TI-82
TI-83
TI-83 Plus

Euler's Method

This program uses Euler's Method to approximate the particular solution of a differential equation. To use this program, be sure to enter the differential equation y' as Y1 in the equation editor. Then the program will prompt you to enter the starting x- and y-values and the step size. Press [ENTER] after each screen display to see more approximations. For the *TI-82*, press [ON] [2] to quit the program. For the *TI-83* and *TI-83 Plus*, press [ON] [1] to quit the program.

```
PROGRAM: EULERMET
:Input "INITIAL X=", X
:Input "INITIAL Y=", Y
:Input "STEP SIZE H=", H
:Lbl A
:Y+Y1*H→Y
:X+H→X
:Pause
:Disp "(X, Y)="
:Disp X, Y
:Goto A
```

Midpoint Rule

This program uses the Midpoint Rule to approximate the definite integral $\int_a^b f(x)\,dx$. You must store the function $f(x)$ as Y1 before executing the program. The program itself will prompt you for the limits of a and b and for the number of subintervals n.

```
PROGRAM:MIDPOINT
:Disp "LOWER LIMIT"
:Input A
:Disp "UPPER LIMIT"
:Input B
:Disp "N DIVISIONS"
:Input N
:0→S
:(B-A)/N→W
:1→J
:Lbl 1
:A+(J-1)W→L
:A+JW→R
:(L+R)/2→X
:S+WY1→S
:IS>(J, N)
:Goto 1
:Disp "APPROXIMATION"
:Disp S
```

Simpson's Rule

This program uses Simpson's Rule to approximate the definite integral $\int_a^b f(x)\,dx$. You must store $f(x)$ as Y1 before executing the program. The program itself will prompt you for the limits a and b and for *half* the number of subintervals you want to use.

```
PROGRAM: SIMPSONS
:Disp "LOWER LIMIT"
:Input A
:Disp "UPPER LIMIT"
:Input B
:Disp "N/2 DIVISIONS"
:Input D
:0→S
:(B-A)/(2D)→W
:1→J
:Lbl 1
:A+2(J-1)W→L
:A+2JW→R
:(L+R)/2→M
:L→X
:Y1→L
:M→X
:Y1→M
:R→X
:Y1→R
:W(L+4M+R)/3+S→S
:IS>(J, D)
:Goto 1
:Disp "APPROXIMATION"
:Disp S
```

Newton's Method

This program uses Newton's Method to approximate the zeros of a function. You must store the expression $f(x)$ as Y1 before executing the program and use a standard viewing window. Then graph the function to estimate one of the zeros. The program will prompt you for this estimate.

```
PROGRAM:NEWTON
:Disp "ENTER"
:Disp "APPROXIMATION"
:Input X
:(Xmax-Xmin)/100→D
:1→N
:Lbl 1
:X-Y1/nDeriv(Y1,X,X,D)→R
:If abs(X-R) ≤ abs(X/1E10)
:Goto 2
:R→X
:N+1→N
:Goto 1
:Lbl 2
:Disp "ZERO="
:Disp R
:Disp "ITER="
:Disp N
```