

$$Y_1 = f(x)$$

Prgm: INTEGRAL	Where to Find The Commands
: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.
:1 → I	Initialize I, the counter for the loop.
:Lbl P	Label for top of loop. Lbl is accessed via PRGM, CTL.
:L + H * Y ₁ → L	Increment L by Y ₁ H, the area of one more rectangle. (Y ₁ is accessed via Y-VARS, or 2nd 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.
:IS > (I, N)	IS > (is accessed via PRGM, CTL. This is the most difficult step in the program: adds 1 to I and does the next step if I ≤ N (i.e., if haven't gone through loop enough times); otherwise, skips next step. Thus, if I ≤ N, goes back to Lbl P and loops through again. If I > N, loop is finished and goes on to print out results. Continue here if I > N, in which case the value of X is now B.
:Goto P	Goto is accessed via PRGM, CTL. Jumps back to Lbl P if J ≤ N.
:Disp "LEFT/RIGHT"	
:Disp L	L now equals the left sum, so display it.
: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.
:Disp R	R now equals right sum, so display it.
:(L + R)/2 → T	Trap approximation is average of L and R.
:Disp "TRAP/MID/SIMP"	
:Disp T	Display trap approximation.
:Disp M	Display midpoint approximation.
:(2M + T)/3 → S	Simpson is weighted average of M and T.
:Disp S	Display Simpson's approximation.

$$Y_1 = f(x)$$

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.
:1 → I	Initialize I, the counter for the loop.
:Lbl P	Label for top of loop. Lbl is accessed via PRGM, CTL.
:L + H * Y ₁ → L	Increment L by Y ₁ H, the area of one more rectangle. (Y ₁ is accessed via Y-VARS, or 2nd 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.
:IS > (I, N)	IS > (is accessed via PRGM, CTL. This is the most difficult step in the program: adds 1 to I and does the the next step if $I \leq N$ (i.e., if haven't gone through loop enough times); otherwise, skips next step. Thus, if $I \leq N$, goes back to Lbl P and loops through again. If $I > N$, loop is finished and goes on to print out results. Continue here if $I > N$, in which case the value of X is now B.
:Goto P	Goto is accessed via PRGM, CTL. Jumps back to Lbl P if $J \leq N$.
:Disp "LEFT/RIGHT"	
:Disp L	L now equals the left sum, so display it.
: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.
:Disp R	R now equals right sum, so display it.
:(L + R)/2 → T	Trap approximation is average of L and R.
:Disp "TRAP/MID/SIMP"	
:Disp T	Display trap approximation.
:Disp M	Display midpoint approximation.
:(2M + T)/3 → S	Simpson is weighted average of M and T.
:Disp S	Display Simpson's approximation.

TI-85
 $y1 = f(x)$

Name=INTEG	<i>Where to Find The Commands</i>
:Disp "LOWER LIMIT"	Disp, Input, and " " 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 right sums, to zero.
:1 → I	Initialize I, the counter for the loop.
:Lbl P	Label for top of loop. Lbl is accessed via CTL
:L + H * y1 → L	Increment L by H * y1, the area of one more rectangle. (y1 is typed in as y and then 1.)
:x + 0.5H → x	Move x to middle of interval.
:M + H * y1 → M	Evaluate y1 at the middle of interval and increment M by a rectangle of this height.
:x + 0.5H → x	Move x to start of next interval.
:IS > (I, N)	Access IS > (from under CTL. This is the most difficult step in the program: adds 1 to I and does the next step if $I \leq N$ (i.e., if haven't gone through loop enough times); otherwise, skips next step. Thus, if $I \leq N$, goes back to Lbl 1 and loops through again. If $I > N$, loop is finished and goes on to print out results.
:Goto P	Goto is accessed via CTL. Jumps back to Lbl P if $I \leq N$.
:Disp "LEFT/RIGHT"	
:Disp L	L now equals the left sum, so display it.
:L + H * y1 → R	Add on area of right-most rectangle, store in R.
:A → x	Reset x to A.
:R - H * y1 → R	Subtract off area of left-most rectangle.
:Disp R	R now equals right sum, so display it.
:(L + R)/2 → T	Trap approximation is average of L and R.
:Disp "TRAP/MID/SIMP"	
:Disp T	Display trap approximation.
:Disp M	Display midpoint approximation.
:(2 * M + T)/3 → S	Simpson is weighted average of M and T.
:Disp S	Display Simpson's approximation.

CASIO fx-7700GB

$$f_1 = f(x)$$

Numerical Integration Program (CASIO)

This is a Casio fx series calculator program for various numerical integrals. It will display the Left and Right Riemann Sums and the Trapezoid Rule, Midpoint Rule, and Simpson's Rule approximations all at once. The way this program evaluates integrals is by keeping a running total of function values on n subintervals, and then multiplying by the width of the rectangles to obtain the area at the very end. At the end of the program, hitting EXE will let you reevaluate the integral with a different number of subdivisions, and hitting AC will let you out of the program.

<i>Program</i>	<i>Comments</i>
"INTEGRAL"	
"L-LIM"?→A	Integrate from X=a
"U-LIM"?→B	to X=b
"DIVNS"?→N	over N subdivisions.
(B-A)div (2N)→H	calculates half the width of a subdivision.
0→L	initialize L, which will keep track of the left sums.
0→M	initialize M, the midpoint sum.
A→X	place X at A, the beginning of the interval.
Lbl 1	top of loop:
f ₁ +L→L	evaluate the function at the left edge of the interval add the result to the left-hand sum running total.
X+H→X	move X to the middle of the interval.
f ₁ +M→M	evaluate the function at the middle of the interval add the result to the midpoint running total.
X+H→X	move X to the beginning of the next interval.
Dsz N	decrease N by 1; if N=0, skip the next step and go on.
Goto 1	bottom of loop.
"LEFT,RIGHT,TRAP"	
2HL→L	multiply the sum of the left-hand function values by width 2H.
L▲	display the left-hand sum.
L+2Hf ₁ →T	evaluate the function at X=b: the rightmost function value add the area of the rightmost rectangle with the left-hand sum.
A→X	put X back at A.
T-2Hf ₁ →T	evaluate the function at the left-most edge of the interval take the area of the leftmost rectangle out of T.
T▲	display what is now the right-hand sum.
(L+T)div 2→T	average the left- and right-hand sums.
T▲	display the trapezoid approximation.
MID,SIMP"	
2HM→M	multiply the midpoint values sum by the interval width.
M▲	display the midpoint approximation.
(2M+T)div 3→M	calculate Simpson's Rule by weighted averaging.
M	display Simpson's Rule approximation.

<i>Program</i>	<i>Where the Commands are</i>
Goto start	Goto is in BRANCH menu
Label eqn	Label is in BRANCH menu
f=...	replace "..." by the integrand $f(x)$
Return	Return is in BRANCH menu
Label start	
Print "l-limit	Print and " are in PROG menu
Input a	Input is in PROG menu
Print "u-limit	
Input b	
Print "divns	
Input n	
x=a	= is also in INEQ menu
s=0	reset memory s to zero
m=0	reset memory m to zero
i=1	
h=(b-a)/n	
Label l	
Gosub eqn	Gosub is in BRANCH menu
s=s+f*h	
x=x+.5h	
Gosub eqn	
m=m+f*h	
x=x+.5h	
i=i+1	
If i <= n Goto l	If and Goto are in BRANCH menu; <= is in INEQ menu
Print "left/right	
Print s	
Gosub eqn	
r=s+f*h	
x=a	
Gosub eqn	
r=r-f*h	
Print r	
Wait	Wait is in PROG menu
Print "trap/mid/simp	
t=.5(s+r)	
Print t	
Print m	
s=(2m+t)/3	
Print s	
End	End is in PROG menu