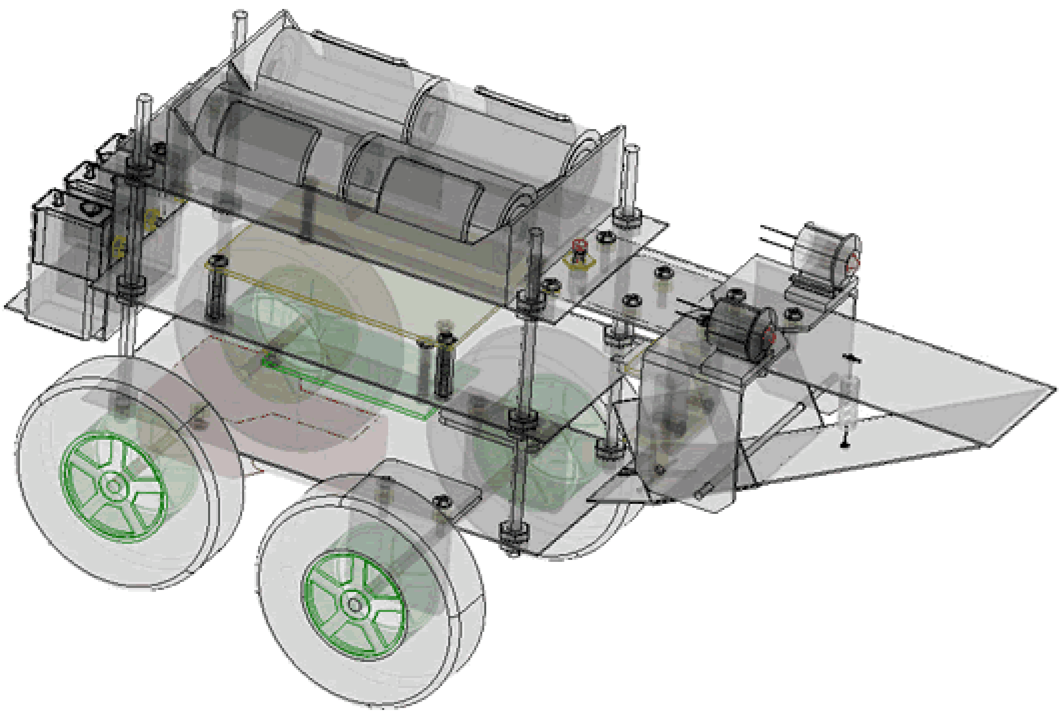
Tools for Advanced Math on the TI-89

Featuring

Vector Calculus, Linear Algebra, and Differential Geometry



(This is my robot. It has nothing to do with this program, but I like it.)

Eric Beltt
belt0033@tc.umn.edu
http://home.earthlink.net/~ebeltt/
October 26, 2002

Table of Contents

Introduction

Disclaimer

The topics addressed by this program are relatively complicated. The functions simply perform a set of operations on the input parameters, so the answer is only as good as the input. As such, you should take this program for what it is. I designed it as a set of tools to check my answers on homework and exams. It won't do the problems for you, and it won't replace the good old-fashioned hard work required to succeed in whatever application you are using the program for.

I spent a good deal of time checking the functions to make sure I could get the correct answer on sample problems. This doesn't mean that the program is bug free. Since it is written in basic, it's very unlikely that it will damage your calculator, but it may not produce the correct answer under certain circumstances. If you are going to blame me for incorrect answers, please don't use this program. If you believe you found a mistake, please email me so I can fix it immediately. Check to make sure I haven't already fixed it in a new version as well. The same goes for this document.

Lastly, I would like to point out that this document isn't a math textbook. If you don't know what the functions are for, you probably won't learn by reading through this. The notation I used might also be different than what you are used to. If you would like to know more about a formula that I used, you can consult the specific textbook where I found it. Each of them is listed at the end of this document.

Message from the Author

My motivation for designing the set of functions that this program is based on was rather selfish. I just wanted to make it a little easier to get through multivariable calculus. I decided to release the set of functions, hoping it would do the same for some other people. Since then, I have had several people email me asking for a better explanation of how the functions work, and I suspect that a lot more people just gave up without asking for help. That's why I decided to write this program. My hope was that by making the functions easier to use, they would be more useful to people. I sincerely hope that I have accomplished that.

If you get a lot of use out of this program, I would love to hear from you. As I say in the "About" section of the program, questions, comments and requests are welcome and appreciated. If you find a bug or if there is something you would like me to add to the next release, please let me know. I hope this document answers all of your questions, but feel free to email me if you still have some.

Basics

Installation

You need to install the program and all of the functions into a folder called mv. Beyond that, there is nothing special you need to do.

Running the Program

The main program is called tool( ). Just enter this in the calculator and press enter. You will have to wait about 30 seconds the first time you run it, but it should start up quickly when you run it in the future. If you are not in the mv folder when you start the program, you will need to type mv\tool( ) to start the program. The program runs from that folder, but will return to your current folder when you exit. If the program doesn't stop normally, you will need to change the folder back manually.

Entering Parameters

Almost every parameter you will enter while using this program needs to be in the form of a vector. Even if there is only one element, you still need to put brackets [ ] around it. You can determine the proper form from the input example in the dialog box. If there are brackets, you need to use them. If there are none, you shouldn't use them. I would have made it simpler, but the nature of advanced math requires the use of vectors.

Results

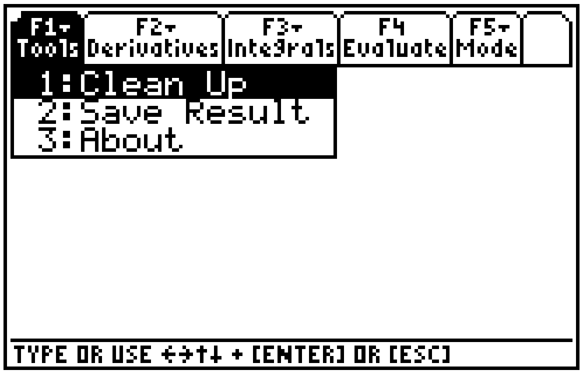
Each time a computation is performed, the result is temporarily stored to a variable called mv_ans. If you want to save the result permanently, use the "Save Result" feature under the "Tools" menu. Remember that your saved variable will be in the mv folder unless you specify the folder you want in the variable name (ex: main\var).

Notes

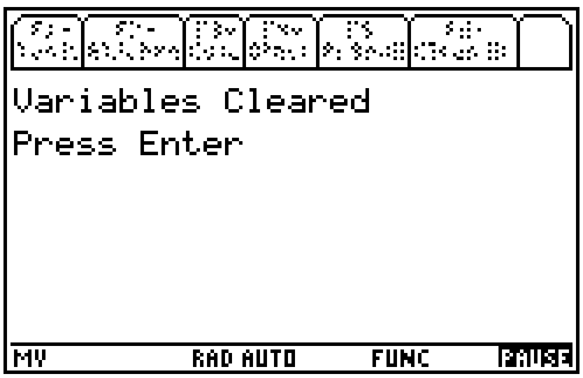
You may notice that there is an annoying "feature" built into the TI-89. Every time a dialog box appears, Alpha-Lock is turned on. If you want to disable this, there is at least one program out there to do it. The one I found is called autoaoff. I haven't tried it and I am not associated with it, nor do I take any responsibility for what it does to your calculator. If you really find auto Alpha-Lock annoying though, this is the way to disable it.

Graphical User Interface

Tools

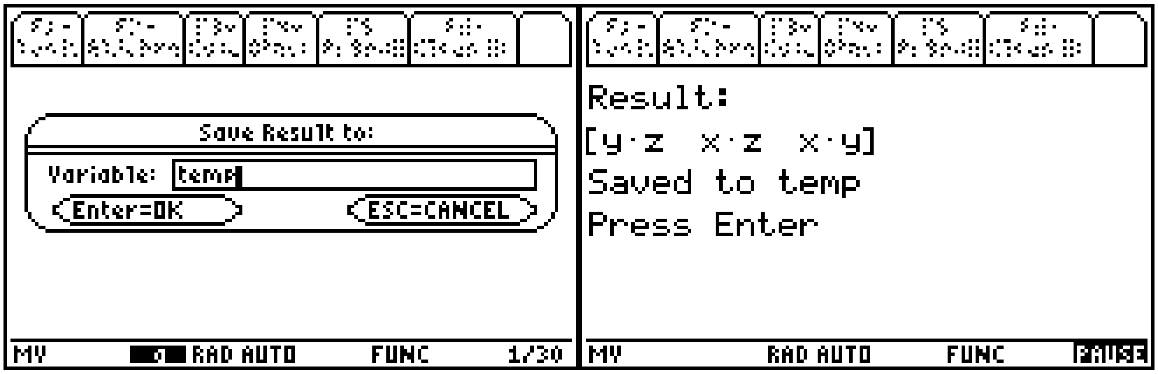


Clean Up



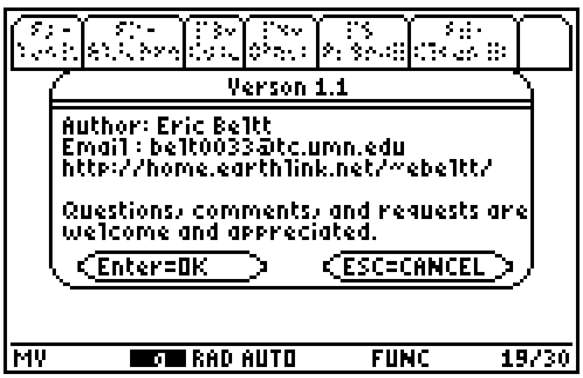
This feature deletes all of the unnecessary variables that are generated while using the program. The program saves the information you enter into a dialog box as various strings to save you from unnecessary repetition. If you want the dialog boxes for a function to start empty, just use Clean Up before hand. This does not delete saved variables. Clean Up also clears the screen.

Save Result



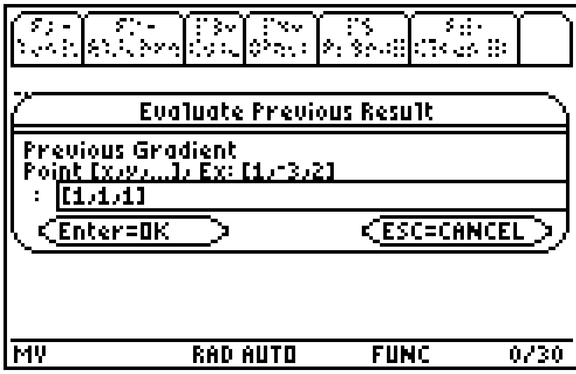
This feature allows you to save the previous result to a permanent variable. If you want the result saved to a folder other than mv, you need to specify that in the variable name. For example, main\temp would save the contents of mv_ans to a variable in the main folder called temp.

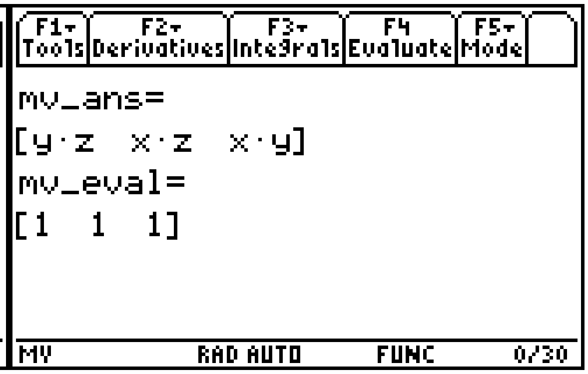
About



About displays what is shown in the picture above. Notice that the program's version number is listed in the title of the dialog box.

Evaluate



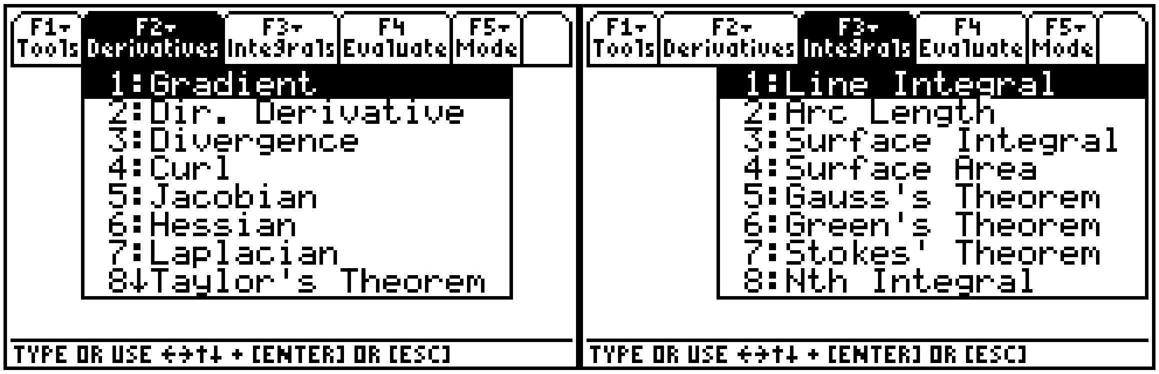


Evaluate takes the previous result and evaluates it at the given point. It uses the function fofg to accomplish this. The result is stored in a variable called mv_eval. Notice that title of the previous computation is displayed in the dialog box. You will need to know the appropriate number of coordinates for the point.

Mode

Mode is where you choose what set of functions will be displayed in the toolbar. The default is Vector Calculus. If you want to change the default mode, you will need to edit the program. There is a variable responsible for the default mode is called mv_mode, and it is located near the top of the code.

Vector Calculus



Derivatives

<u>Gradient</u>

$$\vec{\nabla} f(\vec{x}) = \left(\frac{\partial f}{\partial x_1}(\vec{x}), \frac{\partial f}{\partial x_2}(\vec{x}), \ldots, \frac{\partial f}{\partial x_n}(\vec{x}) \right)$$

$$grad(f, \vec{x})$$

<u>Directional Derivative</u>

$$D_{\vec{u}} f(\vec{x}) = \vec{\nabla} f(\vec{x}) \cdot \vec{u}$$

$$dirder(f, \vec{x}, \vec{u})$$

<u>Divergence</u>

$$div(\vec{F}(\vec{x})) = \frac{\partial F_1}{\partial x_1} + \frac{\partial F_2}{\partial x_2} + \frac{\partial F_3}{\partial x_3}$$

$$div(\vec{F}, \vec{x})$$

Curl

$$curl(\vec{F}(\vec{x})) = \left(\frac{\partial F_3}{\partial x_2} - \frac{\partial F_2}{\partial x_3}, \frac{\partial F_1}{\partial x_3} - \frac{\partial F_3}{\partial x_1}, \frac{\partial F_2}{\partial x_1} - \frac{\partial F_1}{\partial x_2} \right)$$

$$curl(\vec{F}, \vec{x})$$

Jacobian Matrix

$$J\vec{f} = \left[\frac{\partial f_i}{\partial x_j}(\vec{x}) \right]$$

$$jacobian(\vec{F}, \vec{x})$$

Hessian Matrix

$$Hf(\vec{x}) = \left[\frac{\partial^2 f}{\partial x_j \partial x_i}(\vec{x}) \right]$$

$$hessian(f, \vec{x})$$

Laplacian

$$\vec{\nabla}^2 f(\vec{x}) = \frac{\partial^2 f}{\partial x^2}(\vec{x}) + \frac{\partial^2 f}{\partial y^2}(\vec{x}) + \frac{\partial^2 f}{\partial z^2}(\vec{x})$$

$$laplace(f, \vec{x})$$

Taylor's Theorem (First Order)

$$f(\vec{x}) = f(\vec{a}) + \vec{\nabla} f(\vec{a}) \cdot (\vec{x} - \vec{a}) + \frac{1}{2}(\vec{x} - \vec{a})^T Hf(\vec{x}_0)(\vec{x} - \vec{a})$$

$$vtaylor(f, \vec{x}, \vec{a})$$

Unit Normal

$$U_{\vec{x}} = \frac{\vec{g}_u \times \vec{g}_v}{\left\| \vec{g}_u \times \vec{g}_v \right\|} : \vec{g}_u = \frac{\partial \vec{g}}{\partial x_1}, \vec{g}_v = \frac{\partial \vec{g}}{\partial x_2}$$

$$unit\_n(g, \vec{x})$$

Integrals

<u>Line Integral</u>

$$\int_C \vec{F} \cdot d\vec{x} = \int_L \vec{F}(\vec{f}(t)) \cdot \vec{f}'(t)dt$$

$$vline(\vec{F}, \vec{x}, \vec{f}, t, L)$$

<u>Arc Length</u>

$$L(C) = \int_C \left\| \vec{f}'(t) \right\| dt$$

$$arclngth(\vec{f}, t, C)$$

<u>Surface Integral</u>

$$\iint_M g\, d\sigma = \iint_R g(\vec{f}(s,t)) \left\| \frac{\partial \vec{f}}{\partial s} \times \frac{\partial \vec{f}}{\partial t} \right\| dsdt$$

$$vsurface(g, \vec{\sigma}, \vec{f}, [s \quad t], R_s, R_t)$$

<u>Surface Area</u>

$$\sigma(M) = \iint_R \left\| \frac{\partial \vec{f}}{\partial s} \times \frac{\partial \vec{f}}{\partial t} \right\| dsdt$$

$$surfarea(\vec{f}, [s \quad t], R_s, R_t)$$

<u>Gauss's Theorem</u>

$$\oiint_{\partial S} \vec{F} \cdot \vec{n} = \iiint_S div(\vec{F}) dV$$

$$gauss(\vec{F}, \vec{x}, S_1, S_2, S_3)$$

<u>Green's Theorem</u>

$$\oint_{\partial R} F_1 dx + F_2 dy = \iint_R \left(\frac{\partial F_2}{\partial x} - \frac{\partial F_1}{\partial y} \right) dA$$

$$green(\vec{F}, \vec{x}, R_1, R_2)$$

<u>Stokes' Theorem</u>

$$\oint_{\partial M} \vec{F} \cdot d\vec{x} = \iint_M curl(\vec{F}) \cdot \vec{n} d\sigma = \iint_R curl(\vec{F}(\vec{f}(s,t))) \cdot \left(\frac{\partial \vec{f}}{\partial s} \times \frac{\partial \vec{f}}{\partial t} \right) dsdt$$

$$stokes(\vec{F}, \vec{x}, \vec{f}, [s \quad t] R_1, R_2)$$

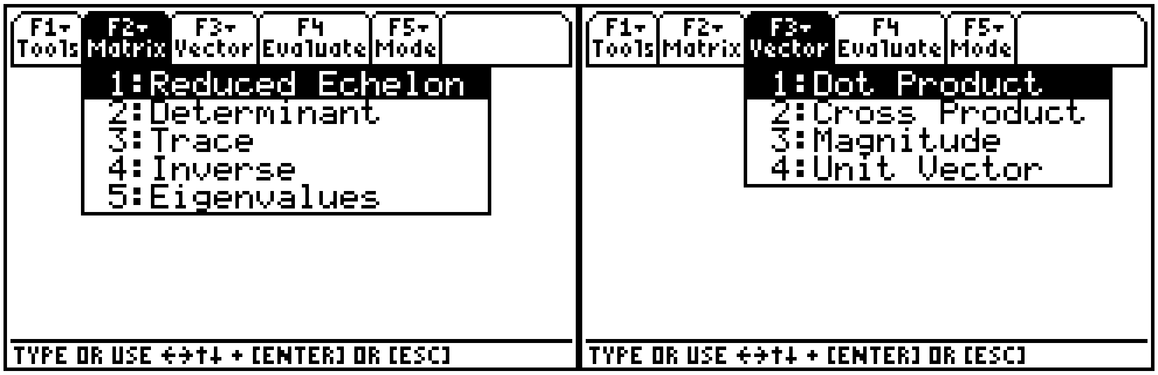
<u>Nth Integral</u>

$$\int FdL, \iint FdA, \iiint FdV, ...$$

$$nth(F, \vec{x})$$

Note: There isn't actually an "nth" function. The code is included in the main program. The number of integrations and their order are determined by the vector x. Bounds are gathered through successive dialog boxes.

Linear Algebra



Matrix

<u>Reduced Echelon Form</u>

$$\left[\begin{array}{ccc:c} a_1 & 0 & 0 & x_1 \\ 0 & a_2 & 0 & x_2 \\ \hdashline 0 & 0 & a_n & x_n \end{array}\right]$$

$rref(A)$ (TI-89)

<u>Determinant</u>

$$\det([A]) = \prod \lambda_i$$

$det(A)$ (TI-89)

<u>Trace</u>

$$tr([A]) = \sum \lambda_i$$

$trc(A)$

<u>Inverse</u>

$$[A] \cdot [A]^{-1} = 1$$

A^{-1} (TI-89)

Eigenvalues

$[A]\vec{v} = \lambda\vec{v}$

$eigenval(A)$

Vector

$\vec{x} : \Re^n \to \Re^m$

Dot Product

$dotP(x, y) = \vec{x} \cdot \vec{y}$

$dotP(\vec{x}, \vec{y})$ (TI-89)

Cross Product

$crossP(x, y) = \vec{x} \times \vec{y}$

$crossP(\vec{x}, \vec{y})$ (TI-89)

Magnitude

$\|\vec{x}\| = \sqrt{\vec{x} \cdot \vec{x}}$

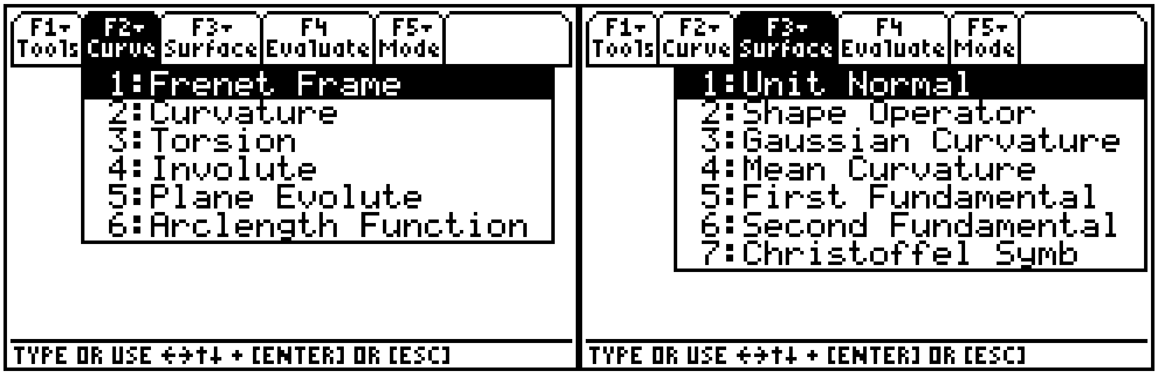
$mag(\vec{x})$

Unit Vector

$$u_{\vec{x}} = \frac{\vec{x}}{\|\vec{x}\|}$$

$unitV(\vec{x})$

Differential Geometry



Curve

$\alpha(t) : I \to \Re^3$

Frenet Frame

$$\begin{bmatrix} T \\ N \\ B \end{bmatrix} = \begin{bmatrix} \dfrac{\alpha'}{\|\alpha'\|} \\ \dfrac{\alpha' \times \alpha''}{\|\alpha' \times \alpha''\|} \times \dfrac{\alpha'}{\|\alpha'\|} \\ \dfrac{\alpha' \times \alpha''}{\|\alpha' \times \alpha''\|} \end{bmatrix}$$

$frenet(\alpha, t)$

Curvature

$$\kappa = \frac{\|\alpha' \times \alpha''\|}{\|\alpha'\|^3}$$

$frenet\_k(\alpha, t)$

Torsion

$$\tau = \frac{(\alpha' \times \alpha'') \cdot \alpha'''}{\|\alpha' \times \alpha''\|^2}$$

$$frenet\_t(\alpha, t)$$

Involute

$$\mathrm{I}(t) = \alpha(t) - s(t)\frac{\alpha'(t)}{\|\alpha'(t)\|} = \alpha(t) - \int_0^t \|\alpha'(u)\| du \frac{\alpha'(t)}{\|\alpha'(t)\|}$$

$$involute(\alpha, t)$$

Plane Evolute

$$\varepsilon(t) = \alpha(t) + \frac{1}{\kappa(t)} N(t)$$

$$evolute(\alpha, t)$$

Note: This function is only valid for plane curves.

Arclength Function

$$s(t) = \int_0^t \|\alpha'(u)\| du$$

$$arclngth(\alpha, u, t)$$

Note: This function doesn't exist as it is shown above. The third parameter of the arclngth function is actually the integration bounds. The main program uses the arclngth function and, in the place of t above, supplies the bounds [0,t].

Surface

$$\vec{x}(u,v) : D \to \Re^3, \vec{x}_u = \frac{\partial \vec{x}}{\partial u}, \vec{x}_v = \frac{\partial \vec{x}}{\partial v}, \vec{x}_{uu} = \frac{\partial^2 \vec{x}}{\partial u^2}, \vec{x}_{uv} = \frac{\partial^2 \vec{x}}{\partial u \partial v}, \vec{x}_{vv} = \frac{\partial^2 \vec{x}}{\partial v^2}$$

Unit Normal

$$U_{\vec{x}} = \frac{\vec{x}_u \times \vec{x}_v}{\left\| \vec{x}_u \times \vec{x}_v \right\|} = (u_1, u_2, u_3)$$

$$unit\_N(\vec{x},[u \quad v])$$

Shape Operator

$$S_p(\vec{v}) = -\nabla_{\vec{v}} U = -\left(\frac{d}{dt}(u_1(\alpha(t)), \frac{d}{dt}(u_2(\alpha(t)), \frac{d}{dt}(u_3(\alpha(t)) \right)\Big\|_{t=0}$$

$$shape(\vec{x},[u \quad v])$$

Gaussian Curvature

$$K(p) = \det(S_p) = \frac{ln - m^2}{EG - F^2}$$

$$gauss\_k(\vec{x},[u \quad v])$$

Mean Curvature

$$H(p) = \frac{1}{2} trace(S_p) = \frac{Gl + En - 2Fm}{2\left(EG - F^2\right)}$$

$$mean\_h(\vec{x},[u \quad v])$$

First Fundamental Form

$$\begin{bmatrix} E \\ F \\ G \end{bmatrix} = \begin{bmatrix} \vec{x}_u \cdot \vec{x}_u \\ \vec{x}_u \cdot \vec{x}_v \\ \vec{x}_v \cdot \vec{x}_v \end{bmatrix}$$

$$diff\_efg(\vec{x},[u \quad v])$$

Second Fundamental Form

$$\begin{bmatrix} l \\ m \\ n \end{bmatrix} = \begin{bmatrix} S(\vec{x}_u)\cdot\vec{x}_u \\ S(\vec{x}_u)\cdot\vec{x}_v \\ S(\vec{x}_v)\cdot\vec{x}_v \end{bmatrix} = \begin{bmatrix} U\cdot\vec{x}_{uu} \\ U\cdot\vec{x}_{uv} \\ U\cdot\vec{x}_{vv} \end{bmatrix}$$

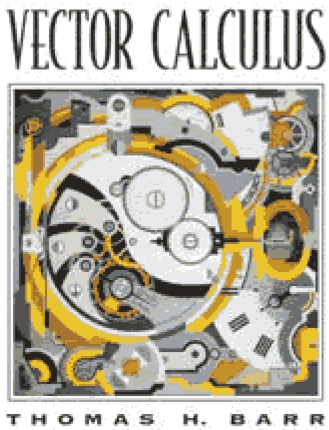
$$diff\_lmn(\vec{x},[u \quad v])$$

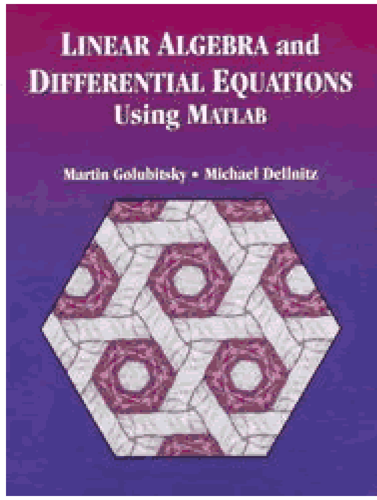
Christoffel Symbols

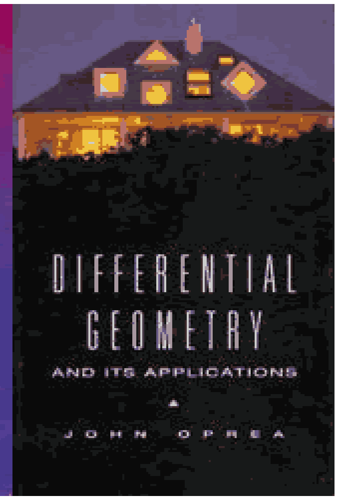
$$\begin{bmatrix} \Gamma^u_{uu} & \Gamma^v_{uu} \\ \Gamma^u_{uv} & \Gamma^v_{uv} \\ \Gamma^u_{vv} & \Gamma^v_{vv} \end{bmatrix} = \begin{bmatrix} \frac{E_u}{2E} & -\frac{E_v}{2G} \\ \frac{E_v}{2E} & \frac{G_u}{2G} \\ -\frac{G_u}{2E} & \frac{G_v}{2G} \end{bmatrix}$$

$$christof(\vec{x},[u \quad v])$$

References







Vector Calculus

Barr, Thomas H. Vector Calculus. Upper Saddle River, NJ: Prentice Hall, 1997

Linear Algebra

Golubitsky, Martin, and Michael Dellnitz. Linear Algebra and Differential Equations using Matlab. New York: Brooks/Cole, 1999

Differential Geometry

Opera, John. Differential Geometry and its Applications. Upper Saddle River, NJ: Prentice Hall, 1997