatible.
	HDS 200	1,200+	As above plus 15" screen; viewports; smooth scroll; NVM storage; 55 PF keys; Tek 4014, VT640/DQ640 & Visual 500 graphics compatible.
Textronix	4114B	13,500+	19" D.V.S.T.: Graphics: 3120 x 4096 displayable; Intelligent: up to 800K memory; APL keyboard (option 4E)
	4125	21,550+	19" 2D colour graphics; Workstation (1280 x 1024); Intelligent: up to 800K memory; APL keyboard (mod AP)
	4128	26,822+	As 4125 plus 3D wireframe

APL PRINTERS

COMPANY	PRODUCT	PRICES £	DETAILS
Datatrade	Datasouth DS180+	1,295	180 cps matrix printer with 4K buffer, 9 x 7 dot matrix and APL option.
	Datasouth DS220	1,695	Letter quality; graphics capability, APL option (both available with IBM Twinex or Coax interface).
Inner Product	Epson FX80	500	Soft char. set, 160 cps, 80 column
	Anadex 9620	1,150	200 cps., 132 col., tractor feed
	Siemens PT88	620	180 cps., 80 col., silent
	TGC Starwriter	1,180	40 cps., letter quality
M.B.T.	Facit 4565	poa	40 cps letter-quality
	Facit 4510/11/12	poa	Matrix printers
MetaTechnics	Quen-data	295	Low-cost APL Daisy-wheel printer
MicroAPL	Datasouth DS180+	1,295	See Datatrade entry
	Philips GP300	1,798	Matrix printer with letter & draft quality and APL.

APL PACKAGES

COMPANY	PRODUCT	PRICES £	DETAILS	
APL ◊ 385	FSM 385	PC: 50	Screen development	
	DRAW 385		Screen design	
	DB 385	Mainframe: 125	Relational W.S.	
	GEN385		Utilities	
APL Software Technology	POWERTOOLS/PC	295	Assembler written replacement function for commonly used CPU-consuming APL functions. Includes extended full-screen Forms Processor.	
Boeing	TABAPL	poa	Hierarchical Planning System	
Cocking/Drury	<i>Mainframe</i>			
	STSC's SHAREFILE	poa	Component files, quad- functions & nested arrays for IBM VSAPL under VM/CMS & MVS/TSO	
	SHAREFILE AP	5,000	STSC's sharefile for APL2	
	EMMA	poa	STSC's database package.	
	CALL/AP	3,000	Non-APL program execution from VS-APL or APL2 under VM/CMS & MVS/TSO	
	AFM	10,000	High-performance shared files for VS-APL.	
	FORMAT	2,250	Enhanced report formatting.	
	WSAIDS	795	Workspace doc. & development aids.	
	<i>Microcomputer</i>			
	STATGRAPHICS PC	995	Statistics & Graphics for IBM PC, AT, etc.	
	APL*PLUS PC Tools			
		VOL 1	325	Incl. 327x IRMA support, RAM disk, full screen data entry, menu input, report generation, games.
		VOL 2	125	Incl. file documentor, screen editor, exception handler.
	APL*PLUS PC Fin & Stat. Library		350	Financial & statistical routines
	APL/DIF		145	Interface between APL*PLUS PC and DIF file supporting spreadsheets.
	MetaScreen		245	See MetaTechnics entry.
MetaPack		95	See MetaTechnics entry.	
POWERTOOLS		295	See APL Software Technology entry.	
E&S	PROTOPAK consisting of:		Packages for prototyping management information systems - PC & mainframe	
	RMS	Modules	Relational databases.	
	AMS		Multi-dimensional arrays	
	RAMS		Combined RMS & AMS.	
	BMS		Dynamic financial modelling & forecasting	
	FMS	Full-screen handler for APL*PLUS/PC. (AP 124-based)		
	CMS	Communications package.		
SOS	poa	Scheduled ordering and stock control.		
Gen. Software	PROPS	500+	Spreadsheet system for Product and/or Project Planning.	
H.M.W.	INPUT	poa	Matrix manipulation package for data entry & report generation	
	PRINTPAK	poa	Block printing for VM/CMS	
	VIEWPAK	poa	AP 124 Protocol emulator for IBM/PC	
Holtech	CASH	3,500	Accounting package & hotel management system on MicroAPL	
		-10,000	SPECTRUM & SAGE CPUs.	
Inner Product	Viewcom	150	Control Viewdata from APL	
	APL/dBASE II	150	Interface APL with dBase II	
	APL/dBASE III	150	Interface APL with dBASE III	
	APL/LOTUS	150	Interface APL with Lotus	
	APL/WORDSTAR	150	Interface APL with Wordstar	
	APL/MULTIPLAN	150	Interface APL with spreadsheet	
	CEMAS	3,500	EEC monetary and agrimonetary analysis.	

M.B.T.	RHOMBUS HASLEMERE	poa poa	Integrated Office System Hotel Accounting System
Mercia	STATGRAPHICS APL*PLUS tools	995	Integrated stat. graphic system for PCs.
		VOL 1 295	IBM PC Utilities: IRMA3270 comms, full screen, RAM Disk report generator
		VOL 2 95	File documentation, screen editing, Exception handling
	FINANCIAL AND STATISTICAL LIB. INFO CENTRE	2,000	Full-screen entry, display & multi-dimensional analysis. Interfaces to other I.C. products. Runs under VM VSAPL on IBM mainframes.
	EXECUCALC	-20,000 4,000	Mainframe Spreadsheet with VisiCalc and Lotus 1-2-3 functionality requires VSAPL under TSO or VM.
	EXECUPLOT	3,200	Mainframe Graphics display system with VisiPlot functionality requires VSAPL under TSO or VM and GDDM.
MetaTechnics	MetaScreen	245	Full-screen handler for APL*PLUS/PC, based on VSAPL AP124
	MetaPack	495	Comprehensive utilities package for APL*PLUS/PC. Includes MetaScreen, MetaWS, Browse, Toolbox, Numeric Editor
	ADAPTA DLS	poa	Production & purchasing scheduling for process manufacturing.
	ADAPTA MSP	poa	Job-shop loading & scheduling for multi-stage production.
MicroAPL	MicroTASK	250	Product development aids
	MicroFILE	250	File utilities and database
	MicroPLOT	250	Graphics for HP plotters etc
	MicroLNK	250	General device communications
	MicroEDIT	250	Full screen APL editor
	MicroFORM	250	Full screen forms design
	MicroSPAN	250	Comprehensive APL tutor
	MicroGRID	poa	Ethernet & other networking
	APLCALC	400	APL spreadsheet system
	MicroPLOT/PC	250	For APL*PLUS/PC product
MicroSPAN/PC	250	For APL*PLUS/PC product	
Parallax	ExecuCalc	\$5,000	Mainframe-based electronic spreadsheet for VM/CMS & MVS/TSO with links to micro products.
	ExecuPlot	\$5,000	Mainframe-based colour graphics with micro links.
I.P. Sharp	ACT	poa	Actuarial system
	APS	poa	Financial Modelling
	BOXJENKINS	poa	Forecasting technique
	CONSOL	poa	Financial Consolidation
	COURSE	poa	APL Instruction
	EASY	poa	Econometric Modelling
	FASTNET	poa	Project Management
	GLOBAL LIMITS	poa	Exposure management for banks
	MABRA	poa	Record maintenance/reporting
	MAGIC	poa	Time series analysis/reporting
	MAGICSTORE	poa	N-dimensional database system
	MAILBOX	poa	Electronic Mail
	MICROCOM	poa	Mainframe to micro link
	SAGA	poa	General graphics, most devices
	SIFT	poa	Forecasting system
	SNAP	poa	Project management
SUPERPLOT	poa	Business graphics	
VIEWPOINT	poa	4GL - Info centre product	
X TABS	poa	Survey Analysis	

APL CONSULTANCY

(prices quoted are per day unless otherwise marked)

COMPANY	PRODUCT	PRICES £	DETAILS
APL People	Consultancy	poa	All levels, most APL systems
APL Software Technology	Consultancy	poa	Technical & business systems, micros, networking & communications a speciality.
Boeing	Consultancy	poa	
Camacho	Consultancy	poa	Specialising in programming & manual writing.
Cocking/Drury	Consultancy	120-150 140-200 185-300 275-400	Junior consultant Consultant Senior consultant Managing consultant
	Courses	375 330 595 poa poa	3 day Fundamentals 3 day Advanced 5 day System Design Introduction to APL2 APL2 in depth
Delphi	Consultancy	poa	Specialising in management reporting systems and APL on microcomputers.
Dyadic	Consultancy	poa	APL system design, consultancy, programming & training for Dyalog APL, VSAPL, APL*PLUS, IPSA APL etc.
E & S	Consultancy	150 -250	System prototyping: all types of information system.
Gen. Software	Consultancy	100+	
H.M.W.	Consultancy	100-250	System design consultancy, programming.
InnerProduct	Consultancy & Training	200	On-site micro-mainframe APL, PC/DOS & Assembler
M.B.T.	Consultancy & Courses	poa	
Mercia	Consultancy & Training	poa	APL*PLUS & VSAPL consultancy. Onsite & Offsite APL courses.
MetaTechnics	Consultancy	poa	Management Information & Production. Management Systems & Process Control.
	MetaSys	poa	Prototyping APL-Hybrid Custom Programming Service.
MicroAPL	Consultancy	poa	Technical & applications consultancy.
Parallax	Consultancy & Training	\$750 poa	Introductory APL, APL for End-user & Advanced Topics in APL
QB On-Line	Consultancy	200	Specialising in Banking, Financial & Planning Systems.
I.P. Sharp	Consultancy	poa	Consultancy & support service world-wide.

OTHER PRODUCTS

COMPANY	PRODUCT	PRICES £	DETAILS
APL People	Employment Agency	poa	Permanent employees placed at all levels. Contractors supplied for short/long-term projects, supervised.
Mine of Information	APL Book Service		See booklist
I.P. Sharp	Productivity Tools	poa	Utilities for systems, operations, administration & analysts; auxiliary processors, comms software, international network.
	Databases	poa	Financial, aviation, energy and socioeconomic.

VENDOR ADDRESSES

COMPANY	CONTACT	ADDRESS & TELEPHONE No.
Analogic Corporation	Denise Favorat	8 Centennial Drive, Centennial Industrial Park, Peabody, Mass. U.S.A. 01961 ☎ 617-246-0300
APL 385	Adrian Smith	Brook House, Gilling East, York. ☎ 04393-385
APL People	Valerie Lusmore	17 Barton Street, Bath, Avon. ☎ 0225-62601
APL Software Technology	John Hagger or Per Hultin	14 Rosewood Avenue, (UK) Ltd. Alveston, Bristol BS12 2PP. ☎ 0454-415737 46 Vicarage Hill South Benfleet, Essex SS7 1PB. ☎ 03745-50501
Boeing Computer	Anne Harding	19 Fitzroy Street, Services (Europe) Ltd., London W1P 5AB. ☎ 01-631 0808
Anthony Camacho		2 Blenheim Road, St. Albans, Herts AL1 4NR. ☎ St. Albans 60130
Cocking & Drury Ltd.	Romilly Cocking Brian Drury	16 Berkeley Street, London W1X 5AE. ☎ 01-493 6172 155 Friar Street Reading RG1 1HE. ☎ 0734-588835
Datatrade Ltd.	Tony Checksfield	38 Billing Road, Northampton, NN1 5DQ. ☎ 0604-22289
Delphi Consultation Ltd.	David Crossley	Church Green House, Stanford-in-the-Vale, Oxon SN7 8LQ. ☎ 03677-384
Dyadic Systems Ltd.	Peter Donnelly	Park House, The High Street, Alton, Hampshire. ☎ 0420-87024
E & S Associates	Frank Evans	19 Homesdale Road, Orpington, Kent BR5 1JS. ☎ 0689-24741
Farnell International Instruments Ltd.	R. Fairbairn or Roger Attard	Jubilee House, Sandbeck Way, Wetherby, W. Yorks. ☎ 0937-61961 Davenport House, Bowers Way, Harpenden, Herts. ☎ 05827-69071
General Software Ltd.	M.E. Martin	22 Russell Road, Northolt, Middx. UB5 4QS. ☎ 01-864 9537
H.M.W. Programming Consultants Ltd.	Ken Jackson	142 Feltham Hill Rd, Ashford, Middx. TW15 1HN. ☎ 07842-41232
Holtech Ltd.	Jan Bateman	'O' Block 4th Floor, Metropolitan Wharf, Wapping Wall, London E1 9SS. ☎ 01-481 3207
IBM UK Ltd	Chris Sell	PO Box 32, Alencon Link, Basingstoke, Hants. RG21 1EJ. ☎ 0256-56144
Inner Product Ltd.	Dominic Murphy	Eagle House, 73 Clapham Common Southside, London SW4 9DG. ☎ 01-673 3354
Mercia Software Ltd.	Gareth Brentnall Barrie Webster	Aston Science Park, Love Lane, Birmingham B7 4BJ. ☎ 021-359 5096
MetaTechnics Systems Ltd	Steve Margolis	Unit 216, 62 Triton Road, London, SE21 8DE. ☎ 01-670 7959
MicroAPL Ltd.	Richard Nabavi Paul Thornton	Unit 1F, Nine Elms Industrial Estate, 87 Kirtling Street, London SW8 5BP. 01-622 0395
Mine of Information	Richard Ross-Langley	PO Box 1000, St. Albans, Herts AL3 6NE. ☎ 0727 52801
Modern Business Technology Ltd. (MBT)	Michael Branson	P.O. Box 87, Guildford, Surrey GU4 8BB ☎ 04868-23956
Parallax Systems Inc.	Kevin Weaver	60 West 9th Street, New York, New York 10011, U.S.A. ☎ 212-475-4001
Portable Software	Richard Smith	60 Aberdeen Ave, Cambridge, Mass. U.S.A. 02138. ☎ 617-547-2918
QB On-Line Systems	Phillip Bulmer	5 Surrey House, Portsmouth Rd Camberley, Surrey, GU15 1LB. ☎ 0276-20789
Shandell Systems Ltd.	Maurice Shanahan	12 High Street, Chalfont St. Giles, Bucks HP8 4QA. ☎ 02407-2027
I.P. Sharp Associates Ltd.	David Weatherby	10 Dean Farrar Street, London SW1. ☎ 01-222 7033
Tektronix UK Ltd.	Paul Morgan	Fourth Avenue, Globe Park, Marlow, Bucks SL7 1YD. ☎ 06284-6000

APL BOOKLIST

List of books stocked and on sale

Title	Price	UK Post & Pack (£)
An Interactive Approach, Gilman & Rose	24.35	2.65
APL: An Introduction, H. Peele	15.00	2.25
Introducing APL to teachers, K. Iverson	3.00	.40
Introduction to APL for Scientists & Engineers	3.00	.40
An APL Notebook, Barrie Wetherill	1.90	.50
A Source Book in APL, A. Falkoff & K. Iverson	8.75	1.80
APL: Design Handbook for Commercial Systems, Smith	13.10	1.50
APL and Insight, P. Berry and G. Bartoli	4.50	.55
APL in Practice, STSC	40.00	2.50
APL The Language and its Usage, R. Polivka & Skin	36.00	2.70
APL Quote-quad the Early Years	31.50	2.70
APL Business Technology '83 Proceedings	11.20	2.70
APL SV Reference Card	.50	0.50
Designing APL Systems Toronto Procs 82 Vol 2	8.50	1.80
FinnAPL Idiom Library	11.20	1.30
Sharp APL Reference Book	1.50	.50
Sharp APL Reference Manual, P. Berry	10.50	2.70
Yale University Idiom list	2.00	.50
APL in Exposition, K. Iverson	3.00	.50
Algebra, K. Iverson	12.00	2.30
Solutions to Algebra, J. Iverson	3.95	.50
Elementary Analysis, K. Iverson	13.50	1.50
Resistive Circuit Theory, R. Spence	20.00	2.80
Star Map, P. Berry & J. Thorstetsen	6.00	.50
Reliable Software Through Composite Design, Myers	15.00	1.30
APL Lapel Pin with gripping backplate	2.00	0.50
Whizzbangs Volume II, R. Sykes	10.00	1.80
Whizzbangs Volume I, R. Sykes	10.00	.40
Information Centres Toronto Procs 84	8.00	1.80
Application Systems in APL, Gibson Levine Metger	30.00	2.70

Please order direct from Mine of Information:

APL Book Service, P O Box 1000, St. Albans, AL3 6NE, U.K. ☎ 0727-52801

This list replaces all previous lists

Prices are in pounds sterling and are as at 1st January 1986

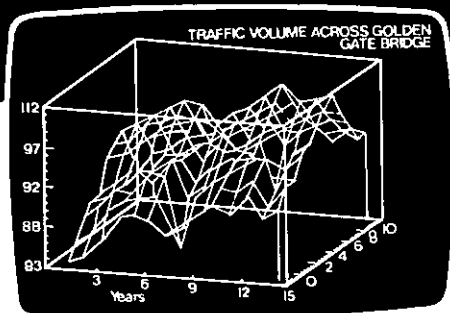
Terms: Access & Visa accepted; sterling cheque with order; invoices payable on receipt of books.

Outside UK add an extra amount per book

£1 to Europe £2.50 Africa & Middle East

£5 elsewhere (all sent airmail).

STATGRAPHICS™ INTRODUCTORY FORUMS



STATGRAPHICS, the interactive PC-based system for Statistical and graphical data analysis.

Written in APL, it supports all major PC hardware and interfaces to ASCII files, DIF™ files and spreadsheet software.

STATGRAPHICS offers over 340 Statistical and graphics routines for analysing commercial, financial, technical and scientific data.

To obtain a more detailed insight into STATGRAPHICS, why not attend a half-day introductory forum at one of the following venues:

25th March	LONDON	15th May	EDINBURGH
		19th May	LONDON
17th April	MANCHESTER	5th June	BIRMINGHAM
30th April	LONDON	19th June	LONDON

The cost is £40 per attendee. To reserve your place now, contact Liz Swann at Cocking & Drury on 01-493 6172.

Trademarks/Owners: Statgraphics/Statistical Graphics Corporation · DIF/Software Arts, Inc.



COCKING & DRURY LTD
THE APL PROFESSIONALS

16 BERKELEY STREET LONDON · W1X 5AE
Tel: (01)493 6172 · Tlx: 23152 MONREF G

Book Reviews

by David Preedy

Diagnosing the System for organisations
by Stafford Beer, John Wiley & Sons, ISBN 0 471 90675 1

Diagnosing the System is the latest in a series of books in which Professor Beer sets out to explain the theories of the managerial cybernetics of organization, i.e. what are the fundamental requirements of an organization's structure in order for it to be viable. The companion books in the series – *Brain of the Firm* and *The Heart of Enterprise* – are both large volumes which develop the subject in principally theoretical terms, explaining the rationale behind the model for a viable system and the basis for the various cybernetic laws underlying it. However the drawback with this type of theoretical approach is that it does not provide the reader with the right sort of understanding to go away and start using the approach. This book attempts to redress the balance by showing how to design a suitable organizational structure and how to diagnose a faulty one.

The first impression of the book – at least to a reader of its precursors – is its size, a mere 150 pages compared to the several hundred in Professor Beer's other works. The second impression is of a highly visual presentation of the subject material. The previous books present the viable system model in diagrammatic form, but *Diagnosing the System* takes the graphics a stage further by liberal use of red ink and expressive hand-drawn diagrams. The book looks as though the author has personally taken your copy and added chosen comments as an afterthought.

The style of the content is equally distinctive with continual emphasis on the practical. By page 6, under the bold red heading

“NOW DO THIS”

the reader is instructed to look at his personal position as a viable system, and on page 16 the tone of the rest of the book is set with a similar instruction to choose and define a specific system that the reader is to model throughout the rest of the book. As each stage of the theory unfolds revealed, so the exercises require the reader to relate it to his chosen system.

This strong emphasis on the practical is reinforced by a punchy, almost abrupt presentation. The style is more like a lecturer's notes than an academic treatise and shows evidence of its origins in the material for a post-graduate course in the cybernetics of organizations. Few paragraphs are longer than 5 or 6 lines; sections and subsections are not numbered, but are distinguished by colour, indentation, etc.; diagrams abound; and the jargon is kept to a minimum and is summarised, along with the major conclusions, at the end of each chapter.

This practical style enables the book to be much shorter than a corresponding purely theoretical work, but puts obligations on the reader, since the principal message comes from the application of the concepts to the reader's own chosen system. In this sense the reader can out take out of the book as much as he is willing to contribute.

The target audience for this book are those readers who face a real organizational problem, either as managers or their consultants, and who use the book as a handbook to diagnose and redesign their own systems. This will include people who have already read up the subject of the cybernetics of organizations (perhaps in the previous books in this series) as well as those who are new to it. However the book should also provide a valuable introduction for newcomers to the subject whose present interest is more conceptual than practical, but who find the practical style of this book easier than the more conventional approach of its predecessors.

References.

Brain of the Firm, S. Beer (1972), Allen Lane, Penguin, Harmondsworth.
The Heart of Enterprise, S. Beer (1979), Wiley, Chichester.

The Visual Display of Quantitative Information by Edward R Tufte Graphics Press, Box 430, Cheshire, Connecticut.

This book is not a recent issue (it first appeared in 1983), but is well worth a belated review because its valuable contribution to an increasingly important area deserves a wider readership than it seems to have achieved. It is one of the few books that provide the development of a theoretical approach to a subject in a book that is visually attractive and a pleasure to read.

The subject area is the effective presentation of data in a visual form. Whilst this is not specifically concerned with computer graphics, Tufte emphasises its importance is increased by the potential of computers to generate mediocre graphics.

“At least a few computer graphics only evoke the response ‘Isn’t it remarkable that the computer can be programmed to draw like that?’ instead of ‘My, what interesting data.’”

The book comprises two main sections, entitled *Graphical Practice* and *Theory of Data Graphics*. The first section starts with a chapter on Graphical Excellence, which contains some of the best examples of graphical displays of data from the last 200 years. (1986 is in fact the bicentenary of the publication of the first known time-series). My personal favourites illustrate the Paris-Lyon train schedules of 1880 and Napoleon's Russian campaign of 1812-13. It is a well-worn cliché that "A picture is worth a thousand words"; Tufte shows how a graph is worth 700 words by comparing one with a newspaper report exposing how US representatives exploit their privilege of free postage in the run-up to elections.

The next chapter (on Graphical Integrity) introduces the Lie-factor, i.e. the ratio of the size of an effect as shown by a graph to its size in the data, again illustrated by published examples mostly from an assortment of the world's best-known newspapers. Tufte makes the point that one lie in a newspaper is a lie repeated hundreds of thousands of times – a message that should be taken to heart by those designing chart layouts for use on computer systems. He also points out that the occurrence of even the well-known abuses of data is continuing to increase.

The first half of the book provides a very clear and enjoyable coverage of material that is, on the whole, available from other well-known books. It is when Tufte starts to develop his theory of graphic design that new ideas are expressed. There are many publications that offer rules for good graphical design, including some from computer graphics equipment suppliers and most of the advice that they give is good. However their recommendations are simply rules, and don't allow much scope for imagination; properly applied they will prevent the worst excesses of bad design, but they tend to suppress the innovation needed to produce excellent graphics. This book differs in that it establishes a clear theory explaining why some charts are misleading and some are particularly effective.

The central themes of this theory are the maximisation of the "data-ink ratio", i.e. the proportion of ink devoted to the non-redundant display of data information, and the elimination of chartjunk. The reader is rapidly led to question the need for boxes, grids, axes and tick-marks and to keep a watchful eye for the three main forms of chartjunk. These are described as unintentional optical art, grids and "ducks". Any user of the standard computer graphics packages and devices will have encountered the range of "highland-tartan" patterns which can automatically shade an otherwise conventional bar-chart and serve principally to distract from the message of the data. Fewer will have detected the less immediate impact of grids which, unless used with caution, can dominate the information content of a chart. The "duck" is the example where the construction of the decoration of a chart has taken precedence over the presentation of the information:

"The symptoms of the We-Used-A-Computer-To-Build-A-Duck Syndrome appear in a display from a professional journal: the thin substance; the clotted, crinkly lettering all in upper-case sans serif; the pointlessly ordered cross-hatching; the labels written in computer abbreviation; the optical vibration."

The final three chapters concentrate on how to use the theory to create excellent graphics – the benefits and risks of multifunctioning graphical elements; the advantages of high information graphics using data density and multiple small charts; and the incorporation and aesthetics and good technique.

This book should be compulsory reading for anybody actively involved in the design of quantitative graphics, particularly for computer systems which will be released to untrained users. My major complaint concerns availability as the book used to require a direct order to the publishers and was hindered by the presence of at least two Graphics Presses in the U.S.A. However I understand that it can be obtained within the U.K. from Graphics Press UK, PO Box 8, Godalming, Surrey.

Graphics & Image in Office Systems
by S.J. Newton
N.C.C. Publications. ISBN 0 85012 425 5

This book is one of the N.C.C. series surveying various computer-related topics. Its stated aim is to “highlight developments in the hardware and software of business graphics, explore the use of graphics systems and consider the relevance of standards.” However the book goes further by including sections on image processing as well as more conventional graphics.

A list of the chapter headings should indicate the extent of the coverage of the book. They are:

Introduction	(10 pages)
Business Graphics Hardware and Software	(24 pages)
Choosing and Using Charts	(20 pages)
Business Graphics Systems – Types and Examples	(26 pages)
Business Graphics Case Studies	(12 pages)
Justifying Business Graphics	(8 pages)
Graphics Standards	(10 pages)
Image Systems – Types and Examples	(26 pages)
Developments in Graphics and Image	(4 pages)

This is a very ambitious list of contents for a fairly short book, and my initial fear was that the task of compressing the material into so short a space would reduce the book to an unacceptable degree of generality.

The hardware section starts by identifying three basic issues – the method of generation (vector or raster), resolution, and the choice of colour or monochrome. Newton then categorises current graphics devices under the headings of displays, input devices and output devices. I detected some confusion between devices for graphics input and for control of graphics systems. For instance a light pen can be regarded as a graphics device when used for drawing images, but for selecting from a series of options it is irrelevant whether the system under control is a business graphics system or not. Consequently I found much of the discussion about the relative merits of keyboards and voice input irrelevant to the subject of business graphics.

Having just finished re-reading Tufte's *Visual Display of Quantitative Information* (reviewed above), I was inevitably going to be disappointed by Newton's section on choosing and using charts. Newton's is a much more straightforward approach describing the different types of chart in common usage and giving some of the basic rules for good chart design. I remain unconvinced that a 'word chart' is really a business graphics application, and would strongly disagree with almost any advocate of the pie-chart. The rules for good graphics were perfectly reasonable – Determine your message, Choose the appropriate chart form, Keep it simple, . . . – but their impact would have been much better if the first charts in the book (from another NCC publication) had actually obeyed them.

The subsequent chapters continue in like vein. They provide a fairly simple coverage of the subject in a reasonably conventional approach without providing any startling revelations. Most readers will find some new material in the book, but it won't change their thinking overnight.

The book fails for me because it tries to cover far too wide a subject area and has no clear audience. It attempts to combine a state-of-the-art review of current hardware with recommendations on chart design, a justification of the use of graphics and a review of standards – each of which could merit a book of its own. The explanation of how to create a pie-chart:

“the data is computed as a percentage of a total and each data value (percent) is converted to its appropriate segment of a full 360 degree circle”

does seem somewhat unnecessary when the facing page shows the result. On the other hand the organisation structure of the International Standards Organisation and the levels of adoption of six different standards are likely to interest only the specialists in the area.



Welcome To The APL86 Conference

On behalf of the Organising Committee and the British APL Association, I am both proud and happy to invite you to participate in APL86 – **APL In Action**. This, the latest in the series of International APL Conferences, will be held at the University of Manchester Institute of Science and Technology (UMIST) from 7-11 July 1986, organised by the British APL Association in co-operation with the British Computer Society and the Association for Computing Machinery.

The conference will be of high quality; response to our initial call for papers was over 150 contributions and our major regret is that we can include only a limited number of these in our timetable. In addition to these contributed papers we have much more lined up in the way of invited speakers, panel discussions, formal debates and Birds of a Feather sessions; last, but by no means least, there will be the usual major APL Exhibition.

Manchester itself is situated in the North West of England, with excellent travel facilities, and may be considered among its other attributes to be the birthplace of British computing. Away from the conference context you will find that Manchester provides an excellent base for visiting some of the most scenic parts of Britain, among them the Lake District, North Wales, the Peak District and the Potteries. Further afield, but within easy reach, are York, London and the Scottish border.

The UMIST conference facilities are virtually ideal for a conference like APL86, with major lecture theatres equipped with excellent audio-visual aids grouped around a central exhibition area; there is copious space for all manner of the fringe events which go toward making an event like this a resounding success. Accommodation may be either on-site (a two minute stroll from the main APL86 location), or in one of Manchester's excellent hotels (again, only a few minutes' walk away).

The Organising Committee is very excited about the programme we are putting together and will be doing its best to ensure that you will have a truly enjoyable stay; we look forward to seeing you in Manchester.

Dick Bowman – APL86 Chairman

Organising Committee

Chairman	Dick Bowman, Central Electricity Generating Board
Programme	Anthony Camacho, Consultant
Publicity	Philip Goacher, British Computer Society
Technical	Dave Ziemann, Cocking and Drury
Treasurer	Mel Chapman, North Staffs Polytechnic
Exhibition	Christine McCree, Beechams
Administration	Cathy Darue, Tesco
Secretariat	Josy Roberts, BISL

APL86 – Programme and Timetable

Scope Of The Programme: Why should you come to APL86? Because you will learn how to use your current APL better; find out about other APLs that you might use; see how the best APL users go about their work – and whatever your work, you can bring away a briefcase full of useful functions.

We have a dozen papers on how to write APL better: how best to provide full screen interaction: what productivity tools are really useful: how to integrate graphics into the dialogue.

APL can be badly written and get the reputation of fragility. We have papers on how to improve the reliability and robustness of your code and your systems. We have several papers on prototyping and commercial systems, including APL typesetting – come and see how others do it.

There are of course the traditional uses of APL in logic, statistics and mathematics: if Quaternions or Tensor analysis is your field we have papers for you too. The Logic programmers have a field day. Apart from papers on APL as a logic programming language we have 'Expert Systems', 'Knowledge Based Systems' and a half day tutorial on using the power of APL2 in Artificial Intelligence.

As one of our authors has put it – 'A Language for Thoughts and Dreams'.

Programme Outline: The conference begins on the morning of Monday 7 July and the formal sessions continue until lunch on Friday 11 July. There are three main themes upon which the programme streams are based, namely: APL in Action, APL Environment and APL Notation. Each session in these streams has been given a title to reflect its particular emphasis – eg the Monday stream for APL in Action is entitled 'Commercial Uses'.

An 'Education' tutorial day on Sunday 6 July will precede the conference.

In addition to the prearranged programme of invited speakers, semi-formal meetings will be set up during the conference to discuss topics of special interest to delegates. Rooms will be available for such meetings as well as a fully equipped rehearsal room for speakers.

If you have a particular interest in a stream then please complete the boxes marked 'Special Interest' on the booking form. Likewise, if you would like to book a 'Birds of a Feather' session please enter the topic in the box marked BOF on the booking form.

Exhibition: Throughout the conference a comprehensive exhibition will be open to delegates. Exhibitors will be able to demonstrate various hardware and software products discussed during the conference. Product forums will be run by some exhibitors, giving delegates the opportunity to follow items of particular interest.

The following exhibitors have so far reserved space at APL86:

APL People
Cocking and Drury
Dyadic Systems
IBM

IP Sharp Associates
Mercia Software
MicroAPL
STSC

Conference Highlights

- ★ APL86 Education Open Day
A pre-conference event to focus attention on APL in education
- ★ Product Review Sessions
There will be a number of product forums, where APL software and hardware products will be reviewed
- ★ A State-of-the-Art Exhibition
Already 26 booths taken
- ★ A Number of Invited Talks and Tutorials, including

Jim Brown	AI and APL2
Andrew Dickey	Linking to Assembler Code
Dieter Latterman	Graphics
Ray Polivka	Introduction to APL
- ★ Contributed papers on a wide range of topics, including
 - Extending APL to Logic Programming, M Alfonseca
 - APL and AI – A study of Search, Ed Eusebi and Jim Brown
 - IOTA3: The Integration of Spreadsheets and APL, John Searle
 - APL in Quality Business Management, Terry McKee
 - LAPLAN – A Corporate Planning Model for Laura Ashley, Adam Dakin
 - PC Local Area Network – APL Manufacturing Resource Planning, H Clark Key and R L Post
 - Writing Multi-User Applications in APL2, Jim Brown
 - APL: A Prototyping Language, Bob Bernecky
 - APL Graphics Representation and Analysis of Space-Based Observation, Jack Rudd
 - Typesetting APL Manuscripts, John Howland
 - Crash Proofing Programs, Otway Pardee
 - APL2 – A Mathematicians Delight, Norman Thompson
 - Style and Literacy in APL, Rowland Pesch and Mike Berry
 - An APL/370 Compiler. Performance and Comparison with an APL Interpreter and Fortran, Wai-Mee Ching
 - Field Results with the APL Compiler, Clark Wiedmann
 - Japanese APL Language System on IBM Multistation 5550, M Udo and M Alfonseca

APL 86 TIMETABLE

[provisional provisos]

Sunday 6 July

**A special event for those interested in
APL and Education**
**A day of presentations and discussions on
using APL to teach and teaching APL**
 Details to be announced
 Conference Registration

Monday 7 July

9.00 – 1.00	Conference Registration		
10.30	Exhibition Opens		
11.30 – 1.00	Opening Plenary Session		
1.00 – 2.00	Lunch		
2.00 – 3.00	APL Philosophy	Better APL	Commercial Uses
4.00 – 5.30			
6.00 – 7.15	Civic Reception		
6.00	Exhibition closes		
7.30 – 8.30	Dinner		

Tuesday 8 July

8.00 – 9.00	Breakfast		
9.00	Exhibition opens		
9.30 – 11.00 &	Workshop Better APL	Discussion The APL Entrepreneur	Mathematics & Statistics
11.30 – 1.00			
1.00 – 2.00	Lunch		
2.00 – 3.30	Workshop cont.	Debate to be announced	Mathematics & Statistics
4.00 – 5.30			
5.45 – 7.15	Product Forums		
7.15	Exhibition closes		
7.30 – 8.30	Dinner		
8.30 –	Real Ale Evening for connoisseurs of the hop		

Wednesday 9 July

Breakfast			8.00-9.00
Exhibition opens			9.00
Tutorials in various subjects	APL Theory	APL in Business	9.30-11.00 & 11.30-1.00
Lunch			1.00-2.00
Tutorial: APL2 and AI (assuming no knowledge of either - all you need is standard APL)			2.00-3.00 & 4.00-5.00
Product Forums			5.45-7.15
Exhibition closes			7.15
Dinner			7.30-8.30
"Birds of a Feather"			8.30-

Thursday 10 July

Breakfast			8.00-9.00
Exhibition opens			9.00
APL for Prototyping	Discussion Non-serial machines	Enhanced APL's & their uses	9.30-11.00 & 11.30-1.00
Lunch			1.00-2.00
Better Tools in APL	Debate to be announced	Logic Programming	2.00-3.00 & 4.00-5.30
Product Forums			5.45-7.15
Exhibition closes			7.15
Conference Banquet			7.30-9.30

Friday 11 July

Breakfast		8.00-9.00
Exhibition opens		9.00
APL is wonderful	How to print in APL	9.30-11.00
Exhibition closes		11.30
Closing plenary session		11.30-1.00
Round-up and awards		
Lunch		1.00
Farewell		2.00


We can fly
more people to
more conferences
than any
other airline.



British Airways is the official airline for this conference. We are the world's largest international airline and in the best position to offer you the most comprehensive range of services from Concorde through to Economy travel. For further details, contact your local British Airways office or appointed Travel Agent.

BRITISH AIRWAYS

The world's favourite airline.



How To Get To APL86

Manchester lies at the heart of the United Kingdom. From it radiates a vast network of air and surface routes, both national and international.

By Air: Manchester's international airport is the principal airport in the UK outside London. International flights are available to many parts of the world on the services of the large number of international airlines operating from the airport.

For APL86 delegates travelling from Europe, direct flights link Manchester with Belgium, France, West Germany, Italy, Ireland, The Netherlands, Scandinavia and Switzerland. Direct flights are also available to Canada and the USA.

Regular shuttle flights run between London Heathrow and Manchester to extend considerably the number of destinations that can be easily reached.

Buses linking the airport to the air terminal in central Manchester stop quite near the UMIST campus.

By Rail: With two Inter-City rail terminals, Manchester connects with all parts of the UK. For example, London is only two and a half hours away.

Discount fares are available to all delegates travelling to APL86 by train, and application forms will be sent to those who request them.

By taking advantage of cheap, daily return fares, accompanying persons can use the Inter-City rail network to travel to many interesting towns and cities – even to shop in London's West End.

By Road: Motorways give fast and speedy access to Manchester from almost any part of the country. There are direct connections to Scotland, the North, London and the South East with links to the East and West coasts, including the main channel ports.

The motorway spur linking Manchester to the main M6 motorway will bring APL86 delegates very close to the UMIST campus. Local maps will be supplied to those wishing to drive to the conference.

Please indicate on the booking form which form of transport you will be using so that we can send the appropriate information.

British Airways: The official airline for APL86 can offer a fly/drive scheme, allowing delegates to spend a few days sightseeing before or after the conference. Contact your local British Airways office or appointed travel agent to arrange your itinerary.

Accommodation

UMIST: Accommodation in the halls of residence on the campus at UMIST has been included in the delegate and accompanying person fees. Rooms are available for the duration of the conference only (6-10 July inclusive) and consist of single study bedrooms with washbasin. Shared bath/shower and WC facilities are available on each floor – usually serving four rooms.



Also included in the fees (both delegate and accompanying person) are all meals from the evening of Sunday 6 July to lunch on Friday 11 July. A cold buffet will be available for those arriving late on Sunday 6 July. All residents have access to kitchen facilities (tea/coffee provided) and to communal TV lounges.

A limited number of twin rooms are available (conference rate per person) and these will be allocated on a first come, first served basis. A full guide to the campus will be provided with confirmation of booking.

Hotel Supplement: Delegates wishing to stay at one of the nearby four-star hotels will be charged a supplement in addition to the conference fee. Please note that this charge is to supplement the amount allocated for accommodation in the conference fee. The rates per room, per night are £50 for a double/twin room and £36 for a single room. This price includes breakfast, private bath, telephone, colour TV, tea and coffee making facilities and VAT at 15%. Extras, such as drinks are settled with the hotel directly. These rates also apply to accompanying persons.

Hotel bookings must be made before 7 April, after which time delegates must make their own hotel arrangements.

Please indicate on the booking form if you wish to book a hotel.

Facilities: All delegates and accompanying persons will be entitled to use the UMIST sports facilities during the week of the conference. Automatic membership of the campus bars also applies.

Registration Information and Terms

Registration: Delegates are requested to register as early as possible using the enclosed Registration Form. A separate form is required for each delegate – please photocopy the form as required. Completed forms together with payment in full should be sent as soon as possible to;

Conference Department (APL86)
The British Computer Society
13 Mansfield Street
London W1M 0BP, England
Telephone 01 637 0471

Fees: Reduced fees are available to delegates holding membership of any of the following: British APL Association (BAA), British Computer Society (BCS), SigAPL and the Association of Computing Machinery (ACM). The rates in pounds sterling are:

	Before 7 April	Before 6 July	On-Site
Delegate – Member	316.25	345.00	373.75
– Non-Member	345.00	373.75	406.50
Accompanying Person	152.50	181.25	210.00
Banquet (Accompanying Person)	20.00	20.00	20.00

Delegate and accompanying person fees include accommodation at UMIST and all meals from the evening of Sunday 6 July to lunch on Friday 11 July. In addition, the delegate fee includes attendance at conference sessions, access to the exhibition areas, conference documentation and attendance at the conference banquet. All fees are inclusive of Value Added Tax at 15%.

Payment: All fees are due with the application and must be in £ sterling. Cheques should be drawn on a British clearing bank and be made payable to the 'British Informatics Society Limited'. Confirmation of the booking will be sent to each delegate along with relevant maps and instructions. Early payment discounts will apply according to the date of the postmark on the envelope in which the application has been sent.

Payments still outstanding after 11 July will be subject to a 5% surcharge.

Substitution: A registered delegate may nominate a substitute providing that the conference office is informed no later than 4 July 1986. The amount of fees then due is according to the eligibility of the substitute, who will be invoiced for any difference owing.

Cancellations: Cancellations will be accepted only if received in writing not later than 6 June 1986. A 10% handling charge will be deducted from any refund.

Programme: The Organisers reserve the right to make changes to the advertised programme.

**APL
SOFTWARE
FOR THE 1990s**

Senior Programmers from £16,000

Programmers from £12,000

Graduates from £9,000



Metapraxix is a dynamic young company providing advanced management information systems for large organisations. We are rapidly establishing a base of blue-chip clients who are using our current generation of products to set new standards for financial control and the presentation of information to Board Directors.

We are now looking for exceptional individuals to join our software development and technical consultancy teams. You will be involved in developing innovative new products, some of which border on the expert system field, as well as enhancing our existing systems and providing in-depth technical support to our clients. Working with advanced colour graphics workstations, you will use a mix of skills in areas such as APL, Assembler, graphics, hardware device control and online data communications.

Graduate applicants should have a first or upper second class honours degree in a relevant discipline. Programmers should be similarly qualified and have at least two years experience in the APL language. Senior Programmers should combine these skills with a demonstrable track record in the implementation of large commercial projects.

This is a unique opportunity to influence and share in the growth of a new industry, and to make a major contribution in an environment that is free from large company rules and regulations. Please write to me with comprehensive career details, indicating which position is of interest, at the following address:

David Preedy

Development Director

**Metapraxix Limited; Hanover House, Coombe Road,
Kingston-upon-Thames, KT2 7AH, Surrey.**

RECENT MEETINGS

This section of VECTOR is intended to document the seminars delivered at recent meetings of the Association, particularly for the benefit of those members based away from London who often find it hard to find the time to attend. It also covers other selected events which are likely to be of interest to the wider APL community.

We are dependent on the willingness of speakers to provide us with a written version of their talk, and we would remind them the "a picture's worth a thousand words". Copies of slides and transparencies will enhance their articles.

The Activities Officer (see inside back cover) will respond enthusiastically to offers from individuals who wish to contribute seminars and supporting papers.

Introductory Notes

by Adrian Smith

In this issue I have included notes on two very successful meetings, both at the now well-established venue of the The Royal Over-Seas League. The first was a series of (mostly) short papers on applications on micros; the second a much more lightly programmed set of papers on commercial APL systems. Readers are referred to the Case Study for a fuller version of Christine Brewster's talk on Sales Analysis.

Besides our coverage of British APL Association meetings, we also have a report on the APL Statistics User Group and a review of some of the sessions from the Operational Research Society conference held in Durham.

APL Systems on Micros October 18th 1985

Introduction

I would like to thank Eileen Dyson for taking such thorough notes on this very heavily scheduled meeting. The overall impression was of a set of well constructed and relevant talks; towards the end this was unfortunately thrown by one speaker who went way over his allocated time, and hopelessly cramped those who followed him.

Maybe to have five talks in a single afternoon was pushing things a bit anyway? It may be a super way of filling up VECTOR, but I can't believe that all the speakers were really given the attention they deserved.

Developments in PEFAC

by Chris Staffurth

This paper gives an update of the talk on PEFAC that Herbert Walton gave to the APL Users Group in 1981, and how PEFAC – Production Engineering Fast Accurate and Consistent – has been transferred from the I P Sharp time-sharing environment to the IBM PC.

PEFAC is an estimating package that does all the arithmetic for the industrial engineer when he has to estimate the time that would be taken to machine a given component. It is particularly useful for:

- ▶ the company that does sub-contract machine shop work and has to provide a quotation from the drawing of the part to be machined. That is, it needs to be Fast and Accurate.
- ▶ the company that pays its machine shop operatives on a bonus scheme. That is, it needs to be Accurate and Consistent.

PEFAC consists of 5 modules, for:

- ▷ Turning
- ▷ Milling
- ▷ Grinding
- ▷ Drilling
- ▷ Boring

It can also cope with Ancillary modules that can be written to meet customer's particular methods and environment. For example, Sheet Metal work is done in a variety of methods, and it is not really possible to program the general case.

PEFAC is now sold on an IBM PC/XT with 10mb. hard disc, a minimum of 384k bytes memory, mono or colour VDU and preferably an Epson FX printer. It uses the APL* PLUS interpreter.

There is a PEFAC workspace which contains the basic functions for input, output and other housekeeping tasks. Functions relating to individual modules are called into memory when required (and the functions relating to the previous module if different are expunged). A special pair of functions containing Assembler segments has been written. One function packs objects, either functions or variables into a single object, and the other unpacks them. This is reckoned to be 5 – 10 times faster than storing functions in quad-CR form and then using quad-FX to fix them.

The PEFAC system is menu driven and the user is unaware that the functions have been written in APL. Even the keyboard is set to run in the ASCII mode. This provides an interesting situation for the maintenance programmer who has to alternate between the two modes: the overstruck characters are in different positions on the keyboard. Maintenance work on site in the absence of a chip for an APL character generator is even more interesting. Characters like Spades, Hearts, Diamonds and Clubs and others appear on the screen.

Being menu driven, the user has usually to answer Yes or No or 1, 2 or 3 to questions. But where there is a list of things to look up, for example, a list of materials or a list of available machines, then use of a function key will display the whole list on to the screen in inverse video.

If the user is uncertain of what input is expected, there is a set of Help screens that are invoked with a function key, and these explain the requirements in detail.

PEFAC generates a report showing the basic machining time of each element of an operation. Each operation report is stored in a temporary file. This temporary file is a security measure, in case for any reason the estimate for a complete job is interrupted part way through. Then only the part-completed operation is lost. It also serves as a What If report, and allows the user to try out more than one method of machining the part. When the user is satisfied with the report, it can be kept permanently, and PEFAC transfers it to a separate Drawings directory.

Engineering Change Orders are a frequent occurrence in a machine shop, in which maybe only one dimension is altered. A revised estimate can be easily calculated by retrieving the drawing from the Drawings directory and by re-processing the element which calls for the particular dimension. If a material specification only is changed, the report of the drawing can be similarly retrieved and after the material has been changed PEFAC will silently reprocess the whole estimate, without the user having to re-input the answers to all the questions. This means that besides packing away the text of the report for each operation, numerous parameters relating to operations and elements have also to be stored. In fact, PEFAC has to work very hard behind the scenes, but it saves the estimator a lot of time and trouble.

As a security measure against corruption of the hard disc, the reports may be backed up on to diskettes, and there is a comprehensive index to keep track of the whereabouts of all the drawings in the system. When there are more than 50 drawings in the Drawings directory, PEFAC starts to erase them from the hard disc, but only if they have been backed up on a diskette.

Although every user has the standard functions for each module, there is usually a different set of machines and materials. Therefore different data is required for each customer. Another workspace exists to allow the user to input or edit any of the data relating to his machine shop. Full screen editing is obtained from a set of functions that has been written. The screens are arranged hierarchically. The user is asked successively for the module, machine group within the module, the actual machine within the group, and then the table within the actual machine. Functions are required to edit character and numeric arrays, and to interpret the action of some 20 non-alphanumeric keys. It is possible to insert or delete new tables or even data relating to a new machine.

At the bottom of the screen are simple prompts for going forward, going backwards, saving and quitting. There is an Edit Help screen which explains the editing in detail, and also an Engineering Help screen which explains the engineering terms used.

Although the PC might be called a micro computer, workspace size or WS FULL messages are not a problem. By overlaying the module functions and storing the report after each operation, there is sufficient space for all intermediate calculations. As a final measure, a check is kept on quad-WA, and then when it gets rather low, the user is prompted to end that operation and to start another.

The response time is generally 2-3 seconds and never more than 5 seconds. This is adequate since the estimator has to think of the job in engineering terms as well as inputting dimensions etc. The reading of the module functions takes about 30 seconds, but since it does not occur very often, it is no great hardship.

Software errors do present a problem. It is estimated that there is somewhere between 10^6 and 10^8 separate logical paths through PEFAC, and therefore impracticable to test it for every eventuality. In the time-sharing environment it was fairly simple to log on to the user's CONTINUE workspace and sort out his problems. But that is not possible on the PC.

On the PC there is a general error trapping function which does two things on encountering an APL error. It sends to the printer the APL error message and the offending line in the function that caused it, together with the contents of the SI stack, plus some parameters relating to the element in hand. This is posted to us and is used as a guide in reproducing the error condition. A corrected file is copied on to a diskette which is then posted back to the user. The error trapping also escapes from the current operation, or if the error is severe, re-loads the complete PEFAC workspace. In either event, the user is not left entirely hanging in mid-air. At least he can continue with his work.

Development of PEFAC continues, and there are plans to add further in due course.

Acknowledgements were given to Herbert Walton for starting PEFAC and making it into a commercial proposition, and to Ian Kemmish who masterminded the installation on the PC.

Corporate Planning at Laura Ashley

by Adam Dakin

The speaker worked for Deloitte, a London based computer company which had been involved in developing a corporate planning system for Laura Ashley. The system was developed in eight months by a team of four; it was written in APL and ran on an IBM PC/AT.

History of the Project

To quote the speaker:

“LA is a fast growing company, receptive to new technology. It is an exciting place to work”

LA make corporate plans every five years. In 1980, the plan was worked out by hand; it was a long hard slog with a lot of number-crunching. Top management then vowed that a computer model would be developed for next time, both to do the number crunching and to allow ‘What if’ questions to be evaluated more easily. The model was therefore specified in 1981 but development did not begin until 1984.

LA produce a wide range of products which they sell worldwide. They have manufacturing sites in the UK, Holland and USA and sell a diversified product range through a range of outlets. The company also buy in from third parties if they feel they can sell goods which they cannot produce internally.

Important Factors

1. Structural – The structure of the company may well change over the next 5 years. For example LA are currently moving into Japan.
2. Sales Targets – LA is a sales led company and again sales patterns may well change in the next 5 years.
3. Manufacturing and Retailing Operation – Manufacturing practises and technology could well change in 5 years as could the balance of stock.
4. External – typically exchange rates

Factors 1-3 are all within LA control and can therefore be predicted whilst LA has little control over factor 4.

The first stage in developing the model was to decide the level of detail to be used. Similar outlets and products were grouped together to make the data more manageable, but at the same time a tradeoff was needed to avoid the data becoming meaningless. It was decided to group the outlets nationally and to group products into six categories. The products and locations were then linked in a diagram; even at this early stage simply seeing a diagrammatic representation of the company was useful.

LAPLAN- Modules

The system was divided into 8 modules. Modules 1-4 could be used to simulate the company whilst 5-8 provide a means of investigating the financial implications of these simulations.

1. Calculate sales revenue & stocking
2. Routing (Demand from factory A to location B)
3. Manufacturing Costs – Unit cost of production
4. Transfer costs between groups
5. Trading account by country
6. Manufacturing account
7. Company results
8. Group results

Implementation

Data required for the module was in several dimensions and it was therefore felt that APL was the most appropriate language to use being more flexible than WIZARD or other modelling languages available on micros or mainframes. It was decided to use APL for modules 1-4 and SYMPHONY for 5-8.

The first three months of development were spent getting the algorithms working correctly by means of a prototype.

Benefits of a prototype

1. Proves model algorithms
2. Gives user familiarity and confidence
3. Highlights data requirements
4. Promotes discussion of possible use
5. Offers smaller version of the model earlier at a lower cost

Current System

The system is a 'Black Box' menu driven system which is easy for the user and makes APL invisible to him. It now requires some 17Mb and therefore had to be switched from RAM to filing. Also in an attempt to conserve space the four-dimensional data is stored as several three dimensional arrays with pointers to them in the workspace. One of the more tricky parts of the development involved setting up print files to link to SYMPHONY.

In summary, the model is a success. It shows the value of prototyping and also that micro-based APL is powerful. It also opposes the theory that APL is an unfriendly language to use; the speaker had no previous experience of APL, yet he managed well with it and would quite happily use it in the future.

An Integrated Manufacturing System on a Local Area Network

by Ed Shaw Junior (Bristol Myers)

Bristol Myers is a large American-based company producing a variety of products including pharmaceuticals and medical equipment. The Corporate Systems department employs about thirty people, all writing APL. This makes them the second largest APL users in New York.

The system which the speaker described was currently being installed in one of the Dutch factories which manufactured milk-based products.

It was complex and was copiously described to us by means of a very colourful but totally incomprehensible overhead slide. The idea was that the system had to be designed for all areas of the company to use. At the purchasing end it was used for purchase orders and goods inwards. The information was then used for inventory and production control, and also to specify the formulae for the products and for daily production scheduling. Quality control also got their hands on the data, and eventually it was used for sales information, packing lists and invoices.

The system had to run in real time. When the milk was delivered production needed to know what the protein and solid content of each batch was so that they could adjust their recipes accordingly.

Other than the gory details of what each function of the company used the system for, I did glean that it used an STSC tool box of APL'isms and ran on a LAN which had a four hour 'fix it or else' maintenance agreement with the suppliers.

Migration from Mainframe to Micro

by John Birch

The speaker was from Premium Management, a company in the City concerned with managing investments for the insurance industry. He felt that to be successful a company should consider three questions:

1. Where are we now?
2. How did we get here?
3. Where do we go next?

Because of statutory restrictions his company had to concentrate on points 1 & 2 even though 3 is probably the most interesting.

The fixed investment market involves a great deal of arithmetic. Until 1983 the company ran APL on an IBM mainframe when they decided to migrate to a micro. The micro which they chose was the MicroAPL Spectrum.

The Migration

Three points arose during the migration:

1. The IBM APL was compatible with APL.68000. However the execution of some code was accelerated by using the APL.68000 quad functions.
2. The file system was incompatible and had to be rewritten for the Spectrum. The data base also had to be redesigned to obviate problems created by the relatively slow input/output processes.
3. The Full Screen systems were compatible as the company had moved from AP124 to AP126.

It took just 15 days to migrate from the mainframe to an operational system on the Spectrum (including the Christmas break). It then took 2 people a further 5 weeks to enhance it to a satisfactory system.

Before the migration the company had 1 printer and 1 VDU on their mainframe. They now have 5 VDUs connected to the Spectrum and the system is also linked, or is planned to be linked, to various external services such as those offered by Citibank, Don Jones and the Telex network.

Investors like an easy life and therefore any systems developed for them must be simple to use with minimum effort. For this reason the company is looking at using Telecom touch screens.

Why APL?

APL is used within the company because it is quick and easy to write. The strange hieroglyphics and intricacies of APL can be hidden from computer illiterate users. It is also easy to amend reports to please the users and can be done relatively quickly. It is however the only language the speaker has used so he may be slightly biased!

Problems with the Spectrum?

Multi-usage is not a problem on the Spectrum. If there is a problem it lies not with APL but with the speaker's secretary who tends to get rather carried away when using the machine as a wordprocessor!

The only real problem is that response is poor if several users are all accessing disks at once because the Spectrum reads sequentially.



APL People Ltd.

17 Barton Street,
Bath,
Avon,
BA1 1HQ
Tel: 0225 62601

APPOINTMENTS

A placement service solely for the APL community.

Use our extensive knowledge, contacts and experience in the APL world.
We have vacancies NOW, both permanent and contract and our confidential advice on career moves is always freely available.

CONSULTANCY

A consultancy service designed to solve your APL problems.

Consultant's Toolkit and Supervision available.

Current projects include:-

- * graphics
- * project management
- * databases
- * system design
- * system 'rescue'
- * documentation
- * VSAPL under VSPC, CMS, TSO
- APL* Plus/PC, Sharp APL

To discuss your requirements: — Call Valerie Lusmore on 0225-62601

Registered employment agency licence SE6440

An Overview of APL Applications on Micros

by Paul Thornton (MicroAPL Ltd)

The speaker gave a very brief overview of the types of applications MicroAPL have been involved with in recent years. In general there is a broad spread of applications across a variety of industries. These applications are in fields such as OR, Information Services, Maths, Commerce, Science, Education and Real Time applications (primarily Process Control).

National Westminster Bank	- Dial in data bulletin board.
	- Developing funds transfer via a dial-in system.
Bank of England	- Small Multi-user systems.
Golden Wonder	- Distribution Applications
	- What is the best method of transferring goods from A to B?
St Thomas's Hospital	- Collating data from a questionnaire on diabetes
British Telecom	- Training staff on electronic theory
Mobil Oil	- Calculating wastage when passing oil through pipes.
	- Route Planning
	- APL Simulation of a complex Fortran simulation program
Cornhill	- Modelling
Heineken	- On-line production monitoring

Before 1982 there were no full implementations of APL on micros, today there are several.

Features of micros

1. Small desk top unit – The user therefore feels more secure.
2. Ease of access encourages people to dabble.
3. Fast data output to screens, a feature only previously available from IBM.
4. Excellent for device control – it is relatively easy to connect to Reuters etc.
5. APL is used extensively for prototyping. It is now possible to link in Assembler and use non APL packages thus increasing the range of usage.
6. More people though not a significant number are using APL.
7. IBM PC APL was the cheapest available but now APL is available on the QL for around £300.

Taking all this into account, the future looks promising.

APL in Practice – Commercial Applications, 15th November 1985

Introduction

Thanks go to Bern King for these notes. We have included his comments on Christine Brewster's talk, although readers should note that there is a much fuller version of her paper in the next issue of VECTOR.

Apologies to Peter Evans & Simon Letchet (Kodak); their talk has not (as far as I know) been recorded for posterity, and all I can report is that at the end, there was a brief demonstration of Instagraphic, a customised camera which converts a display on a terminal into a slide in a couple of minutes. Cost £400 . . . every home should have one!

A Sales Forecasting System

by Patrick Rushton (Nestle)

Scope

The system was developed to allow production to be more closely geared to the changing market. The products were highly seasonal. Forecasters wanted to use their knowledge of hierarchical structure of the Nestle products (e.g. to improve estimates of frozen fish sales in light of projected trends in whole frozen food sector). To complicate the whole process, divisions had different accounting periods (some monthly, some 4-weekly).

History

Some of the talk outlined the evolution of the system through computers, APL's and Operating Systems. Briefly, the original system (using VSAM under CMS) did not allow multi-user update, so they moved to SHAREFILE (a 'crash-resistant' filing system). They are considering moving to APL2 and DATABASE (DB2) in the future.

The original aim was to use the APL development as a prototype for a more formal COBOL system, as the need for an evolutionary approach was foreseen. One year after the system had been finalised, the transfer to COBOL was abandoned and the APL system installed in four weeks! Compatibility with other company systems was maintained by continuing to create VSAM files in the same format as the previous system, rather than altering all dependent systems.

Algorithms

The aids for producing forecasts appeared flexible, including options to: (i) view and alter forecasts on a 4-weekly or a monthly basis or by normalised months. (ii) use the system as a simple calculator (e.g. to reduce forecasts by 10% or add 10). (iii) alter any elements and row/column totals in a 2-dimensional display; the system would jiggle it about to find a sensible result.

Conclusion

This system seems to embody many of the features that we are attempting to put into our own sales forecasting system, so it can be done . . .

APL in Use with a Large Commercial Database

by Christine Brewster (Whitbread & Co)

A large sales analysis system was described. The aim was to access and summarize sales data to produce comparisons of budget against current year and current year against last year. The size of the system was stressed: estimated at 12Mb new data each week and prints up to 40,000 pages!

Scope

The system is designed to answer queries by outlet, product or depot. 'Trees' of categories of outlet are used to indicate the summaries required. For example, outlets are grouped into company regions and areas but also into customers and regions used by customers.

The advantage of this tree structure is that an outlet could be switched groups at the lowest level and all the rest of the tree would be right. Keeping the tree up-to-date is seen as a substantial problem. An 'undefined' category was kept in the tree at all levels, so that every outlet would appear in reports, and outlets that needed information updated would be obvious.

Development

Before APL was chosen two other options were tried and rejected: (i) The original intention was to write the programs in COBOL, and to store all the subtotals that would be needed in reports. (rejected as 40 hours CPU time needed for weekly update) (ii) Bureau solution. No package was found that could cope with complexity. (She did admit that if they were starting again they might choose a different solution.)

I wonder what happened to all the complex Sales Analysis Problems which were solved by these approaches, and didn't end up as talks at BAA meetings . . . Anyway, APL was chosen for the following reasons: (i) cheap – APL was already available and only 4 people were needed to implement the system (ii) quick – live data from the start allowed an immediate return (first summary reports after 3 months and fully implemented after 6 months) (iii) low risk – it would work, probably (iv) experimental – they didn't have to finalise design at the start and it could be used as a prototype.

Reports

Reports are produced in batch runs of APL under TSO as there was so much data. The data was sorted into the appropriate order for a report outside APL. Each day's reports were therefore reordered and submitted in the sequence that needed least sorting. The final result was a JCL program requesting the following steps: (i) invoke APL and read in data (7Mb workspace) (ii) sort data (iii) return to APL and generate report (iv) repeat from steps i,ii, or iii as required.

Conclusion

No system would have been possible without APL. They knew it was a success as the user had requested extra features!

BACK NUMBERS OF VECTOR

Back numbers of VECTOR are available from the BCS. If you don't have them all, now is the time to complete your collection. There is an index to volume one in this issue of VECTOR to tempt you. Apart from the technical contents every issue includes book and product reviews, letters, news and a competition. Send in your order before they run out. These will one day be unobtainable collectors' items like the early issues of Quote Quad.

The prices inclusive of postage and packing are as follows:

	UK	Prices in Pounds Sterling	
		Surface (inc. Europe)	Airmail (outside Europe)
Single issues	3	3.75	5.75
Volume I	10	14.00	22.00

Please send sterling cheques or money orders payable to The British APL Association to:

The British APL Association, 13 Mansfield Street, London W1M 0BP

Don't forget to include your name and address and to be clear which VECTORS you want.

APL Statistics Users Group

Meeting 5th June, 1985

at Customer Centre, IBM, South Bank.

by Jake Ansell

Barry Wetherill spoke on 'Intelligent Statistical Software', illustrating initially how 'dumb' software failed to highlight problems in data sets. For these various problems he suggested appropriate diagnostic tests which could be used. He concluded by showing how the measures are included in the Survey Package he designed which included data validation.

The second talk by Alan Hawkes was on the 'Role of APL in Teaching and Research'. This talk detailed the approach used at Swansea to teach statistics students through the medium of APL. He then illustrated his own use of APL in research emphasising the speed of using APL compared with other languages.

After the talks a discussion of the future role of the group took place, and format of its future meetings.

One suggestion which arose was the formation of a Program Subgroup to consider the possibilities of a standardized APL statistics workspace for statistics. The group were not totally in agreement over the role of this group, but thought it worthy of pursuit.

Another suggestion was the meeting should allow for 'hands-on' experience of software. It was thought that future meetings would have a workshop session in the morning followed by talks in the afternoon. To resolve other problems it was decided to send out a survey.

For further details of the APL Statistics Users Group please contact Jake Ansell at the following address:

Department of Statistics and O.R.,
Coventry Polytechnic,
Priory Street,
Coventry.
CV1 5FB. Tel. (0203) 24166 570

Application Prototype Environment

*an IBM program product for prototyping
and developing screen-based applications under VS APL or APL2*

This product is intended to be used both by the professional programmer and by business professionals, such as managers, planners, scientists, and engineers. The product is particularly suited for developing applications in Information Centre and Application Development Centre environments.

Three interactive functions are provided to easily and quickly define the end user interface.

Panel Design: For creating menu selection, action, and tutorial panels.

Chart Design: Charts (Plot, Surface, Bar, and Pie) are designed via an interactive dialogue.

File Handling: The interactive dialogue allows users to specify APL File, CMS, VSAM, QSAM, BDAM, and INSTORAGE access methods.

Over 70 cover functions are provided for building the application program. Some examples are:

Panel Operations

- Automatic Screen I/O.
- Automatic display of tutorials.
- Program function key control.

File Handling

- Reading/writing records from/to a file.
- Positioning the record pointer.
- Relative record access.

Object Library

The object library is used for storing, retrieving, and status checking of objects such as panels, functions, declarations, and variables.

Free 30-day Test Period

You have 30 days to test Application Prototype Environment to see how well it performs for you. If you are not completely satisfied, just send all materials back to your local IBM Branch Office within 30 days and you will not be billed.

For More Information: Contact your IBM sales representative or the nearest IBM Branch Office.

Technical Data

Application Prototype Environment is an IBM Licensed Program Product, Program Number 5668-896.

The program runs under CMS and TSO together with the following IBM programs or their equivalents: APL2 or VS APL, GDDM (Graphical Data Display Manager). Some examples of terminals supported are: 3277, 3279, 3270 PC/G and CX.



IBM United Kingdom Limited

OR-85 (Durham): Notes on Major Sessions

compiled by Adrian Smith

Disclaimer

These notes cover the four plenary sessions held at the conference. They are obviously a personal view of what was said, rather than an accurate record. If any of the speakers feel grossly misrepresented, I can only offer my apologies, and suggest that they make use of the letters page to redress the balance.

General Introduction

If you skim the surface of the notes which follow, I hope you will become aware that there is a common thread running through all four sessions. Put simply, it is the need for vision and genuine strategic thinking, whether in telecoms, personnel management, building a transport network, or just life in general. As you might expect, it was left to Dr David Jenkins (Lord Bishop of Durham) to tackle this last one; the other three speakers were much more specific in the areas they addressed.

Dr Peter Keen was primarily interested in the development of global telecommunications, and the impact of this on people like banks, retailers, airlines and travel agents. For me, the key message of his talk was the need to think 5 – 7 years ahead; it is hard to counter a pre-emptive strike which is based on this kind of technology.

Considerable vision must have been needed to launch a totally new inner-city light railway, just when the motorway boom was at its height. There is no doubt that the Tyne and Wear Metro is an outstanding success, and with hindsight the reasons aren't hard to see. It is clean and efficient, and above all it is fully integrated with existing public transport.

As a born and bred Wearsider, I obviously have a personal interest in the success of the Nissan car plant at Washington. The final session of the conference described a vision which is yet to be realised; the application of Japanese management to car building in the UK. It will be fascinating to look back on these notes in (say) 10 years time, and see just how close they have come.

The Planning of Change – any Hope?

by the Right Revd Dr David Jenkins (Lord Bishop of Durham)

First some definitions:

“OR – people who make a speciality of analysing what is going on, and helping to manage it better”

“What is going on – the bits of reality you decide to define as real, and promptly make unreal”

“Reality – people tend to define reality as what they see; more often it is what they miss”

“Manage – making things happen on purpose”

I think that sets the scene nicely for Dr Jenkins talk. It was a splendidly entertaining hour, full of memorable epigrams as well as sound practical advice. The four definitions above illustrate his highly recursive style of discourse! Taken as a series of points in order, these notes will make very little sense; taken as an overall nested structure they probably won't either, unless you did a degree in computer science and can write LISP in your sleep!

“One of the reasons people don't believe in God is the way people who believe in God carry on!”

Enough said.

The Value Content of Technical Problems

Suppose you are looking for a simple model as a way of getting a grip on some part of reality. You might start with a premise like “. . . pragmatically, based on reasonably observable facts, . . .”. Already you have allowed value judgements to creep in to your model! In fact it is increasingly being recognised (see the conference abstracts) that values come in to the modelling process at a very early stage.

This awareness of the value-content of problems raises some interesting questions:

- 1 Can OR throw any light on previously assumed ‘value-free’ techniques, which are now recognised as ‘value-full’?
- 2 How do you get a real discussion of value-full issues?
- 3 Does God make a difference via ‘values’ and ‘visions’?

In a turbulent world, there is no doubt that a shared set of values is much more important to society than any number of institutions or regulations.

The Creative Exploitation of Randomness

Of all the phrases used by Dr Jenkins, this is the one most quoted in the bar that evening. To put it in context: the world is a turbulent place where most of the important things happen by accident. The great skill of leadership, management and prophecy lies in the creative exploitation of randomness.

Those who aspire to this skill must know how to leave a system alone for long enough to work out what is going on. They must also pay continual attention; no amount of theorizing will get you anywhere in a stream of truly random events. They should appreciate that there are limits, both to the information which is obtainable, and which is handleable:

“Does the college need a new kitchen?”

“... waffle ... waffle ... rhubarb ... rhubarb ...”

“If you don't take a decision today, you won't get any dinner tomorrow!”

A statement like this radically reduces the amount of information considered relevant to the problem!!

To find out more about our turbulent environment we need plenty of far-out hypotheses and risky experiments. To make it habitable, we need to minimize the perceived randomness (and risk) with a shared set of values; definitely not by regulations and probably not with 'hard' OR models.

OR and Social Problems

Can OR help with the management of social problems? Probably not – we tend to work in clearly defined sub-sub-systems which may themselves be harmful. Maybe we 'define away' the real world with models (e.g. cost-benefit analysis) which assume (and depend on) a particular theory of economics.

What about governments? The present one has a pathological dislike of planning, yet is quite willing to regulate for its own benefit! In 1969 there was published a list of “10 critical continuous problems”; all 10 of these are still well and truly with us! How can society get out of this if we don't plan? The Bible is pretty clear on judging societies on what they do to their 'marginals'.

Maybe OR Can Help, After All

How? Simply by rehabilitating the idea of planning. Companies appreciate the need for planning and co-ordination; no-one would try and operate a 'free market' economy within a group of factories. OR makes its living by building tools to help companies plan better; surely we should be able to hold up the results as a positive reminder that planning and co-ordination are a vital concern of good government.

In summary, the Lord Bishop had these messages for the OR community:

- ★ Whatever you do, be alert to the context of your work. Don't forget about 'meta-things' in your eagerness to crack 'micro-things'.
- ★ Always be ready to respond creatively to random events.
- ★ Listen to your own sense of values.

If we can discover and build a sufficient consensus about what reality is like, maybe we can help to generate the trust which is needed before people will once again accept the idea of planning.

Telecommunications and Business Strategy

by Dr Peter Keen (Managing Director; Infoscope Ltd)

“What happens when a large part of your future depends on a market that doesn't exist?”

The meat of Dr Keen's talk was a series of examples showing the way some US companies have faced up to this question. In particular he focussed the talk around the subsidiary question “What are they doing with Telecoms?”

Setting up the Infrastructure

The key factor in the new marketplace is the customer access-point; the ‘workstation’ can be a VDU in an office, a bank cashpoint, a supermarket checkout, or an airline reservation desk. This explains why the airlines and the banks have suddenly begun to see each other as competitors! Once you have the workstation established, your next move is to start adding more services to the network.

The reason that the access-point is a key to success is that service is fast becoming the only way to differentiate between commodities (e.g. banking) in the marketplace. Cashpoints could never be ‘cost-justified’ on normal accounting; any bank without them is as good as dead! How do you take that kind of decision?

“American Hospital Supplies” is a pharmaceutical company which used to have around 5% of the US market. It spent 5 – 7 years on the infrastructure; building a network which put terminals into every major American hospital. It now has nearer 55% of the market – why?

Clearly it is marginally easier to buy from AHS than from competitors, but that is not the key reason. More important is the added service that if a hospital buys everything through AHS it gets a free information system, and has all its stock control done for it! By adding the element of feedback into the ordering process (with MIS-type reports on usage of supplies) AHS can exploit its network to provide a unique service.

One of the most important lessons from the AHS experience is the value of a pre-emptive strike. If your strike is based on technology (typically with a 5 year lead time) you have an unassailable lead in the marketplace. This is a far more effective strategy than ‘sales gimmicks’ (free glasses with petrol) where everyone joins in and they all get nowhere. The problem is that most technological decisions are made on a 1 year cycle and are ‘cost-justified’; an effective technological strike is a 5 year venture of faith!

Getting a Focussed View of your Customers

Once you have a ‘point of sale’ system, you have the potential to find out both who your customers are, and what they are buying. American Airlines used their ‘frequent flyer’ gimmick to pick up just this kind of information! In the same way it probably provides the motivation for the M&S charge-card. Once you know who your customers are, you can start grafting on all sorts of extra services.

Examples include Merrill Lynch in the US; their 'cash management account' is precisely the all-in-one for those rich Americans who they already had signed up on their network. Scandinavian Airlines let you book your hotel from the airport desk, and check in your bags from the hotel! Look at Reuters; when everyone else was scrabbling to get terminals into the corporate HQ they already had one!

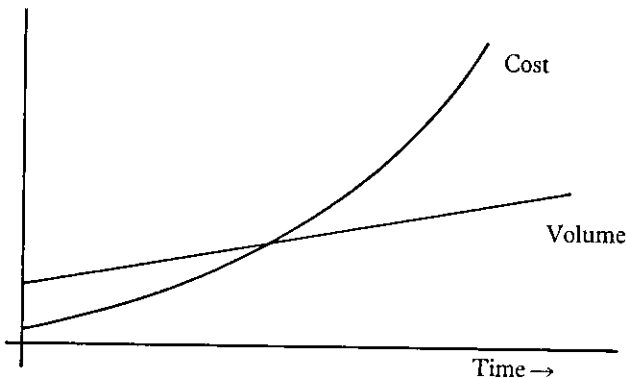
Moral: Own the infrastructure and the delivery point.
Be there when demand takes off!

40% of the US population wanders past a Sears terminal (a department store till) in a typical week. In 10 years time Sears (a retailer) could well be the leading US bank, because they own the network. Similarly AMEX could be the leading travel company, Citibank the leading electronic publisher. If that 5-year lead-time is accurate, it is probably too late to stop them.

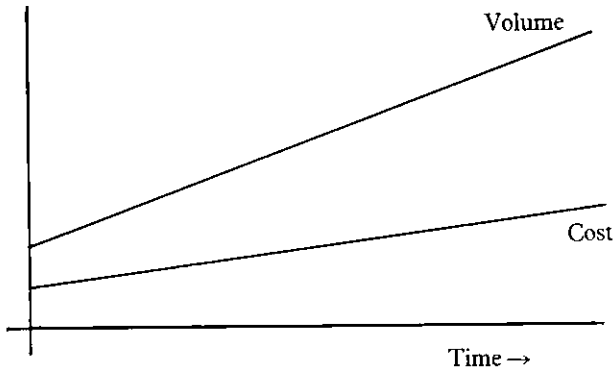
Digression: Whatever Happened to the Electronic Office?

Companies the world over launched enthusiastically into pilot schemes, with networks of communicating PCs handling word processing, email and so on. This was 4 or 5 years ago; why has there been so little follow-up?

The problem is that personal computers are personal. WP may well reduce typing costs by 20%, but how much better to stop all that paper flying around (and then getting archived for 7 years at vast expense). In a typical service industry the interconnections multiply very rapidly as the organisation grows, so you get a cost/volume graph like:



With enough of the right technology, economies of scale are possible, and the two curves come out more like:



For example, transactions that can cost you £2 – 3 in low-tech London might be nearer 15 cents in the US. Little airlines boomed in the late 70s, but the big ones are coming back as they get the benefit of fixed-cost technology.

The Dynamics of Innovation

Dr Keen concluded his talk with a four-point plan for strategic innovation. He also noted the problems faced by DP, where delegation down to middle-management has tended to be a myopic disaster. In particular the rules of the game have changed; cost displacement was the old yardstick, now technological innovation is much more about cost-avoidance, and this is much harder to quantify and predict.

The requirements for successful strategic innovation in this technological world are:

- Vision** A clear picture of where you want to be in (say) 15 years time. This is phrased in terms of 'what market will we be in', and 'who will our competitors be'. Forecasts are a total waste of effort on this timescale.
- Policy** A strong sense of business direction. This is emphatically not planning, which is inevitably based on the status quo. However there is an obvious conflict here: DP have the know-how but lack the authority; few directors (who obviously have the authority) have a strong enough grasp of the technology.
- Architecture** A blueprint for the evolution of an integrated resource over unknown volumes. Motorways generate traffic-flow, not the other way round!

Mobilization Getting people to go along with the (often unintended) rate of social change. The telephone took 40 years! The users of IT are no longer buffered by the DP department; whereas early DP only disrupted clerical staff, IT will disrupt everyone. What's more the customers of the future will have a choice (unlike the clerks).

To make it all work, we need a lot more 'hybrids'. We need people who are literate about computers, and who know about the business. We also need people who have a real feel for computers, but are literate about the business. Any Questions?!

The Tyne and Wear Metro

by David Howard (Director-General; Tyne and Wear Transport)

Background

In 1969 an integrated authority was set up to co-ordinate transport policy in Newcastle, South Shields and Sunderland. To begin with, this simply ran the buses, but it immediately commissioned an overall transport and land-use study covering an area roughly 20 miles in radius round all the major centres of population.

The emphasis was already beginning to switch from road-building to opening up an under-used railway system (which no longer delivered passengers to the right places); it was quickly becoming clear that all that road construction had done little to ease congestion in the city centres.

This general change in strategy was confirmed by a micro study into the 'North Tyne Loop' and its buses. The 'light railway' option came top of the list, showing an 11% rate of return over the next best alternative (an improved bus network). The idea of Metro was born.

Timetable

In 1972-3 Parliament gave outline approval to the entire scheme. The local authority insisted on being fully involved in the planning process, and began by visiting other metro developments overseas. By 1976-7 development was well under way, just in time for a total government moratorium on spending! As the speaker rather delicately put it:

"This does interesting things to your critical path network!"

In tandem with the Metro launch, the bus network was completely re-planned, with a number of strategically sited interchanges to channel traffic flows as efficiently as possible into the new system. Obviously there were extensive surveys, and much vigorous bashing of data, before all the details of this could be worked out.

Film

Basically this was a well constructed piece of propaganda. It did make the whole process look remarkably trouble-free, in a way which ardent followers of "Look North" well remember it wasn't! However if it helps Tyneside sell its expertise to a world market, then good luck to them.

Performance to Date

Typically, inner city public transport is carrying 20 – 25% fewer people than it was 10 years ago. In Tyne and Wear usage has actually grown by 10%, in other words the Metro has boosted the whole public transport system to some 50% above expectation.

This is definitely not a result of absurdly low fares! The key factors are:

- ★ the fully integrated network.
- ★ through ticketing
- ★ aggressive marketing.

The redeployment of buses to support Metro railheads has had other benefits; there are now fewer than 15 buses per hour crossing the Tyne Bridge – there used to be over 100!

As for the future, the Metro may grow some new branches (although the disbanding of the Tyne and Wear authority might confuse the issue), and the existing service will continue to be fine-tuned. One of the great advantages of automatic ticketing is that it allows continuous monitoring of usage, and (to pinch a quote from Peter Keen) it gives you that “focussed view of your customers”.

Comments from the Floor

As I remember it, there were relatively few genuine questions; what did happen was that a series of people stood up and gave unsolicited testimonials to Metro! Personally, I am planning a trip just north of Newcastle in the near future; I shall look forward with considerable anticipation to my first encounter.

Nissan – so Far

by Peter Wickens (Director of Personnel, Nissan UK)

“The Japanese don’t have a monopoly on good management”

Introduction

Peter Wickens has a long history of personnel management in the motor industry. His views on the Japanese experience, and the most effective way of applying their ideas in the UK, were lent a great deal of weight by the simple fact that he is the one who carries the can at Nissan UK. If anyone has a strong personal interest in getting it right, Peter Wickens does!

Working Practices

It is here that we have most to learn from the Japanese, but not everything they do will transfer automatically to Europe. We also need to be very careful in finding out what they really do; the Japanese are very protective of their skills, and academics often only hear and see a carefully circumscribed view! When in doubt talk to the foremen; as well as being the key figures in the Japanese system, they are far less likely to give you the party line.

A good example is the famous 3-minute exercise period. This is led by the foremen in groups of around 15, and has very little to do with physical well-being. The point is much more to build a feeling of teamwork, and to allow the foreman to observe the physical and mental well-being of his working group.

“In Ford we took pride in having no breaks between shifts. Not so at Nissan UK”

The Japanese Tripod

A key feature of all tripods is that they need all three legs if they are to stand up! The three legs of the Japanese tripod are:

- ★ Teamwork. A few minutes per day when the foreman and his group actually talk together. There is a world of difference between this and ‘briefing groups’ where coded messages are handed down from on high.
- ★ Quality consciousness. If you set up lots of inspection points, the first result is to engender an “Oh, it’ll be picked up” attitude. Often it isn’t. The Japanese take the opposite approach; if quality is suffering the job must be made easier. Quality is built in, not inspected in.
- ★ Flexibility. The team itself is responsible for cleaning up, painting the workstation, improving the process engineering and so on. Often they handle minor breakdowns on their own, and would always help with routine maintenance or major repairs. This all looks so obvious to the Japanese!

None of these elements will have any real effect unsupported. In the UK we hear lots of talk about quality circles; these are total bunk on their own! They fall naturally into place when every individual has a genuine interest in quality; they failed totally at Ford because nothing else was changed.

Job Structuring at Nissan

For a long time, the best people haven't gone into production because they don't enjoy being kicked around. At Nissan the supervision are at the same level as the professional engineers, and terms and conditions are common across all levels. Selecting the 25 key supervisors (from 3,500 applicants) took a long time, but was an essential base for the new approach.

The 'single union' deal with the AUEW is also unusual. Nissan rejected the idea of a 'no union' site, as it would have lead to several years of unnecessary antagonism; the 'one union' deal was the obvious alternative.

At Nissan UK there are just two job titles:

Technician
Manufacturing Staff

This is in remarkable contrast with Ford, where there are some 560 fully described jobs! Similarly there are only two sorts of engineer at Nissan; and there are no job descriptions or job grades. Everyone is on a salary range; progression is on merit, and there are no overlaps between ranges. The only concession to the British way of doing things is a gradual cutover from overtime payments to company cars, there are no other distinctions between process workers and white collar.

Questions from the Floor

Does Nissan have an OR Department?

"No. Maybe the OR philosophy does not sit comfortably with the Japanese 'consensus' style of problem solving"

What about bonus systems, share option schemes and so on?

"A definite 'no' to piecework on a day-to-day basis, but a probable 'yes' on the broader scale. This sort of thing is still several years down the road, and is a secondary consideration."

Can the same style of management be introduced anywhere other than on a green-field site (e.g. Fords)? What single change would you make?

“Top level management must strategically plan its employee relations. Where do they want to be in 10 years time? Is harmonization achievable/desirable/inevitable? Let them get a broad consensus with the unions and then start working in a sensible co-ordinated way, not the piecemeal haggling that goes on now. In 10 years time most companies will get there anyway, having paid all the costs and reaped none of the benefits.”

Can Nissan keep to their aims and objectives in times of stress as well as on the ‘upside’?

“The first thing is that all employees should have real information on where the company is. Understanding is always better than ignorance and suspicion. At Nissan there is an agreed 4 weeks notice of termination of the guaranteed week; in virtually all recent cases by the end of 4 weeks the layoff was no longer necessary”

How does the small working group affect job satisfaction?

“In general positively, although the very simplified production might work the other way. As long as you have flexibility, it matters much less that you go for highly automated plant.”

What about component supplies?

“Typically in the UK one supplier supplies all the manufacturers. The Japanese way is to have one supplier per manufacturer, often financially linked. This leads to a long-term relationship, and the manufacturer can insist on quality control and research and development by the suppliers. A pre-requisite of the ‘just in time’ philosophy is that there is no ‘goods inwards’ inspection.”

What about career development and ‘aspirations’ in the very flat job structure?

“The structure does cause problems, but a combination of broad salary ranges and rapid growth (phase-2 will treble output) means that it is a potential problem only! The sort that can be left until 1995”!

Postscript

Phase-2 of the Nissan development was still to be agreed when Peter Wickens gave his talk. It was formally announced early in November, and the scale of the operation exceeded all expectations.

GENERAL ARTICLES

This section of VECTOR is oriented towards readers who may neither know APL, nor be interested in learning it. However we hope you are curious about how, under the right conditions, such impressive results can emerge so quickly from APL programmers.

Introduction to General Articles

by David Preedy

This issue, we have three general articles covering a variety of topics.

Katie Williamson and Geoff Wells, of Imperial Group, bring us up-to-date on how to get the best out of the growing number of databases that are now available. Their article examines the growing range of database services and gives some useful technical advice to others who may want face the same problems.

Our second article is written by Martyn Adams of Metapraxis. Martyn explains why his experience in developing high-quality graphics systems has led him to reject the concept of G.K.S. as a suitable standard for developing such software.

Finally, we rejoin Anthony Camacho's series which indicates how APL provides *Steps towards a better BASIC*. This issue he explores the area of sorting.

External Databases

by Katie Williamson and Geoff Wells

Introduction

This paper describes the experience of Imperial Group Head Office in using external databases. A couple of years ago they were hardly used at all; they now form a key part of the operations of several departments. The purpose of this paper is the modest one of listing some of the main uses that we make of databases to ensure that VECTOR readers are aware of the new possibilities.

The explosive increase in the use of databases seems to have occurred for 3 main reasons:

- there are now many more databases available.
- the means of accessing them is much improved
- there is a general trend in companies to put more emphasis on the external world. This has important implications for management services in that the data one needs to handle is not now confined to internally generated material.

It is possible to access information on:

- several years' editions of newspapers
- company accounts
- company share prices
- Government statistics
- Legal case histories
- Abstracts of articles

How does one access the databases?

We have experienced several methods of physical access; firstly using specific hardware supplied by the database vendor and equipped with special function keys; secondly using a general terminal or micro such as the IBM PC, thirdly using a large micro to download data from the bureau prior to internal manipulation and fourthly taking data on a regular basis via a floppy disc.

We have found that some departments need access to several databases and the second of the alternatives is proving the most popular. A small menu has been written for the IBM PC so that the user merely indicates which database he needs and then the autodial modem connects him to the appropriate database. This is clearly less cumbersome than having many dedicated terminals and usually quite a bit cheaper. Traditionally dial up services have been available using V21 modems running at 300 baud (30 characters per second). Most bureaux now support V22 modems which run at 1200 baud (120 c.p.s.) and this represents a significant improvement when the volume of data transmission is at all significant.

One of the major problems with dialling into databases through the telephone network is the occurrence of noise leading to the possible loss or corruption of data. One method of avoiding this is to use an additional box called an error corrector (costing about £400) alongside the modem. This normally works well. The snag is that relatively few bureaux support this facility, and those that do have not adopted one standard error corrector. However our experience using the Tulsedata error corrector into the IP Sharp databases has been most encouraging. A further method of error avoidance is to use the PSS network of BT. This ensures error-free transmission once you are into their network (by dialling into the nearest PSS node). Errors can still happen between your terminal and the nearest node and we have found this to be a problem. A further problem has been the overloading of the PSS nodes such that one receives the engaged tone when dialling in. All of these problems have been overcome by the purchase of a PSS PAD (for around £3000). This piece of hardware is linked by private wire to the PSS network and gives instant access to the bureau with no transmission problems. This system gives most benefit for multiple users accessing multiple databases, but one does not need a heavy usage to justify the convenience of having a PAD. At one of our sites we have also been able to make use of our modern PABX and we have installed Dataplugs. These are cheaper than modems and enable the user to work on one phone extension – he is able to use this for voice while at the same time accessing databases through the Dataplug and the PAD PSS.

In brief, one is now able to go to a terminal and press one button and select the database of one's choice and get access (phone numbers, passwords etc. are stored in one's machine). This is all possible but it does need specialist knowledge to set up in view of the differing protocols of each service. Gone are the days of looking up phone numbers and pressing data buttons!

APL and Databases

Most of the Group Head Office access is done by non-technicians who are happy to take the data in whatever form the bureaux supplies it. Bureaux differ in their user friendliness but most of them are satisfactory for the regular user if a little daunting for the irregular user. Our main experience with APL has been with IP Sharp. This bureau is especially suitable for access by APL users. We access IP Sharp through our MicroAPL Spectrum and pull off the data we require. Getting hold of the data in a form where we can manipulate the numbers is very easy. We regularly pull off data on share prices of our own (and other)

companies and we also can access the government CSO Statistics (example enclosed) which are held for some series by UK region. We hope to use our graphical facility on this to produce maps of the UK shaded according to regional data. There is also a wealth of other databases (e.g. Singapore Stock Exchange prices!).

Which databases do we use?

Our most popular databases are

IP Sharp	– to extract share prices and economic data
Lexis	– heavy usage in our Legal Dept.
Textline	– to search newspaper articles
Datastream	– company accounts
Dialog	

We also make use of many other systems including Datastar, Microexstat, Dunnsprint, Jordans.

The cost structure of the bureaux vary widely as do the likely bills. With Textline one pays an annual fee and the cost therefore does not vary with usage. For others one pays as one goes either on a purely connect time or a combination of connect and CPU time. In general the costs are modest but it is possible to knock up very heavy bills when searching through some of the larger databases. Many organisations are now dialling up databases for only short periods to pull off data for local manipulation and bureaux are modifying their operations and charging structure as a consequence.

Getting started

More and more companies are now offering databases and their range is expanding all the time. It pays therefore to study the market, contact other users or join the On-line Information Centre (phone 01-430-2502). Whatever systems you use it is well worth addressing the issue of mechanics and communications early on in order to avoid the frustrations of poor lines.

We hope that this article has helped stimulate some interest in databases, whether you want to know the share price history of British Telecom, all newspaper references to your MP in the last year, or find articles on an obscure topic.

Database access is now very easy compared with a few years ago when you needed both the patience of Job and the skills of a telecommunications engineer.

P.S. Since writing this article, we have discovered some serious inaccuracies with the share price data from I.P. Sharp and as a result we understand the service is being withdrawn.



COCKING & DRURY LTD. THE APL PROFESSIONALS

16 BERKELEY STREET · LONDON · W1X 5AE

TELEX: 23152 MONREF G
TELEPHONE: (01) 493 6172

COURSES

VS APL Fundamentals	Feb 11 - 13	3 days	£375.00
VS APL Fundamentals	Feb 25 - 27	3 days	£375.00
VS APL Fundamentals	Mar 11 - 13	3 days	£375.00
VS APL Fundamentals	Apr 8 - 10	3 days	£375.00
VS APL Fundamentals	Apr 22 - 24	3 days	£375.00
VS APL Fundamentals	May 6 - 8	3 days	£375.00
VS APL Fundamentals	May 20 - 22	3 days	£375.00
VS APL Fundamentals	June 3 - 5	3 days	£375.00
VS APL Fundamentals	June 24 - 26	3 days	£375.00

All prices include course materials, refreshments and lunches, but exclude VAT

Discounts are available to companies making more than one booking at a time:

Number of bookings	Total Discount
2	£50 (ie £25 per course)
3	£90 (ie £30 per course)
4	£140 (ie £35 per course)

If you require courses on APL2 or other specialised topics we will be happy to provide them on your premises. In-house courses are also likely to be cost effective if you have 4 or more people to train at any one time. For further information, please ring Liz Swann on 01-493 6172

Why GKS is unsuitable for Business Graphics

by Martyn Adams

Abstract.

This article is aimed at anyone who writes business graphics software. It describes some of the failings of GKS for business applications and gives some hard earned advice on graphics design. It is not aimed at engineers or technical graphics designers whose problems of pictorial representation are slightly different from those working in the business community.

This article is based on research and development into business graphics generation and presentation. The problems encountered with GKS highlight an important design flaw in the use of GKS concepts for business graphics. The alternatives proposed are currently under development.

Introduction.

RESOLVE is a Corporate Control System written in APL. It runs on the IBM PC AT and many of its clones. One of the main features of the system is its ability to generate high quality graphics from a large multi-dimensional database. RESOLVE also needs to support a broad range of high quality graphics devices which, invariably, communicate by using different protocols.

This paper highlights some of the problems found in using GKS to produce business graphics on devices of differing architecture, and outlines proposals for an alternative approach to designing device drivers.

The Problem with GKS.

“The wonderful thing about Tiggers is,
Tiggers are wonderful things.”

Practically every sort of graphics device has its own graphics language. Some manufacturers are trying to standardise the way their devices talk and some have developed their own graphics languages, such as HPGL (Hewlett-Packard Graphics Language).

Unfortunately few graphics devices actually have exactly the same capabilities. Screens may be able to display 2, 4, 8, 16, 256 or any other number of colours simultaneously. Pen plotters certainly find it difficult to draw 256 colours from a palette of 4096, but they have far better resolution than screens. Screens can dynamically change colours being displayed but plotters find that impossible of course. Screens are usually raster scan devices whereas plotters are vector scanned. Aspect ratios are different for many devices, and so on.

GKS was developed as a standard method of describing graphics pictures. In theory one can write a graphics program to generate a GKS file and then use a device driver to read that GKS file and generate a graph, pie-chart or whatever, on any appropriate device. Of course, IBM has implemented their version of GKS on the IBM PC and called it VDI (Virtual Device Interface). VDI supports the PGS (Professional Graphics System), EGA (Enhanced Graphics Adapter) and the usual IBM colour/graphics adapter, all of which are graphics cards which slot into IBM PCs.

GKS for Business Graphics

GKS has one basic flaw. It is too simple. It enables you to define lines, polygons, piecharts, boxes and so forth but when you try to draw an oval on two different devices then the cracks start appearing. Ovals can appear upright on one device but sideways on another which has a different aspect ratio and perhaps no viewport facility. Similarly, some devices have fixed character sizes or, particularly in the case of screens, allow only multiples of the minimum character size. You can easily draw a box and easily write text to the device, but GKS doesn't let you draw a box that fits neatly around your text.

Plotters on the other hand can allow truly variable character sizes and can even allow slanted and rotated text which is difficult if not impossible for some screens. In fact some plotters even allow proportional character spacing! Try putting a GKS specified box around them! Laser printers further confuse the issue by introducing new attributes, such as changing typefaces, whilst other graphics printers have yet another set.

I suspect GKS has been adopted as the lowest level of compatibility which manufacturers are prepared to accept. It does not offer user-level compatibility, but instead gives them a marketing claim to standardisation. This may help them shift boxes, but does not help us create effective graphics. Perhaps a higher level of GKS would solve a lot of these problems – icon definition perhaps. But this is yet a long way off.

Defining the problem.

Why does GKS not resolve the problems of displaying pictorial images? The problem is not with the hardware or the software or even within GKS. The problem arises because GKS does not match the concepts used when drawing graphs. We do not draw a mass of lines and shading which happen to make up a graph. We draw a graph which happens to comprise a mass of lines and shading. GKS solves the wrong problem! When we draw a graph presenting a picture of cash flow or stock holdings, we are trying to pictorially represent a problem in a standard form so that a user can interpret it. It is the picture which conveys the

message and therefore it is the picture which is all important; not the colours and thicknesses of the lines drawn. It would be nice if the device draws the picture to the best of its ability though – the better the quality of the image then the clearer the message.

The purpose of a device driver should be to accept only the information it needs to display a specific picture. Then it should generate that picture using the most appropriate techniques available for the device for which it was written. A new data stream with an additional set of data may cause a screen driver with 256 colours to shade a histogram in a new special colour whereas a plotter driver with no new colours may adopt a new shading pattern instead.

These sorts of decision on data presentation have to be made specifically by the device driver concerned and cannot successfully be generalised. This approach also lets the driver optimise performance to the particular characteristics of the device it supports. A plotter driver may minimise the elapsed time taken to draw a graph by drawing each colour at a time – thus reducing the number of slow pen-changes mid-graph. Drawing overlaid bars on a chart is easy on a screen where each bar simply overwrites anything drawn earlier; it needs more careful planning on a plotter so as to ensure that the background bars are doctored to leave gaps for bars that will be drawn in front.

This does not mean that graphs should be different on each device. When you display a bar-chart on a screen and press the HARDCOPY button you should not watch the plotter draw a pie-chart. The graph formats should be designed so as to be recognisably the same graph on whatever device is chosen. It is up to the device driver, however, to understand that if the selected device can only display large characters, then extra space needs to be made for the titles, labels and headings before the core of the graph itself is drawn. The device driver may even make the decision to move headings and titles to another area on the display if it finds that it has not got enough room for them where they are normally drawn.

An alternative approach, apart from completely redefining GKS, would be to encode the IF statement within GKS files for the GKS device drivers to dynamically interpret. One might then program:

```
IF COLOUR 27 AVAILABLE THEN SET COLOUR TO 27 ELSE SET SHADING PATTERN TO COARSE  
DASHED MODE 3 AND LINE TYPE TO 11.
```

This would be interpreted by GKS device drivers which would take the correct action. Of course this can become very complex code if you want to determine global coordinates for a circle on a device with an horrendous aspect ratio. Then you decide to stick text in the middle of it and suddenly your GKS device driver becomes a command interpreter bigger than the application program which generated the graph in the first place.

Whilst this approach might work in theory, I think conditional coding in GKS files is a dead end approach to true device independence.

Solving the problem.

The earlier discussion has shown that a device driver needs to know about an image as a whole before it can start creating that image on a particular device. In this case we have to code separately each combination of image and device.

The contents of the file created by the graphics software and given to the device driver are very different from the GKS standard. GKS requires the specification of each polyline, polygon or text output that will constitute the finished chart. Our new drivers take as input a selection from the styles of chart available, the values of the data-series to be graphed, the descriptions of those series and the standard styles to be used for each one. The driver has to carry out all the stages of processing to convert this information into a picture.

This means that the driver can no longer be a small and relatively simple program for drawing lines and polygons like the GKS-compatible device drivers, but must be a large and fairly complex program which calculates optimum colour usage, character size, line thickness, text positioning, and so on.

In order to facilitate this we have to introduce a new concept which we have called Standard Styles. A Standard Style is the preferred combination of colour, line-type, marker-type and shading-pattern that should be used to represent a particular type of data. For instance, we may decide to represent budget data as standard 137, which is defined as green, using a dot-dash line pattern, with asterisks as markers and vertical, densely-packed shading for area-fill. This standard will apply across all charts generated by the system. Whenever we want to draw a specific chart on a specific device, the device driver has to match all the standards required by the data with the capabilities of the equipment; if the device cannot support enough colours for the chart, then compensating changes may have to be made to line-types, markers and fill-patterns to ensure that the data series are clearly distinguishable.

Even this is a simplification, as we have to handle issues such as the background colour. Standard Colour 0 is the same colour as the device background (white for normal paper, transparent for foils, black for screens, etc.); Standard Colour 1 is the opposite of 0 (black for white paper, white for screens). The other Standard Colours have two specifications (e.g. their RGB – Red Green Blue – and intensity); one is the ideal choice on a device with a black background, the other for a white background. It is up to a device driver to select from its palette the combination of colours which most closely match the requirements of the chart.

All of this requires yet another design exercise. You must first define exactly what your graphs should look like. Business users like to use standard formats, ones with which they are already familiar, so it is often quite easy to get a specification from them of the sort of graph layout they prefer. Your own input would be to try to clarify the messages the graph conveys by, for instance, removing spurious information. A graph that takes more than 5 seconds to understand is next to useless. Different messages may require different graph formats but there are (mercifully) few significantly different graph formats.

One of the benefits of this standardisation is that it helps managers to interpret charts more quickly and accurately. You are forced to offer a limited range of chart formats; but this lets you design each one carefully in order to make its point effectively.

For example, red is probably best avoided in a financial context. A manager who sees a red line immediately associates it with bad news and this may bias his interpretation of the graph. Budgets are always be presented in their standard style; so are actual data and forecasts. This means that a user viewing a long term forecast chart will not have to reorient himself to a new colour scheme when he looks at an historical chart. Incidentally, even senior managers can be afflicted with colour blindness so it makes sense to draw lines using different line-types as well as different colours.

After the initial work of specifying your standard colours, line types and so on, and then specifying your standard graph formats, writing the device drivers should be relatively easy. The graphics will be of better quality than GKS-generated ones and will probably be generated faster. The users will appreciate the ability to get good quality graphs on any supported device but will probably not appreciate the amount of work you put into it.

Disadvantages of this approach.

I have outlined why we are developing an alternative approach to GKS. It is, like GKS, a compromise and so has some associated disadvantages. There are three major problems:

- the device drivers will require considerably more code than under GKS; they will also have to be written by ourselves, rather than supplied by the hardware manufacturer;
- the range of equipment supported by any system will be restricted; in practice this is less of a problem as most organisations have standardised on certain types of kit;
- the range of types of chart supported by any system will be restricted; this is almost always good practice in any case as it prevents users be bombarded with totally new chart formats every time they use the system; however it would plainly be unacceptable in some other application areas.

This approach will introduce a degree of resistance to frequent requests to support new devices or make major changes to graph formats. You will have to change all of the device drivers for that graph in order to keep them in line. It is worth designing some flexibility into the device drivers so that they can accept options such as '2 to a page' or HEBREW character sets or whatever if the need arises. Fortunately the device drivers can be easily generalised with higher level languages such as APL.

Exploring other possibilities.

It is interesting to note that the intermediate files containing data for the device drivers actually contain the source data from the database. Eye-ball checking of the data for accuracy is possible. Line definitions and viewport specifications and palette settings are not present in that file since all of that low-level specification (that currently has to be specified outside GKS) is now generated in the device driver.

Later developments will probably include a Graph Definition Database (GDD) which all of the device drivers can interrogate when drawing their graphs to examine how the 'ideal' graph should look. The device drivers can then be generalised to accept many different forms of graphics pictures, all of which are maintained by the user on his GDD. As the data in the intermediate file is so close to the real source data then it should be possible to generalise the data extraction routines prior to graph generation.

Finally . . . I probably need to explain why I quoted from 'Winnie The Pooh' at the beginning. Just because GKS salesmen and engineers say it is a wonderful thing doesn't mean that GKS necessarily is a wonderful thing for business graphics programmers. In my view it has its uses but not in proper pictorial representation. It is a crude attempt at getting completely different devices to draw similarly, and for that it is by far the simplest and perhaps only alternative in a world where we have such different machines with different characteristics. The approach we are developing for defining device drivers is far less flexible than GKS in the short term and is unlikely ever to support as wide a range of devices as GKS. However it will provide faster, better quality pictures for our users and in the future may lead to generalised graphics displays from generalised data extracts. That has to be the way forward.

Steps toward a better BASIC

by Anthony Camacho

In this further extract from Anthony Camacho's series of articles aimed at introducing the delights of APL to BASIC users, the facilities for sorting in the two languages are compared.

This article is reproduced with the kind permission of DATALINK magazine.

A High Grade SORT of Language

Sorting in APL and BASIC

In Sinclair BASIC T\$(3) is the third character in the string T\$. In APL A[3] is the third item in the one dimensional array A, which could hold a string of letters or a row of numbers. APL can deal with as many dimensions as you like so A[1;3] is the third item in the first row of a table. If the index is omitted the symbols refer to the whole of the unspecified dimension. So A[;3] is the third value in every row and A[1;] is the whole of the first row.

Indexing parts of arrays like this is a powerful way of sorting them. If A contains: 5 7 1 2 9 8 6 3 4 then A[3] is the smallest item, A[4] is next, A[8] next and so on. APL allows a row of items in an index so A[3 4 8 9 1 7 2 6 5] will give a sorted version of A.

APL has a function called 'grade' which creates just such an index. The symbol looks like a triangle with a vertical line through it; if the point of the triangle is up it is called 'upgrade' and produces an index in ascending order; if the point of the triangle is down it is called 'downgrade' and the index is in descending order.

```

A←5 7 1 2 9 8 6 3 4
A
5 7 1 2 9 8 6 3 4
▲ ▲ IS UPGRADE ▼ ▼ IS DOWNGRADE
▲▲
3 4 8 9 1 7 2 6 5
▼▼
5 6 2 7 1 9 8 4 3
A[▲▲]
1 2 3 4 5 6 7 8 9
A[▼▼]
9 8 7 6 5 4 3 2 1

```

So much for the standard BASIC exercise 'Write a program to sort ten numbers into numerical order' – five symbols on one line! This approach to sorting is not only brief but fast too. On a run of the mill Z80 micro it took eleven seconds to sort a thousand random integers into order. My APL can hold numbers in IBM 64-bit double precision form; a thousand of these took 22 seconds to sort. The indexing method of sorting works equally well on letters, but to prepare the index the letters have to be converted to numerical form. The \square AVI function used below converts a character into the decimal value of its 8-bit code. This shows the unshuffling of an alphabet.

```

      B
TICKPUEMDVSRAGFXZNLOBHYJW
 $\square$ AVI B
14 22 3 9 7 16 15 23 2 25 4 20 8 19
      21 5 12 13 11 1 6 10 26 17 24
      18
      B[ $\square$ AVI B]
ABCDEF GHIJ KLMNOPQRST UVWXYZ

```

But of course usually it is not so much letters that need sorting as words. Some versions of APL have versions of upgrade and downgrade which will do this. They create an index to the rows in a two dimensional array of characters. They also allow you to specify the order in which the characters are to be sorted. On the left of the grade symbol is a character variable (the collating sequence) and on the right of it the table of characters to be indexed by rows.

The following is an elementary example. Note the ';' to pick up the whole of each row. Five rows is all there is room for here but the advantage really shows with large word lists. An array of four hundred five-letter words picked at random took 8 seconds!

```

      A A CONTAINS THE NORMAL
      COLLATING SEQUENCE
      B CONTAINS FIVE WORDS
      B
CHAIR
TABLE
SHELF
STOOL
BCASE
      B[A $\square$ B;]
BCASE
CHAIR
SHELF
STOOL
TABLE

```

One of the nice things about this is that you aren't restricted to putting the field you took the index from into order. You can sort the sales figures into product number order or the product numbers into order based on last months sales. If your file has large records and you can't hold them all in memory you only need space for the keys in memory and then you can extract a record at a time in graded order from the disk.

TECHNICAL SECTION

This section of VECTOR is aimed principally at those of our readers who already know APL. It will contain items to interest people with differing degrees of fluency in APL.

Experienced APL Consultants Required

Up to £16,000 + benefits

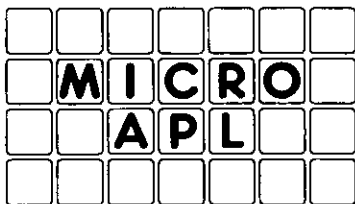
MicroAPL require APL consultants with a minimum of 3 years' experience, preferably including user-support, to work on project development and client training and support.

Applicants must be self-motivated and able to act on their own initiative. The work will be on a variety of projects, the emphasis being on city-based applications.

An excellent salary is offered, together with free health insurance, a pension scheme and 25 days' annual holiday.

If interested, please apply in writing, enclosing a C.V.,
to Denise Sullivan,

MicroAPL Limited, Unit 1F, Nine Elms Industrial Estate,
87 Kirtling Street, London SW8 5BP. Telephone 01-622 0395.



Consultancy with Confidence

Technical Editorial: More on APL2

by Jonathan Barman and David Ziemann

We are starting to get feedback from you on APL2. In this issue we are pleased to include two very different views – a report of the problems migrating to APL2 and a look at the joys of APL2 operators.

The most extensively used new feature of the language must surely be the 'each' operator, which loops through an array applying a function to each of its items. Of course, it's still early days, but already we're getting a feel for how 'each' is being used; in one sense it takes all the clever bits out of APL, as one user commented. APL often attracts those who enjoy being intellectually stimulated, and if there's a faint possibility that a problem can be solved without looping they will spend hours searching for an arcane parallel construct. With the new operator the looping solution can be coded directly by using a defined function with 'each'. In this way the thought processes of ordinary mortals can be implemented in the most obvious way. Even so, we expect that there will be plenty of amazing code written in APL2!

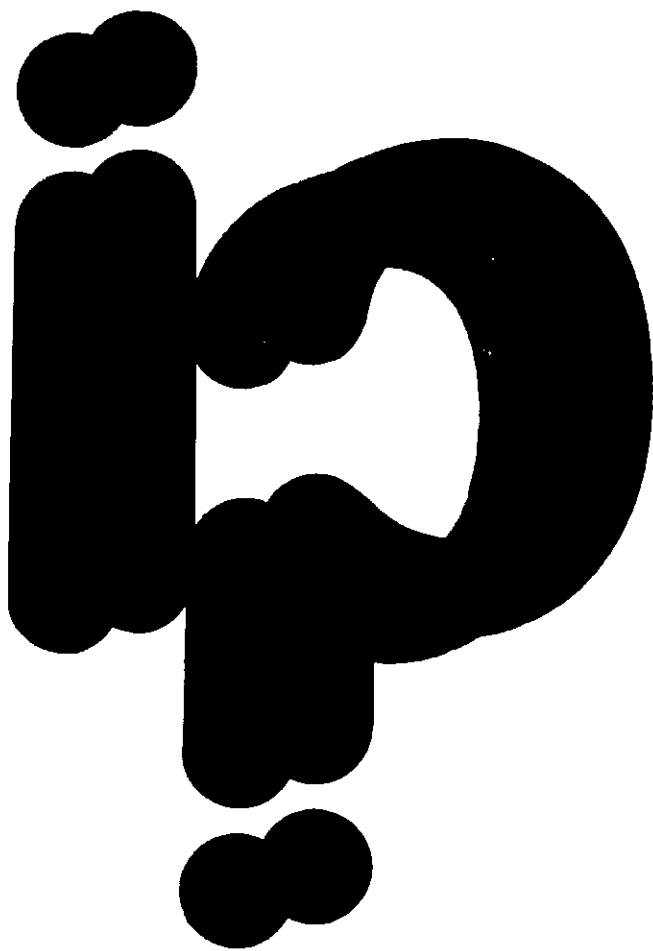
What about the question of APL2 CPU requirements? Has anyone made any studies yet? In general, nested arrays will need more CPU and memory, and the old problem of equating the cost of people with the cost of hardware resurfaces. Good programmers are getting more and more expensive, while the hardware (and software?) is getting relatively cheaper, but if a big new machine is required to run the language then careful evaluation is called for. The argument hasn't really changed –

The cost of running a system includes the cost of its development, and APL2 can be expected to show an overall improvement in many cases.

What of the future of APL2? There are some encouraging signs that links to the APL2 environment are attracting users of other programming languages to APL2. In particular, the ability to access ISPF service routines through a shared variable, and the SQL/DB2 database should cause users of those products to take a serious look at APL2. Also the 'name association' feature of the newly announced Release 2 of APL2 further increases the ability of APL2 to communicate with its environment. The quadNA system function allows you to treat an external program as if it were a defined function within the APL2 workspace – a very powerful feature. (That's how IBM provide the valuable 'partitioned-enclose' facility, sadly missing from the base product).

Another big boost for APL2 would be its availability on a personal computer, and it should now be possible to run APL2 under VM on an AT/370. If only IBM can keep the price of the software low enough this might be the route to the mass market we've all been waiting for.

We hope that VECTOR will become a useful forum for discussing APL2, so please share your views and discoveries with the other readers. If you've found any incompatibilities from VS APL, or bugs in APL2 we'd like to hear from you too.



***THE SOLUTION TO
YOUR APL QUESTIONS
on IBM PC or Mainframe.
INNER PRODUCT***

For further information, contact JAMES MANNING, Marketing Manager, Inner Product, Eagle House, 73 Clapham Common Southside, London SW4 9DG, or telephone 01-673-3354.

Technical Correspondence

This column features letters which are unlikely to interest readers who do not already know APL. Writers are requested to observe the requirements for the inclusion of APL code in VECTOR. All correspondence should be addressed to the Editor, who will forward technical letters to the technical editorial team.

From John Sullivan

30th August 1985

Sir: Surely there must be a better way? Why didn't you ask me this before!

Seriously, though, I enclose some code which, on the face of, seems to satisfy Andrew Wiggins' problem from Vol 2 No. 1. I'm sorry this doesn't match Greg Mateja's 12-1 comments/statements ratio but as you know we programmers hate writing documentation.

Yours faithfully,

John Sullivan,
Research Officer,
Statistics & Market Research Section,
Business Development Division,
National Westminster Bank PLC,
41 Lothbury, London, EC2P 2BP.

Editor: John's solution is published in 'Surely There Must Be a Better Way'. Never mind John, 10-1 isn't bad either.

From John Buckland

27th August 1985

Sir: I was browsing through VECTOR (July, 1985), enjoying my first issue as a new member, when I saw a chance to give my favourite function 'maximum scan' another airing.

You may like to offer the function REDUCE, of which I enclose a listing, to Andrew Wiggins. It seems to meet the needs of his accounting problem which is described on page 97 under the heading 'Surely There Must Be a Better Way'!

The function is just bare bones. I have omitted normal checks for empty and unequal vectors and so on, so as not to obscure its main point.

I hope this is of interest.

Yours Sincerely,

John Buckland,
Westwood,
9 Grove Road,
Camberely,
Surrey, GU15 2DN.

Editor: John's function is reproduced in 'Surely There Must Be a Better Way'.

From D Horton

22 October 1985

Sir: I enclose my solution to the competition 'Wrap Up'. I was interested in this problem as I had previously faced the analogous task of inserting page headings into a continuous block of text, certain lines of which must start on a new page.

I failed then and now to find a non-looping solution to the problem of not ending a page (line) halfway through a block of information (word). I shall await publication of the solutions with interest.

Yours faithfully,

D.J. Horton,
Budget Manager,
Pfizer Limited,
Sandwich,
Kent, CT13 9NJ.

From Phil Last

2nd September 1985

Sir: Sour grapes notwithstanding I should like to point out a few discrepancies in Dave Ziemann's competition result: Test Your Skill.

The winning entry, as printed, does not fulfill the problem as specified. Nor is it correctly described in the text below it. Nor is the text below it correct in itself.

The list of primes must be extended to include the tenth prime, viz. 29, in order to process 10 skills.

The description refers to the required number of primes being, in origin zero, one plus the highest skill number. The code actually contains one less than the highest skill number. The required number of primes is actually equal to the highest skill number, in origin zero, one or minus ninety-nine for that matter.

The use of origin zero does not provide a neat way of ignoring the zeros used for padding. It provides a normal way of avoiding either an index error or the addition of quad-IO to each array. The leading one in the vector of primes is a (fairly) neat way of ignoring the zeros used for padding. The indexing is a mapping of skills onto primes. Zero is not a skill, one is not a prime.

Sincerely,

Phil Last,
19 Stanley Road,
Lower Edmonton,
London N9.

Editor: Glad we've got that straight!

From Phil Last

2nd September 1985

Sir: I wrote the functions documented below several weeks ago following the production of an ad hoc fix to an unacceptably slow database indexing routine.

At the time they were envisaged as a stop-gap solution until we upgraded to VS APL release 4.

I then tried them out on a version of release 4 we had that someone had inadvertently left lying around and found that there was no improvement in the primitives my functions are designed to replace. Performance of these functions was still far better where it counts. Well perhaps APL2 would improve matters! Not according to 'INTERLINK' (Published by Interprocess Systems Inc.) where it is reported that all the relevant primitives are slower with the villains of this piece being among the worst.

The problem was caused by the poor performance of membership and dyadic iota on long numeric vectors.

We had a DUPSOUT function using the traditional algorithm:

```
((V1V)=1ρV)/V
```

which was very slow when the vector was large and contained many unique elements.

It seemed that a partition operation on the ordered vector could also do the job (possibly quicker) and hence this cumbersome looking DUPSOUT.

```
R←DUPSOUT V;L
R←(L×-1+(1+-1L),L+V[L])[ΔL+ΔV]/V
```

In short the negative not-equals difference of the sorted vector is re-sorted by the ranking vector and applied as a compression on the original.

The ensuing benefits are heavily dependent on the length of, and the distribution and frequency of the elements within, the argument. But for vectors of length greater than 300 elements with average frequency of less than two the new algorithm starts to show significant speed up. By the time length is 5000 with 2500 unique elements the new is taking 1 cpu second compared with 17 seconds for the old.

Taking the max scan of the product of the dupsout boolean with its index set produces a replication index vector to be applied after re-sorting to the upgrade vector.

```
R←DYOTA V;L
R←R[(\L×1ρL+L×-1+(1+-1L),L+V[R])[ΔR+ΔV]]
```

This result is identical with that of dyadic iota when both arguments are the same. To apply the same technique to differing arguments requires only catenation of the two, selection of that part of the result referring to the right argument and controlling the maximum value of the result.

```
R←V DIOTA A;L
R←(ρA)ρ(ΠIO+ρV)[R[(\L×1ρL+L×-1φL+L[R])[(ρV)+ΔR+ΔL+V, .A]]]
```

Again a membership function does the same but compares indices with the element count of the right argument returning that part referring to the left.

```
R←A MEMBER B;L;S;ΠIO
ΠIO←1
R←(ρA)ρS≥R[(\L×1ρL+L×-1+(1+-1L),L+L[R])[(S×/ρB)+ΔR+ΔL+(.B), .A]]
```

These last two functions show similar speed up characteristics to those of the DUPSOUT algorithm above. The primitives perform well when either argument is small or the left argument of dyadic iota or the right of membership contains many well dispersed duplicates, and most of the other argument is actually contained in the target.

Nevertheless the overhead of upgrade, drop, max scan and indexing can be counted in a few milliseconds while the speed advantage gained with less convenient data must be counted in seconds.

An aside: As coded DUPSOUT probably won't work on APL*PLUS/PC because of the re-assignment to L while the upgrade of L is still pending and the other functions will probably produce value error on R left of the index brackets as its value is assigned within them.

To conclude, the two functions DIOTA and MEMBER are either immensely superior in performance to the primitives (up to 12 times as fast in some cases, on IBM APL's at least) or are not significantly worse, depending on the size and structure of the arguments. No mention has been made so far but it should be fairly apparent that they will work only for simple numeric arrays. In VS/APL-3 a conditional branch after a test on data type to a line which merely runs the primitive adds 2 milliseconds to the execution of the code.

What I really can't understand is how I can run upgrade twice and still outstrip the machine code when the desired results could easily be (but manifestly aren't) temporary values thrown out of the upgrade itself.

And why IBM (and others) haven't done anything about this before?

Sincerely,

Phil Last,
19 Stanley Road,
Lower Edmonton,
London N9.

Editor: We hope the code has been reproduced correctly, lengthy lines of APL can be difficult to copy. Phil's comment about running grade-up twice and still coming out faster can be explained by the fact that the time taken for sorting is approximately linear; double the number of elements and the cpu time will be only a little more than double. Searching in the way dyadic iota is implemented will tend to increase with the square of the number of elements. These functions are good candidates for the 'Surely There Must Be a Better Way' column. Has anyone got a solution that eliminates the second grade-up?

MERCIA

Software Limited

*We would like to make a
few comments about our*

APL SOFTWARE AND SERVICES

FOR THE PC

- A** APL★PLUS/PC rel. 5.0 (upgrades for rel. 4 available)
 - A** STATGRAPHICS
 - A** TOOLS VOL 1 and VOL 2
 - A** FINANCIAL and STATISTICAL LIBRARY
-
-

FOR THE MAINFRAME

- A** STSC's INFO-CENTRE – Multi dimensional analysis
 - A** PARALLAX's EXECUCALC – 1-2-3 and VisiCalc
spreadsheet functionality
EXECUPLOT – Mainframe Graphics
-
-

SERVICES

- A** STSC's APL★PLUS timesharing service
 - A** APL Courses: INTRO TO APL★PLUS –
Mar. 18-20, May 13-15
APL★PLUS ENHANCEMENTS –
Feb. 20-21, April 9-10
 - A** APL Consultancy
-
-

FOR MORE DOCUMENTATION

★ Write, or call **MERCIA SOFTWARE** ★
Aston Science Park, Love Lane, Birmingham B7 4BJ
'phone 021-359 5096, telex 334535

Prize Competition Result: Wrap Up

by David Ziemann

The problem was to write a function `WRAP` which can fold delimited long lines as follows:

```

      ,S+ 'IN PARADISE',cr,'HERONS',cr,'APPROACH FROM THE LEFT.'
IN PARADISE
HERONS
APPROACH FROM THE LEFT.
      10 0 WRAP cr,S
IN PARADISE
E
HERONS
APPROACH F
ROM THE LE
FT.

      11 2 WRAP cr,S
IN PARADISE
HERONS
APPROACH FR
OM THE LE
FT.

```

The delimiter used to separate the lines is determined by the first character of the vector right argument. The left argument specifies the required width and the indentation for folded lines.

So what were the results like? One entry was disqualified because folded lines were themselves allowed to exceed the required width limit. Two solutions were disqualified for assuming that the delimiter was a carriage return character, one included a reference to the VS APL system variable `quadTC` (not standard-conforming) and one referred to a global variable `cr`.

This well commented solution was sent in by Heinz Reutersberg from Cologne in West Germany:

```

V OUT←N WRAPHR S;I;L;P;ΠIO
[1] A VECTOR 2 1 105: WRAP TEXT 1+S TO 1+N COLUMNS, USING DELIMITER
[2] A CHARACTER 1+S AND 1+N INDENTATION BLANKS FOR FOLDED LINES
[3] A (0<1+N), (0≤1+N), (0<-/N) ARE ASSUMED BUT NOT CHECKED
[4] ΠIO←0
[5] P←(S=1+S)/1P←,S
[6] L←1+(1+P,ρS)-P
[7] I←(1+N)+(-/N)×1f((I/L)-1+N)+-/N+2, N
[8] P←(,L°.>I)/,P°.+I
[9] OUT←1+(S, ' ')[ρS]L4(1P),((1+1+N)×ρP)ρP
[10] OUT[P+(1P)×1+1+N]+1+S
V

```

which produces the desired results. Zeke Hoskin from Vancouver in Canada extended the problem specification to optionally fold lines on word breaks, thereby simultaneously satisfying the competition requirements as well as Mark Bassett's Surely There Must be a Better Way problem from VECTOR 2.1. The word break option is specified as the third element of the left argument, which instructs the function to perform word breaking within that distance from the end of the line fragment. Here are some samples, using Zeke's own sample text:

```

V VEC=WIB WRAPZH VEC;P;BOO;DELIM;IIO;W;I;B;LEN;LONG;BRK;DEPTH;FOUND;MASK;MX
[1] A wrap character vector to specified width, indenting and breaking
[2] A preferentially at blanks
[3] AVEC=a character vector delimited by 1st character
[4] AWIB=width of output(not counting delimiters),indentation,blankdepth
[5] A where blankdepth = how far from right-hand end of line to seek a blank
[6] A Indentation and blankdepth default to 0
[7] IIO+1 I WB+3+,WB I W+WIB[1] I 1+WIB[2] I B+WIB[3] I DELIM+VEC[1]
[8] TOP: A each pass breaks a given line at most once
[9] P+BOO/I PBOO+VEC=DELIM
[10] Ameasure line lengths and quit if none is greater than W
[11] -(v/LONG+W<LEN+(I+P,I+PVEC)-P+1)+0
[12] Acreate matrix of beginnings of lines to be broken
[13] MX+VEC[(P+LONG/P)*. +IW+1]
[14] Abreak before last blank(that's why (v+1)th char included)
[15] BRK+P+W-(FOUND+DEPTH<B+1)*DEPTH+/\A\QMX' '
[16] Asince replicate is not in all APLs, construct expansion mask
[17] MASK+(PVEC)P0 I MASK[BRK]+1
[18] MASK+1,MASK\(((PBRK),I)P1),~FOUND
[19] MASK+{,MASK)/.1,((PVEC),I+1)P0
[20] VEC+MASK\VEC
[21] VEC[BRK+1+~1+0,+ \I+~FOUND]+DELIM
[22] ->TOP
V

```

```

VERSE+cr,'The beer was spilled',cr,'On the barroom floor'
VERSE+VERSE,cr,'And the joint was closed for the night'
VERSE

```

```

The beer was spilled
On the barroom floor
And the joint was closed for the night

```

```

12 3 WRAPZH VERSE A Indent by 3, no blank search

```

```

The beer was
  spilled
On the barro
  om floor
And the join
  t was clo
    sed for t
      he night

```



```
12 3 5 WRAPZH VERSE A Indent 3, break at blanks within 5 of limit
```

```
The beer was
  spilled
On the
  barroom
  floor
And the
  joint was
  closed
  for the
  night
```

```
12 0 5 WRAPZH VERSE A No indent, break at blanks within 5
```

```
The beer was
spilled
On the
barroom
floor
And the
joint was
closed for
the night
```

Notice that the function leaves a leading delimiter character on its result. The extra generality of this function means that it runs a lot slower than Heinz's when ignoring word breaks. By the way, the sample verse above wasn't the most esoteric; one solution contained a sample of what appears to be late 19th century romantic French poetry! (Thanks to D Horton for that).

Maurice Jordan provides us with the following APL*PLUS nested array solution:

```

V TV←N WRAPNAPS TV;LINES
[1] A Solution for APL*PLUS nested array production system
[2] TV←⊂TCNL TV
[3] LINES←(TV←⊂TCNL)⊂TV           A split into lines
[4] LINES←(⊂2+N) WRAPLINE" LINES  A wrap each line
[5] TV←1+⊃,/LINES                A Reassemble
v

V TV←N WRAPLINE TV;STARTS;LINES;INSERT
[1] A Wrap 1 line TV. N+2 integer vector (max length)(offset for new line)
[2] +((1+≡N)≥ρTV)/0 A No need to do processing
[3] STARTS←(ρTV)+1.((INSERT+1+2≡N)+0).(ρTV)ρ(--/N)+1 A Where to wrap
[4] LINES←STARTS⊂TV A Split into new lines
[5] LINES←LINES,"←INSERT+⊂TCNL A Append insert to each line
[6] TV←(-INSERT)+⊃,/LINES A Reassemble and drop last insert
v

```

Once you know what the symbols mean, it's a lot easier to understand than the standard APL solution, and was probably a lot easier to code too.

Prizes of thirty, ten and ten pounds respectively go to Heinz Reutersberg, Zeke Hoskin and D Horton.

Competition Update: Test your Skill

by David Ziemann

Maurice Jordan from British Airways has kindly sent in some timings for the skillmatch functions presented in VECTOR 2.1. Here are his results for Adrian Smith's looping solution <SM9>, Maurice's own APL*PLUS nested arrays production system answer <SMNAPS> and the winning entry <SM10>. By the way, as Phil Last has correctly pointed out, the function <SM10> requires the addition of a 29 to the vector of prime numbers if 10 skills are to be correctly processed.

```

      JOBS
1 2 3
7 9 0
1 4 0
1 3 0
1 6 3

      CONS
1 2 3 7 0 0
1 3 7 9 2 6
7 9 0 0 0 0
1 6 5 3 9 0

      [VVR'SM9'
      ▽ Z←JB SM9 CN;SK;LAB;CT
[1] SK←[/,JB
[2] Z←{(1↑ρJB),1↑ρCN)ρ1
[3] LOOP←LAB+1+(SKρLOOP),END,CT+1
[4] Z←Z^(JBV.=CT)°.A(CNΛ.*CT)
[5] END←LAB[CT+CT+1]
      ▽
      [VVR'SMNAPS'
      ▽ Z←J SMNAPS C
[1] Z←Λ/↑(↓J)°.e+ C,0
      ▽
10 TESTTIME 'Z+JOBS SM9 CONS'
4.5
10 TESTTIME 'Z+JOBS SMNAPS CONS'
2.3
      A But still the best
10 TESTTIME 'Z+JOBS SM10 CONS'
0.4
      [VVR'SM10'
      ▽ Z←J SM10 C;P;DIO
[1] DIO←0
[2] P←1 2 3 5 7 11 13 17 19 23
[3] Z←0=(*/P[J])°. [*/P[C]
      ▽

```

Prize Competition

My other Car's a Porsche

by David Ziemann

In the September issue of that excellent journal, the *Communications of the ACM* Jon Bentley presents a review of his round-up of 'Bumper-sticker' advice on computing. To give you a feel for what he received, here are some samples, firstly from the 'Coding' section:

"The sooner you start to code, the longer the program will take."

Roy Carlson University of Wisconsin

"If the code and comments disagree, then both are probably wrong."

Norm Schryer Bell Labs

"If you have too many special cases, you are doing it wrong."

Craig Zerouni Computer FX Ltd, London, England

"Don't get suckered in by the comments – they can be terribly misleading. Debug only the code."

Dave Storer Cedar Rapids, Iowa

Possibly of more interest to APLers are some of the 'Performance' rules:

"The fastest algorithm can frequently be replaced by one that is almost as fast and much easier to understand."

Douglas W. Jones, University of Iowa

"LISP programmers know the value of everything but the cost of nothing."

Alan Perlis, Yale University

... and for all you system developers out there:

"The first 90 percent of the code accounts for the first 90 percent of the development time. The remaining 10 percent of the code accounts for the other 90 percent of the development time."

Tom Cargill, Bell Labs

"Good judgement comes from experience, and experience comes from bad judgement."

Fred Brooks, University of North Carolina

"Whenever possible, steal code."

Tom Duff, Bell Labs

"I'd rather write programs to write programs than write programs."

Dick Sites, Digital Equipment Corporation

[Thompson's rule for first-time telescope makers]

"It is faster to make a four-inch mirror than a six-inch mirror, than to make a six-inch mirror."

Bill McKeeman, Wang Institute

Now these rules are no doubt useful in their generality, but naturally we would like to see some principles which are more pertinent to APLers. Which, by the way, doesn't just mean programmers, but implementors, users and managers (etc) as well.

So the competition this issue is to dream up some APL 'Bumper-sticker' advice. Of course it needn't be trivial – you should endeavour to crystallise those years of experience into a crisp principle or two. Whether it be at the level of the nitty-gritty symbols or a high-level view of APL systems design is up to you. It could be an aid to memory, a piece of sage advice, a warning, a rule or even a mere observation.

We want all your entries by April 30th, so start thinking and sending. As usual, fifty pounds in prize money will be distributed among not more than three winners at the discretion of the judges.

Surely there must be a Better Way

compiled by David Ziemann

In VECTOR 2.1 Andrew Wiggins presented a depreciation problem to which he had found only a looping solution. John Sullivan has sent us the following parallel solution (If you search hard you'll find the code):

```

V R←WIGGINS X;A
[1] A <X> is a 2×7 matrix, the first row is 'Income'
[2] A the second is 'Depreciation'
[3] A <R> is a 4×7 matrix containing 'Income', 'Depreciation' as revised
[4] A by the function, 'Profit' and 'Carried forward'
[5] A Function written in APL2, although it does not use anything outside
[6] A standard VS APL
[7] A Method:
[8] A Since the program doesn't work properly unless it starts with a
[9] A 'profit', we add a nominal profit of 0 to the start of the input.
[10] A Since we are taking profits whenever they occur, we calculate a
[11] A Profit/Loss vector (+\-/X), and save the amount of profit taken
[12] A so far (A+[\+\/X). If we subtract this lagged 1 period from itself
[13] A we obtain the profit taken in each period (A+A-0, 1+A). Now we can
[14] A revise the 'Depreciation' figures (X[1;]-A) and prepare the first
[15] A three lines of the table at the bottom of page 97.
[16] A R←X[1;],[1](X[1;]-A),[0.1]A+A-0, 1+A+[\+\/X+0,X
[17] A Last of all we calculate the 'Carried Forward' figures, and drop
[18] A the data we added to make the program work
[19] R← 0 1 +R.[1]+\X[2;]-R[2;]
V

```

Here is John's function in action:

```

ID
100 110 120 130 140 150 160 170 180 190
70 130 140 130 120 60 180 170 150 230
WIGGINS ID
100 110 120 130 140 150 160 170 180 190
70 110 120 130 140 80 160 170 170 190
30 0 0 0 0 70 0 0 10 0
0 20 40 40 20 0 20 20 0 40

```

Another John, this time John Buckland, furnished us with this:

```

V R←INC REDUCE DEPN;A;B
[1] A Reduces income <INC> by depreciation <DEPN> as required.
[2] A Returns <R> with two rows holding the amounts deducted
[3] A and carried forward in each period
[4] R←(2,ρB)ρR,B←+R+A-0, 1+A+A-[D](A++\INC)-B++\DEPN
V

```

```

      ID[1:] REDUCE ID[2:]
70 110 120 130 140 80 160 170 170 190
 0  20  40  40  20  0  20  20  0  40

```

Notice that this function has the advantage of being origin independent.

We trust that these solutions will satisfy Andrew, or any other interested readers, but hope that more suitable names will be chosen for the functions. A better name might be `⟨CARRYFORWARD⟩` for example, rather than `⟨DEPRECIATE⟩` say, because this name matches the generality of the function more closely. In this way the reader will be led to believe that the function has a more general usage than just in calculating depreciation.

Mark Bassett's plea in VECTOR 2.1 for the `⟨WORDWRAP⟩` function has been answered by Zeke Hoskin. His function is printed in the competition result page. Has anyone managed to solve this problem without looping? Maybe a good application for 'matrix-divide'?

Making effective use of GDDM and AP126

by David Piper

1. Abstract.

Full screen management is in many ways the natural way for APL to communicate with the user. The screen looks like a window onto an APL array, why not use it as such? In the IBM world, full screen management is limited to the use of auxiliary processors (APs), usually the Graphical Data Display Manager (GDDM) AP – AP126.

In order to implement efficient, easily used systems, screen dialogue must be managed as efficiently as possible in order to provide adequate response times. This article shows that functions provided by GDDM can be easily used to improve the efficiency of screen dialogues. A future article will discuss the use of functions provided by GDDM R4.0 and by AP126 itself.

2. Introduction – Facilities Discussed.

The facilities discussed fall into two basic categories:

- Those integral to GDDM.
- Additional facilities provided by AP126.

The facilities provided by GDDM are available to any program using GDDM (even those not written in APL), the power of APL makes these especially easy to use. The extra functions provided by AP126 are unique to the APL environment. This article discusses those functions provided by GDDM; it is hoped another article will describe the use of the facilities provided by AP126, and additional features of GDDM (R 3.0 and 4.0) which are of use in the development of complex dialogues.

2.1 Facilities provided by GDDM.

The following facilities provided by GDDM are not commonly used, and will be the subject of this discussion:

- Use of logical pages (all GDDM releases)
- Logical pages larger than the terminal (GDDM R3.0 and higher)

The discussion applies specifically to VS APL R4.0, running under MVS/SP and TSO/E, but should apply to other operating systems too, since no file handling is involved. 2.2 Facilities provided by AP126.

2.2 Facilities provided by AP126

The additional facilities available using AP126 are referred to in the VS APL Terminal Users Guide as 'AP126 Service Requests'. There are 10 of these, given call values from -1 to -10, the majority of these are not of particular use in application development. The most useful services are mentioned here for the sake of completeness, but are not discussed in detail in this article.

Service -8 (Query subset of fields for modification), allows an application to read data from the screen as required by the application, rather than all at once. More importantly, it also allows only modified fields to be read rather than all fields on the screen.

Service -9 (Query current hardcopy destination), allows an application to check for a currently open printer. A printer may therefore be left 'open' between print requests, avoiding the large overhead of being 'opened' and 'closed' repeatedly.

Service -10 provides a useful method for implementing graphics in a system. This service request allows the Interactive Chart Utility (ICU) to be invoked from APL, using APL vectors or arrays of numbers as the data to be charted.

3. Logical Pages.

The usual way of organising a full screen dialogue can be summarised as follows:

DIALOGUE:

- DISPLAY a panel
- READ the panel
- VALIDATE the data
- PROCESS the data
- REPEAT

This concept seems perfectly adequate for managing a screen dialogue, until the overheads involved in the DISPLAY process are considered. For the sake of clarity, the DISPLAY process can be broken down further:

DISPLAY:

- VALIDATE the format array
- FORMAT the screen buffer
- WRITE fixed text (prompts etc)
- WRITE dynamic text (date, time etc)
- WRITE default input values
- TRANSMIT panel to screen

If the panel format could be stored in some way, so that it only has to be re-sent to the terminal, the first three processes could be omitted from the DISPLAY process (after the first DISPLAY). The use of logical pages allows precisely this. Each screen format, with its fixed text, can be placed on a separate page, and restored to the terminal as required. The new DISPLAY process could be summarised as follows:

DISPLAY:

- GET page
- WRITE dynamic text
- WRITE default input values
- TRANSMIT panel to screen

The GET process either creates and formats a new page, or re-selects an existing page as required.

GET:

- IF page EXISTS
- SELECT page
- ELSE
- CREATE page
- VALIDATE the format array
- FORMAT the screen buffer
- WRITE fixed text (prompts etc)

3.1 Managing Logical Pages.

The use of logical pages does introduce the problem of management of the pages. This problem can be broken into two parts:

- Selection of pages for display of new panels.
- Re-selection of existing pages to re-display panels.

The first problem is one of selecting a page not previously used, the second is one of selecting the correct page from a list of existing pages. The examples that follow assume that a share with AP126 has already been established, using the variables:

- CTLs – the AP126 control variable.
- DATs – the data variable associated with CTL126.

3.2 Creating New Logical Pages.

GDDM has a function which returns a page number not currently in use – FSQUPG (FS – Full Screen, Q – Query, UPG – Unique PaGe. AP126 Code 307). This function returns the highest possible unused page number. The first call returns page number $(-1)+2*31$, the highest 4 byte integer.

Note: In early versions of release 3.0 of GDDM, the FSQUPG function returns LOW page numbers first, there is a PTF to correct this error.

A major benefit of using the FSQUPG function, is that no matter many pairs of variables are shared with AP126, unique page numbers are always returned. Thus several applications (systems, utilities etc), all using different sets of shared variables, can run in the same workspace without risk of using pages already acquired by other facilities. This principle extends to using GDDM with the APL Session Manager, which uses page number 1001.

Once the unique page number has been found, the page must be created. This is performed using the FSPCRT function (P – Page, CRT – CReaTe. AP126 code 302). The arguments to the FSPCRT function are the page number, the depth and width of the page, and the page type. This last parameter is ignored by GDDM release 4.0, but is required for compatibility with earlier releases of GDDM. A function to query for and create a page with a new page number (CNPAGE) is listed in appendix A.

The act of creating a page causes the new page to become current, so the panel format/write activities can proceed normally. Once complete, the panel can be displayed and read in the normal manner, all screen I/O functions addressing the current GDDM page.

3.3 Re-displaying Existing Pages.

Once a page/panel couple has been formatted and displayed, the only activity that has to occur before re-display of the page is changing the current page number to the one containing the required panel. To do this effectively, a method of relating panel identities (names, numbers or some other ID) to GDDM page numbers must be used.

The IBM supplied workspace FSC126 allows panels to be designed, each panel having a name recognised by all the utility functions in the workspace. The names can easily be used to build a table relating panel names to page numbers. When a panel is displayed for the first time (ie its page is created), the panel name is appended to the bottom of the table, and the page number is appended to the vector of page numbers. The index of the name in the table can then be used to reference the page number vector. Once the page number has been looked up, the page can be reselected by using the FSPSEL function (P – Page, SEL – SElect. AP126 code 305). The FSPSEL function takes the page number to be selected as its only argument.

A function to perform these actions is shown in appendix A (CSPAGE), the function either selects an existing page, or creates a new page if required. This function could be combined with IBM's DISPLAY function (from FSC126) if the new page was to be formatted. An example of this is shown in the HOWTO function in appendix A. Note, the code in the REDISP function is specific to the use of the utilities provided by the FSC126 workspace. This function would have to be changed if different screen management utilities were to be used.

It must be ensured that the panel name table and the page vector are nullified whenever GDDM is re-initialised. The easiest way to ensure this is to reshape the table and vector into nulls in the function run as the latent expression in the workspace.

3.4 Integrating with Existing Systems.

The above discussion shows that it is relatively easy to implement paging techniques within an existing screen management system such as that provided in the FSC126 workspace. The only change required is to ensure the correct page is selected before any screen I/O processing occurs.

In most cases, it will be possible to combine functions using logical paging with functions that do not in the same workspace. This is because, by default, page 0 is used. Functions using paging **MUST** ensure that the default page (page 0) is re-selected after their processing is complete.

4. Large GDDM Logical Pages

From release 3.0 onwards, GDDM allows the use of pages which are longer than the physical terminal. Pages wider than the physical terminal are not supported. Using long pages allows vertical scrolling to be performed very easily. GDDM only requires the line to be placed at the top of the terminal to be passed as an argument to the FSPWIN function (P – Page WIN – WINDOW. AP126 code 309). If the bottom or top of the logical page would appear in the middle of the physical page, GDDM adjusts the position so that the logical top/bottom coincides with the physical top/bottom.

The use of long pages reduces the need to re-write text to the page every time a scroll operation is performed. The AP need only be invoked to perform the scrolling activity, not to perform I/O to the logical page. If some of the text has to be refreshed on the page then the correct GDDM functions must be invoked to perform this.

4.1 Using Large Logical Pages

Large pages will be of most use for browsing APL objects, rather than for activities requiring text to be refreshed as lines are inserted, deleted or changed by the user. Data being browsed does not need to be refreshed. Even so, if a very simple editor is being used, which does not allow lines to be deleted or inserted from the text, the use of a large page will make the editor more efficient by removing the need to refresh text on the page.

Using screen sized pages, a BROWSE function can be summarised as follows:

BROWSE:

- GET browse panel
- WRITE a panel full of text
- READ the panel
- SCROLL text

The SCROLL function is changed by the use of long pages:

- | | |
|----------------------------|--------------------------|
| SCROLL: (normal page size) | SCROLL: (long pages) |
| • CALCULATE new top line | • CALCULATE new top line |
| • WRITE new panel of text | • POSITION window |

When using long pages, all the text is written to the panel during the initialisation stage, not just one panel full. This increase in panel I/O is offset by reductions during scrolling activity.

4.2 Horizontal Scrolling

Logical pages wider than the physical screen size are not supported by GDDM. The limitations imposed by this restriction can be obviated by allowing horizontal scrolling to occur only in units of one screen width. The BROWSE function must create sufficient pages to contain the width of the text being browsed. Scrolling horizontally then simply becomes a matter of selecting the correct logical page. This is a more efficient process than continually performing I/O operations to the same logical page.

The 'page mode' scrolling can also be used for vertical scrolling in releases of GDDM not supporting large logical pages. An ideal application for this type of browse function is to preview a paginated report. Whether all pages should be formatted in advance is application dependent and will typically depend on the volume of text being browsed.

For a 'page mode' browse function allowing both vertical and horizontal scrolling, the SCROLL function becomes:

- BROWSE:
- CALCULATE new top line
 - CALCULATE new horizontal page
 - GET page
 - POSITION window

The GET function is described in section 3.

5. Conclusions.

The use of logical pages can make a considerable difference to the speed of dialogues in systems using GDDM for screen management. In addition, the use of logical pages is quite straightforward to implement within a set of existing dialogue management utilities.

By removing the need to re-format a panel every time it is to be displayed, response times can be improved by as much as 50%. Logical pages can also be of use elsewhere in full-screen dialogues, for example when implementing 'browse' or 'edit' utilities, enabling scrolling to be carried out extremely rapidly.

FSCMD

```

V RC+DAT ΔFSCMD CTL
[1] A Version: 29.11.85      FN: Communicate with APL26 shared vars
[2] A DAT - Character vector  Value for DAT shared variable
[3] A CTL - Numeric vector    Value for CTL shared variable
[4] A RC - Numeric vector     Value returned by CTL shared variable
[5] DATs+DAT
[6] CTLs+CTL
[7] RC+1+CTLs
V

```

CNPAGE

```

V PAGE+CNPAGE;FSPCRT;FSQUPG;RC
[1] A Version: 29.11.85      FN: Query for and create new page
[2] A PAGE - Numeric vector   Page number or null if error
[3] A Assign GDDM function codes
[4] FSQUPG+307
[5] FSPCRT+302
[6] A Query for unused page number
[7] RC+' ' ΔFSCMD FSQUPG
[8] A Store the page number returned
[9] PAGE+1+4+RC
[10] A Create the page with default size.
[11] RC+(2+RC).2+' ' ΔFSCMD FSPCRT.PAGE, 0 0 1
[12] A Return page number if all codes from GDDM/AP126 are zero
[13] PAGE+(A/0=RC)/PAGE
V

```

CSPAGE

```

V RC+CSPAGE PANEL;FSPSEL;MAX;PAGE
[1] A Version: 29.11.85      FN: Select/Create page based on PANEL
[2] A PANEL - Character vector Panel name to be created/selected
[3] A RC - Numeric vector     Return code (0=Fail 1=New page 2=Old page)
[4] A ptab - Global table of panel names
[5] A pvec - Global vector of page numbers
[6] MAX+(pPANEL)I l+ptab
[7] A Check if PANEL is already in <ptab>
[8] PAGE+((((l+ptab).MAX)+ptab)A.=MAX+PANEL)/pvec
[9] +(0=pPAGE)/pc
[10] A Page number non zero. select it. Assign GDDM command code.
[11] FSPSEL+305
[12] A Page exists, reselect and return code=2
[13] RC+2*A/0=2+' ' ΔFSCMD FSPSEL.PAGE
[14] A Run dialogue manager specific code to re-init global vars etc.
[15] REDISP PANEL
[16] +0
[17] A No page number returned, create page. Exit if error
[18] pc:PAGE+CNPAGE
[19] A Return 0 if no page number returned, or 1 to indicate new page.
[20] +0 IF~RC+0≠pPAGE
[21] A Update <pvec> and <ptab>, and exit
[22] pvec+pvec.PAGE
[23] ptab+(((l+ptab).MAX)+ptab),[1]MAX+PANEL
V

```

REDISP

```

V REDISP PANEL:LL
[1]  A Version: 09.12.85      FN: Reset globals used by FSC126 utilities
[2]  A PANEL - Character vector   Panel name
[3]  A Flag no fields as modified
[4]  rmod+10
[5]  A Check for lable array for this panel
[6]  +ex 1F 2*[INC 'LL',,PANEL
[7]  A Set up label globals (eXPAND and cN are utilities from FSC126)
[8]  LL+eXPANDs'LL',,PANEL
[9]  nlabel+cN label+((1+pLL),6)+LL
[10] +0
[11] A Expunge label global, to indicate no labels found
[12] LL+□EX 'label'
V

```

HOWTO

```

V RC+HOWTO;INT;PANEL
[1]  A Version: 29.11.85      FN: Illustrate combined PACING/FSC126 utilities
[2]  A RC - Numeric vector      0 if ok, 1 if error in execution.
[3]  A Initialise return code, assume all ok
[4]  RC+0
[5]  A Set up current panel name
[6]  PANEL+'MYΔPANEL'
[7]  A Create/Select page (Note: GDDM must already be initialised)
[8]  +(er,di,rd)IF 0 1 2 =CSPACE PANEL
[9]  A Now use the FSC126 utilities as normal
[10] di:DISPLAY PANEL
[11] A Show screen and wait for ENTER or PF Key
[12] rd:INT+SREA
[13] A ...
[14] A More processing ..... (SWRI, SCET, exit on PF 3 etc)
[15] A ...
[16] A Reselect the page in case we changed to another panel
[17] +(er,di,rd)IF 0 1 2 =CSPACE PANEL
[18] A Re-read the screen
[19] +rd
[20] er:□+'Failed to display panel ',PANEL
[21] RC+1
V

```



H.M.W. PROGRAMMING CONSULTANTS LTD

Why not discover more about

- ★ **Consultancy/Support Service**
- ★ **APL on the IBM PC**
- ★ **VM/CMS Packages**

Ring Ken Jackson at

H.M.W. PROGRAMMING CONSULTANTS LTD

142 FELTHAM HILL ROAD,
ASHFORD, MIDDLESEX TW15 1HN
Telephone: Ashford 41232

A Guide To Using Operators In APL2

by Norman Thomson

Nested arrays are the feature of APL2 which seems to generate most enthusiasm amongst its devotees, as witness for example the twenty or so pages devoted to that topic in Vector Vol 2 no.1. On the other hand operator extensions seems to receive relatively scant attention in spite of the fact that they provide at least as great an advance in linguistic power. There are two broad aspects of operator extension – first the provision of user-defined operators, and secondly the extension of existing operators. These two features, both separately and together increase by a huge factor the expressiveness of APL in describing programming ideas with succinctness and clarity.

In general if APL objects are the ‘nouns’ of the language and functions the ‘verbs’, then operators are the ‘adverbs’. They direct *how* to apply or combine functions in manners which are common across a range of functions. In APL1 the adverbial aspect of functions is achieved by ‘embroidering’ the function symbols with other symbols such as / and . . For example, +/V describes *how* to add the elements of the vector V, i.e. how to add ‘through’ V, in the sense of inserting the function “+” into all the available spaces (1 less than the number of elements in V) and evaluating the resulting expression. After defining reduction to describe the adverbial concept of ‘through’, it makes sense to talk about ‘multiplying through’, ‘dividing through’, and so on. It is logically perfectly possible to apply reduction to non-scalar dyadic functions, so that one can ‘reshape through’ and ‘join through’ vectors:

```
ρ/3 2 4 ↔ 3ρ2ρ4 ↔ 4 4 4
,/'NOT' 'WITH' 'STANDING'
↔ 'NOTWITHSTANDING'
```

Such extensions provide in total a large amount of extra function in APL2. They are however conceptually straightforward, and I shall concentrate here on some of the subtler aspects of operator extension, which it will transpire are closely interwoven with enclosure and disclosure and thus with nested arrays.

User-Defined Operators

It is well known that in the case of non-commutative functions such as subtract and divide the effect of ‘through’ is to produce respectively the alternating sums and products. To many APL learners this result is counter-intuitive in that they feel that -/6 2 3 should mean 6-2-3, i.e. 1 rather than 7. This is indeed a reasonable variant of subtraction – call it subtraction ‘along’ a vector. The following lines describe the process for a general scalar dyadic function F, which could be either primitive or user-defined:

```
Z←+R
LO:→(0=ρR+1+R)/O
Z←Z F+R
→LO
```


They define a new function called the 'derived function' with R as its right argument. We can generalise the process by specifying an axis G, and at the same time giving a name 'along' to what we now recognise as a new operator, with F and G as its two *operands*.

```

      ∇Z+(F ALONG G)R
[1]  Z+[G]R
[2]  L0:→(0∈ρR+1+[G]R)/0
[3]  Z+Z F 1+[G]R
[4]  →L0
      ∇
    
```

Some applications are:

```

      (+ALONG 1) \ 4
10
      (-ALONG 1) \ 4
-8
      (-ALONG 2) 2 3p \ 6
-4 -7
      +(+ALONG 1) \ 4
24
    
```

(note that index-origin will be taken as 1 throughout.)

Now consider the + outer product of a vector V with itself. We define an operator directing *how* to add V in this context, i.e. by separately adding *each* of its elements with *all* the elements. The 'each with all' clause is an embroidery which could equally well be applied to a general scalar dyadic function F, and is defined by

```

      ∇Z+(F TABLE)R
[1]  Z+R◦.F R
      ∇
    
```

Thus

```

xTABLE \ 2
    
```

is the ordinary school multiplication table.

Commuting function arguments provides another opportunity to embellish a range of functions with a common adverbial embroidery.

```

      ∇Z+L(F CON)R
[1]  Z+R F L
      ∇
2 *CON \ 4 ↔ 1 4 9 16
    
```

What is common to the operators ALONG, TABLE, and COM is that they are applicable across a *range* of functions – if the function variation is really relevant to only one function, there is no need to traffic with operators – user-defined functions can be used in the usual way.

Of the operands F and G, either or both may be functions or arrays, but for meaningful uses at least one should be a function. The derived function then takes arguments L and R. If G and R are both arrays as in the case of the function ALONG then brackets are necessary to show where G stops and R begins.

If the operator is monadic, then the single operand appears to the left, thus reversing the rule for arguments. This is because non-ambiguity of syntax demands that operators follow the reverse rule for functions, that is they are *read* from right to left and *executed* from left to right. Thus in the expression

`(-COM ALONG 1),4`

the operator COM is executed before the operator ALONG. More specifically the derived function (-COM) is constructed and passed as an operand to ALONG to obtain the derived function (-COM) ALONG 1. Applying this derived function to the right argument, the successive execution steps are:

1(-COM)2 = 1
 1(-COM)3 = 2
 2(-COM)4 = 2

giving 2 as the final answer.

Further examples of situations in which user-defined operators suggest themselves naturally are given in three conference papers, my own on 'Cultivating Trees' in APL84, and those on 'Operators for Program Control' and 'Operators for Recursion' by E. Eusebi in APL85.

Extension Of APL1 Operators

The standard APL1 operators can be used to demonstrate further that the execution rule for operators is the opposite to that for functions. As in the previous article, I proceed by giving basic exercises, which you may like to work out by covering up the answers before reading them and the expository notes which follow. As before I shall use the function DR:

`DR: (≡R),ρR`

to obtain the depth-followed-by-rho of an APL object.

What are:

$$(A) \quad 1 \ 2 \ +.x.- \ 3 \ 4$$

$$(B) \quad 1 \ 2 \ +.(x.-) \ 3 \ 4$$

It may be useful to enunciate the APL1 rule for an inner product F.G, viz. first perform G on an element by element basis for the last dimension of the left argument, and the first dimension of the right argument, then carry out F reduction. (Think of +.x and the 'row-into-column' rule for algebraic matrix multiplication). If the arguments of an inner product are scalars then the function F plays no part, i.e. F.G is equivalent to G.

In APL2 this rule has subtly changed – it is now :

For those arguments which are *not* scalars enclose elements along the last axis of the left argument and the first axis of the right argument, then evaluate the 'F reduction each' of the G outer product, enclosing every element.

Just as inner product evaluation begins with enclosure, so outer product begins with *disclosure* of the elements, followed by enclosure of the separate function evaluations.

In (A) the leftmost inner product is executed first, and so the final derived function is (+.x).- . The APL1 rules would suggest an answer

$$4 \quad (+.x)/\ ^{-2} \ ^{-2}$$

The APL2 rule for inner product first obtains the outer product

$$\begin{matrix} (-1 \ 2) \circ. \ ^{-3} \ 4 \\ ^{-2} \ ^{-2} \end{matrix} \quad DR = 2$$

Notice that there is no initial enclosure because both elements are scalars. Evaluating the outer product however involves an initial disclosure, and a final enclosure which accounts for the depth of 2. '+.x reduction each' then leads to the answer 4, and so the APL2 rule follows the old rule in this case – the subtle but important difference being that an enclosure and an 'each' have become involved, even although neither was present in the original expression.

In (B), following the APL1 rules, (x.-) should be equivalent to -, leading to a final result 1 2 +.- 3 4, namely -4. The APL2 rules, however, lead to an initial evaluation of the outer product

$$4 \quad (-1 \ 2) \cdot (x, -) \leftarrow 3 \ 4 \quad DR = 0$$

The '+.x reduction each' is then a null function and the final result is +4, which is *not* the same as the APL1 rules would have us expect.

Now introduce nesting. What are:

$$\begin{aligned} (C) \quad & (-1 \ 2) \times \leftarrow 3 \ 4 \\ (D) \quad & (-1 \ 2) +.x \leftarrow 3 \ 4 \\ (E) \quad & (-1 \ 2) \cdot .x \leftarrow 3 \ 4 \\ (F) \quad & +.x / (-1 \ 2), \leftarrow 3 \ 4 \end{aligned}$$

The answers are:

$$\begin{aligned} (C) \quad & 3 \ 8 \quad DR = 2 \\ (D) \quad & 3 \ 8 \quad DR = 3 \\ (E) \quad & 3 \ 8 \quad DR = 2 \\ (F) \quad & 11 \quad DR = 0 \end{aligned}$$

(C) illustrates the pervasiveness of the scalar dyadic functions, i.e. element by element multiplication takes place through the 'barrier' of enclosure. We shall assume henceforward that the concept of pervasiveness is well known and understood.

The answers (D) to (F) may seem surprising at first.

(i) Why is (D) depth 3 and not 2? APL1 intuition suggests that since both arguments are scalars the + should be inoperative and the result the same as (C). However the APL2 rule for outer product results in a singly enclosed vector as per example (E), while that for the inner product shows that each element of the final result is the *enclosure* of the + reduction - a consideration necessary to guarantee scalarness of elements in cases where the function is not scalar dyadic.

(ii) The non-equivalence of (D) and (F) violates the APL1 rule for reduction of a vector, that is obtain the F/ by placing the function F in all the available (possibly zero) spaces, and evaluating the result. This is because ensuring that the reduction operator always *reduces* rank requires a subtle variation for APL2, namely the '*F each*' should be inserted between adjacent items, and the final evaluation enclosed. Since '*each*' has no effect when no nesting is present this rule completely subsumes the APL1 rule.

Now add the 'each' operator explicitly into the mix. What are:

- (G) (←1 2) ×" ←3 4
 (H) (←1 2) +. ×" ←3 4
 (I) (←1 2) *. ×" ←3 4
 (J) (←1 2) +. (×") ←3 4

The answers are:

- (G) 3 8 DR = 2
 (H) 11 DR = 0
 (I) 3 4 DR = 2
 6 8
 (J) 3 8 DR = 3

The 'each' in (G) forces disclosure of the two arguments, with a final enclosure after multiplying.

(G), (H) and (I) demonstrate the 'shell-penetration' role of each. Comparison of (G) and (H) demonstrates how the implicit 'each' in the definition of inner product achieves an extra level of 'shell-penetration'. Note the equivalence of (H) and (F).

Comparison of the results of (E) and (I) is also revealing. (E) is a straightforward multiplication table for the two scalar arguments. At first sight the 'each' in (I) ought not to make any difference since both arguments are scalars. This is however a circumstance in which the apparent simplicity of the name 'each' can be deceptive. The rule for 'each' must be strictly applied, that is, disclose both elements, apply the operand of 'each' (in this case outer product multiply), then enclose. Similar considerations apply to the pairs (H) and (D). Now comparing (H) and (J), in (H) the +.x outer product (i.e. the first step of the inner product) is in fact applied to *simple* objects following the implicit disclosures with 'each', whereas in (J) the initial 'times each' outer product of the two arguments leads to a nested vector, and the final enclosure implicit in the inner product gives rise to an ultimate depth of 3.

Consistency with Existing Concepts of Reduction and Inner Product

The discussion so far might suggest that the amendments to the rules for inner product and reduction impose an intolerable new layer of complexity on the APL2 user. They are however no more than necessary adjustments which ensure that what is intuitively obvious and desirable should happen in the presence of nestedness. This is illustrated by the remaining examples for which we define B as the nested array:

$$\begin{bmatrix} 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 2 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 2 \end{bmatrix}$$

that is

$$B+2 \rho (c0 \ 1), (c2 \ 0), (c1 \ 0), c0 \ 2$$

What are:

$$(K) \ 1 \ 2 \ +. \times \ B$$

$$(L) \ (c1 \ 2) \ +. \times \ B$$

$$(M) \ (c1 \ 2) \ +. \times'' \ B$$

$$(N) \ (c1 \ 2) \ +. (\times''') \ B$$

Answers and notes:

$$(K) \ (2 \ 1)(2 \ 4) \quad DR = 3 \ 2$$

that is

$$((1 \times c0 \ 1) + 2 \times c1 \ 0) \quad ((1 \times c2 \ 0) + 2 \times c0 \ 2)$$

$$(L) \ (1 \ 2)(2 \ 4) \quad DR = 3 \ 2$$

scalar left argument forces expansion to vector of length 2. Result is

$$\begin{aligned} &(((c1 \ 2) \times c0 \ 1) + (c1 \ 2) \times c1 \ 0) \\ &(((c1 \ 2) \times c2 \ 0) + (c1 \ 2) \times c0 \ 2) \end{aligned}$$

If the elements of B are taken to represent displacement vectors in 2D space (or forces, velocities, etc.) then (K) obtains $V+2W$ where V and W are two elements belonging to the same column, and (L) gives $V'+W'$ where V' and W' represent V and W with x- and y-'stretch factors' 1 and 2 respectively applied to each displacement.

$$(M) \ \begin{matrix} 2 & 2 \\ 1 & 4 \end{matrix} \quad DR = 1 \ 2 \ 2$$

that is the $+. \times$ inner products of 1 2 with each of the 4 elements of B.

(N) – same as (L) on account of pervasiveness of multiplication.

The next two examples extend examples (A) and (B) and demonstrate the working of consecutive inner products.

$$(O) \ (c1 \ 2) \ +. \ +. \times \ B$$

$$(P) \ (c1 \ 2) \ +. \ (+. \times) \ B \ ?$$

Answers and notes:

$$\begin{array}{ll} (O) & 3 \ 6 \qquad \qquad \qquad DR = 1 \ 2 \\ (P) & 1 \ 2 \qquad \qquad 2 \ 4 \qquad \qquad DR = 4 \ 2 \end{array}$$

In both cases the scalar left argument forces scalar expansion. In (O) the derived function is $(+.+.x)$, and the elements of the result are

$$(c+.+/((0 \ 1)(2 \ 0)) \quad (c+.+/((2 \ 0)(0 \ 4)))$$

from which application of the reduction rule leads to the result.

In (P) the corresponding elements are

$$\begin{array}{l} (c+/((c1 \ 2)+.x<0 \ 1),((c1 \ 2)+.x<1 \ 0)) \\ (c+/((c1 \ 2)+.x<2 \ 0),((c1 \ 2)+.x<0 \ 2)) \end{array}$$

The highest level enclosures arise from the last step of the inner product rule, and give rise to the depth of 4.

Summary

As with nested arrays the power of operators in APL2 is considerable and make the operator capabilities of APL1 look minute! At first sight the user may be understandably daunted by the sheer amount of variation afforded by the different possible ways of combining APL2 operators, quite apart from the effects of mixing these with operators of his or her own definition. Also the subtle variations in the inner product, outer product and reduction rules demand careful study in the first place, but will ultimately reveal a beautiful consistency with the simpler APL1 ideas. Some of the language extensions of APL2 provide a substantial intellectual challenge and require a systematic and disciplined approach to master them, these articles may hopefully have provided some pointers.

NEW MAINFRAME SOFTWARE FOR IBM'S® APL

Now available in the UK, two new offerings from STSC that enhance IBM's mainframe APL implementations

If you're staying with VS APL ...

COMPILER

The first commercial compiler for APL compiles functions individually. Results in significantly faster execution. Interpreted functions can call compiled functions and vice versa.

If you're migrating to APL2 ...

SHAREFILE/AP

STSC's popular APL component file system is now available under APL2. Multi-user, nested array storage, libraries, access matrices. Multiple file system support. International language translations.

For full information, contact the APL*PLUS™ Product Group, Cocking & Drury on 01-493 6172.

Trademarks/Owners: IBM/International Business Machines Corporation · APL*PLUS/STSC, Inc.



COCKING & DRURY LTD.
THE APL PROFESSIONALS

16 BERKELEY STREET LONDON W1X 5AE
Tel: (01)493 6172 Tlx: 23152 MONREF G

Index to Advertisers

Ampere	6
APL People	58
British Airways	44
Cocking & Drury	2, 32, 82, 126
Dyadic Systems	18
HMW Programming Consultants	117
IBM (Sweden)	64
Inner Product	94
Mercia Software	17, 100
Metapraxix	48
MicroAPL	20, 92
Mine of Information	31
VECTOR – back numbers	62

All queries regarding advertising in VECTOR should be made to the advertising editor, Steve Lyus, at the following address:

Group Management Services,
Imperial Group plc.,
East Street,
Bedminster,
Bristol,
BS99 7JR.

BRITISH APL ASSOCIATION**Membership Application Form**

Please read the membership information in the inside front cover of VECTOR before completing this form. Use photocopies of this form for multiple applications. The membership year runs from 1st May – 30th April.

Name: _____
 Department: _____
 Organisation: _____
 Address line 1: _____
 Address line 2: _____
 Address line 3: _____
 Address line 4: _____
 Post or zip code: _____
 Country: _____
 Telephone number: _____

Membership category applied for (tick one):	85/86	86/87
Non-voting student membership (U.K. only)	£ 5	£ 5
UK private membership	£ 9	£ 10
Overseas private membership	£ 16	£ 18
Airmail supplement (not needed for Europe)	£ 8	£ 8
Corporate membership	£ 75	£ 85
Corporate membership Overseas	£120	£140
Sustaining membership	£325	£360

For student applicants:

Name of course: _____
 Name and title of supervisor: _____
 Signature of supervisor: _____

PAYMENT

Payment should be enclosed with membership applications in the form of a UK sterling cheque or postal order made payable to "The British APL Association". Corporate or sustaining member applicants should contact the Treasurer in advance if an invoice is required. Please enclose a stamped addressed envelope if you require a receipt.

Send the completed form to the Treasurer at this address:

Mel Chapman, 12 Garden Street, Stafford, ST17 4BT, UK.

The British APL Association

The British APL Association is a Specialist Group of the British Computer Society and a member of EuroAPL, an organisation supported by the Commission of the European Communities. It is administered by a Committee of eight officers who are elected by the vote of Association members at the Annual General Meeting. Working groups are also established in areas such as activity planning and journal production. Offers of assistance and involvement with any Association matters are welcomed and should be addressed in the first instance to the Secretary.

1986/87 Committee

Chairman:	Dick Bowman 01-634 7639	CEGB, 85 Park Street, London SE1.
Secretary:	Anthony Camacho 0727(56 from London)-60130	2 Blenheim Road, St. Albans, Herts AL1 4NR.
Treasurer:	Mel Chapman 0785-53511	12 Garden Street, Stafford, ST17 4BT
Activities:	Stan Wilkinson 01-286 7068	26 Leith Mansions Grantully Road London W9 1LQ.
Publicity:	Romilly Cocking 01-493 6172	Cocking & Drury Ltd. 16 Berkeley Street, London W1X 5AE.
Journal Editor:	David Preedy 01-541 1696	Metapraxix Ltd. Hanover House, Coombe Road Kingston KT2 7AH.
Education:	Dick Gray 0476-860483	Horseshoe House, Sproxtton, Melton Mowbray, Leicestershire LE14 4QB
Technical:	David Ziemann 01-493 6172	Cocking & Drury Ltd., 16 Berkeley Street, London W1X 5AE

Activities Working Group

Peter Donnelly	0420-87024
Steve Margolis	01-670 7959
Tim Perry	04626-77375
Roy Tallis	01-405 7841
Stan Wilkinson	01-286 7068

Journal Working Group

Jonathan Barman	01-493 6172
Anthony Camacho	0727(56 from London)-60130
Steve Lyus	0272-666961
David Preedy	01-541 1696
Adrian Smith	0904-53071
David Ziemann	01-493 6172

WP disks converted to photostetting by Capella House, Stowmarket (0449) 677663.

Printed in England by Short-Run Press Ltd., Exeter.

VECTOR

VECTOR is the quarterly Journal of the British APL Association and is distributed to Association members in the UK and overseas. The British APL Association is a Specialist Group of the British Computer Society and a member of EuroAPL, an organisation supported by the Commission of the European Communities. APL stands for "A Programming Language" — an interactive computer programming language noted for its elegance, conciseness and fast development speed. It is supported on many timesharing bureaux and on most mainframe, mini and micro computers.

SUSTAINING MEMBERS

The Committee of the British APL Association wish to acknowledge the generous financial support of the following Association Sustaining Members. In many cases these organisations also provide manpower and administrative assistance to the Association at their own cost.

APL People

17 Barton Street
Bath, Avon.
Tel. 0225-62601

APL Software Technology

14 Rosewood Avenue,
Alveston, Bristol, BS12 2PP.
Tel. 0454-415737

Cocking & Drury Ltd.

16 Berkeley Street, London W1X 5AE.
Tel. 01-493 6172

Dyadic Systems Ltd.

Park House, The High Street, Alton, Hants.
Tel. 0420 87024

Inner Product Ltd.

Eagle House, 73 Clapham Common Southside,
London SW4 9DG.
Tel. 01-673 3354

MetaTechnics Systems Ltd.

Unit 216, 62 Tritton Road,
London SE21 8DE
Tel: 01-670 7959

MicroAPL Limited

Unit 1F, Nine Elms Industrial Estate,
87 Kirtling Street, London SW8 5BP.
Tel. 01-622 0395

I.P. Sharp Associates

10 Dean Farrar Street,
London SW1H 0DX.
Tel. 01-222 7033

