

## RETURN

## Appendix

To install the spline programs manually do the following steps:

1. Create a directory called SPLINE.
2. Within SPLINE:
  - a. Create a directory called FUNC.
  - b. Create a variable called ACTM. At this point just store a 0 in it. The program will make use of it when it is executing.
  - c. Create a directory called USR.
  - d. Enter and store the program **S** shown below  
**S**  
<< FUNC S1 >>
  - e. Enter and store the program **INPT** shown below.  
**INPT**  
<< FUNC P1 >>
3. Within FUNC enter and store the following eight programs.

### P1

```
<< { } { } 0. → X Y typ
<< UPDIR "Please choose action:" { { "Load existing spline" 1. } { "Create free spline"
2. } { "Create clamped spline" 3. } } 1. CHOOSE
IF 0. == THEN
  "ERROR: No option selected" MSGBOX
ELSE
  'typ' STO
  IF 'typ==1.' THEN
    USR "Please choose spline:" "" INPUT OBJ→ UPDIR
  ELSE
    "Enter filename:" { α "" } INPUT OBJ→
    IF 'typ==3.' THEN
      "DERIVATIVE VALUES" { "DERIV 1:" { "DERIV n" } } { 2. 1. } { } { }
      INFORM DROP SWAP
    END
    typ SWAP "Please enter number of data points" "" INPUT OBJ→ 1. SWAP
    FOR j "INPUT VALUES " j R→I + { "X:" { "Y:" } } { 2. 1. } { } { }
    INFORM
    IF 0. == THEN
      "WARNING: No value" MSGBOX
    ELSE
      DUP 1. GET SWAP 2. GET Y SWAP + 'Y' STO X SWAP + 'X' STO
    END
  END
NEXT
X Y FUNC P2 UPDIR
END 'ACTM' STO
END
>>
>>
```

**P2**

&lt;&lt;

IF 4. ROLL 2. == THEN

OVER OVER P3

ELSE

OVER OVER 6. ROLL ROT ROT P4

END

TRIDI 3. PICK ΔLIST { } { } 1. 6. PICK SIZE 1 -

FOR j

5 PICK DUP j 1 + GET SWAP j GET - 4 PICK j GET / 4 PICK j GET 3 / 6 PICK

DUP j 1 + GET SWAP j GET 2 \* + \* - + SWAP 4 PICK DUP j 1 + GET SWAP j

GET - 4 PICK j GET 3 \* / + SWAP

NEXT

3 ROLL DROP 3 ROLLD P5

&gt;&gt;

**P3**

&lt;&lt;

0. 0. → x a h n

&lt;&lt;

x a SR34 'n' STO 'h' STO 1. SWAP + 1. + SWAP 0. + 3. ROLL 0. SWAP + 4. ROLL

0. SWAP + 0. +

&gt;&gt;

&gt;&gt;

**P4**

&lt;&lt;

0 0 → y x a h n

&lt;&lt;

x a SR34 'n' STO 'h' STO 2 'h(1)' →NUM \* SWAP + 2 'h(n-1)' →NUM \* + SWAP

'h(n-1)' →NUM + 3 ROLL 'h(1)' →NUM SWAP + 4 ROLL 3 'h(1)' →NUM / 'a(2)'

→NUM 'a(1)' →NUM - \* 3 'y(1)' →NUM \* - SWAP + 3 'y(2)' →NUM \* 3 'h(n-1)'

→NUM / 'a(n)' →NUM 'a(n-1)' →NUM - \* - +

&gt;&gt;

&gt;&gt;

**P5**

&lt;&lt; 1. 5.

FOR j

IF j 1. == j 3. == OR THEN

0. +

END

OBJ→ →ARRY 5. ROLLD

NEXT

5. ROW→ USR DUP ROT STO

&gt;&gt;

**S1**

```

<<
UPDIR ACTM SIZE 2. GET 1. → x n j
<<
IF x ACTM { 1. 1. } GET < x ACTM { 1. n } GET > OR THEN
  "Input out of range" MSGBOX
ELSE
  WHILE x ACTM { 1. } j 1. ++ GET > REPEAT
    j 1. + 'j' STO
  END
  x ACTM { 1. j } GET - DUPDUP ACTM { 5. j } GET 4. 2.
  FOR k
    * ACTM { k j } GET + -1.
  STEP
END
>>
>>

```

**SR34**

```

<< 0 0 { } { } { } { } → x a h n α b c d
<< x SIZE 'n' STO x ΔLIST 'h' STO
  IF n 2 == THEN
    d c b α h n
  ELSE
    1 n 2 -
    FOR j
      'h(j)' →NUM 'h(j+1)' →NUM + 2 * α SWAP + 'α' STO 'h(j)' →NUM b
      SWAP + 'b' STO 'h(j+1)' →NUM c SWAP + 'c' STO 3 'h(j+1)' →NUM /
      'a(j+2)' →NUM 'a(j+1)' →NUM - * 3 'h(j)' →NUM / 'a(j+1)' →NUM
      'a(j)' →NUM - * - d SWAP + 'd' STO
    NEXT d c b α h n
  END
>>
>>

```

## TRIDI

<<

→ a b c d

<<

d 1. d 1. GET a 1. GET / PUT 'd' STO a SIZE 1. - DUP 1. SWAP

FOR j

c j c j GET a j GET / PUT 'c' STO a j 1. + a j 1. + GET b j GET c j GET \* - PUT  
'a' STO d j 1. + d j 1. + GET b j GET d j GET \* - a j 1. + GET / PUT 'd' STO

NEXT 1.

FOR k

d k d k GET c k GET d k 1. + GET \* - PUT 'd' STO -1.

STEP

d

>>

>>

**RETURN**