Student *VectorCalculus* Package - Overview

Dr. Robert J. Lopez
Emeritus Professor - RHIT
Maple Fellow - Maplesoft

Introduction

Starting Point

Loading Student:-VectorCalculus	
Command	**Result**
Vector([*u*, *v*])	$(u)e_x + (v)e_y$
Vector([*x*, *y*], polar)	$(x)e_r + (y)e_\theta$
MapToBasis(*Vector*([*x*, *y*]), polar[*r*, θ])	$\left(\sqrt{x^2+y^2}\right)e_r + (\arctan(y,x))e_\theta$
VectorField([*x*, *y*])	$(x)\bar{e}_x + (y)\bar{e}_y$
VectorField([*x*, *y*], polar)	$(x)\bar{e}_r + (y)\bar{e}_\theta$
simplify(*MapToBasis*(*VectorField*([*x*, *y*]), polar[*r*, θ]))	$(r)\bar{e}_r$
MapToBasis(*VectorField*([*u*, *v*]), polar[*r*, θ]) $(u\cos(\theta)+v\sin(\theta))\bar{e}_r + (-u\sin(\theta)+v\cos(\theta))\bar{e}_\theta$	

One Sure Thing

$\mathbf{R} = \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} r\cos(\theta) \\ r\sin(\theta) \end{bmatrix}$	$\mathbf{R}_r = \begin{bmatrix} \cos(\theta) \\ \sin(\theta) \end{bmatrix} \equiv \hat{e}_r$	$\mathbf{R}_\theta = \begin{bmatrix} -r\sin(\theta) \\ r\cos(\theta) \end{bmatrix} \Rightarrow \hat{e}_\theta = \begin{bmatrix} -\sin(\theta) \\ \cos(\theta) \end{bmatrix}$

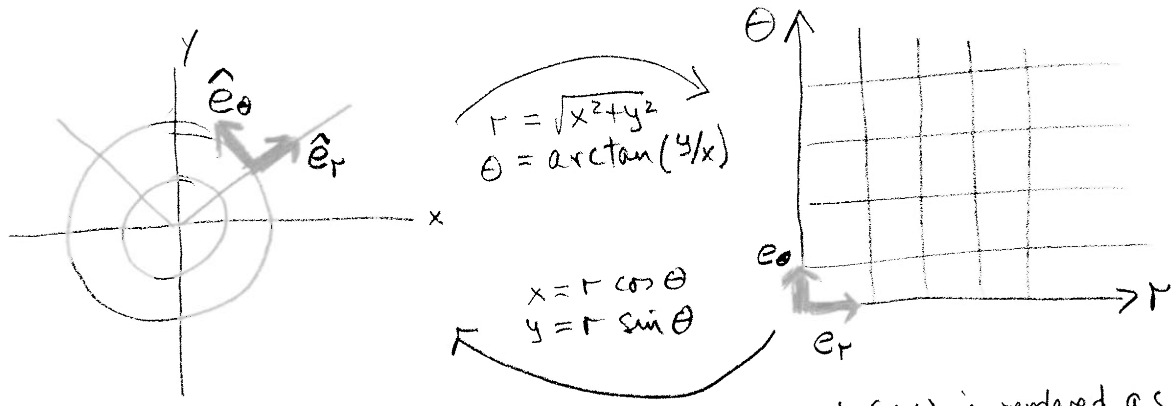
$\mathbf{i}\cos(\theta) + \mathbf{j}\sin(\theta) =$ $-\mathbf{i}\sin(\theta) + \mathbf{j}\cos(\theta) =$	$\Rightarrow$	$\mathbf{i} = \cos(\theta)\ \hat{e}_r - \sin(\theta)\ \hat{e}_\theta$ $\mathbf{j} = \sin(\theta)\ \hat{e}_r + \cos(\theta)\ \hat{e}_\theta c$

$$
\begin{aligned}
u\,\mathbf{i} + v\,\mathbf{j} &= u\left(\cos(\theta)\,\hat{e}_r - \sin(\theta)\,\hat{e}_\theta\right) + v\left(\cos(\theta)\,\hat{e}_r - \sin(\theta)\,\hat{e}_\theta\right) \\
&= \left(u\cos(\theta) + v\sin(\theta)\right)\hat{e}_r + \left(v\cos(\theta) - u\sin(\theta)\right)\hat{e}_\theta
\end{aligned}
$$

Identification of Point and Position Vector

- In Cartesian coordinates, the point (a, b) is identified with the vector $a\,\mathbf{i} + b\,\mathbf{j}$.
- The Cartesian identification of point with position vector is carried over to nonCartesian coordinates.
- A point in nonCartesian coordinates is represented by a fictitious "position vector" in such coordinates.
- At top-level in Maple, points are generally lists: $[a, b]$, but these don't carry a coordinate system.
- Points in the *VectorCalculus* packages, being represented as vectors, carry a coordinate system.

Implications



$\hat{e}_\theta$ rendered as $\bar{e}_\theta$

$\hat{e}_r$ rendered as $\bar{e}_r$

The point (x, y) is rendered as the "vector"

$$\sqrt{x^2+y^2}\, e_r + \arctan(y/x)\, e_\theta$$

Coordinates

The Student *VectorCalculus* package recognizes the five coordinate systems listed in Table 1.

System	Default Names of Coordinate Variables
cartesian	x, y
cartesian	x, y, z
polar	r, θ
cylindrical	r, θ, z
spherical	r, ϕ, θ

Table 1 Coordinate systems recognized by the Student *VectorCalculus* package

Talking Points
• Default names for coordinate variables • Conventions for spherical coordinates • Ambient coordinate system • "Forgiving" nature of the Student package

Table 2 lists the two commands relevant to changing the ambient coordinate system.

Command	Usage
SetCoordinates	SetCoordinates(polar) or SetCoordinates(polar[r,t])
GetCoordinates	GetCoordinates() or GetCoordinates(object)

Table 2 Manipulating coordinate systems

Vector Objects

Table of Basic Vector Objects

Table 3 lists the four basic vector objects in the Student *VectorCalculus* package. These are the free **Vector**, the **RootedVector**, the **PositionVector**, and the **VectorField**.

Object	Usage
Free vector	<a, b> **Vector**([a, b]) **Vector**(<a, b>) **Vector**(<a, b>, polar) **Vector**(<a, b>, polar[r, t])

Rooted vector	**RootedVector**(root = [u, v], <a, b>)
Position vector	**PositionVector**([a, b]) **PositionVector**([f(s), g(s)], polar[r, t]) **PositionVector**([f(u, v), g(u, v), h(u, v)], spherical[ρ, ϕ, θ])
Vector field	**VectorField**(<f(x, y), g(x, y)>) **VectorField**(<f(r, t), g(r, t)>, polar[r, t])
Table 3 Basic vector objects	

Details for the Basic Objects in Table 3

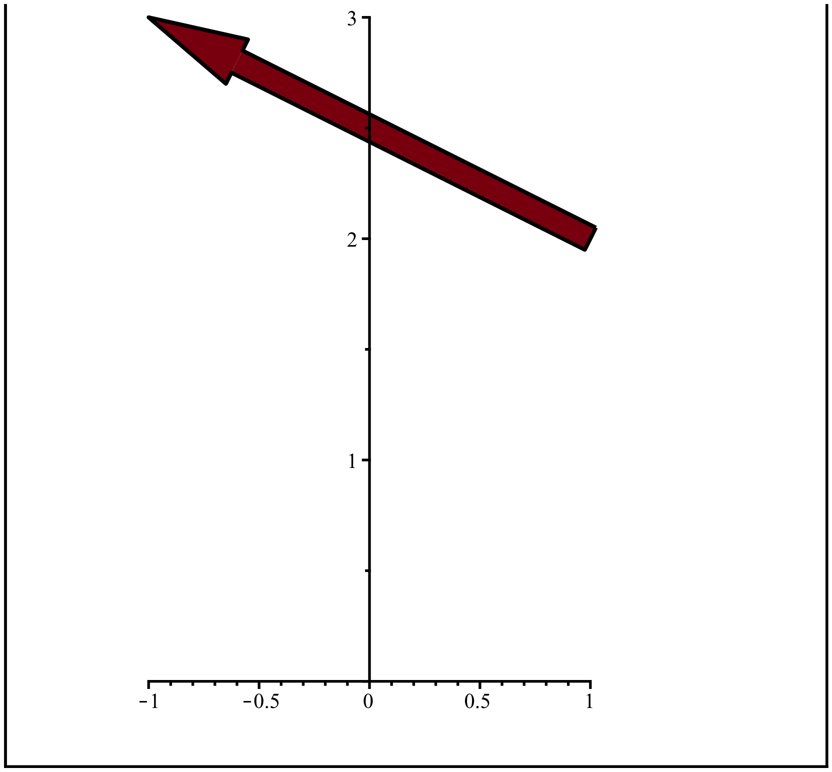
Free Vectors

• Tools>Load Package: Student Vector Calculus	Loading Student:-VectorCalculus
Examples of free vectors	
$\langle a, b\rangle = (a)e_x + (b)e_y$	
$\langle a, b, c\rangle = (a)e_x + (b)e_y + (c)e_z$	
$Vector(\langle a, b\rangle, \text{polar}) = (a)e_r + (b)e_\theta$	
$Vector(\langle a, b, c\rangle, cylindrical) = (a)e_r + (b)e_\theta + (c)e_z$	
$Vector(\langle a, b, c\rangle, spherical[\rho, \phi, \theta]) = (a)e_\rho + (b)e_\phi + (c)e_\theta$	
Table 4 Free vectors	

Rooted Vectors

$with(Student{:}\text{-}VectorCalculus):$

$PlotVector(RootedVector(root = [1, 2], \langle -2, 1\rangle), scaling = constrained, view = [-1..1, 0..3])$



The Position Vector

Loading Student:-VectorCalculus
$PositionVector([s^2, s^3], \text{polar}) = \begin{bmatrix} s^2 \cos(s^3) \\ s^2 \sin(s^3) \end{bmatrix}$
$PositionVector([x, y, f(x, y)]) = \begin{bmatrix} x \\ y \\ f(x, y) \end{bmatrix}$
$PositionVector([1, \phi, \theta], spherical) = \begin{bmatrix} \sin(\phi)\cos(\theta) \\ \sin(\phi)\sin(\theta) \\ \cos(\phi) \end{bmatrix}$
Table 5 Examples of the **PositionVector** command

The real benefit of representing curves and surfaces via the **PositionVector** command is its compatibility with the **PlotPositionVector** command by means of which various vector fields can be superimposed on the curves and surfaces this command draws. (See the section **Visualizations**, below.)

Vector Fields

$with(Student:\text{-}VectorCalculus):$
$VectorField(\langle x+y, x-y\rangle) = (x+y)\bar{e}_x + (x-y)\bar{e}_y$
$VectorField(\langle u, v, w\rangle) = (u)\bar{e}_x + (v)\bar{e}_y + (w)\bar{e}_z$
$VectorField(\langle r+\theta, r-\theta\rangle, \text{polar}) = (r+\theta)\bar{e}_r + (r-\theta)\bar{e}_\theta$
$VectorField(\langle u, v, w\rangle, cylindrical) = (u)\bar{e}_r + (v)\bar{e}_\theta + (w)\bar{e}_z$
$VectorField(\langle u, v, w\rangle, spherical) = (u)\bar{e}_r + (v)\bar{e}_\phi + (w)\bar{e}_\theta$
Table 6 Vector fields with explicit display of "moving basis vectors"

- When a **VectorField** is evaluated at a point, a **RootedVector** results.
- The **evalVF** command is used to evaluate a **VectorField** at a point.
- If an ordinary evaluation (or substitution) were made, only the components of the vector would be pointwise evaluated, and the basis vectors would therefore be incorrect, and a rooted vector would not result.

With the inclusion of the option "output = plot", the **VectorField** command returns a graph of the arrows of the field.

Commands Applicable to Basic Vector Objects

Table 7 lists other commands relevant for the use of the basic vector objects in Table 3.

Command	Comments
BasisFormat	• Changes the display of free vectors and vector fields. • The default is to display basis vectors, either unbarred or barred. • Executing the command with the argument "false" switches the display to column-vector format.
About	• Applied to any of the four basic vector objects, this command returns relevant information for that object.
evalVF	• As noted after Table 6, this command is used to evaluate a vector field at a point, and results in a rooted vector.
MapToBasis	• Change coordinates in a free vector or in a vector field.

	• Does not apply to scalar fields.
ConvertVector	• Converts Cartesian free vector, rooted vector, or position vector to a free, rooted, or position vector.

Table 7 Commands pertinent to use of the basic vector objects in Table 3

Differentiation

Basic Differentiation Commands

Table 8 lists the commands in the Student *VectorCalculus* package that in some way involve differentiation.

Command	Comments
diff	• The top-level diff command is modified so that it automatically maps onto components of vectors.
Gradient	• Computes ∇f, the gradient of the scalar f, returning a vector field.
Divergence	• Computes $\nabla \cdot \mathbf{F}$, the divergence of the vector field **F**.
Curl	• Computes $\nabla \times \mathbf{F}$, the curl of the vector field **F**, returning a vector field.
Laplacian	• Computes $\nabla^2 f$, the Laplacian of the scalar f.
DirectionalDiff	• Computes the directional derivative of the scalar f.
TangentLine	• Returns a representation of the line tangent to a curve.
TangentPlane	• Returns a representation of the plane tangent to a surface.

Table 8 Differentiation commands in the Student *VectorCalculus* package

Frenet-Serret Formalism

Commands relevant to the Frenet-Serret formalism are listed in Table 9.

Command	Comments
Curvature	• Computes κ, the curvature of a curve **R**.
RadiusOfCurvature	• Computes $1/\kappa$, the reciprocal of the curvature of a curve **R**. • With "output = plot", returns a graph of **R** and the circle of curvature.
Torsion	• Computes τ, the torsion of a curve **R**.
TangentVector	• Computes $\mathbf{R}'$, a vector tangent to a curve **R**. • With the option *normalized*, returns **T**, the *unit* tangent vector. • With "output = plot", returns a graph of **R** and representative (unit)

	tangent vectors. • With "output = animation", returns a graph of **R** and a representative **T** traversing **R**.
PrincipalNormal	• For a curve **R**, computes a vector along **N**, the principal normal vector. • With the option *normalized*, returns **N**, the *unit* principal normal. • With "output = plot", returns a graph of **R** and representative (unit) principal-normal vectors. • With "output = animation", returns a graph of **R** and a representative **N** traversing **R**.
Binormal	• For a curve **R**, computes a vector along **B**, the binormal vector. • With the option *normalized*, returns **B**, the *unit* binormal. • With "output = plot", returns a graph of **R** and representative (unit) binormal vectors. • With "output = animation", returns a graph of **R** and a representative **B** traversing **R**.
TNBFrame	• Returns a sequence of **T**, **N**, and **B**, the (unit) tangent, principal normal, and binormal vectors for a curve **R**. • With "output = plot", returns a graph of **R** and representative triples of **T**, **N** and **B** vectors. • With "output = animation", returns a graph of **R** and a representative triple of **T**, **N** and **B** vectors traversing **R**.

Table 9 Commands relevant to the Frenet-Serret formalism

The [Space Curve] tutor implements the graphical aspects of the commands in Table 9. The computational aspects are captured in the Context Panel when the Student *VectorCalculus* package is installed.

Integration

Table 10 lists the commands in the Student *VectorCalculus* package that in some way involve integration.

Command	Comments
int	• The top-level **int** command is modified to recognize the following pre-defined domains: *Circle*, *Ellipse*, *Parallelepiped*, *Rectangle*, *Region*, *Sector*, *Sphere*, *Tetrahedron*, and *Triangle*.
PathInt	• Computes $\int_C f\,ds$, the line integral of the scalar f, taken with respect to arc length s along the curve C. • The following pre-defined paths of integration of recognized: *Arc*, *Circle*, *Ellipse*, *Line*, *LineSegments*, and *Path*. • Access through the Context Panel.
	• Computes along the curve C, $\int_C \mathbf{F}\cdot\mathbf{dr} = \int_C \mathbf{F}\cdot\mathbf{T}\,ds$, the line integral of the tangential component of the vector field **F**, where **T** is the unit tangent vector along

LineInt	C, and ds is the element of arc length along C. • The following pre-defined paths of integration are recognized: *Arc*, *Circle*, *Circle3D*, *Ellipse*, *Line*, *LineSegments*, and *Path*. • A graph of the vector field and the integration path is a possible return for the following pre-defined paths of integration: *Circle*, *Line*, *LineSegments*, and *Path*. • Access through the Context Panel.
SurfaceInt	• Computes $\iint_S f\,d\sigma$, the surface integral of the scalar f taken over the surface S, with $d\sigma$ being the element of surface area for S. • The following pre-defined surfaces are recognized: *Box*, *Sphere*, and *Surface*. • Surfaces specified by the *Surface* option can be defined over the following planar regions: *Circle*, *Ellipse*, *Rectangle*, *Region*, *Sector*, and *Triangle*.
Flux	• In the plane, computes $\int_C \mathbf{F} \cdot \mathbf{N}\,ds$, the flux of the vector field **F** through the plane curve C, where **N** is a unit normal field along C, and ds is the element of arc length along C. • The following pre-defined curves are recognized: *Arc*, *Circle*, *Ellipse*, *Line*, *LineSegments*, and *Path*. A graph of the vector field and the curve is a possible return for the *Circle*, *Line*, *LineSegments*, and *Path* options. One or more representative normal vectors are drawn. • In space, computes $\iint_S \mathbf{F} \cdot \mathbf{N}\,d\sigma$, the flux of the vector field **F** through the surface S, where **N** is a unit normal field on S, and $d\sigma$ is the element of surface area for S. • The following pre-defined surfaces are recognized: *Box*, *Sphere*, and *Surface*. • Surfaces specified by the *Surface* option can be defined over the following planar regions: *Circle*, *Ellipse*, *Rectangle*, *Region*, *Sector*, and *Triangle*. • For the *Box*, *Sphere*, and *Surface* options, a graph of the vector field and surface of integration is a possible return. One or more representative normal vectors are drawn. However, no graphs are drawn for surfaces specified over any of the pre-defined planar regions. • Implemented in a set of Task Templates.
ScalarPotential	Given a vector field **F**, returns (if it exists) the scalar f whose gradient ∇f equals **F**.
VectorPotential	Given a vector field **F**, returns (if it exists) a vector **A** whose curl $\nabla \times \mathbf{A}$ equals **F**.

Table 10 Student *VectorCalculus* commands that involve integration

Visualization

Table 11 lists the commands in the Student *VectorCalculus* package that do, or can, return graphs.

Command	Comment
PlotVector	• Graphs one or more free or rooted vectors.
PlotPositionVector	• Graphs the curve or surface represented by a **PositionVector**, and has options for adding vectors from various vector fields defined along the curve or surface.

VectorField	• Creates a vector-field object, or graphs its arrows. • This graphical functionality can also be accessed through the [Vector Field] tutor.
FlowLine	• Graph arrows of a vector field, and one or more of its flow lines.
SpaceCurve	• Provides a unified interface for graphing planar and spatial curves. • The [Space Curve] tutor provides interactive access to this functionality.
LineInt	• Forms and evaluates line integrals of the tangential component of a vector field, and can also return a graph.
Flux	• Forms and evaluates flux integrals, and can also return a graph of the vector field and the curve or surface.
RadiusOfCurvature TangentVector PrincipalNormal Binormal TNBFrame	• These commands for implementing the Frenet-Serret formalism, can return graphs and animations.

Table 11 Student *VectorCalculus* commands that do, or can, return graphs

Example

Calculate the flux of $\mathbf{F} = (x + y^2)\,\mathbf{i} + (x^2 - y)\,\mathbf{j} + x\,y\,z\,\mathbf{k}$ through that part of the surface $z = 10 - x^2 - y^2$ that sits over the disk whose center is at $(1, 2)$ and whose radius is 1.

Solution via Task Template

Tools>Tasks>Browse:
Calculus - Vector>Integration>Flux>3-D>Through a Surface Defined over a Disk

Flux through a Surface Defined over a Disk

For the Vector Field:

Clear All and Re...

Select Coordinate...

Flux I...

Simplify

Value

Simplify

Float

Solution via Explicit Commands

Initialize

> $restart$

> $with(Student:-VectorCalculus):$

> $BasisFormat(false):$

> $\mathbf{F} := VectorField\left(\left[x+y^2, x^2-y, x\,y\,z\right]\right)$

Obtain the Flux

> $Flux\left(\mathbf{F}, Surface\left(\langle x, y, 10-x^2-y^2\rangle, [x, y] = Circle(\langle 1, 2\rangle, 1, [r, \theta])\right), output = integral\right)$

> $Flux\left(\mathbf{F}, Surface\left(\langle x, y, 10 - x^2 - y^2\rangle, [x, y] = Circle(\langle 1, 2\rangle, 1, [r, \theta])\right)\right)$

From First Principles

> $Z := 10 - x^2 - y^2$

> $X := 1 + r\cos(\theta);$
> $Y := 2 + r\sin(\theta)$

> $d\sigma := \sqrt{1 + \left(\frac{\partial}{\partial x} Z\right)^2 + \left(\frac{\partial}{\partial y} Z\right)^2}$

> $\mathbf{N} := Normalize(Gradient(z - Z))$

> $q := simplify(eval(\mathbf{F} \cdot \mathbf{N} \cdot d\sigma, z = Z))$

> $Q := eval(q, [x = X, y = Y])$

> $\int_0^1 \int_0^{2\pi} Q \cdot r \, d\theta \, dr$