N E W S L E T T E R # 1 4

March 1986

Aloha from Hawaii! The Soft Warehouse Newsletter provides you with information on new Soft Warehouse products, and software extensions or corrections to existing products. In addition, the newsletter is a medium for the exchange of ideas and application programs within the growing community of **muMATH** and **muLISP** users.

If you would like to subscribe, or extend your subscription to the Newsletter for three issues beyond the expiration number on your mailing label, please send $6 ($10 for orders from outside the U.S. or Canada) by check, VISA, or Master Card to Soft Warehouse, Inc., 3615 Harding Avenue, Suite 505, Honolulu, HI, 96816, U.S.A. A complete set of back issues is available on request for $15 ($20 for orders from outside the U.S. and Canada).

C H A N G E O F A D D R E S S

Soft Warehouse, Inc. is moving up in the world: up to the 5th floor of the brand new Kaimuki Business Plaza. Henceforth please address all Soft Warehouse correspondence to 3615 Harding Avenue, Suite 505, Honolulu, HI, 96816, U.S.A.

N E W P R O D U C T A N N O U N C E M E N T S

Microsoft Corporation has just released **Microsoft LISP**[tm]. This product is also being sold under the trade name **muLISP-86**[tm] by Soft Warehouse, Inc. This version adds to the '85 version of muLISP fifty more primitively defined Common LISP functions, a DEFSTRUCT facility, and a help facility. Also the muLISP Reference Manual has been reprinted and repackaged in a more attractive format.

A native code **muLISP-86 compiler** has also just been released by Soft Warehouse, Inc. Depending on the application, it can improve the speed of the already fast muLISP-86 interpreter by a factor of three or more. The machine code produced by the incremental muLISP compiler is virtually 100% compatible with the interpreter.

Microsoft LISP is now available from Microsoft. For pricing and availability contact your local Microsoft dealer. **muLISP-86**, with **or** without the compiler, is available directly from Soft Warehouse, Inc.

U P C O M I N G C O N F E R E N C E S

April 29 - May 1, 1986: The Artificial Intelligence and Advanced Computer Technology Conference & Exhibition (AI '86) will be held in the Long Beach Convention Center, Long Beach, California. For more information, write Tower Conference Management Co., 331 W. Wesley St., Wheaton, IL, 60187; or telephone (312) 668-8100.

July 21 - July 23, 1986: The SYMSAC-86 Computer Algebra Conference will be held at the University of Waterloo. For conference information write Professor Keith Geddes, Computer Science Department, University of Waterloo, Ontario, N2L 3G1, Canada. The program chairman is Professor David Yun, Computer Science Department, Southern Methodist University, Dallas, TX, 75275.

July 30 - August 1, 1986: The Computers in Mathematics Conference will be held at Stanford University, Stanford, California. For conference information write Computers in Mathematics Conference, P.O. Box 218, Yorktown Heights, NY, 10598. The program chairman is Professor James Davenport, School of Mathematics, University of Bath, Claverton Down, Bath, BA2 7AY, England.

August 11 - August 15, 1986: The fifth National Conference on Artificial Intelligence, AAAI-86, will be held in Philadelphia, Pennsylvania. For more information, write American Association for Artificial Intelligence (AAAI), 445 Burgess Drive, Menlo Park, CA, 94025-3496; or telephone (415) 328-3123.

September 23 - September 25, 1986: The Artificial Intelligence and Advanced Computer Technology Conference/Exhibition (AI Europa '86) will be held at Rhein-Main-Halle, Wiesbaden, West Germany. For more information, write TCM Expositions, Ltd., Exchange House, 33 Station Road, Liphook, Hants GU30 7DN, England; or telephone (44) 0428-724660; telex 859438 TOWER. In the U.S.A. telephone (312) 668-8100.

* * * * * * * T h e m u M A T H e m a t i c i a n * * * * * *

Apple II ADIOS to Pascal Conversion Utility

Professor J.A. Burt of York University has been using muMATH to symbolically test algorithms for use in Pascal programs. As part of this project, he has devised a procedure for converting files generated by muMATH running under the Apple II ADIOS operating system to files formatted for Apple Pascal and vice versa.

If you would like a copy of the conversion procedure and a listing of the Pascal routine, you can write Professor Burt at the Department of Physics, York University, 4700 Keele Street, North York, Ontario, M3J 1P3, Canada.

muMATH Users

The following people have expressed an interest in communicating with other muMATH users for the exchange of application programs and user tips. Send the Newsletter editor your name and address if you want to be listed in the next issue.

William G. Ansley, 9 Portmore Lane, Framingham, MA 01701
James Hurt, Cognition, Inc., 50 Cambridge St., Burlington, MA 01803
Kenneth R. Meyer, Boston U., Mathematics, Boston, MA 02215
Eloise Kleban, U. of Maine, Computing Center, Orono, ME 04473
Michael R. Meck, 1199 Whitney Ave. Apt.522, Hamden, CT 06517
Mark S. Procop, 16 College St., Clinton, NY 13323
D.J. Hebert, U. of Pittsburgh, Math Dept., Pittsburgh, PA 15260
Louis M. Pecora, Naval Research Lab, Code 6631, Washington, DC 20375
Doyle Rhynard, Space Applications Corp., 8230 Boone Blvd. #350, Vienna, VA 22180
Terry L. Harter, 202 Overlook Ct., Fredericksburg, VA 22405-1956
Tom Goldring, BDM Corp., 2227 Drake Ave., Huntsville, AL 35802
William W. Rowell, Rt. 14 Box 97-10, Gray, TN 37615
David Dunthorn, C F Systems, 908 W. Outer Dr., Oak Ridge, TN 37830
Roberto Artigas Jr., 7804 Stanley Rd. #160, Powell, TN 37849
Victor E. Wright, 5502 Random Way, Louisville, KY 40291
Donald Wisander, NASA/D23 2, 10016 Valley View Dr., Columbia Station, OH 44028-9610
Dr. Claus Bornakke, U. of Michigan, Dept. of M.E. & A.M., 305 Autolab, Herbert Dr., Ann Arbor, MI 48109-2121
Gordon Kane, U of Michigan, Randall Physics Lab, Ann Arbor, MI 48109
Dr. Harold H. Johnson, Trinity College, 590 S. Brys Dr., Grosse Pointe Wood, MI 48236
Michael C. Kelly, 3836 Red Cedar Lane, Harvey, LA 70058
Gary T. Boswell, 9221 Clover Valley Dr., Dallas, TX 75243
Dr. Keith Williams, Tyler Engr. Dept., POB 2039, Tyler, TX 75710
Chris R. Warber, Pacific-Sierra Res.Corp., 12340 Santa Monica Blvd., Los Angeles, CA 90025
Taylor Trowbridge, 418 Rose Ave. #7, Venice, CA 90291
Stephen Witkowski, Sunburst Assoc, 23265 Cindy Ct., Bend, OR 97701
Grahame Wilson, Satcom Eng. & Comm., P.O. Box 145, Glebe, NSW 2037, AUSTRALIA
Faculty of Engineering, Royal Melbourne Inst. of Tech., 124 LaTrobe St., Melbourne, Victoria 3001, AUSTRALIA
David Halprin, Apple Users Society of Melbourne, P.O. Box 186, North Carlton, Victoria 3054, AUSTRALIA
P. H. Makin, Ander Goldenen Stiege 21/4/3, Moedling A-2340, AUSTRIA
Duncan Murdock, Health & Welfare Canada, Environmental Health Centre Rm. 120, Ottawa, Ont. K1A 0L2, CANADA (math statistics)
Brad Brown, Information Engineering, 175 Glenview Ave., Toronto, Ontario M4R 1R4, CANADA
Minoru Ichimura, Okayama U. of Science, Applied Mathematics, 1-1 Ridaicho, Okayama 700, JAPAN
Robert B. Davies, 16 Gloucester St., Wellington 5, NEW ZEALAND
Jose Ali Moreno, U. Central de Venezuela, Fisica y Matematica, A.P. 47906 Los Chaguaramos 1041 A, Caracas, VENEZUELA
Dr. Georg Erber, Zillestr. 99 D, 1000 Berlin 10, WEST GERMANY
Alex H. Blin, Eichendorffstr. 1, Donaustauf D-84050, WEST GERMANY

CURLy bugs

We thank Peter Joseph of Upper Darby, Pennsylvania for pointing out that the definition of the curl operator in file VECDIF.VEC should be

```
PROPERTY CURL, [, FUNCTION (EX1),
    % fluid: SF, SFPROD & COORDS.  Modified by PMJ, Feb 11, 1986. %
  EX1: ADJOIN ('[, EX1),
  ADJOIN ('[, LIST (
    (DIF (EX1 [3] * SF [3], COORDS [2])
      - DIF (EX1 [2] * SF [2], COORDS [3]))/(SF[2]*SF[3]),
    (DIF (EX1 [1] * SF [1], COORDS [3])
      - DIF (EX1 [3] * SF [3], COORDS [1]))/(SF[1]*SF[3]),
    (DIF (EX1 [2] * SF [2], COORDS [1])
      - DIF (EX1 [1] * SF [1], COORDS [2]))/(SF[1]*SF[2]))),
ENDFUN $
```

After making this change, also change the date at the top of the file to 02/20/86.

Those !#@* characters!

If OBLIST () has names containing unprintable characters such as some control characters used by the PDS editor, then displaying OBLIST () causes some insufficiently IBM-compatible computers to misbehave. The following function alleviates this difficulty by returning a reversed copy of OBLIST () in which names beginning with such characters are omitted:

```
FUNCTION PRTBLOBL (
    % Local: % OBL, ANS),
  OBL: OBLIST (),
  LOOP
    BLOCK
      WHEN 31 < ASCII (FIRST (OBL)) < 127,
        PUSH (POP (OBL), ANS) EXIT,
      POP (OBL),
    ENDBLOCK,
    WHEN ATOM (OBL), ANS EXIT,
  ENDLOOP,
ENDFUN $
```

Overloaded Backquote

Some people would like to use the ` derivative abbreviation and the matrix transpose in the same environment. This can be done by changing every instance of ` to a name such as TP in file MATRIX.ARR, then deleting the line beginning "COND" in that file. If you would then prefer TP to use functional syntax rather than be a postfix operator, then also delete the line

```
PROPERTY LBP, TP, 170 $
```

Tracing numerous functions

For debugging or gathering statistics on a package under development, it is sometimes desired to trace all or most of the functions introduced after some point in a file. With file DEBUG.MUS loaded, the following function will trace the indicated function and all functions having more recently introduced names:

```
FUNCTION TRACEAFTER (FUNNAM),
  APPLY ('TRACE, MEMBER (FUNNAM, REVERSE (OBLIST ()))),
ENDFUN $
```

The following function will clear tracing from the indicated function and from all functions having more recently introduced names:

```
FUNCTION CLEARAFTER (FUNNAM),
  APPLY ('CLEAR, MEMBER (FUNNAM, REVERSE (OBLIST ()))),
ENDFUN $
```

A good place to put these functions is at the end of DEBUG.MUS.

Tracing much-used muSIMP functions

File DEBUG.MUS uses renamed copies of basic muSIMP functions such as APPLY to prevent distracting extra output and infinite looping when you trace one of those functions. However, PARSE, DRIVE and PRTMATH do not use renamed copies, so you may still encounter these difficulties when tracing such functions. If you encounter this problem and PRTMATH is the culprit, then setting MATHTRACE: FALSE will avoid the problem.

It is rarely helpful to trace much-used muSIMP functions such as FIRST, ADJOIN or ': without using the DEBUGIN feature and perhaps other constraints such as MINLEVEL, MAXLEVEL, and MINCALL. Even these techniques may be insufficient to prune the tracing to a succinct but revealing amount. If you are interested in tracing only a few invocation points of such a function, try temporarily renaming those invocations to a unique new name such as MYFIRST, then do

```
MOVD ('FIRST, 'MYFIRST);
   TRACE ('MYFIRST);
```

In the case of an operator such as ':, you must also either modify the altered invocations to functional notation such as

```
MYASSIGN (..., ...)
```

or you must make your altered name be an operator with the same binding powers as the original -- for example

```
PUT ('LBP, 'MYASSIGN, GET ('LBP, ':)) $
PUT ('RBP, 'MYASSIGN, GET ('RBP, ':)) $
```

Automated Building of SYS files

Duncan Murdoch -- Ottawa, Canada

Duncan writes: "If a file being read opens a second file by executing an RDS with an argument, then the RDS() at the end of the second file should act as a return to the first file rather than straight to the console. Where this is very useful is in building SYS files. I've enclosed a new definition for RDS that accomplishes this. ... One problem is that the file EDIT.DIS does not end in an RDS(); ... this should be added to that file ..."

```
COND (WHEN NOT GETD ('OLDRDS, TRUE),
  MOVD ('RDS, 'OLDRDS),

FUNCTION RDS (FNAME, FTYPE, DRIVE,
    % Local: LASTPTR),
    % Alters outside variable FILESTACK. %
  BLOCK
    WHEN RDS, EXIT,
    FILESTACK: '(),
  ENDBLOCK,
  WHEN EMPTY (FNAME),          % Go to next line in previous file. %
    FILESTACK: RRREST (FILESTACK),
    LASTPTR: POP (FILESTACK),
    OLDRDS (FIRST (FILESTACK), SECOND (FILESTACK), THIRD (FILESTACK)),
    READPTR (LASTPTR),
    RDS EXIT,
  PUSH (READPTR (0), FILESTACK),
  PUSH (DRIVE, FILESTACK),
  PUSH (FTYPE, FILESTACK),
  PUSH (FNAME, FILESTACK),
  OLDRDS (FNAME, FTYPE, DRIVE),
ENDFUN, EXIT) $
```

After this redefinition is done, a ".BLD" file can have a sequence of RDS commands that build an environment. For example, starting with ARITH.MUS, ALGEBRA.ARI, and the above redefinition loaded, we could have a file

```
% File  ALG2CALC.BLD %
DEMO: FALSE $
RDS ('ARRAY, 'ARI) $
RDS ('DIF, 'ALG) $
RDS ('INT, 'DIF) $
RDS ('LIM, 'DIF) $
RDS ('SIGMA, 'DIF) $
RDS ('LOG, 'ALG) $
RDS ('TRG, 'ALG) $
RDS ('ATRG, 'TRG) $
COND (WHEN SAVEIT, SAVE ('CALCULUS) EXIT) $
RDS () $
```

The environment will automatically be saved if the value of SAVEIT is nonFALSE. As a convenience, the conditional redefinition of RDS could be placed at the beginning of every such .BLD file.

Picard Iteration

Newsletter 12 listed a program that generates truncated Taylor series solutions to differential equations. Picard Iteration provides an alternative technique for constructing a truncated series solution: Given an equation of the form $y` = f(t, y)$ subject to an initial condition $y(t_0) = y_0$, we can multiply by dt, then integrate both sides, then add y_0 to both sides to derive an equivalent integral equation:

$$y = y_0 + \int_{t_0}^{t} f(t, y)\, dt.$$

This suggests computing succesive approximations y_1, ... by the iteration:

$$y_k \text{ <-- } y_0 + \int_{t_0}^{t} f(t, y_{k-1})\, dt, \quad k=1,2,...$$

The function PICARD accomplishes this with the aid of the prerequisite file INT.DIF:

```
FUNCTION PICARD (RHS, INDEPVAR, DEPVAR, INITINDEP, INITDEP, NITER,
    % Local: %  APPROX),
  APPROX: INITDEP,
  LOOP
    WHEN NEGATIVE (NITER: NITER - 1) OR NOT FREE (APPROX, 'DEFINT),
      EXPAND (APPROX) EXIT,
    APPROX: INITDEP + DEFINT (EVSUB (RHS, DEPVAR, APPROX),
                                     INDEPVAR, INITINDEP, INDEPVAR),
  ENDLOOP
ENDFUN $
```

As examples of its use with the problem $y` = x + y^2$, $y(0) = 0$:

```
? PICARD (X + Y^2, X, Y, 0, 0, 3);
@: X^2/2 + X^5/20 + X^8/160 + X^11/4400

? PICARD (X + Y^2, X, Y, 0, 0, 4);
@: X^2/2 + X^5/20 + X^8/160 + 7/8800 X^11 + 3/49280 X^14 +
   87/23936000 X^17 + X^20/7040000 + X^23/445280000
```

Note that the coefficients of the highest order returned terms may not have completely converged to their correct values. However, comparison of the returned results for successive values of NITER can indicate which terms have converged.

Picard iteration can be adapted to systems of first order equations, hence to systems of higher order equations.

In comparison to the Taylor series method, Picard iteration has the advantage of no need for differentiability, but the disadvantage of requiring repeated integrability.

muLISP Users

The following people have expressed an interest in communicating with other muLISP users for the exchange of application programs and user tips. Send the Newsletter editor your name and address if you want to be listed in the next issue.

Dr. Stanley Schwartz, The Memorial Hospital, Dept. of Pathology, Pawtucket, RI 02860

Mark S. Procop, 16 College St., Clinton, NY 13323

Dr. William Roy Hall, 113 Kilbuck Dr., Monroeville, PA 15146

Michael E. Rothrock, MERTEK, 1301 Serena Dr., Winter Park, FL 32789

David Dunthorn, C F Systems, 908 W. Outer Dr., Oak Ridge, TN 37830

Donald Wisander, NASA/D23 2, 10016 Valley View Dr., Columbia Station, OH 44028-9610

Dr. Harold H. Johnson, Trinity College, 590 S. Brys Dr., Grosse Pointe Wood, MI 48236

Rich Whale, 14278 Watson Rd., Bath, MI 48808

Gary T. Boswell, 9221 Clover Valley Dr., Dallas, TX 75243

Ms. Pat Miller, 6898 S. Broadway, Littleton, CO 80122

Ray F. Shaw/Loren C. Lively, Lively & Associates, P. O. Box 938/3575 Palm Dr., Twentynine Palms, CA 92277

Kenneth Lui, 17331 Poplar St., Fountain Valley, CA 92708

Dr. DeHoff, Systems Control Technology Inc., P.O. Box 10180, Palo Alto, CA 94303

John Covington, 4821 Kent-Des Moines Rd. #306, Kent, WA 98032

Fred J. Martin, British Columbia Inst. of Tech., Comp. Tech., 3700 Willingdon Ave., Burnaby, BC V5G 3H2, CANADA

Prof. Fred Karlsson, U. of Helsinki, Dept. of General Linguistics, Hallituskatu 11-13, SF-00100 Helsinki, FINLAND

Robert Davis, Hewlett-Packard Ltd., Mktg. Sys. Group, Miller House, The Ring, Bracknell, Berks. RG12 1XN, GREAT BRITAIN

A Three-octave Organ Program

The following muLISP organ program is an extension of an idea suggested by John Dargan of Bethesda, Maryland. Using the TONE function provided by the '85 and '86 version of muLISP, it turns the IBM PC keyboard into a three-octave keyboard.

The bottom row keys on the keyboard (Z, X, C, V, B, N, and M) play the notes (A, B, C, D, E, F, and G) of the lowest octave, the middle row keys (A, S, D, F, G, H, and J) play the notes of the middle octave, and the top row keys (Q, W, E, R, T, Y, and U) play the notes of the highest octave. Pressing any other alphabetic key generates a pause in the music.

The control and shift keys are used to vary the duration of a note. Holding the control key down while pressing a note key doubles the time the note is played. Holding the shift key down quadruples the time the note is played.

The organ program is terminated when the "=" key is pressed.

John is currently working on an interface between muLISP and the Texas Instruments speech board. He says the speech board will "offer tremendous opportunities to the creative LISP programmer".

```
; File:  ORGAN.LSP          11/02/85          Soft Warehouse, Inc.


(DEFUN ORGAN (
    READ-CHAR )
  (LOOP
    (SETQ CHAR (READ-BYTE))
    ((EQ CHAR 61))
    ( ((<= 0 CHAR 31)
      (TONE (GET (ASCII (+ CHAR 64)) 'FREQUENCY) 500) )
      ((<= 96 CHAR 127)
      (TONE (GET (ASCII (- CHAR 32)) 'FREQUENCY) 1000) )
      (TONE (GET (ASCII CHAR) 'FREQUENCY) 250) ) ) )

(MAPC '(LAMBDA (PAIR) (PUT (CAR PAIR) 'FREQUENCY (CADR PAIR)))
      '((Z 220) (A 440) (Q 880)
      (X 247) (S 494) (W 988)
      (C 262) (D 523) (E 1046)
      (V 294) (F 587) (R 1175)
      (B 330) (G 659) (T 1318)
      (N 349) (H 698) (Y 1397)
      (M 392) (J 784) (U 1568) ) )

(RDS)
```

muLISP-86 Function Autoload Facililty

The file AUTOLOAD.LSP provides a simple means of not reading in a function's definition from a source file until the function is actually called. This makes it possible to automatically load the necessary parts of application programs that are too big in their entirety.

If <filename> is the name of a muLISP source file, (AUTOLOAD filename) reads through <filename> and finds each symbol defined using a DEFUN. AUTOLOAD puts on the property list of each such symbol, under the property indicator 'OFFSET, the offset in the file of the symbol's function definition. AUTOLOAD also defines each such symbol as a macro that generates a call to the function AUTOSTUB-AUX.

Later, if the function to be autoloaded is actually called, AUTOSTUB-AUX reads in and defines the function using its definition in <filename>. To save space, AUTOSTUB-AUX also removes the offset information stored on the function name's property list.

```
; File:  AUTOLOAD.LSP  (C)       02/13/86       Soft Warehouse, Inc.

(DEFUN AUTOLOAD (FILENAME)
  ((NULL (RDS FILENAME)))
  (SETQ *SOURCE-FILE* FILENAME)
  (LOOP
    (PRINC \!)
    (SETQ OFFSET (READPTR)
        EXPN (READ))
    ( ((EQ (CAR EXPN) 'DEFUN)
      (PUT (CADR EXPN) 'OFFSET OFFSET)
      (MOVD 'AUTOSTUB (CADR EXPN)) )
      (EVAL EXPN) )
    ((NULL RDS)) ) )

(DEFUN AUTOSTUB (MACRO (BODY)
  (LIST 'PROGN (LIST 'AUTOSTUB-AUX (CAR BODY)) BODY) ))

(DEFUN AUTOSTUB-AUX (FUNC
    RDS ECHO )
  (SETQ FNAME (INPUTFILE))
  (RDS *SOURCE-FILE*)
  (READPTR (REMPROP FUNC 'OFFSET))
  (EVAL (READ))
  ((NULL FNAME))
  (RDS FNAME) )

(SETQ MACROEXPAND NIL)        ;Don't expand at compile time

(RDS)
```

Advertisement

M a x F A C T S: an Expert System Engine

Thanks for your interest in MaxFACTS. I'd like to use this space to address the typical questions I have been receiving:

What are the capabilities of MaxFACTS? MaxFACTS is a programming language designed to be specifically useful in Expert Systems and A.I. related work. It aids in decision making tasks by performing automatic backtracking in order to 're-evaluate' possible decisions. This frees the knowledge engineer from low level programming tasks otherwise required in conventional languages. In fact, only the backward chainer of MaxFACTS would be considered very 'Prolog-like'. But MaxFACTS has the same power both backward and forward; plus the advantage of software hooks to conventional languages.

What type of user interface (dialog) does MaxFACTS have? As part of the current MaxFACTS language you have available both input and output facilities. The output facility allows for clean, English sentences, output with the use of TEMPLATES (similar to FORMATS). Of course, the software hooks also allow access to screen handling routines or conventional language I/O. A more complete natural language facility will be in future releases.

How are rules and facts entered into MaxFACTS and how are Expert Systems actually developed with MaxFACTS? MaxFACTS does not provide its own development system. Rules can simply be entered directly into MaxFACTS when it is loaded. OR - simply use your favorite editor! I use Microsoft's WORD^{tm} to create files of many rules and use the LOADTHEORY command to load them. If I add one or two while in MaxFACTS, then I resave the file with the SAVETHEORY command.

What are the future plans for MaxFACTS? Distributed with each MaxFACTS package is a cover letter. It gives users a perspective of where MaxFACTS is and where I plan to take it. One exception to a normal release time table will be the compiled code. The speed increase will be an advantage everyone can use as soon as it becomes available.

What is included in the MaxFACTS package? Two different versions of MaxFACTS are available. The full feature version **requires muLISP-85** and includes: 1) the MaxFACTS source file, 2) the user manual with explanation and reference material, and 3) two demonstration files that contain working examples. The stand alone version is the same except that the benefits of Software Hooks are lost since the system is distributed as an executable COM file rather than as a source file.

As of July 1, 1985 the new price for the full feature package is $240.00 and the price for the stand alone version is $175.00. Please include $10.00 for shipping and handling. To order write to **Robert R. McKenzie**, P.O. Box 1638, Huntington Beach, CA, 92647, U.S.A.

Advertisement

the Grad Student: Rational Calculator

This just released software product facilitates the symbolic manipulation of matricies whose elements may be Cartan differential forms. Written in muLISP-86, it is particularily useful for solving problems in the fields of differential geometry and general relativity theory.

On a 10 MHz 8086 based computer with 320K of RAM, **the Grad Student** requires about 4 1/2 minutes to calculate the complete Riemann tensor for the Kerr rotating black hole metric, proceeding from Kerr's null tetrad and performing the calculation using Cartan's differential forms. [Phys. Rev. Lett. 11, 237 (1963).]

the Grad Student consists of the MS-DOS executable file GS.EXE, more than a dozen illustrative application programs, and 15 pages of documentation. The software is distributed on a 9 sector per track, IBM PC formatted 5 1/4 inch DSDD diskette. It is being licensed at the introductory price of $80, plus $3 for shipping and handling. For more information or to order, write Dr. Frederick J. Ernst, 83 Pierrepont Ave., Potsdam, NY, 13676, U.S.A.

Advertisement

L I S P - W I N D O W S

Designed for the IBM PC family of computers and compatibles, LISP-WINDOWS aids in creating window based user interfaces for your muLISP application programs. LISP-WINDOWS accepts keyboard or Microsoft Mouse input. You can create pull-down menus and multiple split-screen or overlapping windows. LISP-WINDOWS is written in and **requires muLISP-85**. Since the source is included, it can be modified to suit your needs. LISP-WINDOWS features include:

- split-screen and overlapping windows
- pull-down menus
- display page access
- keyboard and mouse input
- color-graphics and monochrome support
- fully commented source code

To receive a copy on an IBM-PC formatted diskette, send $39.95 to **IntelliSOFT**, 777 Kapiolani Blvd., Suite 2603, Honolulu, Hawaii, 96813-5211, U.S.A.