

TAMUCALC 4.2 Command Reference

COPYRIGHT ©2004 by Art Belmonte
Department of Mathematics, Texas A&M University

Tuesday, 20/Jan/2004

ANGVEC

- Description** The **angvec** function computes the exact angle between two vectors using the current angle mode.
- Access** **Vectors** submenu in the **TAMUCALC** custom menu
- Syntax** **angvec(v,w)**
- Input** **v**: A vector having 2, 3, or more components
w: A vector with the same number of components as **v**
- Output** The exact angle $\theta = \cos^{-1} \left(\frac{\mathbf{v} \cdot \mathbf{w}}{\|\mathbf{v}\| \|\mathbf{w}\|} \right)$ between **v** and **w** in the current angle mode
- Examples** **angvec([3,1,-1], [1,-1,2])** returns $\pi/2$ in **RADIAN** mode.
angvec([3,1,-1], [1,-1,2]) returns 90 in **DEGREE** mode.
- See Also** **ANGVECDG, ANGVECRD**
-

ANGVECDG

- Description** The **angvecdg** function computes the approximate angle between two vectors in decimal degrees.
- Access** **Vectors** submenu in the **TAMUCALC** custom menu
- Syntax** **angvecdg(v,w)**
- Input** **v**: A vector having 2, 3, or more components
w: A vector with the same number of components as **v**
- Output** The approximate angle $\theta = \cos^{-1} \left(\frac{\mathbf{v} \cdot \mathbf{w}}{\|\mathbf{v}\| \|\mathbf{w}\|} \right)$ between **v** and **w** in decimal degrees
- Example** **angvecdg([5,-1,3], [8,2,0])** returns 38.84.
- See Also** **ANGVEC, ANGVECRD**
-

ANGVECRD

Description	The angvecrd function computes the approximate angle between two vectors in decimal radians.
Access	Vectors submenu in the TAMUCALC custom menu
Syntax	angvecrd(v,w)
Input	v : A vector having 2, 3, or more components w : A vector with the same number of components as v
Output	The approximate angle $\theta = \cos^{-1} \left(\frac{\mathbf{v} \cdot \mathbf{w}}{\ \mathbf{v}\ \ \mathbf{w}\ } \right)$ between v and w in decimal radians
Example	angvecrdg([5,-1,3], [8,2,0]) returns 0.68.
See Also	ANGVEC, ANGVECDG

AUTOPOLR

Description	The autopolr program draws polar curves in an automatic fashion using 1-1 <i>xy</i> -scaling.
Access	FVMD submenu in the TAMUCALC custom menu
Syntax	autopolr()
Input	There are no arguments.
Output	Polar plots are automatically displayed on the Graph screen using 1-1 <i>xy</i> -scaling. A range of $0^\circ \leq \theta \leq 360^\circ$ in 5° increments is employed. The curves may be traced to analyze intersections, etc.
Example	Set the Graphing mode to Polar, input polar curves using the Y= editor, then invoke autopolr . In order to successfully use this routine, the angle θ must only occur in arguments to trigonometric functions. Otherwise, use manual polar graphing and RADIAN mode. Try graphing the polar curves $r = 3$ and $r = 3 + 2 \cos(3\theta)$ simultaneously.
See Also	Chapter 8, Polar Graphing , in the TI-89 Guidebook
Note	When finished looking at the Graph screen, press ENTER . In the event that the default θ -range didn't produce a satisfactory plot, manually adjust the range and/or θ -increment in DEGREE mode.

CA

Description	The ca function computes the acceleration of a position curve.
Access	CLP submenu in the TAMUCALC custom menu
Syntax	ca(r,t)
Input	r : Position vector expression t : Independent variable (time)
Output	The expression for acceleration, $\mathbf{a} = \mathbf{r}''$
Example	ca([t, 2*t^2, 3*t^3], t) returns $[0, 4, 18t]$.
See Also	CS, CV

CAN

Description	The can function computes the normal component of acceleration of a position curve.
Access	CLP submenu in the TAMUCALC custom menu
Syntax	can(r,t)
Input	r : Position vector expression t : Independent variable (time)
Output	The expression for the normal component of acceleration, $a_N = \ \mathbf{T}'\ \ \mathbf{r}'\ $, where $\mathbf{T} = \frac{\mathbf{r}'}{\ \mathbf{r}'\ }$
Example	can([$t, 2t, t^2$], t) returns $\frac{2\sqrt{5}}{\sqrt{4t^2 + 5}}$.
See Also	CA, CAT

CAT

Description	The cat function computes the tangential component of acceleration of a position curve.
Access	CLP submenu in the TAMUCALC custom menu
Syntax	cat(r,t)
Input	r : Position vector expression t : Independent variable (time)
Output	The expression for the tangential component of acceleration, $a_T = \frac{\mathbf{r}' \cdot \mathbf{r}''}{\ \mathbf{r}'\ }$
Example	cat([$t, 2t, t^2$], t) returns $\frac{4t}{\sqrt{4t^2 + 5}}$.
See Also	CA, CAN

CB

Description	The cb function computes the unit binormal vector to a position curve.
Access	CLP submenu in the TAMUCALC custom menu
Syntax	cb(r,t)
Input	r : Position vector expression t : Independent variable (time)
Output	The unit binormal vector $\mathbf{B} = \frac{\mathbf{r}' \times \mathbf{r}''}{\ \mathbf{r}' \times \mathbf{r}''\ }$ to the curve
Example	cb([$\cos(t), \sin(t), t$], t) returns $\left[\frac{\sin t}{\sqrt{2}}, -\frac{\cos t}{\sqrt{2}}, \frac{1}{\sqrt{2}} \right]$.
See Also	CN, CT

CJ

Description	The cj function computes the jerk of a position curve.
Access	CLP submenu in the TAMUCALC custom menu
Syntax	cj(r,t)
Input	r : Position vector expression t : Independent variable (time)
Output	The expression for jerk, r'''
Example	cj([$t, 2t^2, 3t^3$], t) returns $[0, 0, 18]$.
See Also	CA

CK

Description	The ck function computes the curvature of a position curve.
Access	CLP submenu in the TAMUCALC custom menu
Syntax	ck(r,t)
Input	r : Position vector expression t : Independent variable (time)
Output	The curvature $\kappa = \frac{\ \mathbf{T}'\ }{\ \mathbf{r}'\ }$ of the curve, where $\mathbf{T} = \frac{\mathbf{r}'}{\ \mathbf{r}'\ }$
Example	ck([$t, 2t, t^2$], t) returns $\frac{2\sqrt{5}}{(4t^2 + 5)^{3/2}}$.
See Also	Cτ

CL

Description	The cl function computes the arc length of a position curve.
Access	CLP submenu in the TAMUCALC custom menu
Syntax	cl(r,t,a,b)
Input	r : Position vector expression t : Independent variable (time) a : Lower bound of t -range b : Upper bound of t -range
Output	The arc length $\int_a^b \ \mathbf{r}'\ dt$ of the curve
Example	cl([$t, 2t, t^2$], $t, 0, 1$) returns $\frac{5}{8} \ln 5 + 3/2 \approx 2.51$.
See Also	CK, Cτ, LEN

CLINE2PT

Description	The cline2pt function finds a Cartesian equation of the line between two points in the xy -plane.
Access	CLP submenu in the TAMUCALC custom menu
Syntax	cline2pt(a,b)
Input	a : A point on the line b : A different point on the line
Output	A Cartesian equation of the line
Examples	cline2pt([5,2], [8,2]) returns $y = 2$. cline2pt([3,6], [3,-4]) returns $x = 3$. cline2pt([2,3], [1,1]) returns $y = 2x - 1$.
See Also	LINPTDIR, PLINE2PT

CN

Description	The cn function computes the unit normal vector to a position curve.
Access	CLP submenu in the TAMUCALC custom menu
Syntax	cn(r,t)
Input	r : Position vector expression t : Independent variable (time)
Output	The unit normal vector $\mathbf{N} = \frac{\mathbf{T}'}{\ \mathbf{T}'\ }$ to the curve, where $\mathbf{T} = \frac{\mathbf{r}'}{\ \mathbf{r}'\ }$
Example	cn([cos(t), sin(t), t], t) returns $[-\cos t, -\sin t, 0]$.
See Also	CB, CT

COMP

Description	The comp function computes the scalar projection of one vector onto another.
Access	Vectors submenu in the TAMUCALC custom menu
Syntax	comp(v,w)
Input	v : A vector having 2, 3, or more components w : A vector with the same number of components as v
Output	The scalar projection $\text{comp}_{\mathbf{v}}\mathbf{w} = \frac{\mathbf{v} \cdot \mathbf{w}}{\ \mathbf{v}\ }$ of the vector w onto the vector v
Example	comp([6,1,-2], [0,4,7]) returns $-10/\sqrt{41} \approx -1.56$.
See Also	ORTH, PROJ

COMPOS

Description	The compos function yields the composition of two functions.
Access	FVMD submenu in the TAMUCALC custom menu
Syntax	compos(f,g,v)
Input	f : A string (the name of the “outer” function) g : A string (the name of the “inner” function) v : Either a variable (if g is a function of one variable) or a vector of variables (if g is a function of several variables)
Output	The composition $f \circ g$ evaluated at v ; i.e., $f(g(v))$
Examples	$x^3y^2z \rightarrow f(x,y,z): [2t+3, t^2-1, t^3] \rightarrow g(t): \text{compos}(\text{“f”}, \text{“g”}, t)$ returns $(t^2 - 1)^2 (3 + 2t)^3 t^3$. $[\sin(x*y*z), x^2*y, z^2*e^{(x/5)}] \rightarrow w(x,y,z): [u, \cos(v), 2*\sin(v)] \rightarrow s(u,v): \text{compos}(\text{“w”}, \text{“s”}, [u,v])$ returns $[\sin(2u \sin v \cos v), u^2 \cos v, 4e^{u/5} \sin^2 v]$.
See Also	User-defined functions in the TI-89 Guidebook

CROSS

Description	The cross function computes the cross product of two vectors.
Access	Vectors submenu in the TAMUCALC custom menu
Syntax	cross(v,w)
Input	v : A vector having 3 components w : A vector having 3 components
Output	The cross product $\mathbf{v} \times \mathbf{w} = \det \begin{bmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ v_1 & v_2 & v_3 \\ w_1 & w_2 & w_3 \end{bmatrix}$ of the vectors v and w
Example	cross([-3,8,4], [5,0,7]) returns [56, 41, -40].
See Also	DOT

CS

Description	The cs function computes the speed of a position curve.
Access	CLP submenu in the TAMUCALC custom menu
Syntax	cs(r,t)
Input	r : Position vector expression t : Independent variable (time)
Output	The expression for speed, $s = \ \mathbf{r}'\ $
Example	cs([t, 2*t^2, 3*t^3], t) returns $\sqrt{81t^4 + 16t^2 + 1}$.
See Also	CA, CV

CT

Description The **ct** function computes the unit tangent vector to a position curve.

Access **CLP** submenu in the **TAMUCALC** custom menu

Syntax **ct(*r*,*t*)**

Input *r*: Position vector expression
t: Independent variable (time)

Output The unit tangent vector $\mathbf{T} = \frac{\mathbf{r}'}{\|\mathbf{r}'\|}$ to the curve

Example **ct([cos(*t*), sin(*t*), *t*], *t*)** returns $\left[-\frac{\sin t}{\sqrt{2}}, \frac{\cos t}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right]$.

See Also **CB, CN**

Cτ

Description The **cτ** function computes the torsion of a position curve.

Access **CLP** submenu in the **TAMUCALC** custom menu

Syntax **cτ(*r*,*t*)**

Input *r*: Position vector expression
t: Independent variable (time)

Output The torsion $\tau = \frac{(\mathbf{r}' \times \mathbf{r}'') \cdot \mathbf{r}'''}{\|\mathbf{r}' \times \mathbf{r}''\|^2}$ of the curve

Example **cτ([sin(4*t*), 6*t*², cos(4*t*)], *t*)** returns $\frac{48t}{144t^2 + 25}$.

See Also **CK**

CURL

Description The **curl** function computes the curl (rotational) of a vector field.

Access **FVMD** submenu in the **TAMUCALC** custom menu

Syntax **curl(*v*)**

Input *v*: A vector field having 3 components

Output The curl $\vec{\nabla} \times \mathbf{v} = \det \begin{bmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ \partial/\partial x & \partial/\partial y & \partial/\partial z \\ v_1 & v_2 & v_3 \end{bmatrix}$ of the vector field *v*

Example **curl([x²*y, y*z², z*x²])** returns $[-2yz, -2xz, -x^2]$.

See Also **DIV, LAP**

CV

Description The **cv** function computes the velocity of a position curve.

Access **CLP** submenu in the **TAMUCALC** custom menu

Syntax **cv(r,t)**

Input **r**: Position vector expression
t: Independent variable (time)

Output The expression for velocity, $\mathbf{v} = \mathbf{r}'$

Example **cv([t, 2*t^2, 3*t^3], t)** returns $[1, 4t, 9t^2]$.

See Also **CA, CS**

DBLSUMBL

Description The **dblsumbl** function computes an approximate (numerical) double Riemann sum using the bottom left corners of subrectangles. This is an estimate of the double integral $\int_c^d \int_a^b f(x, y) dx dy$.

Access **LSI** submenu in the **TAMUCALC** custom menu

Syntax **dblsumbl("f",a,b,m,c,d,n)**

Input **f**: A string (the name of the function in the integrand)
a: Lower horizontal (x) value
b: Upper horizontal (x) value
m: Number of horizontal subintervals
c: Lower vertical (y) value
d: Upper vertical (y) value
n: Number of vertical subintervals

Output The double Riemann sum $\sum_{i=1}^m \sum_{j=1}^n f(x_i^*, y_j^*) \Delta x_i \Delta y_j$, where $x_i^* = a + (i - 1)h$ & $\Delta x_i = \Delta x = h = \frac{b - a}{m}$
and $y_j^* = c + (j - 1)k$ & $\Delta y_j = \Delta y = k = \frac{d - c}{n}$, as a floating point decimal

Example **e^(x+y)→f(x,y): dblsumbl("f", 0, 1/2, 4, 3/4, 1, 8)** returns 0.360501; the exact answer is

$$\int_{3/4}^1 \int_0^{1/2} e^{x+y} dx dy = e^{3/2} - e^{5/4} - e + e^{3/4} \approx 0.390064.$$

See Also **DBLSUMBR, DBLSUMMD, DBLSUMTL, DBLSUMTR**

DBLSUMBR

Description The **dblsumbr** function computes an approximate (numerical) double Riemann sum using the bottom right corners of subrectangles. This is an estimate of the double integral $\int_c^d \int_a^b f(x, y) dx dy$.

Access **LSI** submenu in the **TAMUCALC** custom menu

Syntax **dblsumbr("f",a,b,m,c,d,n)**

Input **f**: A string (the name of the function in the integrand)
a: Lower horizontal (x) value
b: Upper horizontal (x) value
m: Number of horizontal subintervals
c: Lower vertical (y) value
d: Upper vertical (y) value
n: Number of vertical subintervals

Output The double Riemann sum $\sum_{i=1}^m \sum_{j=1}^n f(x_i^*, y_j^*) \Delta x_i \Delta y_j$, where $x_i^* = a + ih$ & $\Delta x_i = \Delta x = h = \frac{b-a}{m}$
and $y_j^* = c + (j-1)k$ & $\Delta y_j = \Delta y = k = \frac{d-c}{n}$, as a floating point decimal

Example **e^{x+y}→f(x,y): dblsumbr("f", 0, 1/2, 4, 3/4, 1, 8)** returns 0.408501; the exact answer is $\int_{3/4}^1 \int_0^{1/2} e^{x+y} dx dy = e^{3/2} - e^{5/4} - e + e^{3/4} \approx 0.390064$.

See Also **DBLSUMBL, DBLSUMMD, DBLSUMTL, DBLSUMTR**

DBLSUMMD

Description The **dblsummd** function computes an approximate (numerical) double Riemann sum using the middle (center) points of subrectangles. This is an estimate of the double integral $\int_c^d \int_a^b f(x, y) dx dy$.

Access **LSI** submenu in the **TAMUCALC** custom menu

Syntax **dblsummd("f",a,b,m,c,d,n)**

Input **f**: A string (the name of the function in the integrand)
a: Lower horizontal (x) value
b: Upper horizontal (x) value
m: Number of horizontal subintervals
c: Lower vertical (y) value
d: Upper vertical (y) value
n: Number of vertical subintervals

Output The double Riemann sum $\sum_{i=1}^m \sum_{j=1}^n f(x_i^*, y_j^*) \Delta x_i \Delta y_j$, where $x_i^* = a + (i - \frac{1}{2})h$ & $\Delta x_i = \Delta x = h = \frac{b-a}{m}$
and $y_j^* = c + (j - \frac{1}{2})k$ & $\Delta y_j = \Delta y = k = \frac{d-c}{n}$, as a floating point decimal

Example **e^{x+y}→f(x,y): dblsummd("f", 0, 1/2, 4, 3/4, 1, 8)** returns 0.389795; the exact answer is $\int_{3/4}^1 \int_0^{1/2} e^{x+y} dx dy = e^{3/2} - e^{5/4} - e + e^{3/4} \approx 0.390064$.

See Also **DBLSUMBL, DBLSUMBR, DBLSUMTL, DBLSUMTR**

DBLSUMTL

Description The **dblsumtl** function computes an approximate (numerical) double Riemann sum using the top left corners of subrectangles. This is an estimate of the double integral $\int_c^d \int_a^b f(x, y) dx dy$.

Access **LSI** submenu in the **TAMUCALC** custom menu

Syntax **dblsumtl("f",a,b,m,c,d,n)**

Input **f**: A string (the name of the function in the integrand)
a: Lower horizontal (x) value
b: Upper horizontal (x) value
m: Number of horizontal subintervals
c: Lower vertical (y) value
d: Upper vertical (y) value
n: Number of vertical subintervals

Output The double Riemann sum $\sum_{i=1}^m \sum_{j=1}^n f(x_i^*, y_j^*) \Delta x_i \Delta y_j$, where $x_i^* = a + (i - 1)h$ & $\Delta x_i = \Delta x = h = \frac{b - a}{m}$ and $y_j^* = c + jk$ & $\Delta y_j = \Delta y = k = \frac{d - c}{n}$, as a floating point decimal

Example **e^{x+y} → f(x,y): dblsumtl("f", 0, 1/2, 4, 3/4, 1, 8)** returns 0.371945; the exact answer is $\int_{3/4}^1 \int_0^{1/2} e^{x+y} dx dy = e^{3/2} - e^{5/4} - e + e^{3/4} \approx 0.390064$.

See Also **DBLSUMBL, DBLSUMBR, DBLSUMMD, DBLSUMTR**

DBLSUMTR

Description The **dblsumtr** function computes an approximate (numerical) double Riemann sum using the top right corners of subrectangles. This is an estimate of the double integral $\int_c^d \int_a^b f(x, y) dx dy$.

Access **LSI** submenu in the **TAMUCALC** custom menu

Syntax **dblsumtr("f",a,b,m,c,d,n)**

Input **f**: A string (the name of the function in the integrand)
a: Lower horizontal (x) value
b: Upper horizontal (x) value
m: Number of horizontal subintervals
c: Lower vertical (y) value
d: Upper vertical (y) value
n: Number of vertical subintervals

Output The double Riemann sum $\sum_{i=1}^m \sum_{j=1}^n f(x_i^*, y_j^*) \Delta x_i \Delta y_j$, where $x_i^* = a + ih$ & $\Delta x_i = \Delta x = h = \frac{b - a}{m}$ and $y_j^* = c + jk$ & $\Delta y_j = \Delta y = k = \frac{d - c}{n}$, as a floating point decimal

Example $e^{x+y} \rightarrow f(x,y)$: `dblsumtr("f", 0, 1/2, 4, 3/4, 1, 8)` returns 0.421468; the exact answer is $\int_{3/4}^1 \int_0^{1/2} e^{x+y} dx dy = e^{3/2} - e^{5/4} - e + e^{3/4} \approx 0.390064$.

See Also **DBLSUMBL, DBLSUMBR, DBLSUMMD, DBLSUMTL**

DIV

Description The **div** function computes the divergence of a vector field.

Access **FVMD** submenu in the **TAMUCALC** custom menu

Syntax **div(v,w)**

Input **v**: A vector field with n (typically 3) components
w: A vector of n (typically 3) independent variables

Output The divergence $\vec{\nabla} \cdot \mathbf{v} = \sum_{k=1}^n \frac{\partial v_k}{\partial x_k}$ of the vector field **v**; typically, $\frac{\partial v_1}{\partial x} + \frac{\partial v_2}{\partial y} + \frac{\partial v_3}{\partial z}$

Example **div([x^2*y, y*z^2, z*x^2], [x,y,z])** returns $2xy + z^2 + x^2$.

See Also **CURL, LAP**

DOT

Description The **dot** function computes the dot product of two (real) vectors.

Access **Vectors** submenu in the **TAMUCALC** custom menu

Syntax **dot(v,w)**

Input **v**: A vector having 2, 3, or more components
w: A vector with the same number n of components as **v**

Output The dot product $\mathbf{v} \cdot \mathbf{w} = \sum_{k=1}^n v_k w_k$ of the (real) vectors **v** and **w**

Example **dot([-3,8,4], [5,0,7])** returns 13.

See Also **CROSS**

DPTHLINE

Description	The dpthline function computes the distance from a point to a (hyper)line.
Access	Vectors submenu in the TAMUCALC custom menu
Syntax	dpthline(p,a,b)
Input	p : A point [not on the (hyper)line] a : A point on the (hyper)line b : A different point on the (hyper)line
Output	The distance d from the point p to the (hyper)line containing a and b , given by $d = \ \text{orth}_{\mathbf{v}}\mathbf{w}\ $, where $\mathbf{v} = \vec{ab}$ and $\mathbf{w} = \vec{ap}$
Example	dpthline([1,0,-1], [5,0,1], [4,3,3]) returns $2\sqrt{5} \approx 4.47$.
See Also	DPTPLANE, ORTH

DPTPLANE

Description	The dptplane function computes the distance from a point to a plane.
Access	Vectors submenu in the TAMUCALC custom menu
Syntax	dptplane(p,a,b,c)
Input	p : A point (not on the plane) a : A point on the plane b : A different point on the plane c : A third point on the plane; a , b , and c are noncollinear.
Output	The distance d from the point p to the plane containing a and b , and c , given by $d = \ \text{proj}_{\mathbf{v}}\mathbf{w}\ = \text{comp}_{\mathbf{v}}\mathbf{w} $, where $\mathbf{v} = \vec{ab} \times \vec{ac}$ and $\mathbf{w} = \vec{ap}$
Example	dptplane([3,-2,7], [1,0,1], [2,1,3], [1,2,13]) returns $26/\sqrt{53} \approx 3.57$.
See Also	DPTHLINE, PROJ

EQUATE

Description	The equate function constructs simultaneous equations from the respective components of two vectors or lists.
Access	Alg submenu in the TAMUCALC custom menu
Syntax	equate(v,w)
Input	v : A vector or list w : Another vector or list with the same number of components as v .
Output	Simultaneous equations joined with ands ; possibly simplified
Example	equate([3*x^2+4y, 4*x-2*y], [0,0]) returns $4y + 3x^2 = 0$ and $2x - y = 0$.
See Also	SOLVE in the TI-89 Guidebook

FTC

Description	The FTC function applies the Fundamental Theorem of Calculus.
Access	Calc submenu in the TAMUCALC custom menu
Syntax	FTC(F,x,a,b)
Input	F : Antiderivative expression x : Variable of integration a : Lower limit of integration b : Upper limit of integration
Output	The value $(F _{x=b}) - (F _{x=a})$
Example	FTC(cos(x)+x*sin(x), x, 0, pi/4) returns $\frac{\sqrt{2}}{2} + \frac{\pi\sqrt{2}}{8} - 1 \approx 0.26$.
See Also	\int (in the TI-89 Guidebook

FTRI

Description	The FTRI function computes the line integral of a conservative vector field \vec{w} by applying the Fundamental Theorem for Line Integrals.
Access	FVMD submenu in the TAMUCALC custom menu
Syntax	FTRI("f","g",a,b)
Input	f : A string (the name of a potential function for the conservative vector field \vec{w}) g : A string (the name of a parameterization for the curve C) a : Lower bound of the range of the independent variable in the line intergral b : Upper bound of the range of the independent variable in the line intergral
Output	The value of the line integral, $\int_C \vec{w} \cdot d\vec{g} = \int_C \vec{\nabla} f \cdot d\vec{g} = f(\vec{g}(b)) - f(\vec{g}(a))$.
Example	mf([x,y,z], pot([4*x*e^z), cos(y), 2*x^2*e^z], [x,y,z], "f"): [t, t^2, t^4]→g(t): FTRI("f", "g", 0, 1) returns $2e + \sin 1 \approx 6.28$.
See Also	POT

GRAD

Description	The grad function computes the gradient vector of a scalar function.
Access	FVMD submenu in the TAMUCALC custom menu
Syntax	grad(f,v)
Input	f : A functional expression v : A vector of n (typically 2 or 3) independent variables
Output	The gradient vector $\vec{\nabla} f = \left[\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \dots, \frac{\partial f}{\partial x_n} \right]$; typically $\left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right]$ or $\left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z} \right]$.

Example `grad(x^2*sin(y)+4*x*z^3, [x,y,z])` returns $[2x \sin y + 4z^3, x^2 \cos y, 12xz^2]$.

See Also **HESS**

HESS

Description The **hess** function computes the Hessian matrix of a scalar function.

Access **FVMD** submenu in the **TAMUCALC** custom menu

Syntax **hess(f,v)**

Input **f**: A functional expression

v: A vector of n (typically 2) independent variables

Output The Hessian matrix $H = \begin{bmatrix} f_{x_1x_1} & \cdots & f_{x_1x_n} \\ \vdots & \ddots & \vdots \\ f_{x_nx_1} & \cdots & f_{x_nx_n} \end{bmatrix}$; typically $H = \begin{bmatrix} f_{xx} & f_{xy} \\ f_{yx} & f_{yy} \end{bmatrix}$.

Example `hess(x^3+4*x*y-y^2, [x,y])` returns $\begin{bmatrix} 6x & 4 \\ 4 & -2 \end{bmatrix}$.

See Also **GRAD, LPMD**

IMPLDIFF

Description The **impldiff** function computes a derivative of an implicitly defined function.

Access **Calc** submenu in the **TAMUCALC** custom menu

Syntax **impldiff(q,v,x)**

Input **q**: An equation relating dependent and independent variables

v: The dependent variable

x: An independent variable

Output The (partial) derivative of **v** with respect to **x**.

Example `impldiff(x*z=sin(x+y+z), z, y)` returns $\frac{-\cos(x+y+z)}{\cos(x+y+z)-x}$.

See Also **d/** in the TI-89 Guidebook

INTPARTS

- Description** The **intparts** program repeatedly performs integration by parts at the direction of the user. At any time, the user may choose to have the routine automatically finish the integration.
- Access** **Calc** submenu in the **TAMUCALC** custom menu
- Syntax** **intparts(): antider**
- Input** There are no arguments; the user is prompted for input.
- Output** An antiderivative of the user-supplied integrand with respect to the specified independent variable is stored in the global variable **antider**, then displayed on the Home screen.
- Example** **intparts(): antider** (See **NOTE** below.)
- See Also** $\int()$ in the TI-89 Guidebook

Note Here is a transcript of the steps involved in computing $\int (x^2 + 4x - 7) e^{3x} dx$ via integration by parts. TI-89 prompts are left-justified and user responses are indented. (**Remark:** Whenever the PAUSE annunciator appears, the user simply presses ENTER to continue. This allows time to read and/or scroll intermediate results.)

intparts(): antider
Variable of integration? (or ENTER to default to x)
x
Integrand?
(x^2+4*x-7)*e^(3*x)
Cumulative antiderivative: 0
Current integrand: $(x^2 + 4x - 7) e^{3x}$
 u : Part to differentiate? (or ENTER to finish)
x^2+4*x-7
Cumulative antiderivative: $\frac{(x^2 + 4x - 7) e^{3x}}{3}$
Current integrand: $\frac{-2(x + 2) e^{3x}}{3}$
 u : Part to differentiate? (or ENTER to finish)
x+2
Cumulative antiderivative: $\left(\frac{x^2}{3} + \frac{10x}{9} - 25/9\right) e^{3x}$
Current integrand: $\frac{2e^{3x}}{9}$
 u : Part to differentiate? (or ENTER to finish)
ENTER
Final antiderivative: scroll then ENTER for HOME.
 $\left(\frac{x^2}{3} + \frac{10x}{9} - 73/27\right) e^{3x}$

INVFUNC

- Description** The **InvFunc** function attempts to compute the inverse of a function $y = f(x)$.
- Access** **FCS** submenu in the **TAMUCALC** custom menu
- Syntax** **InvFunc(f)**
- Input** f : A functional expression in x

Output The inverse functional expression f^{-1} , if successful

Example **InvFunc((2+5*x)^(1/3))**functional returns $\frac{x^3 - 2}{5}$.

See Also **SOLVE** in the TI-89 Guidebook

JMATRIX

Description The **jmatrix** function computes the Jacobian matrix of a transformation.

Access **FVMD** submenu in the **TAMUCALC** custom menu

Syntax **jmatrix(T,v)**

Input **T**: A vector of expressions giving old variables x_1, \dots, x_n in terms of new ones u_1, \dots, u_n ;
i.e., $[x_1(u_1, \dots, u_n), \dots, x_n(u_1, \dots, u_n)]$
v: A vector of new variables, $[u_1, \dots, u_n]$

Output The Jacobian matrix $J = \begin{bmatrix} \partial x_1 / \partial u_1 & \cdots & \partial x_1 / \partial u_n \\ \vdots & \ddots & \vdots \\ \partial x_n / \partial u_1 & \cdots & \partial x_n / \partial u_n \end{bmatrix}$

Example **jmatrix([r*cos(theta), r*sin(theta)], [r,theta])** returns $\begin{bmatrix} \cos \theta & -r \sin \theta \\ \sin \theta & r \cos \theta \end{bmatrix}$.

See Also **DET** in the TI-89 Guidebook

LAP

Description The **lap** function computes the Laplacian of a scalar field.

Access **FVMD** submenu in the **TAMUCALC** custom menu

Syntax **lap(f,v)**

Input **f**: A scalar field with n (typically 3) components
v: A vector of n (typically 3) independent variables

Output The Laplacian $\bar{\nabla}^2 f = \sum_{k=1}^n \frac{\partial^2 f}{\partial x_k^2}$ of the scalar field f ; typically, $\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} + \frac{\partial^2 f}{\partial z^2}$

Example **lap(sin(x)*cosh(y)+cos(x)*sinh(y), [x,y])** returns 0.

See Also **CURL, DIV**

LEN

Description	The len function computes the length of a vector.
Access	Vectors submenu in the TAMUCALC custom menu
Syntax	len(v)
Input	v : A vector having 2, 3, or more components
Output	The Euclidean length $\ \mathbf{v}\ = \left(\sum_{k=1}^n v_k^2 \right)^{1/2}$ of the (real) vector v
Example	len([-3,12,4]) returns 13.
See Also	UNITVEC

LINPTDIR

Description	The linptdir function finds a parametric representation of the (hyper)line through a point in the direction of a vector.
Access	CLP submenu in the TAMUCALC custom menu
Syntax	linptdir(a,v)
Input	a : A point on the (hyper)line v : A direction vector
Output	A parametric representation $\vec{L}(t) = \vec{a} + t\vec{v}$ of the (hyper)line
Example	linptdir([8,1,6], [2,-5,1]) returns $[2t + 8, 1 - 5t, t + 6]$.
See Also	CLINE2PT , PLINE2PT

LIS

Description	The lis function computes the line integral of a scalar field along a curve C with respect to arc length.
Access	LSI submenu in the TAMUCALC custom menu
Syntax	lis("f", "g", a, b)
Input	f : A string (the name of the scalar field) g : A string (the name of the parameterization $\vec{g}(t)$ of the curve C) a : Lower bound of the range of the independent variable in the line intergral b : Upper bound of the range of the independent variable in the line intergral
Output	The line integral $\int_C f ds = \int_a^b f(\vec{g}(t)) \ \vec{g}'(t)\ dt$
Example	7+x^2+y^4 → f(x,y) : [2*cos(t), 2*sin(t)] → g(t) : lis("f", "g", 0, pi/4) returns $\frac{3}{2}(5\pi - 4) \approx 17.56$.
See Also	LIV ; FCS and FVMD submenus of the TAMUCALC custom menu

LIV

Description The **liv** function computes the line integral of a vector field along a curve C .

Access **LSI** submenu in the **TAMUCALC** custom menu

Syntax **liv("w", "g", a, b)**

Input **w**: A string (the name of the vector field)
g: A string (the name of the parameterization $\vec{g}(t)$ of the curve C)
a: Lower bound of the range of the independent variable in the line intergral
b: Upper bound of the range of the independent variable in the line intergral

Output The line integral $\int_C \vec{w} \cdot d\vec{g} = \int_a^b \vec{w}(\vec{g}(t)) \cdot \vec{g}'(t) dt$

Example **[x*y, 4*z-x, x^2*z] → w(x,y,z): [3*t^2-8, 4*t^3, 6*t+7] → g(t):**
liv("w", "g", -1, 1) returns $\frac{30804}{7} \approx 4400.57$.

See Also **LIS**; **FCS** and **FVMD** submenus of the **TAMUCALC** custom menu

LPMD

Description The **LPMD** function computes the leading principal minor determinants of a (numerical Hessian) matrix.

Access **FVMD** submenu in the **TAMUCALC** custom menu

Syntax **LPMD(a)**

Input **a**: A square matrix; typically a numerical Hessian matrix

Output A list of the leading principal minor determinants of the input matrix **a**; i.e., the determinants of the upper left square submatrices of **a**: $|a_{11}|$, $\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}$, etc.

Example **mf([x,y], hess(x^3+4*x*y-y^2, [x,y]), "h")**: **h(-8/3, -16/3)** returns $\begin{bmatrix} -16 & 4 \\ 4 & -2 \end{bmatrix}$.
Then **LPMD(h(-8/3,-16/3))** returns $\{-16, 16\}$.

See Also **HESS**

LSUM

Description The **lsum** function computes an approximate (numerical) Riemann sum using left endpoints of subintervals. This is an estimate of $\int_a^b f(x) dx$.

Access **Calc** submenu in the **TAMUCALC** custom menu

Syntax **lsum(f,a,b,n)**

Input	f : A string (either the name of a function or the name of a list of function values) a : Lower horizontal (x) value b : Upper horizontal (x) value n : Number of horizontal subintervals
Output	The Riemann sum $\sum_{k=1}^n f(x_k^*) \Delta x_k$, where $x_k^* = a + (k-1)h$ and $\Delta x_k = \Delta x = h = \frac{b-a}{n}$, as a floating point decimal
Examples	1/x → f(x) Isum("f", 2, 3, 10) returns 0.413914; the exact answer is $\int_2^3 \frac{1}{x} dx = \ln \frac{3}{2} \approx 0.405465$. DelVar f: {0, 1, 3, 4.5, 5, 4.5, 3, 2, 1} → f Isum("f", 0, 4, 0) returns 11.50. The routine in this case automatically computes n to be one fewer than the number of elements in the list f ; in other words, the number of equal-length subintervals of $[a, b]$ determined by the data.
See Also	MSUM, RSUM, SIMP, TRAP

MF

Description	The mf program makes a user-defined function after evaluating its defining expression.
Access	FVMD submenu in the TAMUCALC custom menu
Syntax	mf(i,o,f)
Input	i : An independent variable or a vector of independent variables (the input to the function) o : Functional expression; a scalar, vector, or matrix (the output of the function) f : A string (the name of the function)
Output	The user-defined function is stored in the variable specified by the string f .
Examples	mf(x, x^2+9, "f") : f(x) returns $x^2 + 9$. This is equivalent to x^2+9 → f(x) : f(x) . mf(t, [cos(t), sin(t), 4*t], "g") : g(t) returns $[\cos t, \sin t, 4t]$. This is equivalent to [cos(t), sin(t), 4*t] → g(t) : g(t) . mf([u,v], [u*cos(v), u*sin(v), v], "s") : s(u,v) returns $[u \cos v, u \sin v, v]$. This is equivalent to [u*cos(v), u*sin(v), v] → s(u,v) : s(u,v) . mf(t, pline2pt([2,-6,8], [-4,0,5]), "g") : g(t) returns $[2 - 6t, 6t - 6, 8 - 3t]$. This is NOT equivalent to pline2pt([2,-6,8], [-4,0,5]) → g(t) : g(t) . Here mf constructs the function correctly; e.g., $g(1) = [-4, 0, 5]$. However, the conventional method fails; e.g., $g(1) = [2 - 6t, 6t - 6, 8 - 3t]$, which is clearly incorrect. Accordingly, mf will <i>always</i> construct functions correctly, whereas the conventional method may not if the function definition has embedded user-defined function calls. If the function definition does <i>not</i> have such calls, the conventional method is quicker if the TAMUCALC FCS submenu is employed (due to predefined templates).
See Also	FCS submenu in the TAMUCALC custom menu; user-defined functions in the TI-89 Guidebook

MSUM

Description	The msum function computes an approximate (numerical) Riemann sum using midpoints of subintervals. This is an estimate of $\int_a^b f(x) dx$.
Access	Calc submenu in the TAMUCALC custom menu
Syntax	msum(f,a,b,n)
Input	f : A string (either the name of a function or the name of a list of function values) a : Lower horizontal (x) value b : Upper horizontal (x) value n : Number of horizontal subintervals
Output	The Riemann sum $\sum_{k=1}^n f(x_k^*) \Delta x_k$, where $x_k^* = a + (k - \frac{1}{2})h$ and $\Delta x_k = \Delta x = h = \frac{b-a}{n}$, as a floating point decimal
Examples	1/x → f(x) msum("f", 2, 3, 10) returns 0.405407; the exact answer is $\int_2^3 \frac{1}{x} dx = \ln \frac{3}{2} \approx 0.405465$. DelVar f: {0, 1, 3, 4.5, 5, 4.5, 3, 2, 1} → f msum("f", 0, 4, 0) returns 12.00. The routine in this case automatically computes n to be one fewer than the number of elements in the list f ; in other words, the number of equal-length subintervals of $[a, b]$ determined by the data.
See Also	LSUM, RSUM, SIMP, TRAP

ORTH

Description	The orth function computes the orthogonal projection of one vector onto another.
Access	Vectors submenu in the TAMUCALC custom menu
Syntax	orth(v,w)
Input	v : A vector having 2, 3, or more components w : A vector with the same number of components as v
Output	The orthogonal projection $\text{orth}_v \mathbf{w} = \mathbf{w} - \text{proj}_v \mathbf{w}$ of the vector w onto the vector v , where $\text{proj}_v \mathbf{w}$ is the vector (parallel) projection of w onto v
Example	orth([6,1,-2], [0,4,7]) returns $[60/41, 174/41, 267/41] \approx [1.46, 4.24, 6.51]$.
See Also	COMP, PROJ

PERP

Description	The perp function computes an orthogonal complement of a two-dimensional vector.
Access	Vectors submenu in the TAMUCALC custom menu
Syntax	perp(v)

Input \mathbf{v} : A vector having 2 components
Output An orthogonal complement $\mathbf{v}^\perp = [-v_2, v_1]$ of the vector $\mathbf{v} = [v_1, v_2]$
Example `perp([6,1])` returns `[-1, 6]`.
See Also **ORTH**

PLANE3PT

Description The `plane3pt` function finds a Cartesian representation of the plane through three points in space.
Access **CLP** submenu in the **TAMUCALC** custom menu
Syntax `plane3pt(a,b,c)`
Input \mathbf{a} : A point on the plane
 \mathbf{b} : A different point on the plane
 \mathbf{c} : A third point on the plane; \mathbf{a} , \mathbf{b} , and \mathbf{c} are noncollinear.
Output A Cartesian equation of the plane, $\vec{n} \cdot \langle x, y, z \rangle = \vec{n} \cdot \vec{a}$, where $\vec{n} = (\vec{b} - \vec{a}) \times (\vec{c} - \vec{a})$
Example `plane3pt([0,4,6], [5,1,-1], [2,6,0])` returns `32x + 16y + 16z = 160`.
See Also **PLANENPT**

PLANENPT

Description The `planenpt` function finds a Cartesian representation of the plane through a point having a given normal vector.
Access **CLP** submenu in the **TAMUCALC** custom menu
Syntax `planenpt(n,p)`
Input \mathbf{n} : A normal vector to the plane
 \mathbf{p} : A point on the plane
Output A Cartesian equation of the plane, $\vec{n} \cdot \langle x, y, z \rangle = \vec{n} \cdot \vec{p}$
Example `planenpt([3,-4,-7], [8,2,0])` returns `3x - 4y - 7z = 16`.
See Also **PLANE3PT**

PLINE2PT

Description	The pline2pt function finds a parametric representation of the (hyper)line between two points.
Access	CLP submenu in the TAMUCALC custom menu
Syntax	pline2pt(a,b)
Input	a : A point on the (hyper)line b : A different point on the (hyper)line
Output	A parametric representation $\vec{L}(t) = \vec{a} + t(\vec{b} - \vec{a})$ of the (hyper)line
Example	pline2pt([-3,5,-2], [-4,6,7]) returns $[-t - 3, t + 5, 9t - 2]$.
See Also	CLINE2PT, LINPTDIR

POT

Description	The pot function determines whether a vector field is conservative. If so, it returns a potential function of the vector field; if not, FALSE is returned.
Access	FVMD submenu in the TAMUCALC custom menu
Syntax	pot(w,v)
Input	w : A vector field v : A vector of independent variables with the same number of components as w
Output	A potential function f for w (i.e., $\mathbf{w} = \vec{\nabla} f$) if w is conservative; otherwise, FALSE
Example	pot([4*x*e^(z), cos(y), 2*x^2*e^(z)], [x,y,z]) returns $2x^2e^z + \sin y$.
See Also	FTLI

PROJ

Description	The proj function computes the vector projection of one vector onto another.
Access	Vectors submenu in the TAMUCALC custom menu
Syntax	proj(v,w)
Input	v : A vector having 2, 3, or more components w : A vector with the same number of components as v
Output	The vector (parallel) projection $\text{proj}_{\mathbf{v}}\mathbf{w} = \left(\frac{\mathbf{v} \cdot \mathbf{w}}{\ \mathbf{v}\ ^2}\right) \frac{\mathbf{v}}{\ \mathbf{v}\ }$ of the vector w onto the vector v
Example	proj([6,1,-2], [0,4,7]) returns $[-60/41, -10/41, 20/41] \approx [-1.46, -0.24, 0.49]$.
See Also	COMP, ORTH

RSUM

Description	The rsum function computes an approximate (numerical) Riemann sum using right endpoints of subintervals. This is an estimate of $\int_a^b f(x) dx$.
Access	Calc submenu in the TAMUCALC custom menu
Syntax	rsum(f,a,b,n)
Input	f : A string (either the name of a function or the name of a list of function values) a : Lower horizontal (x) value b : Upper horizontal (x) value n : Number of horizontal subintervals
Output	The Riemann sum $\sum_{k=1}^n f(x_k^*) \Delta x_k$, where $x_k^* = a + kh$ and $\Delta x_k = \Delta x = h = \frac{b-a}{n}$, as a floating point decimal
Examples	1/x → f(x) rsum("f", 2, 3, 10) returns 0.397247; the exact answer is $\int_2^3 \frac{1}{x} dx = \ln \frac{3}{2} \approx 0.405465$. DelVar f: {0, 1, 3, 4.5, 5, 4.5, 3, 2, 1} → f rsum("f", 0, 4, 0) returns 12.00. The routine in this case automatically computes n to be one fewer than the number of elements in the list f ; in other words, the number of equal-length subintervals of $[a, b]$ determined by the data.
See Also	LSUM, MSUM, SIMP, TRAP

SIMP

Description	The simp function numerically approximates the definite integral $\int_a^b f(x) dx$ using the Simpson's Rule.
Access	Calc submenu in the TAMUCALC custom menu
Syntax	simp(f,a,b,n)
Input	f : A string (either the name of a function or the name of a list of function values) a : Lower horizontal (x) value b : Upper horizontal (x) value n : Number of horizontal subintervals (n must be even!)
Output	The expression $\frac{1}{3}\Delta x \left(f(x_0) + f(x_n) + 4 \sum_{j=1}^{n/2} f(x_{2j-1}) + 2 \sum_{k=1}^{n/2-1} f(x_{2k}) \right)$ where $x_i = a + ih$, $i = 0, 1, \dots, n$ and $\Delta x = h = \frac{b-a}{n}$
Examples	1/x → f(x) simp("f", 2, 3, 10) returns 0.405465; the exact answer is $\int_2^3 \frac{1}{x} dx = \ln \frac{3}{2} \approx 0.405465$. DelVar f: {0, 1, 3, 4.5, 5, 4.5, 3, 2, 1} → f simp("f", 0, 4, 0) returns 11.83. The routine in this case automatically computes n to be one fewer than the number of elements in the list f ; in other words, the number of equal-length subintervals of $[a, b]$ determined by the data.
See Also	LSUM, MSUM, RSUM, TRAP

SIS

Description	The sis function computes the surface integral of a scalar field.
Access	LSI submenu in the TAMUCALC custom menu
Syntax	sis("f", "S", u, g, h, v, a, b)
Input	f : A string (the name of the scalar field) S : A string (the name of the parameterization $\vec{S}(u, v)$ of the surface) u : Inner variable of integration g : Lower limit of the inner range of integration h : Upper limit of the inner range of integration v : Outer variable of integration a : Lower limit of the outer range of integration b : Upper limit of the outer range of integration
Output	The surface integral $\iint_S f \, dS = \int_a^b \int_g^h f(\vec{S}(u, v)) \ \vec{S}_u \times \vec{S}_v\ \, du \, dv$
Example	7+x^2+y^2 → f(x,y,z): [r*cos(θ), r*sin(θ), 36-4*r^2] → s(r,θ): sis("f", "s", r, 0, 3, θ, 0, 2*π) returns $\frac{(381397\sqrt{577} - 373)\pi}{5120} \approx 5621.18$. (On a TI-89, an uppercase "S" is regarded as a lowercase "s." The latter is one less keystroke.)
See Also	SIV ; FCS and FVMD submenus of the TAMUCALC custom menu

SIV

Description	The siv function computes the surface integral of a vector field.
Access	LSI submenu in the TAMUCALC custom menu
Syntax	siv("w", "S", u, g, h, v, a, b)
Input	w : A string (the name of the vector field) S : A string (the name of the parameterization $\vec{S}(u, v)$ of the <i>oriented</i> surface) u : Inner variable of integration g : Lower limit of the inner range of integration h : Upper limit of the inner range of integration v : Outer variable of integration a : Lower limit of the outer range of integration b : Upper limit of the outer range of integration
Output	The surface integral $\iint_S \vec{w} \cdot d\vec{S} = \int_a^b \int_g^h \vec{w}(\vec{S}(u, v)) \cdot (\vec{S}_u \times \vec{S}_v) \, du \, dv$
Example	[e^y, y*e^x, x^2*y] → w(x,y,z): [x, y, x^2+y^2] → s(x,y): siv("w", "s", x, 0, 1, y, 0, 1) returns $\frac{11}{6} - \frac{5e}{3} \approx -2.70$. (On a TI-89, an uppercase "S" is regarded as a lowercase "s." The latter is one less keystroke.)
See Also	SIS ; FCS and FVMD submenus of the TAMUCALC custom menu
Note	If the parameterization of $\vec{S}(u, v)$ is such that the normal to the surface, $\vec{S}_u \times \vec{S}_v$, has the opposite orientation to the desired one, simply negate the siv command. That is, specify -siv(...) .

SMI

Description	The smi (stepwise multiple integration) program shows the steps involved in multiple integration by repeatedly computing antiderivatives and applying the Fundamental Theorem of Calculus. This is done in a fully automatic fashion. The routine also handles single integrals.
Access	Calc submenu in the TAMUCALC custom menu
Syntax	smi(): solution
Input	There are no arguments; the user is prompted for input.
Output	The value of a (multiple) definite integral is stored in the global variable solution , then displayed on the Home screen. Along the way, every step of the integration is shown. In addition to the exact value of the integral, a decimal approximation is also given for convenience.
Example	smi(): solution (See NOTE below.)
See Also	Muint submenu in the TAMUCALC menu

Note Here is a transcript of the steps involved in the integration of the triple integral $\int_1^2 \int_3^4 \int_5^6 x^2y + 5xz \, dz \, dy \, dx$. Typically, the user first computes the integral using one of the templates in the **Muint** submenu of the **TAMUCALC** menu, copies the input sitting in the entry line, then pastes it in the Program IO area at the first **smi** prompt. (Alternatively, one may provide the integrand, then the ranges from innermost to outermost when prompted.) Below, TI-89 prompts are left-justified and user responses are indented. (**Remark:** Whenever the PAUSE annunciator appears, the user simply presses ENTER to continue. This allows time to read and/or scroll intermediate results.)

smi(): solution

Enter complete integral or just the integrand.

$\int(\int(\int(x^2*y+5*x*z, z, 5, 6), y, 3, 4), x, 1, 2)$

Antiderivative w.r.t. z: $x^2yz + \frac{5xz^2}{2}$

When z = 6: $6x^2y + 90x$

When z = 5: $5x^2y + \frac{125x}{2}$

Difference: $x^2y + \frac{55x}{2}$

Antiderivative w.r.t. y: $\frac{x^2y^2}{2} + \frac{55xy}{2}$

When y = 4: $8x^2 + 110x$

When y = 3: $\frac{9x^2}{2} + \frac{165x}{2}$

Difference: $\frac{7x^2}{2} + \frac{55x}{2}$

Antiderivative w.r.t. x: $\frac{7x^3}{6} + \frac{55x^2}{4}$

When x = 2: $\frac{193}{3}$

When x = 1: $\frac{173}{12}$

Difference: $\frac{593}{12}$

Answer ↑ and approximation ↓

49.42

STP

Description The **stp** function computes the scalar triple product of three vectors.

Access **Vectors** submenu in the **TAMUCALC** custom menu

Syntax **stp(a,b,c)**

Input **a**: A vector with 3 components
b: Another vector with 3 components
c: A third vector with 3 components

Output The scalar triple product, $\mathbf{a} \cdot (\mathbf{b} \times \mathbf{c})$

Example **stp([3,-2,7], [-4,0,5], [-6,9,8])** returns -391 .

See Also **VTP**

TAYLOREM

Description The **taylorem** function gives the absolute value of Taylor's remainder formula

Access **Calc** submenu in the **TAMUCALC** custom menu

Syntax **taylorem(f,x,n,a)**

Input **f**: Functional expression in **x**
x: Independent variable
n: Order
a: Center of the Taylor expansion

Output The absolute value of the remainder term in Taylor's theorem, $\left| \frac{f^{(n+1)}(\xi) (x-a)^{n+1}}{(n+1)!} \right|$, where ξ is between x and a .

Example **taylorem(sin(x), x, 4, 0)** returns $\left| \frac{x^5 \cos \xi}{120} \right|$ where ξ is between x and 0.

See Also **TAYLOR** in the TI-89 Guidebook

TRAP

Description The **trap** function numerically approximates the definite integral $\int_a^b f(x) dx$ using the Trapezoidal Rule.

Access **Calc** submenu in the **TAMUCALC** custom menu

Syntax **trap(f,a,b,n)**

Input **f**: A string (either the name of a function or the name of a list of function values)
a: Lower horizontal (x) value
b: Upper horizontal (x) value
n: Number of horizontal subintervals

Output The expression $\left(\frac{1}{2}(f(x_0) + f(x_n)) + \sum_{k=1}^{n-1} f(x_k)\right) \Delta x$ where $x_k = a + kh$, $k = 0, 1, \dots, n$ and $\Delta x = h = \frac{b-a}{n}$

Examples $1/x \rightarrow f(x)$

trap("f", 2, 3, 10) returns 0.405581; the exact answer is $\int_2^3 \frac{1}{x} dx = \ln \frac{3}{2} \approx 0.405465$.

DelVar f: {0, 1, 3, 4.5, 5, 4.5, 3, 2, 1} → f

trap("f", 0, 4, 0) returns 11.75. The routine in this case automatically computes n to be one fewer than the number of elements in the list f ; in other words, the number of equal-length subintervals of $[a, b]$ determined by the data.

See Also **LSUM, MSUM, RSUM, SIMP**

TRIPSUMD

Description The **tripsumd** function computes an approximate (numerical) triple Riemann sum using the middle (center) points of subboxes. This is an estimate of the triple integral $\int_r^s \int_c^d \int_a^b f(x, y, z) dx dy dz$.

Access **LSI** submenu in the **TAMUCALC** custom menu

Syntax **tripsumd("f",a,b,L,c,d,m,r,s,n)**

Input **f**: A string (the name of the function in the integrand)

a: Lower x -value

b: Upper x -value

L: Number of x -subintervals

c: Lower y -value

d: Upper y -value

m: Number of y -subintervals

r: Lower z -value

s: Upper z -value

n: Number of z -subintervals

Output The triple Riemann sum $\sum_{i=1}^L \sum_{j=1}^m \sum_{k=1}^n f(x_i^*, y_j^*, z_k^*) \Delta x_i \Delta y_j \Delta z_k$, where

$$x_i^* = a + \left(i - \frac{1}{2}\right) h_1 \text{ \& } \Delta x_i = \Delta x = h_1 = \frac{b-a}{L} \text{ and}$$

$$y_j^* = c + \left(j - \frac{1}{2}\right) h_2 \text{ \& } \Delta y_j = \Delta y = h_2 = \frac{d-c}{m} \text{ and}$$

$$z_k^* = r + \left(k - \frac{1}{2}\right) h_3 \text{ \& } \Delta z_k = \Delta z = h_3 = \frac{s-r}{n}, \text{ as a floating point decimal}$$

Example **e^(x^3+y^3+z^3) → f(x,y,z): tripsumd("f", 0,1,4, 0,1,4, 0,1,4)** returns 2.308752.

See Also **DBLSUMBL, DBLSUMBR, DBLSUMMD, DBLSUMTL, DBLSUMTR**

UNITVEC

Description The **unitvec** function computes the unit vector in the same direction as a given vector.

Access **Vectors** submenu in the **TAMUCALC** custom menu

Syntax	unitvec(v)
Input	v : A vector having 2, 3, or more components
Output	The unit vector $\hat{\mathbf{v}} = \frac{\mathbf{v}}{\ \mathbf{v}\ }$ in the same direction as the vector v
Example	unitvec([-3, 12, 4]) returns $[-3/13, 12/13, 4/13] \approx [-0.23, 0.92, 0.31]$.
See Also	LEN

VIEW

Description	The view program displays a large expression and permits easy scrolling.
Access	Calc submenu in the TAMUCALC custom menu
Syntax	view(x)
Input	x : An expression, typically the one just computed, ans(1)
Output	The expression is displayed in the Program IO window in regular (as opposed to reverse) video. The usual cursor keys allow for easy scrolling. Press ENTER when viewing is complete.
Example	solve(x^2+y^2=1 and x=λ*x and y=-λ*y, {x,y,λ}) then view(ans(1)) allows one to view the expression $x = 1 \text{ and } y = 0 \text{ and } \lambda = 1 \text{ or } \dots \text{ or } x = -1 \text{ and } y = 0 \text{ and } \lambda = 1$ by using the cursor keys to scroll.
See Also	The Keyboard section in Chapter 2 of the TI-89 Guidebook

VTP

Description	The vtp function computes the vector triple product of three vectors.
Access	Vectors submenu in the TAMUCALC custom menu
Syntax	vtp(a,b,c)
Input	a : A vector with 3 components b : Another vector with 3 components c : A third vector with 3 components
Output	The vector triple product, $\mathbf{a} \times (\mathbf{b} \times \mathbf{c})$
Example	vtp([3,-2,7], [-4,0,5], [-6,9,8]) returns $[58, -207, -84]$.
See Also	STP

XLSUM

Description	The xlsun function computes an exact (symbolic) Riemann sum using left endpoints of subintervals. This is an estimate of $\int_a^b f(x) dx$.
Access	Calc submenu in the TAMUCALC custom menu
Syntax	xlsun(f,a,b,n)
Input	f : A string (the name of a function) a : Lower horizontal (x) value b : Upper horizontal (x) value n : Number of horizontal subintervals, typically as a symbol (variable with no value)
Output	The Riemann sum $\sum_{k=1}^n f(x_k^*) \Delta x_k$, where $x_k^* = a + (k-1)h$ and $\Delta x_k = \Delta x = h = \frac{b-a}{n}$, as a symbolic expression
Example	DelVar n: x^2+4*x-7→f(x) xlsun("f", 3, 5, n) returns $\frac{4(38n^2 - 18n + 1)}{3n^2}$. Then limit(ans(1), n, ∞) yields $\frac{152}{3}$, the <i>exact</i> value of $\int_3^5 x^2 + 4x - 7 dx$.
See Also	XMSUM, XRSUM

XMSUM

Description	The xmsun function computes an exact (symbolic) Riemann sum using midpoints of subintervals. This is an estimate of $\int_a^b f(x) dx$.
Access	Calc submenu in the TAMUCALC custom menu
Syntax	xmsun(f,a,b,n)
Input	f : A string (the name of a function) a : Lower horizontal (x) value b : Upper horizontal (x) value n : Number of horizontal subintervals, typically as a symbol (variable with no value)
Output	The Riemann sum $\sum_{k=1}^n f(x_k^*) \Delta x_k$, where $x_k^* = a + (k - \frac{1}{2})h$ and $\Delta x_k = \Delta x = h = \frac{b-a}{n}$, as a symbolic expression
Example	DelVar n: x^2+4*x-7→f(x) xmsun("f", 3, 5, n) returns $\frac{2(76n^2 - 1)}{3n^2}$. Then limit(ans(1), n, ∞) yields $\frac{152}{3}$, the <i>exact</i> value of $\int_3^5 x^2 + 4x - 7 dx$.
See Also	XLSUM, XRSUM

XRSUM

Description	The xrsum function computes an exact (symbolic) Riemann sum using right endpoints of subintervals. This is an estimate of $\int_a^b f(x) dx$.
Access	Calc submenu in the TAMUCALC custom menu
Syntax	xrsum(f,a,b,n)
Input	f : A string (the name of a function) a : Lower horizontal (x) value b : Upper horizontal (x) value n : Number of horizontal subintervals, typically as a symbol (variable with no value)
Output	The Riemann sum $\sum_{k=1}^n f(x_k^*) \Delta x_k$, where $x_k^* = a + kh$ and $\Delta x_k = \Delta x = h = \frac{b-a}{n}$, as a symbolic expression
Example	DelVar n: x^2+4*x-7→f(x) xrsum("f", 3, 5, n) returns $\frac{4(38n^2 + 18n + 1)}{3n^2}$. Then limit(ans(1), n, ∞) yields $\frac{152}{3}$, the <i>exact</i> value of $\int_3^5 x^2 + 4x - 7 dx$.
See Also	XLSUM, XMSUM

ZO

Description	The zo ("zero or one") function is an indicator function.
Access	Just type it in or use the VAR-LINK menu.
Syntax	zo(b)
Input	b : Boolean expression
Output	If b is TRUE , then 1 is returned; if b is FALSE , 0 is returned.
Examples	zo(5>4) returns 1, whereas zo(5=4) returns 0.
See Also	Conditional Tests in the TI-89 Guidebook
