

PART VI

CALCULATOR PROGRAMS

CALCULATOR PROGRAMS

for all models of the

TI - 84, TI - 83 and TI - 82

Introduction to the TI-84, TI-83 and TI-82

The following are programs for the TI-84/83/82 calculator to do numerical integration, graph slope fields, demonstrate solutions to differential equations using Euler's method, and plot trajectories and solutions to systems of differential equations.

Notes on Using the TI-84, TI-83 and TI-82

1. The commands on these models are the same, though the TI-82 has one key named differently and this will be noted in the program listing.
2. Press PRGM to get the program menu, move to NEW to enter a new program. You must first give it a name (for example, RSUMS) when prompted; to finish entering/editing a program, hit 2nd QUIT.
3. To delete a program, press 2nd MEM and select [2:Delete].
4. The quantity AB represents the product of A and B. All variables are single capital letters. Some built-in variable names use lower case, for example Xmin.
5. Writing $Y_3 = Y_1 Y_2$ defines Y_3 as the product Y_1 and Y_2 . Writing $Y_3 = Y_1(Y_2)$ defines Y_3 as the composition of Y_1 and Y_2 .
6. The TI-83 Plus and all TI-84 models have an APPS key. This means that programs can be downloaded from the internet, stored, and run on these models. The TI-84 has a USB connection which makes it the most convenient model to acquire programs from the internet.
7. The official TI site is:

<http://education.ti.com>

This site includes information for instructors to obtain free TI equipment, lesson plans, programs, and the latest information about all of the TI graphing calculators.

Program to Calculate Riemann Sums to Evaluate a Definite Integral (TI-84/83/82)

Press PRGM to get the program menu, move to NEW to enter a new program. Give it a name, RSUMS, when prompted; to finish entering/editing a program, hit 2nd QUIT. The variable names, such as A and B listed below, are shown in italics only for ease of identification. They show as A and B on the TI screen.

| | |
|-------------------------------|--|
| Name= RSUMS | <i>Where to Find The Commands</i> |
| :Disp "LOWER LIMIT" | Disp and Input are accessed via PRGM, I/O, pressed while in the middle of a program. |
| :Input A | The " " are on the + key when you have pressed the ALPHA key. |
| :Disp "UPPER LIMIT" | |
| :Input B | |
| :Disp "DIVNS" | |
| :Input N | |
| : $A \rightarrow X$ | \rightarrow is on the STO key. |
| : $0 \rightarrow S$ | 0 is Zero, not oh. |
| : $(B - A)/N \rightarrow H$ | |
| :For($I, 1, N$) | For(is on the PRGM, CTL menu. |
| : $S + H * Y_1 \rightarrow S$ | Y_1 is on the VARS, Y-VARS, Function menu. (On a TI-82 use 2nd Y-VARS, Function.) (Do not use Y followed by 1, which means Y multiplied by 1.) |
| : $X + H \rightarrow X$ | |
| :End | End is on the PRGM, CTL menu. |
| :Disp "LEFT SUM" | |
| :Disp S | |
| : $S + Y_1 * H \rightarrow S$ | |
| : $A \rightarrow X$ | |
| : $S - Y_1 * H \rightarrow S$ | |
| :Disp "RIGHT SUM" | |
| :Disp S | |

Things to watch for:

1. Difference between $-$, which means subtract, and $(-)$, which means negative. For example, $-2 - 3$ must be entered as $(-)2 - 3$.
2. Disp and Input are not typed in letter-by-letter, but are obtained by highlighting them under PRGM, I/O and hitting ENTER.

To run this program:

1. The integrand (the function you want to integrate) $f(x)$ must be entered into Y_1 , with X as the independent variable.
2. Make sure the lower limit of the integral you're approximating is less than the upper limit.
3. Test on $\int_1^3 x^3 dx = 20$ with 100 subintervals. You should get left- and right-hand sums of 19.7408 and 20.2608, respectively.

Numerical Integration Program (TI-84/83/82)

The next program calculates left- and right-hand Riemann sums, and the trapezoidal, midpoint and Simpson approximations. Since there's not room on the calculator to label each approximation separately, we use a compressed method of displaying the results. For instance, the label LEFT/RIGHT indicates that the next two numbers are the left- and right-hand Riemann sums, respectively.

Notes:

1. Press PRGM, NEW and enter the program name (for example, INTEGRAL). To finish entering/editing a program, hit 2nd QUIT.
2. The function to be integrated must be entered as Y_1 (accessed by the "Y =" button). When Y_1 occurs in a program, it is evaluated at the current value of X .
3. The lower limit of integration must be less than the upper limit.
4. To run a program, select PRGM, EXEC. To stop a program while it is running, hit ON. A program is in a running state when there are dots scrolling in the upper right corner of the screen.
5. Test the program by evaluating $\int_1^3 x^3 dx = 20$, using 100 subdivisions. You should get left- and right-hand sums of 19.7408 and 20.2608, respectively. For the trapezoid approximation, you should get 20.0008. For the mid point approximation, you should get 19.9996. For Simpson's rule, you should get exactly 20.

| | |
|-----------------------------|---|
| Name= INTEGRAL | <i>Where to Find The Commands</i> |
| :Disp "LOWER LIMIT" | Disp and Input are accessed via PRGM, I/O |
| :Input A | Enter lower limit of integration. |
| :Disp "UPPER LIMIT" | |
| :Input B | Enter upper limit of integration. |
| :Disp "DIVNS" | |
| :Input N | Enter number subdivisions. |
| :(B - A)/N → H | Stores size of one subdivision in H (Note that → means hit STO button). |
| :A → X | Start X off at beginning of interval. |
| :0 → L | Initialize L, which keeps track of left sums, to zero. |
| :0 → M | Initialize M, which keeps track of midpoint sums, to zero. |
| :For(I, 1, N) | For(is on PRGM, CTL menu. |
| :L + H * Y ₁ → L | Increment L by Y ₁ H, the area of one more rectangle. Y ₁ is accessed via VARS, Y-VARS, Function. (TI-82: Y ₁ is accessed via 2nd Y-VARS.) |
| :X + .5H → X | Move X to middle of interval. |
| :M + H * Y ₁ → M | Evaluate Y ₁ at the middle of interval and increment M by rectangle of this height. |
| :X + .5H → X | Move X to start of next interval. |
| :End | End of For(loop, return if I ≤ N. |
| :L + H * Y ₁ → R | Add on area of right-most rectangle, store in R. |
| :A → X | Reset X to A. |
| :R - H * Y ₁ → R | Subtract off area of left-most rectangle. |
| :(L + R)/2 → T | Trapezoid approximation is average of L and R. |
| :(2M + T)/3 → S | Simpson is weighted average of M and T. |
| :ClrHome | |
| :Disp "LEFT" | |
| :Output (1,6,L) | |
| :Disp "RGHT" | |
| :Output (2,6,R) | |
| :Disp "MID" | |
| :Output (3,6,M) | |
| :Disp "TRAP" | |
| :Output (4,6,T) | |
| :Disp "SIMP" | |
| :Output (5,6,S) | |

Slope Field Program (TI-84/83/82)

This program draws the slope field for the differential equation $\frac{dy}{dx} = f(x, y)$.

| | |
|-----------------------------------|---|
| Name=SLOPES | <i>Where To Find The Commands</i> |
| :ClrDraw | 2nd Draw |
| :FnOff | Use VARS, Y-VARS, On/Off. (2nd Y-VARS on TI-82.) |
| :7(Xmax-Xmin)/83 → H | VARS, Window (2nd Y-VARS on TI-82.) → is STO key. Creates box width for the 96 tiles. |
| :7(Ymax-Ymin)/55 → K | Creates box height. |
| :1/(0.4H) ² → A | |
| :1/(0.4K) ² → B | |
| :Xmin+0.5H → X | |
| :Ymin+0.5K → Z | |
| :For(I, 1, 12) | PRGM, CTL |
| :Z → Y | |
| :For(J, 1, 8) | PRGM, CTL |
| :Y ₁ → T | |
| :1/√(A + B * T ²) → C | |
| :T * C → S | |
| :X → U | |
| :Y → V | |
| :Line(U - C, V - S, U + C, V + S) | 2nd Draw |
| :V + K → Y | |
| :End | via PRGM, CTL; ends J Loop |
| :U + H → X | |
| :End | via PRGM, CTL; ends I Loop |

To run this program:

1. Enter $\frac{dy}{dx} = f(x, y)$ as $Y_1 = f(X, Y)$.
2. Choose an appropriate WINDOW.
3. Run the program.
4. To get rid of previously drawn slope field, press ClrDraw (from the 2nd DRAW menu) or change the WINDOW.
5. Test the program by graphing the slope field for $\frac{dy}{dx} = x$. See Transparency Master for Section 11.2.

Euler's Method (TI-84/83/82)

This program calculates and graphs an approximate solution for the differential equation $y' = f(x, y)$ using Euler's method.

The function f is entered as $Y_1 = f(X, Y)$. When called, the program asks for the coordinates of the starting point and the step size. The graph is then drawn. The program does not clear the screen to start with, so solutions with different starting points or step sizes can be graphed consecutively on the same axes, or solution curves can be superimposed on a slope field. When the program is finished, the final Y -value remains in variable Y .

| | |
|---|--|
| Name=EULER | <i>Where to Find The Commands</i> |
| :FnOff | Use VARS, Y-VARS, On/Off. (2nd Y-VARS on TI-82.) |
| :Disp "INITIAL X" | Disp and Input are on the PRGM I/O menu. |
| :Input X | |
| :Disp "INITIAL Y" | |
| :Input Y | |
| :Disp "STEP SIZE" | |
| :Input H | |
| :While ($X \geq Xmin$)($X \leq Xmax$) | While is on PRGM CTL menu. \geq and \leq are on 2nd TEST menu. This tricky logic stops the trajectory at the X limits. |
| :X + H \rightarrow U | |
| :Y + H * $Y_1 \rightarrow V$ | Y_1 is on VARS, Y-VARS, Function (For TI-82, use 2nd VARS menu.) |
| :Line(X, Y, U, V) | Line(is on the 2nd DRAW menu. |
| :U \rightarrow X | |
| :V \rightarrow Y | |
| :End | End is on the PRGM CTL menu. |

To run this program:

1. Enter $y' = f(x, y)$ as $Y_1 = f(X, Y)$.
2. Choose an appropriate WINDOW.
3. Run the program; in most cases the program ends as the trajectory reaches the X limits. Press CLEAR when ready to return to the home screen. In some cases the trajectory ends in a error (Divide-by-zero), at which point you can select Quit. If you have seen enough of a trajectory and the program is still active (dots scrolling in the upper right) then press ON and select Quit. In all cases, you can rerun the program immediately after a Quit by just pressing ENTER.
4. To get rid of previously drawn graphs, press ClrDraw (from the 2nd DRAW menu) or change the WINDOW.
5. If you get an edgy, disconnected, or otherwise weird graph, try a smaller step size. A negative step size produces a trajectory in the opposite direction from a positive step.
6. Test the program by graphing the solution curve for $\frac{dy}{dx} = y$. Enter $Y_1 = Y$, and select the standard viewing window. Run the program with the initial values of X and Y to be 0 and 1 respectively and a step size of 0.01. Your solution curve should resemble $y = e^x$ for $x \geq 0$.

Program for Trajectories (TI-84/83/82)

The program below graphs the *trajectory* in the xy phase plane of the system

$$\frac{dx}{dt} = f(x, y), \quad x(0) = x_0,$$

$$\frac{dy}{dt} = g(x, y), \quad y(0) = y_0.$$

The functions $f(x, y)$ and $g(x, y)$ are entered into Y_2 and Y_3 . If you want to plot a trajectory on top of a slope field, enter $\frac{dy}{dx}$ as $Y_1 = \frac{g(x, y)}{f(x, y)}$ and run the slope field program first.

Notice that you are graphing the relation between x and y in the phase plane; t does not appear. To graph x and y against t , use the Time Series program on the next page.

| Name= TRAJECT | Command | Menu | Selection |
|-----------------------------|---------------------------------|------------|-------------------|
| :FnOff | FnOff | VARS | Y-VARS, On/Off, 2 |
| :Disp "INITIAL X" | Disp | PRGM | I/O, 3 |
| :Input X | Input | PRGM | I/O, 1 |
| :Disp "INITIAL Y" | While | PRGM | CTL, 5 |
| :Input Y | \geq, \leq | 2nd TEST | 4, 6 |
| :Disp "STEP SIZE" | Xmin, Xmax | VARS | Window, 1, 2 |
| :Input H | $X \geq Xmin)(X \leq Xmax)$ | Y_2, Y_3 | VARS |
| :X + H * Y ₂ → U | Y ₂ , Y ₃ | VARS | Y-VARS, Function |
| :Y + H * Y ₃ → V | Line(| 2nd DRAW | 2 |
| :Line(X, Y, U, V) | End | PRGM | CTL, 7 |
| :U → X | For the TI-82: | | |
| :V → Y | FnOff | 2nd Y-VARS | On/Off, 2 |
| :End | Xmin, Xmax | 2nd Y-VARS | Window, 1, 2 |
| | Y_2, Y_3 | 2nd Y-VARS | Y, 2, 3 |

To run this program:

1. Enter the differential equation as $Y_2 = f(X, Y)$ and $Y_3 = g(X, Y)$.
2. Choose an appropriate WINDOW.
3. Run the program; in most cases the program ends as the trajectory reaches the X limits. Press CLEAR when ready to return to the home screen. In some cases the trajectory ends in a error (Divide-by-zero), at which point you can select Quit. If you have seen enough of a trajectory and the program is still active (dots scrolling in the upper right) then press ON and select Quit. In all cases, you can rerun the program immediately after a Quit by just pressing ENTER.
4. To get rid of previously drawn trajectories, press ClrDraw (from the 2nd DRAW menu) or change the WINDOW.
5. If you get an edgy, disconnected, or otherwise weird trajectory, choose a smaller step size H . If the program runs too slowly, choose a larger step size H . Try starting with $H = 0.05$.
6. Test the program by graphing the trajectory for $\frac{dx}{dt} = y, \frac{dy}{dt} = x$.
Enter $Y_1 = X/Y, Y_2 = Y, Y_3 = X$. Run SLOPES with a $[-3, 3]$ by $[-3, 3]$ window to create a field pattern. Run TRAJECT with $x = 0, y = 1$ and step size 0.01. The trajectory follows the slope field pattern.

Program for Time Series (TI-84/83/82)

This program graphs the *solutions* $x(t)$ and $y(t)$ of the system

$$\frac{dx}{dt} = f(x, y), \quad x(0) = x_0,$$

$$\frac{dy}{dt} = g(x, y), \quad y(0) = y_0.$$

Notice that here we are graphing both x and y as functions of t . Compare the Trajectory program on the previous page, which graphs the relation between x and y , and in which t does not appear.

```
Name= TIMESER
:ClrDraw                      Use 2nd DRAW.
:FnOff                        See TRAJECT for command menu locations.
:Disp "INITIAL X"
:Input X
:Disp "INITIAL Y"
:Input Y
:Disp "STEP SIZE"
:Input H
:0 → T                        Starts graphing from the vertical axis.
:While (T ≥ Xmin)(T ≤ Xmax)
:X + H * Y2 → U
:Y + H * Y3 → V
:T + H → W
:Line(T, X, W, U)
:Line(T, Y, W, V)
:U → X
:V → Y
:W → T
:End
```

To run this program:

1. Enter the differential equation as $Y_2 = f(X, Y)$ and $Y_3 = g(X, Y)$.
2. Choose an appropriate WINDOW (note that Xmin and Xmax refer to the horizontal axis of the window, which is time in our case; Ymin and Ymax refer to the vertical axis: x and y in our case). The graph will start at $t = 0$, so make $Xmin \leq 0$.
3. Run the program; in most cases the program ends as time (T) reaches the X limits. Press CLEAR when ready to return to the home screen. In some cases the graphs end in a error (Divide-by-zero), at which point you can select Quit. If you have seen enough of the graphs and the program is still active (dots scrolling in the upper right) then press ON and select Quit. In all cases, you can rerun the program immediately after a Quit by just pressing ENTER.
4. Test the program by graphing the solutions for the previous settings for TRAJECT. Choose initial (X, Y) to be $(1, 0.5)$ and step 0.05. This shows two parabolic shaped curves in the first quadrant.

CALCULATOR PROGRAMS

for all models of the

TI - 89, TI - 92 and Voyage 200

To enter a program, press **APPS** and choose the program editor. To run a program, type the name, such as euler(), in the home screen.

The four programs that follow in this section are primarily for TI-92 users. The TI-89, TI-92 Plus and Voyage 200 have differential equation features built-in. To access the differential equation features in the newer models press MODE, and change the Graph setting to DIFF EQUATIONS. See your User Guide for more details. For Riemann sums, a menu driven program is available by downloading the CalcTools APP from the TI website. To download the CalcTool APP go to: <http://education.ti.com/us/product/apps/latest.html>.

The TI-92 programs listed in this section can be used on the TI-92 Plus, TI-89 and Voyage 200 without modification, if preferred.

Riemann Sum Program (TI-89)

```

:rsum()
:Prgm
:ClrIO
:Disp "Put function in y1"
:Disp "Set window "
:Input "Enter Partitions",n
:(xmax-xmin)/n → d
:ClrIO
:Output 1,1,"nInt"
:Output 1,50,nInt(y1(x),x,xmin,xmax)
:Output 11,1,"Left"
:sum(seq(y1(xmin+i*d),i,0,n-1))*d → s1
:Output 11,50, s1
:Output 21,1,"Right"
:sum(seq(y1(xmin+i*d),i,1,n))*d → s2
:Output 21,50,s2
:Output 31,1,"Mid"
:sum(seq(y1(xmin+i*d),i,0.5,n))*d → s3
:Output 31,50,s3
:Output 41,1,"Trap"
:(s1+s2)/2 → s4
:Output 41,50,s4
:Output 51,1,"Simpson"
:Output 51,50,(2*s3+s4)/3
:EndPrgm

```

Slope Field Program (TI-89)

This program plots the slope field of the differential equation $\frac{dy}{dx} = f(x, y)$. The function $f(x, y)$ and the window are input by the user during the running of the program.

```

:slope()
:Prgm
:setMode("Graph", "FUNCTION")
:PlotsOff
:FnOff
:ClrDraw
:ClrGraph
:ClrIO
:Request "Enter f(x,y)=", ff
:Define f(x, y) =Func
:expr(ff)
:EndFunc
:Input "Enter xmin =", xmin
:Input "Enter xmax =", xmax
:Input "Enter ymin =", ymin
:Input "Enter ymax =", ymax
:Local h, k, a, b, c, x, y, z, i, j, t, u, v
:7 * (xmax - xmin) / 83 → h
:7 * (ymax - ymin) / 55 → k
:1 / (.4 * h) ^ 2 → a
:1 / (.4 * k) ^ 2 → b
:xmin + .5 * h → x
:ymin + .5 * k → z
:For i, 1, 12
:z → y
:For j, 1, 8
:f(x, y) → t
:1 / (√(a + b * t^2)) → c
:t * c → s
:x → u
:y → v
:Line u - c, v - s, u + c, v + s
:v + k → y
:EndFor
:u + h → x
:EndFor
:EndPrgm

```

Euler's Method (TI-89)

This program plots solutions to the differential equation

$$\frac{dy}{dx} = f(x, y).$$

The function $f(x, y)$ is input by the user during the running of the program.

```

:euler()
:Prgm
:setMode("Graph", "FUNCTION")
:PlotsOff
:FnOff
:ClrDraw
:ClrGraph
:ClrIO
:Local x, y, h, u, v
:Request "Enter f(x,y)=", ff
:Define f(x, y) = Func
:expr(ff)
:EndFunc
:Input "Initial x=", x
:Input "Initial y=", y
:Input "Step size =", h
:Text "Press ON to stop while running"
:While (x ≥ xmin) and (x ≤ xmax)
:x + h → u
:y + h * f(x, y) → v
:Line x, y, u, v
:u → x
:v → y
:EndWhile
:EndPrgm

```

To see the solution curves superimposed on the slope field, comment out (disable) the command lines :ClrDraw and :ClrGraph by putting @ in front of them. (@ is found in the Control menu.) Run slope() followed by euler().

Program for Trajectories (TI-89)

This program plots trajectories of the differential equations

$$\frac{dx}{dt} = f(x, y)$$

$$\frac{dy}{dt} = g(x, y)$$

The functions $f(x, y)$ and $g(x, y)$ are input during the running of the program.

```

:traj()
:Prgm
:setMode("Graph", "FUNCTION")
:PlotsOff
:ClrDraw
:ClrGraph
:ClrIO
:FnOff
:Local x, y, h, u, v
:Request "Enter f(x, y) =", fden
:Define f(x, y) = Func
:expr(fden)
:EndFunc
:Request "Enter g(x, y) =", fnum
:Define g(x, y) = Func
:expr(fnum)
:EndFunc
:Input "Initial x =", x
:Input "Initial y =", y
:Input "Step size =", h
:Text "Press ON to stop while running"
:While (x ≥ xmin) and (x ≤ xmax)
:x + h * f(x, y) → u
:y + h * g(x, y) → v
:Line x, y, u, v
:u → x
:v → y
:EndWhile
:EndPrgm

```

To see the trajectories superimposed on the slope field, comment out (disable) the command lines :ClrDraw and :ClrGraph from the traj() program by putting @ in front of them. (@ is found in the Control menu.) Run slope() followed by traj().

Program for Time Series (TI-89)

This program plots solutions $x(t)$, $y(t)$ to the system of differential equations

$$\frac{dx}{dt} = f(x, y)$$

$$\frac{dy}{dt} = g(x, y)$$

The functions $f(x, y)$ and $g(x, y)$ are input during the running of the program; the window must be set beforehand. Choose an appropriate WINDOW (note that Xmin and Xmax refer to the horizontal axis of the window, which is time in our case; Ymin and Ymax refer to the vertical axis: x and y in our case). The graph starts at $t = 0$, so be sure $Xmin \leq 0$.

```

:times()
:Prgm
:setMode("Graph", "FUNCTION")
:PlotsOff
:ClrDraw
:ClrGraph
:ClrIO
:FnOff
:Local x, y, t, h, u, v, w
:Request "Enter f(x, y) =", fden
:Define f(x, y) = Func
:expr(fden)
:EndFunc
:Request "Enter g(x, y) =", fnum
:Define g(x, y) = Func
:expr(fnum)
:EndFunc
:Input "Initial x =", x
:Input "Initial y =", y
:Input "Step size =", h
:Text "Press ON to stop while running"
:0 → t
:While (t ≥ xmin) and (t ≤ xmax)
:x + h * f(x, y) → u
:y + h * g(x, y) → v
:t + h → w
:Line t, x, w, u
:Line t, y, w, v
:u → x
:v → y
:w → t
:EndWhile
:EndPrgm

```