```
      Y = C(1)
      DO 13 I = 2, TEN
         IF (C(I) .GT. Y)  Y = C(I)
13    CONTINUE
C     FOURTH SQUARE ROOT LOOP
      ROOT = 1.0E+0
      DO 14 I = 1, FIVE
         ROOT = 0.5E+0 * (ROOT + Y/ROOT)
14    CONTINUE
      ACC1 = ACC1 + ROOT * DIVN
      ACC = ACC + ACC1
15    CONTINUE
      WRITE( 6, 100) N, ACC, ACC1
C SHOULD PRINT N THEN 16.73343 22410 90064 71684 80142
C                     13037 73134 63994
C           AND THEN 16.73 ... / N
100   FORMAT( I10, 2E30.22 )
C FORMAT SHOULD BE ADJUSTED TO PRINT TO MAXIMUM PRECISION
STOP
END
```


References

GENTLEMAN, W. M. and WICHMANN, B. A. (1973). Timing on Computers, *SIGARCH*, (ACM) Vol. 2, pp. 20-23.

HEINHOLD, J. and BAUER, F. L. (Eds.) (1962). *Fachbegriffe der Programmierungstechnik*. Ansgearbeitet vom Fachausschutz Programmieren der Gesellschaften fuer Angewandte Mathematik und Mechanik (GAMM) Muechen, Oldenbourg-Verlag.

OSTERWEIL, L. J. and FOSDICK, L. D. (1976). DAVE—A validation error detection and documentation system for FORTRAN programs, *Software—Practice and Experience*, Vol. 6, pp. 473-486.

SCOWEN, R. S. (1969). BABEL, A new programming language, *National Physical Laboratory Report CCU7*.

WICHMANN, B. A. (1973). *ALGOL 60 Compilation and Assessment*, Academic Press, London.

CAD meets AI

Artificial Intelligence and Pattern Recognition in Computer Aided Design, edited by J-C. Latombe, 1978; 510 pages. (*North-Holland*, $60.00)

This book presents the 20 papers and as many discussions of a conference convened in March 1978 by the IFIP working group on Computer Aided Design (CAD). Its purpose was 'to study the impact of Artificial Intelligence (AI) and Pattern Recognition (PR) on CAD'.

Of the three invited papers on CAD, AI and PR, Sandewall's account of AI stands out. It brought home the central message 'the current body of knowledge in AI is mostly a way of thinking' and stressed his own view of AI's deep almost philosophical commitment to advanced programming technology. Warman's paper on CAD failed to give me the picture I needed of CAD practice and problems, though Fenves' paper left me in no doubt about the sheer size of design specifications that are created in building designs and other fields of structural engineering. Nagao's workmanlike review of pattern recognition failed to arouse the audience and PR was clearly a non-issue throughout the conference.

One of the issues that PR might be thought to address was taken up in a number of papers directed towards the man-machine interface. Thus the paper by Mohr and Masini presents the syntax-directed approach to recognising drawings; in contrast Liardets excellent paper shows how knowing the semantics of the pictorial forms being drawn can be mobilised to 'tidy up' and recognise the input. Other presentations concerned with making sense of drawings on paper were consistently attacked by Negroponte (who stands out in many of the discussions) for being the wrong way to go about it: graphical displays will be 'flat portable transparent waterproof' but not like paper capable of being crumpled?

While there can be little doubt that the communication interface can and should require intelligent computation the main emphasis of the conference concerned problem solving. Several papers were directly concerned with the presentation of systems employing problem solving techniques for necessarily limited areas of design (Pereira, Henrion, Perkowski, Tyugu, McDermott), but the significance of the topic lay not so much in particular solutions as in its power to serve as a focus for questions about the nature of designing. Akin examined the behaviour of designers with just such a perspective based upon Newell and Simon's conceptions of human information processing. He believes that many cognitively distinct mechanisms contribute to the emerging solution, an account echoed in another Carnegie Mellon contribution from Eastman who sought to integrate the contribution of different components of a CAD system in a single data base reminiscent of the organisation of the speech-understanding systems built by Reddy and his colleagues at Carnegie Mellon University. Akin notes that 'it is widely accepted that designers use parts of buildings' (as a basis for generating solutions) and it is this notion that is at the centre of what was perhaps the most important contribution to the conference: Sussman's idea of 'problem solving by debugging almost right plans' (PSBDARP). These almost-right-plans or 'answers' provide the basis for organising the design work, specifically in debugging and patching the bits of the answer that don't fit the problem specification. While Sussman's presentation concerned the implementation of his PSBDARP approach in the field of circuit design, it has a wider significance. It represents in part a concept widespread in AI that as intelligent beings we bring to every situation large hunks of organised experience that determine our perceptual expectations, subserve our understanding and guide our thinking. Variously known as schemata, scripts or frames they figure prominently in AI work on vision and natural language. Sussman's distinctive contribution to AI is to introduce his own version the 'plan' into problem solving. Sussman's argument was not just 'get some answers into your systems' but also, 'look at what you are doing as a designer in these terms'. Leaving the 'answers' with the designer of course maintains the balance of design responsibility as CAD has largely conceived it—with the user—but Sussman (supported by Sandewall's account of LISP style programming in AI) sees the prospect of the designer more explicitly formulating his almost-right-answers in computational terms by giving him powerful programming tools rather than problem oriented packages. Programmers are for Sussman designers, and AI programmers write programs to explore the problem often discarding the program to write a better more informed and informative version. The programming technology that supports such an approach to design is perhaps one of the things AI has to offer CAD, and if designing in the new computer age is to become synonymous with debugging then we'll surely need it.

M. CLOWES (Brighton)

An Introduction to Programming and Applications with Fortran by T. E. Hull and D. D. F. Day, 1978; 254 pages. (*Addison-Wesley*, £8·25)

This is really two separate and indifferent volumes stuck together to make one poor but expensive book offering terrible value for money.

The first half is a vague and feeble introduction to FORTRAN (it does not even touch upon COMMON). The second half, which has no need of the first, has its brief moments of value but attempts too much with the result that it is little more than a very superficial survey of various numerical aspects of computation.

D. L. FISHER (Leicester)