DECUS
PROGRAM LIBRARY

DECUS NO.	8-254
TITLE	VECTOR ALGEBRA PACKAGE
AUTHOR	Bryan D. Young
COMPANY	Department of Cardiology Glasgow Royal Infirmary Glasgow, Scotland
DATE	Submitted August 7, 1969
SOURCE LANGUAGE	PAL

VECTOR ALGEBRA PACKAGE

DECUS Program Library Write-up DECUS No. 8-254

ABSTRACT

This package is designed to operate with the Basic Three Word Floating Point Package (Digital 8-5-S) and enables the user to label three dimensional vectors (i.e. three consecutive three-word floating point numbers) by a single symbol. The use of this package is similar to that of the Floating Point Package in that vector operations can be initiated by a single instruction. The operation of addition, subtraction, dot product, cross product and modulus of vectors can be simply programmed with this package, e.g. the operation $\underline{D} = (\underline{A}x\underline{B})x\underline{C}$ would be programmed as:

```
VGET   A
VCROS  B
VCROS  C
VPUT   D
```

Full input and output facilities for vectors are available with this package.

REQUIREMENTS

Storage

This package occupies locations 2-4, 64-74, 5200-5577. The three-word Floating Point Package (Basic) occupies locations 5-7, 40-63, 5600-7577.

Total storage requirements are 2-7, 40-74, 5200-7577.

Equipment

Standard PDP-8, ASR-33 teletype.

USAGE

Loading

This package is loaded with the binary loader (Digital 8-2-U). The binary tape supplied contains only the Vector Algebra Package and hence the three-word Floating Point Package (Digital 8-5-S) must also be loaded separately. This package can also be loaded with the EAE version of the Floating Point Package, thus reducing the execution times of the vector instructions. Note that the Vector Algebra Package cannot be loaded with Floating Point Package with Extended Functions, since the same area of core is used by each package.

Calling Sequence

Location 4 is used as an indirect address for entry to the Vector Algebra Package. Hence JMS I 4 will enter the package. After this instruction all subsequent instructions must be of the "vector form" (see Discussion) until the exit instruction VEXT is encountered.

DESCRIPTION

Discussion

In the following discussion it is assumed that the reader is familiar with the Floating Point Package. The Vector Algebra Package uses locations 64-74 as a vector accumulator, similar to the way in which the Floating Point Package uses location 44-46 as a floating point accumulator.

Op. Code	Mnemonic	Effect
6	VPUT	Vector Put. Store the contents of the vector accumulator in the locations specified by the effective address. The vector accumulator is unchanged.
7	VMOD	Vector Modulus. Compute the modulus or length of the vector in the vector accumulator and store result in the floating point accumulator.
0	VEXT	Exit from the Vector Algebra Package.

Direct or indirect addressing can be used with these instructions. Note that since these instructions are not contained in the symbol tables of the PAL or MACRO compilers they must be defined by the user's program as follows:

```
VADD  = 1000
VSUB  = 2000
VDOT  = 3000
VCROS = 4000
VGET  = 5000
VPUT  = 6000
VMOD  = 7000
VEXT  = 0000
```

Input and Output

Locations 2 and 3 are used for indirect addressing of the vector input and output routines in a manner similar to the way in which locations 5 and 6 are used by the Floating Point Package. Hence the instruction JMS I 2 enables the user to input the three components of a vector via the teletype into the vector accumulator.

Similarly the instruction JMS I 3 enables the user to output the three components of the vector contained in the vector accumulator. The three components are output on a single line with 5 spaces between each component followed by a carriage-return/line-feed.

Since the vector input and output routines utilize the input and output routines of the Floating Point Package the format of each component is that of a floating point number. Note also that the input and output routines use the instructions:

```
TSF
JMP .-1
TLS
```

Hence the teletype flag must be set before input or output.

i.e.

	Location		
Vector accumulator	64	Exponent	X - component
	65	Mantissa	
	66	Mantissa	
	67	Exponent	Y - component
	70	Mantissa	
	71	Mantissa	
	72	Exponent	Z - component
	73	Mantissa	
	74	Mantissa	

Each component is stored in the normal three-word floating point format. All vectors are assumed to be stored in this format where the identifying symbol refers to the location which contains the exponent of the X - component. The user is responsible for ensuring that each vector is allocated sufficient core storage, i.e. 9 locations.

Vector Instructions

Op. Code	Mnemonic	Effect
1	VADD	Vector Addition. Add vectorially the contents of the effective address to the vector accumulator.
2	VSUB	Vector Subtraction. Subtract vectorially the contents of the effective address from the vector accumulator.
3	VDOT	Vector Dot Product. Form the dot product of the contents of the effective address with the vector accumulator. Result in floating point accumulator.

Op. Code	Mnemonic	Effect
4	VCROS	Vector Cross Product. Form the cross product of the vector accumulator with the contents of the effective address.
5	VGET	Vector Get. Load the vector accumulator with the contents of the effective address.

Example

As an example of the use of this package consider the calculation of:

$$D = (A \times B) \times C$$

where A, B, C are vectors to be input via the teletype and D is to be output on the teletype.

```
/EXAMPLE OF PROGRAM TO COMPUTE D=(AXB)XC
A=3ØØ
B=311
C=322
D=333
*2ØØ
CLA
TLS                 /SET FLAG
JMS I 2             /INPUT A
JMS I 4
VPUT A
VEXT
JMS I 2             /INPUT B
JMS I 4
VPUT B
VEXT
JMS I 2             /INPUT C
JMS I 4
VPUT C
VGET A
VCROS B             /AXB
VCROS C             /(AXB)XC
VPUT D              /STORE RESULT IN D
VEXT
JMS I 3             /OUTPUT D
HLT
$
```

METHODS

Discussion

Let $\underline{A} = a_x\underline{i} + a_y\underline{j} + a_z\underline{k}$; $\underline{B} = b_x\underline{i} + b_y\underline{j} + b_z\underline{k}$

where $\underline{i}$, $\underline{j}$, $\underline{k}$ are unit vectors.

The vector operations are performed as follows:

Vector Addition $\underline{A}+\underline{B} = (a_x+b_x)\underline{i} + (a_y+b_y)\underline{j} + (a_z+b_z)\underline{k}$

Vector Subtraction $\underline{A}-\underline{B} = (a_x-b_x)\underline{i} + (a_y-b_y)\underline{j} + (a_z-b_z)\underline{k}$

Dot Product $\underline{A}.\underline{B} = a_x b_x + a_y b_y + a_z b_z$

Cross Product $\underline{A}\times\underline{B} = (a_y b_z - b_y a_z)\underline{i} + (a_z b_x - a_x b_z)\underline{j} + (a_x b_y - b_x a_y)\underline{k}$

Vector Modulus $|\underline{A}| = \sqrt{a_x^2 + a_y^2 + a_z^2}$

```
                    /VECTOR PROGRAMMING PACKAGE

                    /DEFINITIONS
                    SQROOT=2
                    FADD=1ØØØ
                    FSUB=2ØØØ
                    FMPY=3ØØØ
                    FDIV=4ØØØ
                    FGET=5ØØØ
                    FPUT=6ØØØ
                    FNOR=7ØØØ
                    FEXT=ØØØØ

                    /VECTOR ACCUMULATOR
                    V1 = 64         /X-CPT.
                    V2 = 67         /Y-CPT.
                    V3 = 72         /Z-CPT.

                    *2
ØØØ2  5276                          VIN             /ADDRESS OF INPUT ROUTINE
ØØØ3  5314                          VOUT            /ADDRESS OF OUTPUT ROUTINE
ØØØ4  52ØØ                          VPNT            /ADDRESS OF PACKAGE
ØØØ5  74ØØ                          74ØØ
ØØØ6  72ØØ                          72ØØ
ØØØ7  56ØØ                          56ØØ
                    *52ØØ
52ØØ  ØØØØ         VPNT,           Ø
52Ø1  73ØØ                          CLA CLL
52Ø2  16ØØ                          TAD I VPNT      /GET NEXT INSTRUCTION
52Ø3  22ØØ                          ISZ VPNT
52Ø4  745Ø                          SNA
52Ø5  56ØØ                          JMP I VPNT
52Ø6  3252                          DCA JUMP
52Ø7  1252                          TAD JUMP
521Ø  Ø254                          AND PAGENO      /SET PAGE BIT
5211  765Ø                          SNA CLA         /PAGE ZERO?
5212  5215                          JMP .+3         /YES
5213  1257                          TAD MASK5       /NO
5214  Ø2ØØ                          AND VPNT
5215  3261                          DCA ADDR
5216  126Ø                          TAD MASK7       /GET 7 BIT ADDRESS
5217  Ø252                          AND JUMP
522Ø  1261                          TAD ADDR
5221  3261                          DCA ADDR
5222  1255                          TAD INDRCT
5223  Ø252                          AND JUMP
5224  765Ø                          SNA CLA
5225  523Ø                          JMP .+3
```

```
5226 1661                TAD I ADDR
5227 3261                DCA ADDR
523Ø 1261                TAD ADDR
5231 3663                DCA I IU1          /INDIRECT ADDRESS OF X-CPT
5232 1663                TAD I IU1
5233 1262                TAD CONF
5234 3664                DCA I IU2          /INDIRECT ADDRESS OF Y-CPT
5235 1664                TAD I IU2
5236 1262                TAD CONF
5237 3665                DCA I IU3          /INDIRECT ADDRESS OF Z-CPT
524Ø 1252                TAD JUMP
5241 71Ø6                CLL RTL
5242 7ØØ6                RTL
5243 Ø256                AND MASK3
5244 1266                TAD TABLE          /SELECT OPERATION
5245 3253                DCA JUMP2
5246 1653                TAD I JUMP2
5247 3253                DCA JUMP2
525Ø 4653                JMS I JUMP2
5251 52Ø1                JMP VPNT+1
5252 ØØØØ  JUMP,         Ø
5253 ØØØØ  JUMP2,        Ø
5254 Ø2ØØ  PAGENO,       2ØØ
5255 Ø4ØØ  INDRCT,       4ØØ
5256 ØØ17  MASK3,        17
5257 76ØØ  MASK5,        76ØØ
526Ø Ø177  MASK7,        177
5261 ØØØØ  ADDR,         Ø
5262 ØØØ3  CONF,         3
5263 5564  IU1,          U1
5264 5565  IU2,          U2
5265 5566  IU3,          U3
5266 5255  TABLE,        TABLE              /TABLE FOR INTERPRETING
5267 54ØØ                VAD                /PSEUDO-INSTRUCTIONS
527Ø 5415                VSU
5271 5432                VDT
5272 545Ø                VCS
5273 5521                VGT
5274 5533                VPT
5275 5545                VMD
5276 ØØØØ  VIN,          Ø                  /ROUTINE TO INPUT A VECTOR
5277 44Ø5                JMS I 5            /INTO VECTOR ACCUMULATOR
53ØØ 44Ø7                JMS I 7
53Ø1 6Ø64                FPUT V1            /INPUT X-CPT
53Ø2 ØØØØ                FEXT
53Ø3 44Ø5                JMS I 5
53Ø4 44Ø7                JMS I 7
53Ø5 6Ø67                FPUT V2            /INPUT Y-CPT
```

```
5306 0000          FEXT
5307 4405          JMS I 5
5310 4407          JMS I 7
5311 6072          FPUT V3          /INPUT Z-CPT
5312 0000          FEXT
5313 5676          JMP I VIN
5314 0000  VOUT,   0                /ROUTINE TO OUTPUT A VECTOR
5315 7200          CLA              /FROM VECTOR ACCUMULATOR
5316 1055          TAD 55
5317 3361          DCA SAVE
5320 3055          DCA 55
5321 4407          JMS I 7
5322 5064          FGET V1
5323 0000          FEXT
5324 4406          JMS I 6          /OUTPUT X-CPT
5325 4343          JMS SP
5326 4407          JMS I 7
5327 5067          FGET V2
5330 0000          FEXT
5331 4406          JMS I 6          /OUTPUT Y-CPT
5332 4343          JMS SP
5333 2055          ISZ 55
5334 4407          JMS I 7
5335 5072          FGET V3
5336 0000          FEXT
5337 4406          JMS I 6          /OUTPUT Z-CPT
5340 1361          TAD SAVE
5341 3055          DCA 55
5342 5714          JMP I VOUT
5343 0000  SP,     0                /ROUTINE TO INSERT SPACES
5344 1357          TAD NMB          /BETWEEN CPTS
5345 3356          DCA CTR
5346 1360          TAD SPACE
5347 6041          TSF
5350 5347          JMP .-1
5351 6046          TLS
5352 2356          ISZ CTR
5353 5347          JMP .-4
5354 7300          CLA CLL
5355 5743          JMP I SP
5356 0000  CTR,    0
5357 7772  NMB,    -6
5360 0240  SPACE,  240
5361 0000  SAVE,   0
```

```
                    /VECTOR ROUTINES

                    /VECTOR ADD
                    *54ØØ
5400 ØØØØ  VAD,     Ø
5401 4407           JMS I 7
5402 5064           FGET V1
5403 1764           FADD I U1
5404 6064           FPUT V1
5405 5067           FGET V2
5406 1765           FADD I U2
5407 6067           FPUT V2
5410 5072           FGET V3
5411 1766           FADD I U3
5412 6072           FPUT V3
5413 ØØØØ           FEXT
5414 56ØØ           JMP I VAD
                    /VECTOR SUBTRACT
5415 ØØØØ  VSU,     Ø
5416 4407           JMS I 7
5417 5064           FGET V1
5420 2764           FSUB I U1
5421 6064           FPUT V1
5422 5067           FGET V2
5423 2765           FSUB I U2
5424 6067           FPUT V2
5425 5072           FGET V3
5426 2766           FSUB I U3
5427 6072           FPUT V3
5430 ØØØØ           FEXT
5431 5615           JMP I VSU
                    /VECTOR DOT PRODUCT
5432 ØØØØ  VDT,     Ø
5433 4407           JMS I 7
5434 5064           FGET V1
5435 3764           FMPY I U1
5436 6064           FPUT V1
5437 5067           FGET V2
5440 3765           FMPY I U2
5441 6067           FPUT V2
5442 5072           FGET V3
5443 3766           FMPY I U3
5444 1067           FADD V2
5445 1064           FADD V1
5446 ØØØØ           FEXT
5447 5632           JMP I VDT
```

```
                    /VECTOR CROSS PRODUCT
5450 0000  VCS,     0
5451 4407           JMS I 7
5452 5072           FGET V3
5453 3765           FMPY I U2
5454 6305           FPUT TEMP1
5455 5067           FGET V2
5456 3766           FMPY I U3
5457 2305           FSUB TEMP1
5460 6305           FPUT TEMP1
5461 5064           FGET V1
5462 3766           FMPY I U3
5463 6311           FPUT TEMP2
5464 5072           FGET V3
5465 3764           FMPY I U1
5466 2311           FSUB TEMP2
5467 6311           FPUT TEMP2
5470 5067           FGET V2
5471 3764           FMPY I U1
5472 6315           FPUT TEMP3
5473 5064           FGET V1
5474 3765           FMPY I U2
5475 2315           FSUB TEMP3
5476 6072           FPUT V3
5477 5311           FGET TEMP2
5500 6067           FPUT V2
5501 5305           FGET TEMP1
5502 6064           FPUT V1
5503 0000           FEXT
5504 5650           JMP I VCS
5505 0000  TEMP1,   0          /TEMPORARY STORES USED
5506 0000           0          /BY CROSS PRODUCT ROUTINE
5507 0000           0
5510 0000           0
5511 0000  TEMP2,   0
5512 0000           0
5513 0000           0
5514 0000           0
5515 0000  TEMP3,   0
5516 0000           0
5517 0000           0
5520 0000           0
```

```
                    /VECTOR GET
5521 ØØØØ  VGT,     Ø
5522 44Ø7           JMS I 7
5523 5764           FGET I U1
5524 6Ø64           FPUT V1
5525 5765           FGET I U2
5526 6Ø67           FPUT V2
5527 5766           FGET I U3
553Ø 6Ø72           FPUT V3
5531 ØØØØ           FEXT
5532 5721           JMP I VGT
                    /VECTOR PUT
5533 ØØØØ  VPT,     Ø
5534 44Ø7           JMS I 7
5535 5Ø64           FGET V1
5536 6764           FPUT I U1
5537 5Ø67           FGET V2
554Ø 6765           FPUT I U2
5541 5Ø72           FGET V3
5542 6766           FPUT I U3
5543 ØØØØ           FEXT
5544 5733           JMP I VPT
                    /VECTOR LENGTH
5545 ØØØØ  VMD,     Ø
5546 44Ø7           JMS I 7
5547 5Ø64           FGET V1
555Ø 3Ø64           FMPY V1
5551 6Ø64           FPUT V1
5552 5Ø67           FGET V2
5553 3Ø67           FMPY V2
5554 6Ø67           FPUT V2
5555 5Ø72           FGET V3
5556 3Ø72           FMPY V3
5557 1Ø67           FADD V2
556Ø 1Ø64           FAD  V1
5561 ØØØ2           SQROOT
5562 ØØØØ           FEXT
5563 5745           JMP I VMD
5564 ØØØØ  U1,      Ø          /USED AS INDIRECT ADDRESS
5565 ØØØØ  U2,      Ø          /OF OPERAND
5566 ØØØØ  U3,      Ø
```

ADDR	5261
CONF	5262
CTR	5356
FADD	1ØØØ
FDIV	4ØØØ
FEXT	ØØØØ
FGET	5ØØØ
FMPY	3ØØØ
FNOR	7ØØØ
FPUT	6ØØØ
FSUB	2ØØØ
INDRCT	5255
IU1	5263
IU2	5264
IU3	5265
JUMP	5252
JUMP2	5253
MASK3	5256
MASK5	5257
MASK7	526Ø
NMB	5357
PAGENO	5254
SAVE	5361
SP	5343
SPACE	536Ø
SQROOT	ØØØ2
TABLE	5266
TEMP1	55Ø5
TEMP2	5511
TEMP3	5515
U1	5564
U2	5565
U3	5566
VAD	54ØØ
VCS	545Ø
VDT	5432
VGT	5521
VIN	5276
VMD	5545
VOUT	5314
VPNT	52ØØ
VPT	5533
VSU	5415
V1	ØØ64
V2	ØØ67
V3	ØØ72