Can / Am EMTP News

Voice of the Canadian / American EMTP User Group

Publishers and Mailers :
Drs. Kai - Hwa Ger and Tsu - huei Liu
3179 Oak Tree Court
West Linn, Oregon 97068
United States of America

Authorized by Co-chairmen :
Dr. W. Scott Meyer, Editor
Dr. Tsu - huei Liu
E - mail : thliu @ bpa . gov
Vol. 98 - 4; October, 1998

Table of Contents

Salford Compiler and DOS Extender

Salford DBOS compatibility with oriental versions of Windows 98 was reported in public E-mail dated July 29th. First, Dr. Tsuyoshi Funaki of Osaka University explained that the Japanese version went on sale July 25th. He installed and tested a copy on his own computer shortly thereafter, and reported no trouble. Reacting to this report was Sangpil-AHN of Sung Kyun Kwan University in Suwon, South Korea, who observed: *"Yes. ATP works correctly in Win98 (Korean ver.) ...."*

But what about DBOS use with occidental versions of Win98? A complaint has yet to be noted. Recall Laszlo Prikler of the Technical University of Budapest in Hungary is author of detailed instructions for Windows 95 (see ATPSETUP.LIS on the GIVE2 disk). In private E-mail dated September 25th, he advised a potential user as follows: *"I have no personal experience. Win95 is good enough for me and I have no enthusiasm to deal with the same operating system in a more elegant jacket. But, as I learned from others, there is no difference regarding the usage of DBOS under Win98 compared to Win95. If you were able to use Salford ATP under Win95, you should be able to use it under Win98, too."*

The ASSIGN statement associates a statement number with a variable (e.g., ASSIGN 2542 TO NEXT) in order that a later GO TO NEXT will transfer control to that location anywhere within the same module. Trouble storing such information in a vector for later use was described in the GNU story of the July issue. At that time, it was stated that Watcom and Salford compilers handled storage, recovery, and use of the ASSIGN information without difficulty. However, on August 10th, after implementation of the new pocket calculator, it was discovered that Salford did not always store usable information in NEXTOP when there was no /DEBUG qualifier of the compilation command. Precisely what is necessary for correct operation

within SUBROUTINE POCKET remains unclear. Your Editor discovered that he could ensure correct storage if one extra line involving NEXT was added for each ASSIGN statement and storage in vector NEXTOP. But BPA's Walter Powell later suggested a more efficient and sophisticated solution involving EQUIVALENCE, so this improved alternative was adopted. This followed Mr. Powell's investigation of small demonstration program ASSIGN.FTN using the DEC VMS compiler on his workstation. Even with your Editor's extra statements, VMS execution failed to store addresses (all cells of NEXTOP were zero). As with GNU, DEC VMS may have a fundamental problem. In any case, usage continues following the Powell modification (new variables NEXT1, NEXT2, etc.) on August 12th.

Installation-dependence of the new pocket calculator was modularized during the morning of August 22nd. This followed discovery that what previously had been scattered could be concentrated in a single location (new module POCKE1) without significant loss of speed. Yes, the new code **should be** slower for both Salford and Watcom. But the difference seems to be so small it could not be measured using Salford DBOS in the case of those famous 100K steps of MATHPLT4.FTN Both old MATHPLT4 and the new MATHPLT7 were timed at 60.4945 seconds using Dr. Liu's 486. So, the restructuring was performed. The new code is more compact (an average of two lines were saved for each instruction), and the modularized part is small: a mere 2 lines for Salford. The restructuring is important because GNU and VAX VMS should become compatible following variation of only the installation-dependent part. Execution may not be as efficient as for Salford EMTP, but answers should be correct.

The fifth subcase of DC-66 illustrates OBSERVE PARALLEL MONTE CARLO (OPMC) use. It shows the accumulation produced by the preceding four subcases. The total working space is L23TOT, and prior to a reform on August 29th, the advertised size of List 23 (12K for three times default dimensioning) was seen. But Salford EMTP offers many advantages, and one is complete control over the size and location of COMMON blocks. The size of L23TOT at the end of DC-66 was increased to 1544K, which is the correct limit for TPBIG. I.e., the user now has access to all available space of the .BPA file whether he requests it or not in his LISTSIZE.DAT file. This is as was first explained in the July, 1997, newsletter. However, that earlier progress was limited to the first subcase of a disk file; it did not extend to the 5th subcase of DC-66 until August 29th when the shortcoming was realized during PCVP testing (see separate story).

More about the Internet and E-mail

Prof. Bruce Mork of Michigan Tech in Houghton announced the addition of security to his Internet-connected storage of ATP materials. His E-mail dated August 13th explained: *"Password protection has now been added to the master ftp site at Michigan Technological University. This will allow us to make available more material that is restricted to licensed ATP users. Updated Rule Book chapters should soon be supplied, for example. To access the site, the URL is: ftp://atp@ftp.ee.mtu.edu/pub/atp/ .... IMPORTANT NOTE! Access to this site is restricted to licensed ATP users. Retrieval or use of password-protected materials by unlicensed persons is in violation of the licensing terms. Do not share this password with anyone who is not licensed."*

The size of E-mail handled by both NYPA and BPA is limited. The limits are plenty large for practical use, but they exist nonetheless. This was learned on July 15th when BPA's Dr. Tsu-huei Liu tried to send a number of large files including Watcom TPBIG to Robert Meredith. All were attached to the same message, which was rejected by NYPA with the explanation: *"… 552 message size of 6464534 bytes exceeds site limit of 5120000 bytes 554 <meredith.r@nypa.gov>... Service unavailable"* It was learned from Dr. Liu that BPA, too, has a 5-Mbyte limit, but in this case it applied before MIME encoding, which increases the size by 40 to 50%, typically.

Rush Limbaugh, the heavyweight of talk radio (see the October, 1992, and July, 1993, issues), has no Web site. Your Editor heard El Rushbo explain the reason around July 13th. First, everyone else now seems to have one, so this does **not** provide motivation for him (Rush never has copied others). Second, Rush explained that he has not yet figured out what he would do with a Web site! Well, so what? Lack of a good reason has not stopped nearly every other radio program of which your Editor can think (joke)! Rush continues to use E-mail, of course, although he long ago switched from his original CompuServe address to the current **rush@eibnet.com** The eib stands for Excellence In Broadcasting (a joke involving Larry King).

BPA mishandling of E-mail became apparent on August 31st when Dr. Tsu-huei Liu returned from a trip to North Carolina. Your Editor informed others of the problem as follows (remainder of this paragraph). From public E-mail of that same day: Dr. Tsu-huei Liu's computer was successfully used for E-mail six days ago (Tuesday morning). Then some BPA specialist installed MS Office 97, and mail became unusable. Dr. Liu returned from a trip this morning, and attempted to have her mail service repaired. The effort has been only partially successful, and reports received this morning point to a problem bigger than Office 97. There now are several obvious problems with incoming mail. .... This morning, several messages dating to June and July appeared in Dr. Liu's In-box for the first time. For example, there is one from Prof. Akihiro Ametani dated Monday, June 29, 1998 1:28AM. On this end, the date is current (this past week). Has the BPA post office been holding such messages hostage all this time? This morning, Bernd Stein of FGH in Mannheim, Germany, reported that he *"received your mail 7/31/98, 12:22 pm*

with tpbig.zip enclosed, which I missed at the beginning of this month ...." It would appear that mail in both directions sometimes has been detained by BPA's post office. By itself, Bernd Stein's report might be questioned. But Stu Cook of JUST Services in Montreal, Quebec, Canada reported the same phenomenon involving more messages: *"There certainly is a hold-up somewhere in the chain. This morning I received 3 emails from you with date stamps:*

Mon, 10 Aug 1998 08:55:00 -0700
Tue, 18 Aug 1998 13:45:00 -0700
Wed, 19 Aug 1998 17:13:00 -0700

Working my way back in my in-box I find a message dated

Wed, 19 Aug 1998 07:34:00 -0700

which was received when it should have been. Looking at the expanded header for the first message shows .... It looks to me as if the BPA firewall has a rather impervious connection between the brick and the mortar."

Disclaimers at the end of E-mail messages probably make sense in these litigious times. A good illustration was seen September 17th in public E-mail of the Fargo list server. Apparently *"From: YKT (Kwong Tong),"* this message ended with: *"NOTE: This E-mail is private and confidential to the named recipients. Any information provided is given in good faith. However, unless specifically stated to the contrary, National Grid accepts no liability for the content of this E-mail, or for the consequences of any actions taken on the basis of the information provided, unless that information is subsequently confirmed in writing. The unauthorised copying of any information contained in this E-mail to persons other than the named recipients is strictly forbidden."*

Watcom ATP for MS Windows

Environment variable COMPTACS was not being interrogated by Watcom ATP on July 15th when the matter was investigated. Yet, as explained in the January newsletter, the Watcom compiler now **can** handle environment variables. Previously-commented Salford code associated with variable INLINE was converted to Watcom and activated later that same day by BPA's Dr. Tsu-huei Liu in order that the same data could be used for both MAKE and USE phases. This was part of extending compiled TACS to Watcom, whether or not (the latter alternative seems more likely) important use ever is expected.

Watcom's change in the meaning of clock ticks was mentioned in the January newsletter. But in E-mail dated July 23rd, Mr. Meredith observed that his version 10.6 behaves the old way (a factor of 100) rather than the new way (a factor of 1000) as Dr. Liu's quotation from Watcom would have the reader believe. As a result, Mr. Meredith must hand-edit a change in TGRUN1 whenever he receives a copy of the source code from developers in Portland. Not nice.

The storage of Monte Carlo tables was switched from RAM to disk on August 13th. This followed study of the reason Watcom ATP died while simulating disk file BACKFL.DAT, the famous backflash data case from Gabor Furst of suburban Vancouver, B.C., Canada. Masahiro Kan of Toshiba Corporation in Japan first reported the trouble to developers in Portland. E-mail from him dated August 10th contained a table showing time spent in the time-step loop for the case of limiting dimensions. I.e., "without listsize.dat:

```
ATP/djgpp  :  216.868
ATP/Mingw32:  216.372
ATP/Watcom :  cannot run for unknown reason"
```

In Portland, BPA's Dr. Tsu-huei Liu and your Editor found that SUBROUTINE MOVE was corrupting itself (the 3rd argument became garbage in the middle of the transfer) while handling Dube's CHARACTER*1 CSTO vector --- the only CHARACTER*1 block in ATP.

Energization number KNT of a Monte Carlo simulation was incorrectly written to the screen at the end of each restoration of program tables. Daniel Durbak of PTI (Power Technologies) first reported the problem to program developers in E-mail dated August 8th: *"... Ming ATP puts out to the screen a 3 digit value for KNT =, but Watcom ATP only puts out the last digit. (very minor point for those of us entranced on watching the screen during a statistical run)."* Yes, it was quickly found that Watcom ATP function NONBLK erroneously involved use of Watcom LENTRIM. By finding the ending rather than the beginning of each encoded energization number, NONBLK was reporting only the right-most (least significant) digit. Output to the .LIS file was unaffected, however.

Conditional data assembly, distinguished by an IF-THEN-ENDIF statement of data, was introduced in a story of the July newsletter. At the time of that writing, only the Salford compiler had been used. Watcom treatment of the feature was tested and corrected on September 4th, at which time it was learned that Watcom behavior was neither better nor worse than Salford behavior. Following change in the code to detect an undefined variable (Salford blanks are replaced by Watcom null bytes), correct operation was confirmed by a correct 3rd subcase of DCNEW-25. Unfortunately, Watcom FSYSTEM seems no more effective than Salford CISSUE@ to execute a DOS SET command within the program. Anyone having insight into the problem is encouraged to share his understanding with program developers in Portland.

An extraneous line about PEN PLOT cancellation was mentioned by Robert Meredith in E-mail dated September 10th. He reported: *"I do not request pen plot anywhere in the case. The message gets printed if MODSCR = 0, but MODSCR = 2 in STARTUP. I have to assume that it gets set to zero for this output at the last line of NEWPXX, but I do not know why."* Neither could

your Editor or BPA's Dr. Tsu-huei Liu. It is strange no one reported the problem months ago, even though the output is inconspicuous (a single line). Removing the definition in NEWPXX corrected the problem.

Availability of updated Watcom ATP was announced by Mr. Meredith in public E-mail dated September 22nd: *"The Watcom-compiler-based version of ATP for Windows NT 4 and the lesser 32-bit Windows systems has been recently updated and tested by the Can-Am User Group. The atpwnt-e.zip package containing the program has been sent to Dr. Funaki so that he can make it available for download from the secure ftp server … Besides incorporating all the latest ATP features from Drs. Meyer and Liu, this Watcom version (atpwnt.exe) is the first to allow file names of 80 or more characters. Program dimensioning has also been increased substantially in response to the actual and anticipated needs of users. Some of the list sizes have been more than doubled from the 'BPA' dimensions used in 'tpbig'. Users should alert this author, if they require still greater dimensions, since there is no penalty to supplying them to WNT users with adequate RAM -- 64 MB is plenty. (This author asserts that such WNT4 users should simply delete the listsize.dat file, since there are negligible execution time savings from its use in WNT4.) As previously, data files of up to 150,000 lines can be accommodated by Watcom ATP, so it is the preferred ATP program for those making large data cases. Potential users should be aware that atpwnt.exe does not inherently display screen plots. Plot display immediately follows execution when the eatpw.cmd execution script calls a separate program to seamlessly display the plots contained in the Post-script file from the case. Atpwnt-e.zip contains the DisplayNT.exe program, written by this author and Bob Schultz, to perform this function. It may also work on other 32-bit windows systems having the latest OpenGL support. A less-elegant solution is to use (free) Ghostview and Ghostscript to display the Postscript plots. The latter programs do allow plot printing to non-Postscript printers, so many users may need them anyhow. Both of these options are fully described in the atpgraph.pdf file on Bruce Mork's MTU server in the pub/atp/util/atpgraph directory. (Ftp to ftp.ee.mtu.edu. The user id is ….) Installation, configuration, supporting programs and other details are described within atpwnt-e.zip."* About LISTSIZE.DAT, look for comment next time.

News About TACS and MODELS

The optional blank card that follows MODELS data was provided with interpretation on July 3rd. Prior to that, there was no trace of it in the .LIS file, although the optional card was being correctly handled (i.e., it was ignored). This minor progress was the result of study of a suggestion from Guilherme Sarcinelli Luz of Furnas in Rio de Janeiro, Brazil. Public E-mail of the Fargo list server first brought the matter to public attention on June 8th, and this was followed by a private inquiry on July 3rd. Upon study, your Editor could find no need to modify logic of the /-card sorting of data that involves both TACS and MODELS. There already exist separate /TACS and /MODELS sorting directives, and these should correctly separate the two types of control system modeling if BLANK cards end each of the two classes of data. To show this to others, a 5th subcase was added to DC-30. This is the same as the 4th except for the addition of sorting. It works, with the new interpretation reading: *"Optional blank card follows last of MODELS data."*

Random numbers within MODELS are requested by the RAN function, which calls the same random number generator RANDNZ that is used by STATISTICS and TACS data cases. But unlike STATISTICS code, MODELS code failed to initialize the random number generator. The REPEATABLE RANDOM NUMBERS (RRN) and TRULY RANDOM NUMBERS (TRN) requests, mentioned elsewhere in this issue, were in fact added to rectify this shortcoming. This was after observing that Salford ATP produced different numeric results than GNU ATP for that backflash data from Gabor Furst (BACKFL.DAT). Following independent and separate analyses of the problem, BPA's Dr. Tsu-huei Liu and your Editor jointly installed the new controls during the morning of August 14th. New initialization has been added to the top of OVER8, with the RRN case case treated differently for historic reasons. Whereas STATISTICS code of OVER12 uses 14 * Pi for the seed in the RRN case, the new OVER8 code uses zero. This ensures continuity of old answers with new ones for Salford DBOS and all other systems that automatically zero otherwise-undefined cells of a COMMON block. That explains the difference in the RRN case. In the TRN case, there is no difference: the generator is seeded using the same combination of the time of day and the execution time. About location in OVER8, it should be explained that this is as late as possible while still preceding the first possible use and also remaining outside the MODELS code itself. Recall that a phasor solution may be bypassed (e.g., DC-3), or that MODELS might be involved in a FREQUENCY SCAN (FS) or a HARMONIC FREQUENCY SCAN (HFS) simulation. Randomness of the elapsed time should be adequate by overlay 8 to ensure a random seed even if the user began execution at exactly the same time of the day (seconds since midnight). This is for the situation where the user really does want different answers every time he simulates his data. Prior to the addition of a TRN request, Salford EMTP simulation of BACKFL.DAT would produce the same answers every time (not good for real engineering).

PRINTED NUMBER WIDTH controls the precision of printout within the time-step loop. But this was ignored by MODELS author Laurent Dube for output of MODELS itself (the WRITE statement) . Dube performed his own semi-optimal manipulation within USE2 following fixed encoding of FORMAT (G15.8)

at S.N. 1002. To illustrate using output of DC-52 :

```
RE{V1}, IM{V1} = 1, 0
RE{V2}, IM{V2} = 0.2830432, 0.45047724
RE{I1}, IM{I1} = 0.71695682E-01, -0.45037319E-01
```

There are several minor objections. First, precision is limited to the 8 digits of a G15.8 FORMAT. Second, there is a mixture of integers (first row) and floating-point numbers (rows 2 and 3) whereas in fact all numbers are floating-point. Third, one digit of precision (the leading zero of rows 2 and 3) is being wasted. Fourth, there is use of scientific notation (row 3) even though it does not enhance precision within the space that is allowed. In fact, it detracts from precision. Fifth and finally, numbers are not aligned in columns. So, your Editor replaced Dube's logic (62 lines of code) by a simple call to the procedure used elsewhere in ATP. The result for DC-52, using default width 11 and separation 2, follows:

```
RE{V1}, IM{V1} =  1.0     ,  0.0
RE{V2}, IM{V2} =  .2830432,  .4504772
RE{I1}, IM{I1} =  .0716957,  -.045037
```

Local storage KENT(30) in TSTACS limited the number of S-blocks of TACS to 30 prior to variable dimensioning on September 3rd. More precisely, there was no limit on S blocks, which simply overflowed KENT. Trouble first was reported by Claes Hillberg of ABB in public E-mail dated August 31st. Curiously, this had *"Subject: TACS Device 53, Watcom, erroneous result."* When a Type-53 device was added, Watcom ATP died, it was reported. Yet, there was nothing wrong with the Type-53 code or storage. Also, Salford EMTP produced the right answer even though it had the same defect as Watcom ATP (strange, the differences in treatment of local storage). Variable dimensioning uses new offset KENOFF that is shared only by OVER2 and TSTACS (new structure encouraged by the 23-cell limit in TACSAR). Following modification, the only standard test case requiring change was DC47.DAT in which List 19 was increased from two times default dimensioning (3200 cells) to 3400 cells.

Higher - Order Pi Circuits

Bernd Stein of FGH in Mannheim, Germany, is another user of high-order Pi-circuits. In E-mail dated August 6th, he expressed concern about dimensions of LISTSIZE.BPA that were the starting point used to create a special version of Watcom ATP for him: *"With the data case I mentioned above, I reach 915 nodes and 1899 branches, modelling one phase of a transformer. Thinking about modelling three phases ...."* So, BPA's Dr. Tsu-huei Liu and your Editor applied the 400-coil upgrade to the EEUG dimensions rather than the .BPA file as used for the first try. Disk file VARDFGH.BAT connects LISTSIZE.FGH to create tables of size just under 28 Mbytes (7 million words). Compare this with the old TP20 of 1993 and before, which had tables of about 8 Mbytes (2 million words). The Watcom linker is demanding of disk space from WinNT, too. Unlike Salford, it seems to require every byte of the virtual size in real hard disk space. On a computer with 32 Mbytes of RAM, 60 Mbytes of swap space on disk was found to be inadequate! The linking failed with the complaint "*dynamic memory exhausted."* So, after moving some directories to another drive, Dr. Liu was able to increase the local paging file size to 100 Mbytes using the virtual memory button under Control Panel. Then the Watcom linking succeeded. Conclusion: WinNT overhead of Watcom ATP linking has become staggering. Of course, Salford compilation and linking avoid this problem.

Brain - Damaged MS Windows

About insufficient memory, Daniel Durbak of PTI (Power Technologies) provided an interesting observation in E-mail dated August 8th: *"I started running EMTP 20 years ago this month on computer systems that would only allow you to use 64 kb. EMTP has gotten so much better since then. But isn't it strange that we can still have memory problems with computers having 64 Mb of memory?"* True when using MS Windows (Mr. Durbak then was adapting as a new user of NT), which exacerbates the problem by imposing, in complex ways, limits of its own, and masking the consequences from the user. What was obvious using Salford DBOS under DOS becomes complicated using Bill's G's Windows. Of course, eventually the trouble (missing LISTSIZE.DAT for the GNU djgpp and Mingw32 versions of ATP) was understood and corrected. But during the struggle, one had to wonder how this was possible on a computer with 32 Mbytes of RAM (what Mr. Durbak was using prior to doubling in an effort to improve performance). What was unthinkable 5 years ago, Bill G has made possible.

"Installed Office 97 over previous version of Office ..." was the note found on Dr. Tsu-huei Liu's desk at BPA the morning of August 27th. Installation had been scheduled two days earlier, and it began on time (your Editor knows because he then lost use of the computer). Approximately 24 hours later, when your Editor returned, work by BPA's Tyson Cross was still underway during two-plus hours of observation from a distance. This morning, except for all-important E-mail, the computer seems to be generally usable. But during log in, NT opened a small window entitled *"Mail"* to inform your Editor that *"Mail could not connect to your Mail server. The configuration for the Mail server path is missing or invalid."* Since Dr. Liu is some 3K miles away (in North Carolina, no doubt looking for a row boat) this problem may take a while to solve.

If you type *I'd like to see Bill Gates dead* in Word 97, highlight it, and go to the thesaurus, the suggested synonym is *I'll drink to that*! This latest insight into the workings of MS software came from Robert Meredith of the New York City area. His E-mail dated September 20th concluded with the observation that the procedure *"works here."* Well, it works in Portland, too, on Dr. Liu's Pentium. Amazing.

Longer Disk File Names

Long file names were a subject of Salford interest in the preceding issue. The new 80-byte CHARACTER variable CLIK80 was mentioned as part of changes to output in order better to handle those 80-byte input data file names advocated by Robert Meredith of the New York City area.

Expansion from 80 to 132 bytes was progress of the week ending September 12th. Several days of work were involved by both your Editor and BPA's Dr. Tsu-huei Liu, with the latter doing the real testing with file names more than 80 bytes long. This was using Watcom ATP on her Pentium that runs WinNT. Both the input and the output file name now are allowed to be more than 80 bytes in length. Usage has been confined to parameters of the execution command as communicated to ATP by RUNTP. This began as direct, interactive keyboard communication with MS-DOS (yes, this does work), but rapidly was shifted to a batch file LONG.BAT along the lines of RUN.BAT in order to have permanent records and also to avoid keying more than once. The long data file name was created without difficulty using PFE (see the October, 1997, issue), and the different long names were remembered from one use to another, thereby avoiding re-keying.

Robert Meredith sounded positively pleased in his reaction dated September 16th : *"Congratulations to the chefs! … I threw my usual plottest.dat at it. Then the 80-character version named THISISAVERYLONGVERSION OFTHEORIGINALPLOTTESTDATACASEWITHANEXTR ACOLLECTIONOFCH.DAT worked fine as well. I tested TACS with my bobtacl.dat and bob68l.dat cases; they both worked and the type 68 inclusion in the latter shortened execution to 15-16% of the time for the version with TACS. I love it! … I have increased dimensioning to 8000 nodes, 15000 lines, etc. LABCOM for me is 3639004 instead of 233316 at 3X default (a factor of 15.5)."* Both the long file names and larger dimensions were announced to ordinary users by Mr. Meredith in public E-mail dated September 22nd. For more details, see the preceding story about Watcom ATP.

Batch file RUNTP.BAT has been used for many years to communicate input file name and output choices to ATP. Common values for the final two of four arguments are * and -r where the star indicates naming parallel to the input data file name and -r indicates replacement (if a file of that name already exists). GNU ATP has the same procedure but with the star replaced by letter S (think of it as meaning *same*). The concept is powerful and useful. However, it is less than perfectly general. It assumes the user wants the .LIS file in the same directory as his data file. This is not always the case. Why force the user to re-key the file name in order to put it in a different place? For short names, the issue was not large. But for names having length 80 or more bytes, the difference is enormous. It might even be life-saving (there is a 127-byte limit of djgpp running under WinNT DOS). So, beginning September 15th, the procedure was generalized to allow either any directory or the current directory. For example, for Salford EMTP that runs under DOS :

```
runtp  disk  \data\jday.  \test\*  -r
runtp  disk  \data\jday.    .*     -r
```

In both of these, the star is optional, and in fact has no effect. Note the directory must end in a backslash, just as was the case for the ATPDIR definition. Of course, more than one level of directory is allowed, and the specification can begin with a disk drive. A more general example would be c:\data\project\test\ (here the optional star has been omitted). About the period of the second illustration, just as for DOS itself, this indicates the current directory (whatever that might be). Changes are required in installation-dependent SYSDEP because details are computer dependent. But the concept is general, and should be exploitable for any computer. Already, Salford, Watcom, and GNU ATP versions should be empowered with the new logic.

GNU ATP may or may not tolerate names approaching 132 bytes in length. For details, see the GNU story.

FNDFIL was mentioned in the April, 1997, newsletter. Recall the Salford user is allowed to define his own file prefixes at the bottom of STARTUP using the DEC VMS-like :== separator. From the beginning, this applied to the input data file. Then, September 16th, it was extended to $INCLUDE file names. As names are made much longer, this becomes important. In the absence of such extensions, file names obviously are limited by the 80-byte width of data cards. But $PREFIX and $SUFFIX provided one way to extend the name, and FNDFIL now provides a second for those compilers and operating systems that use the feature. Thus far, Salford logic has been made available to Watcom and GNU. I.e., there has been no attempt to modify the DOS logic of Salford. For use under MS Windows, the DOS rules probably are reasonable. But for Linux, logic really should be altered.

Machinery Saturation Questioned, 3

The following paragraph was received from Prof. Hermann W. Dommel on September 25th. Comment will be withheld temporarily due to lack of space in this issue.

"I appreciate the interest of ATP users in the saturation representation of synchronous machines, and I apologize for not having been precise enough in specifying the version in which the corrections were made. They were made in EMTP96 (Version 3 of the DCG/EPRI EMTP). I presented a 14-page report about it at an EMTP96 Seminar in Irving, Texas on May 23-24, 1996, which was written by Jiming Lin, with contributions from S. Lefebvre and J. Mahseredjian. Dr. H. Okamoto of TEPCO is correct in saying that DCG Version 1.1 gave erroneous results,

because that is a very old version before the corrections were implemented. He confirmed to me that he has never tried EMTP96 for the data case from Dr. Haginomori, only DCG Version 1.1 on the TEPCO mainframe, and ATP."

Szymanski C - like .PL4 Structure

A decade ago, manipulations associated with C-like .PL4 files had been modularized in order that computer expert David Szymanski in Wattsburg (suburban Erie), Pennsylvania, could do the work using C language. Recall 1988 was the year your Editor purchased the first half of his 16-MHz, Intel 386-based, Unix computer from Szymanski. Prof. Tom Leskovich of the Beaver campus of Penn State originally owned the other half. The purchase was made to provide Szymanski with a platform for testing his ideas about the adaptation of ATP to standard Unix (then AT&T System V, Release 3). At the time, your Editor was developing ATP using Apollo, for which C-like .PL4 files never were supported. Also used was the Lahey compiler F77L for DOS, available on a 286-based PC. Lahey directly supported C-like files as part of normal READ and WRITE (the OPEN statement involved ACCESS = 'TRANSPARENT' and REC= was required on usage). In 1988 and 1989, the C-like code from Szymanski was nicely isolated in modules.

Next came Salford. It was during September of 1989 that Mustafa Kizilcay, then a doctoral student at the University of Hannover, demonstrated success supporting ATP. One can read about this momentous development in newsletters beginning with the October, 1989, issue. The Salford FTN77 compiler revolutionized both program development and usage. Unfortunately, it ended modularization of the C-like code, too. For Salford as for Lahey, there was no need for separate subroutines with their added inefficiency and complexity. So, when the UTPF was changed from Apollo to Salford (see the October, 1992, newsletter), the modularization was largely lost. Some consideration was given to reviving it for Watcom during 1994 or 1995, but nothing ever was done for two reasons. Most importantly, no one volunteered to handle the C. Second, use of ACCESS='DIRECT' with one record per time step was made to work satisfactorily (except for the second or later subcase of DC-46). So, reconsideration was delayed until more recent application of the GNU compiler to ATP by Masahiro Kan of Toshiba Corporation in Japan. Many months ago, this gentleman had volunteered to do the coding and testing using C. His offer finally was accepted during the Labor Day weekend of September, which your Editor dedicated to re-modularization and testing using Salford.

Nine UTPF segments required modification for the Salford testing. By the morning of the 3rd day, all test cases had been reconciled. Interface requirements then were communicated to Masahiro Kan, and he did the rest for GNU ATP using real C. The next issue should summarize Mr. Kan's innovation, and performance of the new code.

Evaluating Analytical Functions , II

The pocket calculator from BPA's Walter Powell first was mentioned in the April, 1997, issue. Several uses were considered at the time. First, there was the support for mathematics within $INCLUDE files (see the January, 1997, issue). Second, there was the Type-10 user-defined analytical source. Third, there was $PARAMETER. Illustrations of these three ideas were placed in DC-36 ($INCLUDE) and DCNEW-19 (Type-10 source and $PARAMETER). But execution has been imperfect, so real, general use has been limited. A fourth possible use, to convert FORTRAN of TACS supplemental variables into TACS ASSEMBLY LANGUAGE (TAL), never was seriously considered due to slow execution.

Replacement code that *almost worked right* first was demonstrated for BPA's Dr. Tsu-huei Liu on July 23rd. This was program COMPILER.FTN after some four or five working days of effort spread over two weeks. Volume is moderate: 475 lines including occasional blank and comment lines. More work is needed, but success now seems guaranteed. Enough complicated expressions already have been handled correctly to convince your Editor that the project can be completed to handle all cases. The plan is to replace code of the Powell pocket calculator by this new one. That will be the first use in ATP. But it is far from the last. The real motivation for all-new coding is TAL, which until now was written by the human mind. Existing TAL is close to the language of the new pocket calculator. After perfecting the new logic as a replacement for the Powell pocket calculator, it is expected that compilation of TACS supplemental variables will be attempted. This is the long-term plan as writing of this story begins on July 23rd.

Lack of modularization is a feature of your Editor's code. Unlike the Powell pocket calculator, all of your Editor's code is contained in one main program. That is, no SUBROUTINE or FUNCTION module is used, and this should make execution fast enough for use within the time-step loop. That was the fundamental problem of the Powell pocket calculator: it was slow --- presumably because it called subroutines with arguments as part of each evaluation. Even if operation could be made perfect, lack of speed would have prevented widespread use within the time-step loop. This should not be a problem with the new TAL-like code (see MATHPLT3 in the April, 1997, issue).

That inquiry for help (*"What reader knows enough about compilers to offer an efficient solution?"*), publicized in the January newsletter, went unanswered. Your Editor began coding during July only because no one else offered any advice or alternative. Necessity was the mother of invention in the case of faster pocket calculators, too.

COMPILER.LIS is the output file that documents compilation of COMPILER.DAT, the input file that automatically is connected internally. For each line of input, there will be a corresponding output table. On July 25th when improvement of the compiler had stalled, there were 37 such examples. Output can be illustrated by showing the 23rd example, which is relatively simple:

```
          Compile next line  KNT = 23
          ===========================
          88TEST2   = - COS ( THETA )
          ===========================
 1 COS     THETA
 2 -SAVE   TEST2
```

The input (a TACS supplemental variable definition) is shown first, between separator lines of equal signs. Below this will be found the lines of TAL (in this case, just two) that produce the same effect. Note the minus sign in front of SAVE. This is new: saving the **negative** of the accumulator. While writing and testing the compiler, this and several other new assembly language instructions were found to be either convenient or more efficient. Details of TAL have changed, and a few lines of MATHPLT4.DAT differ from MATHPLT3.DAT in small ways. The new TAL allows modified SAVE whereas the old TAL used comparable modified library functions (no longer allowed).

Removal of reliance upon blank separator characters was the first of two projects for the weekend that began August. As this paragraph is written during the evening of August 2nd, blanks can be inserted anywhere except inside constants than involve scientific notation. Alternatively, all blanks can be removed. For Dube-format supplemental variables with a type-code in columns 1-2 and an equal sign in column 11, each input line is processed exactly as it is supplied by the user. But standard FORTRAN lines, which begin in column 7, are compressed by the removal of all blanks prior to processing. Perhaps Dube-format lines later will be compressed. But, for now, this is being avoided as a safety measure. If compressed logic ever might fail, one can switch to Dube-format lines with blanks to separate the troubled items. The possible embedded plus or minus sign of constants involving scientific notation posed the greatest problem of recognition. Either letter D or letter E must immediately precede the plus or minus sign, which in turn must immediately precede a decimal digit. Also, the E or D must immediately be preceded by either a decimal point or a decimal digit. If these conditions are not satisfied, the plus or minus is taken as a request for arithmetic (addition or subtraction).

Unlimited precision of constants was the second of two projects for the weekend that began August. As originally coded, constants were limited to 8 bytes, and TAL involved fields of width 8 for all input. While program variables remain 6 bytes, with blank fill on the right, constants can be as wide as the user wants within the limit of column 80 (continuation lines are not yet allowed). Syntax of TAL has been converted to free-format, with one or more blanks separating parameters. The compiler output begins the first parameter in column 9, and thereafter uses two blanks to separate parameters. The extra blanks are not necessary, but make the lines more readable (a cosmetic consideration).

IF-THEN-ENDIF and IF-THEN-ELSE-ENDIF were the first structured IF statements to be allowed. This was August 3rd. Logic seems to be working correctly, and could be extended to allow an indeterminate number of internal ELSEIF if there is sufficient interest. But your Editor is rapidly losing interest. The single conditional jump handles all cases of interest. Who needs the structure? To your Editor, structure is more of a problem than it is a solution. Long live statement numbers (which remain a cornerstone of TAL in any case).

Speed of compilation seems to be a strength of your Editor's new code. The first timing was performed on August 3rd by looping 10K times over those same 6 famous lines that end with TWOXX (see the April, 1997 issue). After suppressing all printout, the 60K lines were compiled in 23.90 sec on your Editor's 133-MHz Pentium. This implies a compilation rate of 151K lines/minute --- plenty fast for massive usage prior to each simulation.

Type-10 sources are to be the first feature supported by the new compiler. The only rule being changed concerns overflow. Previously, such sources were limited in number to 20 by local storage. That storage now is being removed. There **will** be some limits, but the new ones will be much larger; and they will be associated with TAL creation (by compilation) and execution rather than with the sources themselves. This is as implementation begins on August 6th.

Data of DCNEW-19 provided the first demonstration of correct handling of Type-10 sources. This was on August 8th. For ease of timing, 100K steps to 2 seconds were taken as explained in the April issue. The same math is being performed, only TACS has been replaced by Type-10 voltage sources which drive one-ohm resistors to ground. Time-step loop output is identical, and execution seems believably fast --- and obviously faster than Dube's logic. We now can add a third row to the two that were published six months ago:

```
      1980 data of  MATHTAC1.DAT :   18.077
      1975 data of  MATHOLD.DAT  :   13.077
      Type-10  in   MATHTY10.DAT :    6.538
```

About the structure of execution, there is a single call to the new SUBROUTINE POCKET during each time step. All type-10 source functions are evaluated for that particular instant of simulation time. This is the same as TACS: one call per time step. In order to separate the time of function evaluation from the remaining burden, the code to service TAL within POCKET was temporarily bypassed:

```
      Type-10  without functions :    4.780
```

In round numbers, less than 2 seconds are required for 100K evaluations of that set of six functions. Type-10 sources have been boosted to warp speed.

$PARAMETER use was activated August 11th after struggling with a known ver. 2.66 Salford compiler bug (imperfect operation of virtual scratch files). Unlike TACS supplemental variables and Type-10 sources, speed of execution is not an important consideration for the $PARAMETER application. This is because usage is outside the time-step loop (each evaluation is performed only once). Whereas all Type-10 functions are evaluated during a single CALL to the new POCKET, each line of $PARAMETER definition was permitted to involve its own, separate compilation and execution. A fundamental improvement concerns the handling of SPY symbols (e.g., EPSILN for the familiar singularity tolerance). Previously, each such symbol was eliminated by substitution of its numerical value for the A6 name. This caused two problems: 1) full precision of floating-point values required that the user leave artificial, extra space; and 2) negative values sometimes resulted in an error termination by the Powell pocket calculator (e.g., "- TMAX" would become "- -2.5" if TMAX had value -2.5). Both of these complications have been eliminated by the new code. No longer does extra white space affect precision (parameters automatically are handled as an 18-byte encoded string), and negative parameters always should be handled correctly. If not, complain to your Editor.

Laurent Dube's logical functions and operators were the preoccupation of the weekend of August 15th and 16th. As a first step, in order to gain familiarity, the logical functions and operators were added to MATHPLT4. Next, they were added to COMPILER. Finally, after this simplified and isolated testing, changes of the preceding two programs were transferred to subroutine POCKET of ATP. Recall what is involved. Unlike FORTRAN, Dube's language of supplemental variables allows floating-point variables to be processed by functions .NOT., .AND., .OR., .NAND., and .NOR., and by relational operators .EQ., .NE., .GT., .GE., .LT., and .LE. Meanings correspond to FORTRAN, but no separate variable type LOGICAL is involved. This is the main reason compiled TACS provided an incomplete solution: Dube established his own rules for logical variables. This difference was a primary motivation for allowing the AA and BB alternatives as explained in the April, 1997, newsletter. Well, a strength of your Editor's compiler is that he, unlike Salford or Watcom or GNU, can follow Dube's non-standard rules for logical variables. Anyway, this has been the hope. This provided the motivation for doing the work. About order of evaluation, all NOT are processed first, then all AND and all NAND, then all OR and all NOR, and finally the six relational operators. Despite the separate name, the relational operators, too, are functions, it is to be noted. The logical function evaluation is not started until all numerical operations have been completed. Also, within any one of the four logical classes, processing is done from left to right. These rules seem plausible to your Editor, but part did not come from any book. Any reader who believes them to be incorrect is encouraged to send E-mail correcting the order of evaluation. Change should not be difficult if it is made soon enough, while details remain familiar.

Logical operations have been less efficiently coded than numerical operations, it should be explained. With only one exception, logical operations are lower in level (more RISC-like). Implementation was simplified by assuming that both arguments of all 2-argument functions are stored in memory. That is, neither is allowed to be in the accumulator. As a result, otherwise-extraneous SAVEs of the accumulator will be required in places. The only exception is .NOT., which involves just a single argument. Programming allows this to be either in memory or in the accumulator (i.e., the compiler appropriately avoids the intermediate SAVE if the argument already is in the accumulator). Another extension applied to .NOT. is an optional, automatic SAVE to memory upon completion. As an illustration, consider the FORTRAN line SIGNAL = .NOT. FIRE2 Many comparable statements can be found in Robert Hasibar's DC-1 --- comparable, but different in structure, because his data creation preceded the free-format alternative that Dube introduced during 1980. The stated illustration will be compiled into the single line of assembler : .NOTSV FIRE2 SIGNAL Whether other logical operators might be raised to this higher, CISC-like level will depend on demand. Your Editor is aware of no fundamental theoretical problem (for the Salford compiler, anyway).

Variable Dimensioning of ATP

240000 is the first number on the 4th data card of both the .DAT and the .BPA files of LISTSIZE. Need to expand this value is unlikely for the average user. Yet, an attempt to use a value larger than 240K in just the .DAT file was reported on August 5th. Luis Giraldo, writing from somewhere in Columbia, reported trouble. Later that same day, your Editor explained that protection against overflow had been lacking, but now was being added, to protect users in the future. An illustration was provided.

List number 24 is the number of phases of compensation that are being used. Prior to correction on August 7th, it was not correctly checked for overflow during dynamic dimensioning at the start of program execution. Curiously, for five years, no one has complained. What ATP did was check the user-requested value of LISTSIZE.DAT against the limit that is built into the program (from the .BPA file). This is sufficient to prevent overflow, but is not necessary. Only a single array is involved, and the product of List 1 and List 24 from the .DAT file must not exceed this same product in the .BPA file (used for program creation).

Error messages of dynamic dimensioning were handled informally prior to August 8th. They involved simple WRITE statements to LUNIT6 followed by STOP. For Salford, this was seen on the screen as intended, so was satisfactory. But for Watcom ATP, the output was **not**

seen on the screen, and there was no .LIS file, because the local STOP statement prevented flushing of the output buffer to disk. The error message was found in the .DBG file, of all places! So, the WRITE and STOP within DIMENS were converted to formal output procedures that include flushing of the output buffer prior to an end of execution. The user of BOTH should see the error message both on the screen and in the .LIS file.

Missing LISTSIZE.DAT should result in the use of limiting table sizes. But a recent change (see the opening paragraph of this story) upset this. Robert Meredith reported the bad news about Watcom ATP in E-mail dated September 9th: *"I did not have much success with your latest Watcom code. .... When I attempted to run it, it appears to output an endless succession of blank lines to the screen. The command window cursor just keeps moving down. I have to close the window to stop the action. There are occasionally a few garbage characters output and in one case the name of the ***.tmp file used for debug. Most of it goes by too fast to observe."* This confusing behavior was traced to the combined effect of two errors. Salford ATP had only one: without endless output, it erroneously complained that program dimensions were too large. Analysis and correction was made September 11th. The universal error involved failure to initialize variable N6 in DIMENS whereas Watcom also involved an extraneous CALL in installation-dependent RFUNL1.

LSIZ30, the limiting number of Noda frequency-dependent circuits, was not being correctly reported in some .LIS files prior to correction on September 14th. The problem first was reported by Robert Meredith of the New York City area in E-mail dated two days earlier. He asked: *"... have you just not remembered that the Noda C is now included?*" Actually, the error went unnoticed in Portland because use almost always involves KOMPAR = 4 to ensure easy comparison. In this case, 80-column output of case-summary statistics omits all limiting sizes. Upon switching to value zero, the complaint was quickly confirmed: a zero was being outputted instead of 6 (3 times the default value of 2). This was traced to output of the wrong variable (NODA). As for the number of Noda circuits being used, there seems never to have been a problem. Output of DCNEW-14 correctly shows values 1 and 3 (the latter for the 2nd subcase).

Monte Carlo (STATISTICS)

REPEATABLE RANDOM NUMBERS (RRN) and TRULY RANDOM NUMBERS (TRN) are new English-language requests that control seeding of the random number generator used by ATP. The column-80 punch of the STATISTICS miscellaneous data card controls the same variable NSEED, although it does this numerically rather than using English. Values unity and zero, respectively, correspond to the two request words. If one of the two English-language requests is used, it will override the later numeric specification of NSEED in column 80. This is illustrated by DC-24, which has been modified by the addition of an RRN request and a change of the column-80 punch from unity to zero. Operation was inspired by, and is consistent philosophically with, those decade-old English-language requests of LINE CONSTANTS such as SPECIAL DOUBLE CIRCUIT. While not removing any of the historic numerical control, new English-language alternatives are being offered. They are believed to be easier to remember; and unlike numbers, they should be self-explanatory. See the TACS and MODELS story for a continuation.

STATISTICS or SYSTEMATIC data cases are practical with no output variables provided MODELS is used for the output. This is illustrated by Gabor Furst's BACKFL.DAT data as mentioned elsewhere in this issue. Upon study of the .LIS file, it was noted that output of the time-step loop, and variable extrema, might profitably be suppressed. The former showed nothing other than the step and associated time whereas the latter consisted of nothing more than English-language labeling (normally-associated numbers were missing). So, when the ATP output vector has no variable, a little computation is avoided, and all such broken output now is suppressed. For BACKFL, there now is no output other than that requested by the user's WRITE statements (MODELS data). Changes were made to SUBTS3 and MAIN20 on August 17th. The following day, a new 8th subcase of DC-68 was added to illustrate operation without normal outputs. This has some of the structure of BACKFL without any of the real engineering.

GNU ATP for Linux and DOS

The undefined seed for the GNU random number generator (see preceding mention in the Watcom story) was unexpected. After all, it is carried in a COMMON block, and Robert Schultz's turbo table dumping relies on the operating system setting unused portions to zero. How could this not be working for GNU ATP as it was for Salford ATP? Investigation provided a simple answer: GNU RFUNL1 uses the ATP random number generator RFUNL1 to produce a random number upon which the diagnostic file name can be based. This step was avoided in the Salford version because DBOS TEMP_FILE@ was available as a more convenient alternative.

An ASSIGN statement that involves EQUIVALENCE was the suggestion of BPA's Walter Powell as mentioned in the opening story. The story changed from Salford to GNU August 21st when BPA's Dr. Tsu-huei Liu performed and tested a GNU translation. Special diagnostic had been added to monitor the COPY and A+B instructions, which are used first. But it was not seen in the aborted output of standard test cases DCNEW-19 and 21. Other, earlier diagnostic showed that all addresses NEXTOP and

NEXTXX were zero, just as all VAX addresses had been before the modification. This was using the last subcase of DCNEW-21. To conclude, Mr. Powell's modification was good for Salford, but seemed not to help GNU.

GNU ATP using djgpp in a DOS window of WinNT fails instantly if the execution command is too long. Since the program is slow starting (several seconds are required) whereas the failure seems instantaneous, the operating system would seem to be rejecting the line. The complaint is: *"The system can not execute the specified program."* Quoting from Dr. Liu's September 17th E-mail to Masahiro Kan: *"The total length of that command which starts with TPBIG and ends with -r is 127 characters. Is there anything special about this length on DOS of NT with GNU djgpp? It works when we shorten either the input file name or output file name by one character .... Do you suppose djgpp has a 7-bit limit as the Internet does?"* Mr. Kan answered the question definitively later that same day: *"As you know, the original DOS has a limitation of the command line length (127 bytes). But what is the limitation for command.com of WinNT and Win95? I don't know. I made a test under Win95, WinNT, djgpp, and Mingw32. The results are as follows; 1) Win95 + djgpp: OK; 2) Win95 + Mingw32: OK; 3) WinNT + djgpp: NG (with a message that the system can not execute ....); and 4) WinNT+Mingw32: OK. The conclusion: Djgpp under WinNT has a problem with command line length."* In response, your Editor explained that this paragraph rightly belonged to the story about brain-damaged MS software, even though it is not being placed there!

C-like .PL4 files should be available soon thanks to the C-language programming of Masahiro Kan of Toshiba Corporation in Japan. Read the story about Szymanski's work a decade ago, and expect details in the next issue.

Compiled TACS & Assembly Language

The Type-54 TACS device resulted in undefined variable "I" as mentioned in the previous issue. Yet, there was more to the story. An addendum July 12th: Several days ago (around July 9th), this same error was rediscovered using a new GNU translation. Somehow, the correction never was carried back to the UTPF. Curiously, the July rediscovery did not involve DC-19a, however. This data simulated uneventfully. Instead, it was the simulation of DC-21a that this time was killed by djgpp. Uninitialized variables sometimes behave this way: both unpredictably and inconsistently.

Relational operators such as .EQ. and .GE. of FORTRAN were added to TACS assembly language on July 28th in order to make the execution of TACS supplemental variables conditional. This modification bears clear similarity (e.g., the SN= tag) to experimentation that was described in the October, 1996, newsletter. Yet, there is one fundamental difference. Here, TACS itself is not being changed. It could have been changed, but it has not been changed. Instead, the logic of evaluation using assembly language is being expanded. By using relational operators, TAL has become more than an equivalence; it has become an extension. Motivation is as follows. Why not allow two or more alternatives for the computation of a supplemental variable? For example, there may be one formula if some variable is unsaturated and a second formula for saturation. The number of supplemental variables will not change (it remains one). But before computing it, one does a comparison to see which formula should be used. This is in assembly language, not in Dube's original TACS language. TAL not only is faster (the original motivation), but now it also allows conditional execution, which is an extension. Indirectly (only via TAL), TACS supplemental variables have become more powerful. Yet the added complexity is confined to new code supporting TAL; it is not a burden on Dube's TACS code. Neither data nor code from Dube has changed. This approach is fundamentally different from thinking reported in the October, 1996, newsletter.

Arbitrary precision for constants began July 30th, when the original fixed format for data of assembler commands was replaced by free-format data input. The old remains generally valid, but is a special case of the new. Although full precision of constants was the dominant motivation, at the same time users were freed from the need to align data in particular columns. Parameters remain ordered, and now are separated from one another and the assembler command verb by an arbitrary positive number of blanks. Only the op (operation) code of the instruction remains fixed: it must begin in column one of the input data line. In-line comments are allowed --- and ignored, of course. About 6-character program variable names, the non-blank part must be left-adjusted in the A6 data field that defines them, and must involve no imbedded blanks. Yet, this is nothing new. It has been a requirement of EMTP free-format data for decades, and became the rule for MODELS by Laurent Dube about a decade ago (MODELS data is all free-format). Anyone who wants to position blanks anywhere other than on the right of A6 names will find ATP life increasingly difficult. The new TAL format is just one more way there can and will be trouble.

Efficiency of the modified TAL instruction set seems to be no worse, and even might be a little better, than it was. That is, execution speed seems **not** to have been lost during modification of the language so as to be more compatible with machine compilation. This can be seen by comparing the elapsed time of MATHPLT3 execution with MATHPLT4 execution. With the /DEBUG switch used in both compilations, and with execution under Win95 DOS on your Editor's 133-MHz Pentium (Win95 was not running), six consecutive trials were performed. There was no output other than one timing line during the million time steps that were taken (a 10K-step simulation to 2 seconds was repeated 100 times). The following five elapsed times in seconds are for the five best (smallest) in each case.

```
Old TAL (3) : 6.374  6.374  6.374  6.429  6.374
New TAL (4) : 6.374  6.374  6.374  6.374  6.319
```

Efficiency of the compiler is demonstrated using MATHPLT5.DAT, which was produced by your Editor's compiler. Surprisingly, this simulates no slower, and might even simulate faster. Again, reporting the best 5 of 6 consecutive executions:

```
Compiler TAL: 6.319  6.319  6.374  6.319  6.319
```

It was not expected that computer logic written by your Editor would beat your Editor's mind. Yet, there is a logical explanation besides the debilitating effects of temperature (soon expected to peak at 95 degrees F). The tests of the previous paragraph either date to 1997 (3) or involve only minor modifications of 1997code (4). The compiler, on the other hand, was written to exploit the new language fully. Your Editor might be able to go back and modify (4) to take greater advantage of new TAL structures. But he will not bother. Remember that debilitating temperature. It may be time to shut the computer off (although no error yet has been noted). The old Apollo DN-420 would fail around 90 degrees. Hardware seems to have improved.

About (3), (4), and compiler output, each file of code has 22 non-comment lines. The average level of complexity has not changed, therefore. The average TACS supplemental variable requires 22 / 6 = 3.67 assembly instructions.

Five-digit indexing of memory was found to be inadequate for compiled TACS use by Orlando Hevia of Universidad Tecnologica Nacional in Santa Fe, Argentina. A report of the trouble, and its correction, was received in E-mail dated August 19th from Masahiro Kan of Toshiba Corporation in Japan. *"He commented out the NEW LIST SIZES line from the DC22D.DAT data case. When we run the case, the generated …. file contains vector references with bad subscripts such as XTCS(*****). Obviously, overflow has occurred in writing with FORMAT (I5). I modified the code of tacsup.f and over16.f and have attached these files for your review."* Yes, with the limit on List 19 (working space for TACS) set at 120K, it is easy to overflow 99999. So, Mr. Kan's changes are accepted. They amount to changing I5 to I7, so should last for more than a few years. The limit on List 19 now is ten million, which will address 40 Mbytes of TACS working space. Work using Salford was completed August 20th, and BPA's Dr. Tsu-huei successfully tested the corresponding GNU translation the following day.

Supplemental variable lines that either reached or approached column 80 were incorrectly treated during COMPILED TACS MAKE prior to an update on August 19th. This correction, too, came from Masahiro Kan in Japan. His E-mail dated August 9th included the following: *"2. I modified the code of tacsup.f to cope with the TACS supplemental variable data whose line length is over 72 characters. I attach the diff file to tacsup.f and the test data (testtacs.dat)."* The April newsletter mentioned that *"longer FORTRAN lines were allowed ...."* Unfortunately, your Editor's counting was faulty. With an equal sign in column 11, 69 columns are needed to hold the remainder through column 80. But this will not fit on one continuation card, which offers only the 66 columns 7 through 72. I.e., the concept of splitting at the equal sign was flawed. So, it is being modified. This remains the first choice, if there is room. If not, the program first will look for a blank at least 4 bytes to the right. If that is not found, a plus sign, then a minus sign, and finally, a multiplication sign, will be checked --- in that order. One will be found, and the split will be made immediately following it. In order that correct operation be verified regularly, Mr. Kan's data has been appended as a trivial, new 7th subcase of DC-68.

Stu Cook Uses Apple Macintosh

Two different versions of ATP have been executed successfully on a PowerPC-based Apple Macintosh computer by Stu Cook of Just Services in suburban Montreal, Quebec, Canada. This is a continuation of the story in the preceding issue.

The Absoft compiler, now used by Mr. Cook to provide full-speed simulation, deserves further explanation. Recall that previous reported work by Mr. Cook relied upon the Language Systems compiler, and speed of ATP execution was disappointing. This was documented in some detail for the Motorola 68040-based Macintosh Quadra in the April, 1995, newsletter. Well, if one compiler has drawbacks, why not try another? In any case, Absoft is not a new name to ATP developers. It can be found in the October, 1990, newsletter, which summarized Macintosh ATP testing by John Schwartzenberg at Drexel University in Philadelphia, Pennsylvania. Of course, much time has past, and the hardware is different (now RISC PowerPC rather than the former CISC 68020). Absoft problems with ATP generally seem to have been solved, although some details still are being worked on as this story is closed October 2nd.

Nonzero exponentials of zero numbers are a peculiarity of Absoft ATP output. Consider the DC-6 solution as supplied by Mr. Cook on July 28th. For the capacitor from NODE2 to ground, resistance and inductance are zero. Interpretation of the Absoft ATP line begins as follows:

```
Series R-L-C.   0.000E-01  0.000E-01  1.000E+00
```

Lack of SAVE statements for variables in DATA statements was an early objection of the Absoft compiler. More precisely, it was a discovery of the Absoft user (Stu Cook), at execution time. Although not the correct final solution (your Editor agrees with Robert Schultz of the New York City area on this point), Mr. Cook was able to avoid trouble by careful compilation. Quoting from his report dated July 11th: *"All Mac applications have two areas of memory: the heap and the stack. In Fortran, all local variables are placed on the stack and only exist for the time that the routine containing them is current. Once the routine has finished its work and exits, these local*

variables are gone. .... So the -N1 option tells the compiler to allocate all DATA variables to the heap as if they also had been included in SAVE statements."

Comings and Goings

Tom Field of FREEP fame (see the April and July, 1997, issues) has moved from Nashville, Tennessee to Chicago, Illinois. This according to E-mail dated June 25th, when Mr. Field inquired about the ATP licensing status of his new employer, Commonwealth Edison. Of course, CECO is licensed to use ATP materials. Work at CECO sounds interesting and challenging. Mr. Field wrote: *"I am working with a real time simulator here in what appears to be a supportive environment."* No news about FREEP can be provided, unfortunately, since none of consequence has been received. If and when a publishable account might come to the attention of your Editor, it should be covered by this newsletter. The FREEP story is in need of significant and formal continuation or closure.

BPA's Robert Hasibar died suddenly and unexpectedly on July 17th. Circumstances were summarized by colleague Jules Esztergalyos in public E-mail dated July 20th.

Limits on Library Functions

Should library function usage be limited by ATP code? This question was raised by Robert Meredith of the New York City area in private E-mail dated September 23rd. This related an incident in which a demonstration by a manufacturer had failed because TACS had attempted to take the square root of a negative number. After the addition of protection against negative values solved the problem, Meredith observed: *"I suspect that something like a numerical oscillation produced a slightly out of bounds value, whose correction to zero had no real impact on the system. I doubt that the control logic is so screwed up that an actual negative number would be produced."* The manufacturer was using DCG/EPRI EMTP under Win95 whereas NYPA colleague Liana Hopkins was using Watcom ATP under Windows NT. Execution of both programs halted, although with different symptoms.

Your Editor expressed concern later that same day after performing a quick test using Dr. Liu's 486 at BPA. This is equipped with ver. 2.67 Salford software, and it died on the negative square root without any meaningful diagnostic: *"Might be array bound error or corrupt program."* That was Salford ver. 2.67 Following experimentation at home using ver. 2.66 and the DOS of Windows 95, there was doubt, however. The following day, Dr. Liu and this writer investigated further using different program versions on the two Pentium-based computers at her disposal nearby. As reported to Mr. Meredith, *"the problem seems complicated and serious for Watcom and GNU, but not for Salford. My concern last night was premature."*

At home using ver. 2.66, and across the hall using ver. 3.50, DBOS correctly complained of *"Error: Negative argument to square root."* Furthermore, the same TPBIG.EXE that was not killed properly by DBOS ver. 2.67 **does** die properly when run under DBOS ver. 3.50 across the hall. It would seem that the confusing message of ver. 2.67 is a peculiarity. Since DBOS ver. 2.67 has never been shared with the general public, it is hoped that no one else ever will be bothered by the problem.

GNU ATP execution died violently. Executing in a full-screen DOS window with BOTH, suddenly the DOS screen disappeared, and a new, small window appeared. The window title was *"Dr. Watson for Windows NT."* The message stated: *"An application error has occurred and an application error log is being generated. ntvdm.exe Exception: access violation (0xc0000005), Address: 0x0f003084"* No such file was found locally, so WinNT Explorer was used to search (the **Find** command) for it. Two such files were found searching C:, but associated dates were old. So, testing was abandoned. The only good news had to do with the DOS window, which was not killed (it had become an icon).

Version 11 Watcom ATP behaved even worse. Recall Meredith and Schultz continue to use Watcom 10.5 in New York, where execution was halted with a decent message (*"*ERR* LI-03 argument must not be negative"*). Not so in Portland, where execution did **not** stop! From the complete .LIS file, it appears the function was returning zero and allowing execution to continue. There is no Watcom error message in the .LIS file, although a message was written to the screen --- most likely once for each violation. The message said: *"Domain error in sqrt."* This was not a bad message, if the user happened to see it (an obvious problem for batch usage).

Branch Data Input Restructured

List 26 of case-summary statistics (after batch-mode plotting in the .LIS file) often was undefined (value -9999) prior to an addition to OVER3 on July 6th. Following overflow, an error message explained that *"LIST 26 is used to size the short working vectors VOLT, VOLTI, VOLTK, ..."* These *"vectors of List 26 are also used for matrix manipulations involving coupled phases of mutually coupled elements."* Well, Pi-circuits (including coupled R-L as a special case) never contributed to the printed size of List 26. They should have, but they did not. Of course, nobody much cared, for 2 or 3 phases. But the statistic becomes important, and highly noticeable, for high-order Pi-circuits (see separate story).

A table of phasor branch voltages (left half) and branch

currents (right half) ends printed output when no simulation is involved (i.e., when TMAX is non-positive). On July 1st, BPA's Robert Hasibar first reported a defect in this. Whereas column-80 punches were being honored correctly, the alternative -5 in columns 1 and 2, which requests voltage difference between a named pair of nodes, was being ignored. As illustrated by the 10th subcase of DCNEW-21, the negative of the voltage at node LOAD, requested as the difference between ground and this node, was missing prior to correction on July 3rd.

Parameter Variation Studies

POCKET CALCULATOR VARIES PARAMETERS (PCVP) is a new request word that will result in a family of simulations. There is some similarity to the ancient SYSTEMATIC data cases in that the result usually will be predictable rather than random. But the new PCVP offers much greater flexibility. Reliance upon the new pocket calculator allows the user to modify any parameter of input data according to the formula of his choice as the program loops over the solutions. The already-explained $PARAMETER statement is used to specify the rules of parameter variation as a function of loop index KNT and any other program variables the user may want to involve.

Parametric studies are not new. The concept dates to the early days of ATP when SPY was being enhanced to work well on Apollo workstations with their high-speed, scrollable, simultaneous windows and graphics. Unfortunately, Apollo no longer is with us. The Salford version does support SPY, but not as well as Apollo because display is limited to one window at a time, and windows are not integrated with the operating system. Nonetheless, Salford users can find parameter variation in INCLSPY5.DAT (for simulation) and INCLSPY6.DAT (for LINE CONSTANTS). Better than words, examples such as these explain the concept. However, use of SPY APPEND for arithmetic and data-card modification was primitive. The significance of new PCVP is that it has replaced APPEND programming in assembly language by FORTRAN programming of the new pocket calculator. Also, it avoids the need to locate and specify in detail all points of modification. Instead, reliance upon high-level $PARAMETER is used --- comparable to the way the user locates parameters for $INCLUDE use (see rules of DATA BASE MODULE). To conclude, the new PCVP is much easier to use, particularly for large data cases that might require simultaneous variation of several or many parameters in many locations.

Prof. Mustafa Kizilcay of FH Osnabrueck in Germany wrote about parametric studies in E-mail dated August 16th. No doubt, thinking about this message provided important impetus for the development of PCVP during the second half of August rather than later. About parametric studies, Prof. Kizilcay wrote: *"sometimes similar to STATISTICS switch option, it is needed to execute the same data case repetitively, each time with a different value of a certain element in the network. As an example, the influence of a short-circuit reactance of the source network to energization overvoltages can be investigated in a convenient way, if the reactance is changed automatically and the data case is executed several times using different values at each run. This feature would be also interesting for repetitive phasor solutions. My idea is to specify parametric values of a circuit element between two (starting and ending) cards like ...."* Apparently unaware of SPY use for the task, Prof. Kizilcay had envisioned his own dedicated extension to ATP. It was while thinking of problems associated with such dedicated code that your Editor envisioned a better alternative: reliance upon a general tool, the new pocket calculator. PCVP modeling was the result. It should be far better for production use than either rigid, dedicated code or the low-level SPY APPEND of years past.

.PL4 plot files are handled differently for a PCVP study. Naming that is parallel with the input data file seems to be the only practical alternative for Salford DBOS, which limits file names to the eight characters of MS-DOS. I.e., there generally is not room to append serialization as part of the 8-character name. Besides, parallel naming generally is endorsed as a concept for non-PCVP cases, so it is natural to try to employ it for the new PCVP data cases, too. The following convention has been adopted: the usual file type .PL4 has been sacrificed in favor of 3-decimal-digit serialization. So, the .PL4 file of the first shot will have file type .001, the 2nd will have .002, etc. No problem as long as the total number of shots MAXKNT is inferior to 1000. This should satisfy nearly everyone (handling a thousand disk files of consequence is a huge endeavor). But, there always will be some clown who will want more, so overflow into the file name has been allowed. I.e., the right-most byte of the file name will be overlaid by the thousands digit when it becomes nonzero, the byte preceding it by the ten thousands digit, etc., for as many bytes as might be required. No practical limit in sight. Using 8 + 3 = 11 decimal digits this way, the limit exceeds that of the 32-bit integer that is used to store the number of shots!

Extrema of extrema is a new table for the .LIS file of PCVP studies. It will be seen at the end, after the final shot, provided the user requests the output of extrema by positive MAXOUT (the punch in column 40 of the integer miscellaneous data card). The rows of maxima and minima appear exactly as for normal extrema output, but the times that followed such output have been replaced by rows of corresponding energization numbers. These integers allow the user to identify the shot that produced the most extreme values of all output variables. If the user wants the corresponding time of the simulation, and parameter values associated with the data, these can be found by consulting the preceding output for the shot that is referenced, of course. Because the final table is

fundamentally different from the preceding tables that look similar, it is distinguished by separator lines as demonstrated by the new 4th subcase of DCNEW-25.

Timing as reported by case-summary statistics at the end of the .LIS file deserves explanation. What has been done is the following. Times reported for the first four phases of execution will dominate, since these are the sums of the corresponding figures for all of the MAXKNT shots. The fifth phase is only traversed once, so should be relatively small, typically. All five are scaled uniformly in order that their total agrees with the known, actual time since the start of the first shot. Typically the reported total is slightly less than the total reported to the screen by Robert Schultz at the end of execution ("Total Execution Time was xxxx seconds") because it takes time to halt the program. For example, the output buffer may need to be flushed.

IOPCVP is a new control variable that allows the user to suppress printout (the .LIS file) during the second and later shot. Value zero corresponds to full, normal ATP output (no suppression). This is the way parameter studies using SPY worked: the same output was produced over and over again, once for each shot. But the new PCVP is similar to STATISTICS or SYSTEMATIC in that it allows the limitation of output to extrema only. This is the effect of value unity, as illustrated by the 4th subcase of DCNEW-25. It is the recommended, normal mode of operation when the user is not debugging or verifying procedures. Extrema are preceded by two special new lines of output: 1) A line documenting shot number KNT; 2) a line documenting the replacement symbols being used. Both should be self-explanatory, and the first of these two will be seen on the screen if DISK is used. Just as with STATISTICS or SYSTEMATIC, one status line will reassure the user of progress at the start of each energization.

MAXOUT is the miscellaneous data parameter that requests the output of extrema. Normally, this is optional. But for PCVP simulation, extrema are the **only** output, normally. So, benevolent ATP will set MAXOUT to unity if the user fails to define a positive value. This illustrates one of several mandatory *choices*. Another is the plot frequency IPLOT. If the user is not saving .PL4 files (i.e., if ICAT is not positive), ATP will set IPLOT equal to -1 in order to suppress the accumulation of plot data points in RAM as well as on disk.

Use of $PARAMETER with wide data formats began August 30th. For years, R, L, and C have been varied using the old, narrow, 6-column alternative. But practical engineering sometimes requires more precision. Even if six significant digits might be plenty, there sometimes is loss of precision due to the need for either scientific notation or leading zeros of a number less than unity. So, the procedure was modified to allow wider data fields. Of most common interest should be $VINTAGE, 1, with its width of 16 for each parameter of a series R-L-C element. In surprisingly short order, your Editor made this work. That 4th subcase of DCNEW-25 has been generalized to illustrate both wide and narrow formats. There are two parallel, disconnected circuits that give the same answer.

Publishing Programs and Viewers

Unification of Theory Book chapters was mentioned in the preceding issue, but space was in short supply. The continuation of the story now follows. Just one day after attending a week-long course across the continent at ESCA, Robert Meredith provided his solution on May 16th. Having *"Subject: Consolidated Theory Book for posting,"* Mr. Meredith's E-mail explained: *"The attached represents such a document. With Acrobat it is possible to combine several .pdf files into one with the 'insert pages' option. I collected the .pdf files from the Osaka site and combined them into one file and made a table of contents, with 'goto' links. While I was at it, I made the resulting document read only/print only, so it could not be changed without a password. I will send the owner password under separate cover. If you want to make changes to the protected document, you must 'save as' a new document, click the security box and supply the password. (I tried to remove the Agora e-mail address for Scott, but got a notice that I could not edit it without having the font on my system.) I suppose I could delete the page and put it back in with a new look-alike font, but I did not try that yet. .... I believe nothing has changed, except the addition of the contents page."*

Mr. Meredith provided more insight the following day: *"The only 'work' I did was writing down the chapter titles and typing them in Word. I could not figure how to copy titles directly from Acrobat, which was a disappointment. The linking was easy and limited to just the Contents list. If someone wanted to link each reference, it could be done. The only thing I found awkward was having to return to page 2 to change to another chapter. To change that. I suppose one would have to add 'Return to Contents' lines in the source documents, that could be linked in the pdf. .... If you are concerned about security, there is also a user password option. Make it something like ATP_USER and anyone wanting to use it would have a hard time passing it off as anything other than an ATP effort. A header on every page saying 'Produced by Can/Am ATP User Group' could also be effective -- or at least require masking some 483 pages and producing a low quality scanned reproduction. Putting a header on might have to be done before producing the Postscript files. If there is a real security concern, get the unprotected versions off the net ASAP. Any licensed user might let it get into the wrong hands (innocently, of course)."*

Searching for any character string of interest is a dominant advantage of computer storage. Your Editor was concerned that this not be lost --- ability to search the entire book with a single command. Apparently this is not a problem. Mr. Meredith explained: *"About 'Find' -- getting*

all the pieces together in one file enables that feature in the Acrobat reader, without need for a .pdx (index file). (e.g., there are 15 occurrences of the word 'solutions' in the 483 page pdf.) There are more advanced 'indexing' options in Acrobat that do require a .pdx file be distributed along with the entire collection of .pdf documents. I did not immediately see the use of them here, so I did not investigate them. You may have other ideas."

Robert Meredith's contribution was well received in Japan, and quickly handled. E-mail from Masahiro Kan of Toshiba Corporation reported as follows on May 18th: *"Thanks for your nice contribution of the PDF file representing the Theory Book. It is good to read and browse."* One day later, Mr. Kan confirmed completion of the upgrade: *"Dr. Funaki .... replaced the PDF file of the Theory Book with yours in our secure ftp servers in Japan."*

PDF became the official language of newsletters on July 24th when the July issue was released to distributors. The MS Word .DOC files of the past couple of years are being replaced by Adobe PDF files. About file size, we observed: 1) 138 Kbytes is the size of JUL98.DOC (no longer available to the general public); 2) 1871 Kbytes is the size of the PostScript output, which was used as input to Adobe Distiller; 3) 155 Kbytes is the size of JUL98.PDF as released to others; and 4) 145 Kbytes is the size of the PDF following compression by PKZIP. As well known, the final operation gains little (6%) because new PDF files already are compressed. This seems to be the biggest disadvantage: loss of disk space (the MS .DOC file can be compressed to 53 Kbytes). Yet, most seem willing to pay this price (loss of disk space) for universal, free, undistorted reading. The first to inform others of availability was Prof. Bruce Mork of Michigan Tech in Houghton. Later that same day, he announced: *"The July CanAm EMTP News is now available .... The default format henceforth has been declared as PDF. I've printed it out on a 600 dpi printer and it looks very nice."*

Automatic superscripting for numbers is a nice feature of MS Word 7 as used by your Editor under Windows 95 of his Szymanski-supplied Pentium. For example, a blank keyed following the st of 1st or the nd of 2nd will result in 1st or 2nd --- automatically. But Word 6 for Windows NT, as used by BPA on Dr. Tsu-huei Liu's 133-MHz Pentium, does not offer such assistance. Is this just another fundamental difference between MS programs under different MS operating systems? If any reader knows how to instruct Dr. Liu's Word 6 to perform automatic superscripting, the information would be much appreciated. While your Editor would enjoy writing newsletters at Dittmer using Dr. Liu's 21-inch monitor, the necessity of later searching for all dates has discouraged such practice.

Documentation of STARTUP and GRAPHICS are contained in Section I-E of the Rule Book, which first became available in PDF format on July 31st. Dr. Liu's public E-mail later that same day explained: *"Yesterday, the PDF version of this file, h01e.pdf, was zipped and sent to Dr. Funaki of Osaka University for his verification and subsequent placement on the Secure FTP site at his University. .... To download this file, the password is required. If not known, it can be obtained from the ATP user group that licensed your use."*

Frequency Scans and Harmonics

HARMONIC FREQUENCY SCAN (HFS) by Gabor Furst has been the dominant subject of this same story in the two preceding issues. The feature evolves.

DELTAT, the size of the time step, now can be specified in steps per cycle rather than the usual seconds. This is the inspiration of Gabor Furst of suburban Vancouver, B.C., Canada. For persons concerned with harmonics, the accurate specification of DELTAT is important. Typically, such users want an integer number of steps per cycle at the power frequency. While this is easy for 50 Hz, it typically requires many digits for 60 Hz. As an illustration, note DELTAT = .0000462962962962963 in the 4th subcase of DC-22. This represents nearly full precision for a time step of one degree. But why force the user to key 16 decimal digits of precision when all he wants is an integer number (here, 360) of steps per cycle? If the new alternative data is recognized, ATP will convert it to DELTAT and TMAX in seconds, and the interpretation of columns 1-50 will confirm the conversion (accurate to a few digits, anyway). This alternative began August 18th. For details, see comments in the just-mentioned data case.

Disagreement between FS and HFS solutions of the same network was reported by Gabor Furst in E-mail dated August 25th. Of course, there should be agreement, so his data cases TEST_FS and TEST_HFS demonstrated a contradiction. After some study, this was traced to simple, unintended collision between newer scaling of output voltage and current (see BVIV and BCIA in the October, 1997, issue) with older variation of source angles as a function of frequency for James Randall (see the July, 1997, issue). Variable BASEV erroneously shared the same storage as FUNDFS until the former was moved on August 31st. The delay was the result of E-mail troubles (again) at BPA as reported elsewhere in this same issue (see aftermath of Office 97 installation).

Changes to Universal ATP Graphics

MODHFS is a new appendage to FORMATTED .PL4 files that an alert user might notice. It represents the second such appendage, after the original IHSPL4 (search for *logarithm* or *UNFORMATTED* in the July, 1993, newsletter). The first, which was done for Randy Suhrbier of BPA, began with UNFORMATTED .PL4 files --- all he used with DEC VMS. The second has nothing to do

with either VMS or Mr. Suhrbier. As a result, nothing is being done for the UNFORMATTED case. Whereas Mr. Suhrbier correctly found one unused integer, there were not two of them, and your Editor has not yet taken the bold step of modifying header structure. Maybe later, but not now. The initial implementation is being developed for Salford DBOS, with both the most-efficient C-like and the universal FORMATTED alternative covered during the work of May 17th. The following day, Mr. Furst observed that the universal **widexx** required special treatment, and this was provided in the form of an optional 4th row of the header that will be seen only if multiple (2 or 4) parts of each output variable are involved. For the default case of magnitude only, there has been no change.

Conditional Data Assembly

IF-THEN-ENDIF statements are allowed as part of ATP input data as explained in the preceding issue. The story now is being continued.

No one responded to the call for help with environment variables of DEC VMS. As a result, VMS ATP continues to lack both ATPDIR and IF-THEN-ENDIF support. The DEC ATP ship is sinking. Have all serious ATP users already abandoned it?

Data card interpretation in the .LIS file will contain no trace of any IF - THEN, ELSEIF, ELSE, or ENDIF declarations. The logic of evaluation discards all such control lines. So, if the user wants a record of the control, he is advised to use comment cards or in-line comments to mark the data cards themselves. Of course, NOCOMM equal to zero is required to see the comment cards in the .LIS file. Finally, if debugging is difficult, diagnostic printout that will show each line can be turned on by IPRSUP > 8 in STARTUP. Of course, as done for each $INCLUDE card, ATP could automatically make and display a comment line for each controlling line of an IF block. Is this what many users want?

Illustrations of the new IF-THEN structure are being placed in DCNEW-25.

Power Company Politics & Religion

"The Salt River Project, an Arizona electric and water utility created by farmers with federal funding in 1903," might be providing an illustration of the political problems BPA later might face, as its commercialization proceeds. The story is told on page B18 of the April 2nd issue of *The Wall Street Journal*. This begins with some explanation about *"California's electricity market, now officially open for competition."* Of course, SRP is there competing successfully: *"Its biggest prize is a contract to provide Mobil Corp.'s California operation with electricity and energy services. Competitors are crying foul, and doing their best to block Salt River Project. Their beef: Salt River Project, as a quasigovernmental entity, has built up its electric empire of dams, power plants and nuclear power through cheap sources of capital, namely tax-free bonds. .... Investor-owned rivals argue that they are at a disadvantage ..."* The SRP debt may be small by BPA standards, but sizable nonetheless. The article mentions *"$3 billion in tax-exempt bonds."* State and federal taxes on the income would represent no small amount of money.

Interactive Plotting in MS Windows

GTPPLOT comes from Orlando Hevia of Universidad Tecnologica Nacional in Santa Fe, Argentina, as explained in the preceding issue. Since the initial announcement, the program rapidly was extended to offer numerous power quality indices. Mr. Hevia explained this in public E-mail dated June 27th. He showed illustrative output that included the following labels: 1) RMS value .... 2) THD .... 3) THD EVEN .... 4) THD ODD .... 5) THD balanced .... 6) THD residual .... 7) K factor (for transformer derating) .... 8) Total TIF IEEE Std. 519 .... 9) Balanced TIF IEEE Std. 519 .... 10) Residual TIF IEEE Std. 519 .... 11) Total C message index .... 12) Balanced C message index .... 13) Residual C message index .... 14) Total I.T or V.T product IEEE Std. 519 .... 15) Balanced I.T or V.T product IEEE Std. 519 .... 16) Residual I.T or V.T product IEEE Std. 519 .... 17) Distortion index DIN = THD / SQRT (1 + THD**2) .... 18) Mean value Vmean .... 19) Peak value Vpeak .... 20) Crest factor Vpeak/Vrms .... 21) Form factor RMS/Vmean

Masahiro Kan of Toshiba Corporation has taken Mr. Hevia's source code for GTPPLOT and modified it for use with Mingw32. Quoting from E-mail dated June 12th, changes included: *"*) Enabling a long filename support; *) Fix the abnormal termination for C-like PL4 file; *) Fix the abnormal termination for ATP/Watcom C-like PL4 file; *) Fix the abnormal termination for GNU unformatted PL4 file with a big TACS dimension. It worked fine under Win95 (and maybe WinNT). I attach the modified source code (gtpplot.f)."*

Miscellaneous Intel PC Information

Voice-recognition software was given a strong recommendation by American radio talk show host Art (*"The Quickening"*) Bell, who dominates the nights as Rush Limbaugh once did the days. It was January 10th that your Editor heard the testimonial for PCs that are at least as powerful as Mr. Bell's 133-MHz Pentium. The software is not free. So, not wanting to provide a free commercial, Mr. Bell refused to name the product that he had purchased. But he clearly conveyed the message that it works well for dictation. Of course, this followed substantial voice

training. Mr. Bell explained that he read a chapter from the 1968 novel *"2001: A Space Odyssey."* It also should be explained that Art Bell is a relatively slow and clear talker. Nonetheless, use of voice recognition is intriguing for SPY.

"Microsoft now most valuable company" is the headline of an AP story on page E2 of *The Columbian* (Vancouver's dominant daily newspaper) dated September 15th. MS replaced GE. MS *"is still a significantly smaller company with $14.5 billion in revenue during its past fiscal year compared to $88.5 billion for GE. In July, General Electric became the first company to surpass $300 billion, reaching a high of about $315.3 billion. Its stock ... has since fallen more than 20 percent."*

Miscellaneous Small Items

Removal of the Fontaine address from LICENSE.ZIP was made June 19th --- before your Editor's Unit 6B was sold, in fact. But a copy was not sent to Prof. Bruce Mork until July 24th, unfortunately. Later that same day, Prof. Mork confirmed the update: *"The new license file is in place, renamed as pub/atp/license/canam/license.zip"*

A continuation line of $INCLUDE use required that columns 1 and 2 either be blank or carry the continuation symbol (nominally "C "). This was prior to improvement on August 1st in response to E-mail from Bernd Stein of FGH in Mannheim, Germany. The complaint involved a case with just a single blank on the left, but an equally good case would involve no blanks on the left. Recall no particular number is required since the $$ at the end of the preceding line indicated the need for continuation. So, either no blanks or one blank on the left now is correctly handled --- provided two blanks exist on the right. I.e., columns 79 and 80 must either be blank or be part of an in-line comment of the line. To summarize, a continuation line must have 2 blanks on either the left or the right, if it does not carry the continuation symbol in columns 1-2.

TSTALL of STARTUP was introduced in the preceding issue. But there was not room for a warning. Addendum on July 9th: Use of a new TPBIG with an old STARTUP file leads to obvious trouble, readers are warned. With 40 seconds (the PIXPUN value) of delay accompanying each output line, the user will know something is drastically wrong, but he might not imagine or remember how to correct it. Precisely such a situation was the subject of public E-mail dated July 6th and 7th. Eventually, the user looked inside the .DBG file and understood the meaning of lines such as: "TFLUS2 ready to waste TSTALL = 40.00000000000 seconds." In order that such confusion never occur again, a limit of 0.5 seconds has been placed on TSTALL. This began July 12th. While ATP will allow larger delays, it will issue a 3-line warning to the screen in such cases. The monitor will be notified instantly of the reason for subsequent delays.

Environment variable ATPDIR is supported by all Intel-based PC versions of the program (Salford, Watcom, and GNU), and these dominate ATP usage. So, the average ATP user should know about ATPDIR (for Watcom, the name was changed to WATDIR), whether or not he chooses to use it. Background can be found in previous issues of the newsletter beginning with April, 1996. Protection against one possible case of misuse is being added following the difficult experience of Khin Swe at Ebasco Services in New York City. For Salford EMTP that she had chosen to use, SET ATPDIR=C:\ATP\ is the recommendation in ATPSETUP.LIS on the GIVE2 disk. This bears some similarity to familiar DOS commands such as CD C:\ATP or DIR C:\ATP But note the critical difference: The content of ATPDIR ends with a trailing backslash whereas there is none in the other two examples. It is understandable that users might from time to time forget the trailing slash, and that the resulting rejection might not be understood by a new user. STARTUP is the first file to be connected, and without the trailing slash, DBOS will kill execution while complaining that a file such as C:\ATPSTARTUP can not be found. No longer. Beginning July 15th, all-knowing, all-caring ATP will stop itself after writing the following message to the monitor: "Error. Environment variable ATPDIR is defined, and involves a directory. However, it does not end with the directory symbol as required." Note that a backslash is not mentioned explicitly because the forward slash of Unix (e.g., GNU ATP use under Linux as supported by Masahiro Kan of Toshiba Corporation in Japan) also is considered. The rule is simple: if one such symbol is present, the non-blank string must end with it. About the stop and writing to the monitor, it might be explained why standard, more-elegant procedures were not used. The protection has been added at the start of execution, before variables of STOPTP and OUTSIX have been initialized. The hard way, it was learned that usual modularized procedures are not yet operative.

STARTUP.FRE is the free-format alternative to the original fixed-format STARTUP file. This was explained in the January, 1995, issue. The questions now being asked are these: Does anyone use free-format STARTUP with an ATP version other than Salford? If so, is there any concern about speed or reliability? Some thought is being given to reworking the procedure in RSTART where the distinction between alphanumeric and numeric variables is made. A free-formatted READ now is used, and this leads to a Salford warning: *"The use of list-directed formatting with an internal file is a Fortran 77 extension."* It would be nice to remove the warning and improve universality at the same time. Maybe no reader cares (is there any concern about how this might be changed?).

KOMPAR is the variable of STARTUP that allows easier comparison of old and new solutions. In particular, the date and time can be set to correspond to the armistice that ended World War I. That was all part of the original

meaning, for values between zero (no such help with comparisons) and four (maximum help). Then, on April 19th, the control was extended to allow reduced precision of phasor printout (branch flows and injected currents). Values greater than 6 and less than 17 will be assumed to be a specification of desired output width. Since the default width is 16, any smaller number should ease comparisons that otherwise might be plagued by roundoff. Finally, if this column width is less than KOLWID, the width of time-step loop output, KOLWID will be set equal to this special value of KOMPAR. So, although the new control is limited to two output structures, these structures are both important and repetitive. Unlike program TRUNCATE (see the April, 1995, issue), the new control using KOMPAR should be universal, so can be used with any computer and any compiler.

$DUMMY is the declaration to initialize the serialization of dummy nodes that are created during $INCLUDE evaluation. For an illustration, see standard test case DC-64. About limits, 999 has existed since year one as recognized by Bernd Stein of FGH in Mannheim, Germany. In E-mail dated July 21st, he wrote to report overflow. In response, your Editor expanded the limit to 9999 later that same day, and also added protection against overflow (KILL = 235). For nodes numbered greater than 999, the right-most (third) byte of the root name will be overlaid by the digit 1. This is a new rule. So, for example, after DUM999 will come DU1000. If any reader can think of a better idea, he is welcome to explain his thinking in E-mail of the Fargo list server. About use (when it comes to record-breaking performance, there always is interest in what is being modeled), it was suggested that a formal report could be made to EEUG. If and when it might be, mention should be made in this newsletter, if your Editor is informed.

Text for the .LIS file might erroneously be written to the start of the input data file. This disturbing behavior first was reported to developers in Portland by Bernd Stein of FGH in Mannheim, Germany. In E-mail dated July 23rd, he summarized an incident involving PFE as advocated by Prof. Mustafa Kizilcay of FH Osnabrueck. At first your editor tried to deny responsibility as follows: *"This is not an ATP issue, however. ATP does not know about PFE. This is between you and Mustafa .... He gives you the gun and the bullet; you shoot yourself in the foot."* But Prof. Kizilcay responded as follows: *"I observed the same problem with PFE several times in the past, too. I investigated this behaviour with and without PFE. In the end, I concluded it is really an ATP bug reproducible using both Salford and Watcom ATP. In my case, the trouble occurred when a $INCLUDE file could not be accessed by ATP."* To observe the trouble, a number of unlikely circumstances must converge. Most important is the input file name. Those who end this with a period are protected (Prof. Kizilcay instead ended his input file name with .dat). Second, there is buffered output of LU6VRT > 0 (the problem is associated with such use). It also is associated with the conversion of all input data to upper case as requested by KINSEN = 1. ATP checks for an input file ending in .dat but it failed to find one because Prof. Kizilcay's name by that time had been converted to upper case. Anyway, more protection was added July 27th. Upper case .DAT, too, now is checked. If that fails, any period will be accepted as the location at which .LIS will be overlaid to produce the output file name. Finally, if there is no period at all, the default name dumout.lis will be assigned. This logic, added to universal TFLUSH, should be bulletproof.

The *year 2000 problem* was raised with respect to ATP by Vicente Figueroa-Castillo of the Puerto Rico Electric Power Authority in San Juan. Quoting from E-mail dated May 6th: *"We use the EMTP/ATP and TPPLOT you sent us ... Can you, please send an official notification for the compliance of this software?"* Your Editor was not very sympathetic. Quoting from his response: *"... all bureaucracies must by now have committees on this subject (joke)! .... If you really are concerned, why not issue public E-mail via Fargo? .... If you use Salford DBOS, as I do, at least this ensures that we will be surprised together! No one has raised the issue before because no one believes there will be a problem, as far as I can imagine. .... Having said that, I recall some weeks or months ago that Robert Meredith had tested his computer by setting the calendar ahead to 2002 or some such year. If you are concerned, do it yourself. The only ATP-related effect that I can think of would occur when the date was used to name a .PL4 file. I assume DBOS will return 00 after 99, so alphabetical sorting will wrap the bottom around to the top. But who cares? If this is a problem, files could be renamed."* Your Editor ended his E-mail about the requested *official notification of compliance*: *"We could, but we won't. Sorry, we have better things to do with our time."* Does any reader disagree? If so, let's hear the reasoning (preferably publicly, via the Fargo list server). As for mainframes, is anyone using them any more to support ATP? That's the good news : the change to personal computers (including what once were called workstations) seems about complete. ATP users won while DEC paid with its life. What a decade (1988 to 1998).

An occasional extra blank line might be noted in .LIS files following a change to text handling by TFLUSH on August 26th. Specifically, it was discovered that the request for a blank line at the end of text output was not being honored. The problem is believed to be a decade or more old, so a lot of history is involved. Some of the new blanks were intended, but have been missing for so long that it seems easiest and best to omit them by editing the data (KILLCODE). The most prominent example of this is the second blank line that separated phasor branch flows. It is **not** being added. Another example is some blank lines within the initial-condition printout of the Type-59 S.M (also rejected). On the other hand, some new blanks are viewed as progress, so are being accepted. The most prominent example is the blank line that now follows the

listing of punched cards (previously, there was a blank in front of the listing of $PUNCH output, but none following it). Although unrelated to it, all of these changes occurred in the middle of PCVP changes (see separate story).

Prof. Hermann Dommel's MicroTran has never been used by anyone with whom your Editor is in working contact. So, your Editor is unqualified to offer an opinion of operation, although the program is mentioned from time to time. For example, prices can be found in the July, 1996, issue. About operation and relationship to ATP, an opinion was learned September 1st when Siegfried Ruhland issued E-mail of the Fargo list server to inquire about *"Subject: MicroTran to ATP converter."* He explained his interest as follows: *"Up to now I used MicroTran (MT), from Microtran Power Systems (MPS), for transient simulations. Due to the poor support of MPS I decided to change to ATP. As I have a lot of input files of MT and as there are slight differences bewteen MT and ATP, I am looking for a conversion tool."* About differences, it should be explained that some are small while others are huge, obviously. It depends what modeling is being considered. The nonexistent or small differences would apply to modeling that predates Prof. Dommel's departure from BPA in July of 1973. At least the direction of the proposed conversion is right. I.e., in general, old data can be modified to work with current ATP. The reverse is not generally true. There is a lot of ATP modeling that should be fundamentally incompatible with MicroTran, with Laurent Dube's MODELS probably being the best-known example. If Prof. Dommel does not offer MODELS or anything comparable, how could anyone convert such data for use by his program? About support, readers are reminded that Prof. Dommel does have his own Internet forum, even though it seemed not to be used much at the time of consideration (see the story involving address emtp@ee.ubc.ca in the October, 1997, newsletter).

About EPRI and DCG, it seems official that the former has slipped from its original position as an equal partner dominating the latter to a member of the latter. This first was learned from Masahiro Kan of Toshiba Corporation, who reacted to your Editor's speculation in the April newsletter. From E-mail dated April 28th: *"EPRI signed the new DCG agreement and is a member of the new DCG now. The agreement is valid until Dec. 31, 2000. The objective of the agreement is to maintain EMTP96 and to develop EMTP-RV (restructured version)."* Through BPA, Dr. Liu received photocopy that seems to confirm the change. Included was page 91 of some spiral-bound book, with the bottom half of the page involving EMTP. The top margin shows *"EPRI membership offering 99"* on the left and *"Target 57; Grid Planning and Development; Home Market Segment: Transmission"* on the right. Under the title, one reads the summary line: *"Speed analysis and identification of solutions for power system transients."* In the explanatory paragraph that follows: *"Although EMTP V. 3.0 is being used successfully by utilities, users have noted that usability and integration of new models are hampered by the ancient structure of the program. The Development Coordination Group (DCG), of which EPRI is a member, has undertaken restructuring of EMTP to improve the speed and usability of the product. The EMTP code will be rewritten, using a structured language that will streamline the interaction of various modules such as time-step loop, input and output data, and TACS. The restructured EMTP is expected to reduce simulation and data preparation time by at least 30%. This project is cofunded, with a majority of funding coming from other DCG members."* There follows a table that ends with the date 4Q02 (4th quarter of the year 2002).

START AGAIN may involve a minor error --- an order of DELTAT error --- if tables are saved using a program older than August 20th. Standard test cases DC-32 and 49 illustrated the procedure. I.e., DC49.LIS was imperfect for many years, although no one seemed to have noticed. For engineering purposes, the procedure worked. In fact, the present mention is a continuation of a paragraph in the January, 1994, issue. Recall the assessment that *"the loss had no engineering significance since the first half was merely drifting in the quasi-steady state, anyway."* That 1994 explanation mentioned TACS and DC-33 whereas the present observation involves a similar problem with the multi-phase Pi-circuit (present in DC-32 via the 2x2 [A] and [R] representation inherent in saturable TRANSFORMER modeling). David Francis of BHP Engineering in Australia first reported trouble using public E-mail dated August 16th. In his response, your Editor asked for illustrative data, and this was promptly supplied. But, after the trouble was diagnosed, it was realized that DC-49 should provide a much smaller and simpler illustration; and this proved to be true. Following correction, a copy of Salford TPBIG was E-mailed to Mr. Francis on August 20th along with a request that any further trouble be reported. None has been.

The IPST'97 conference proceedings book is lacking a copyright notice inside the front cover, assuming one can believe public E-mail on the subject. First, your Editor had suggested that papers from conferences generally were copyrighted, so may not, in general, be made available to others via the Internet. But in public E-mail dated March 24th, Prof. Laszlo Prikler of T.U. Budapest indicated that no copyright notice was found inside the cover of the IPST'97 conference record. Any reader who might understand the thinking of IPST management on the issue is urged to share this with others (including your Editor). The only possibility that occurs to your Editor is simple carelessness on the part of whomever had responsibility. The material clearly is not in the public domain, having been authored by many persons in many places. According to Prof. Prikler, no IEEE PES-like transfer of rights from authors to the publisher was involved. So, under U.S. law, the rights associated with each paper would remain with the authors of that paper. Why would IPST management not protect those rights with a customary copyright notice? Is there any explanation other than negligence? What was the situation two years earlier (in Lisbon, Portugal)?