REVA UNIVERSITY

Bengaluru, India

School of Applied Sciences

Department of Mathematics

LAB MANUAL

VI Semester B.Sc. Mathematics Lab V

Contents

1. Evaluation of the line integral with constant limits

If the equation of the curve C in the xy plane is $y = \phi(x)$, then

(1) $\displaystyle\int_c f(x,y)dx = \int_{x_1}^{x_2} f(x,\phi(x))dx$

(2) $\displaystyle\int_c f(x,y)dy = \int_{x_1}^{x_2} f(x,\phi(x))\phi'(x)dx$

(3) $\displaystyle\int_c f(x,y)ds = \int_{x_1}^{x_2} f(x,\phi(x))\sqrt{1+(f'(x))^2}dx.$

1. Evaluate $\displaystyle\int_c ydx - xdy$ along the curve $y = x^2$ from $(0,0)$ to $(1,1)$.

1.1. Program:

```
kill(all)$
load("vect")$
y:x^2$
F:[y,-x]$
dx:diff(x,x)$
dy:diff(y,x)$
dr:[dx,dy]$
I:(F.dr)$
integrate(I,x,0,1);
```

2. Integrate $(x+y)dx + (y-x)dy$ along the curve $x = y^2$ from $(1,1)$ to $(4,2)$.

1.2. Program:

```
kill(all)$
load("vect")$
x:y^2$
F:[x+y,y-x]$
dx:diff(x,y)$
dy:diff(y,y)$
dr:[dx,dy]$
I:(F.dr)$
integrate(I,y,1,2);
```

1.3. Exercise.

(1) Evaluate $\displaystyle\int_c (x^2-y^2)dx + xydy$ along the curve $y = x^3$ from $(0,0)$ to $(2,8)$

(2) Evaluate $\int_c x^2dx + xydy$ along the curve $y = x$ from $(0,0)$ to $(1,1)$.

(3) Evaluate $\int_c x^2dx + xydy$ along the curve $y = \sqrt{x}$ from $(0,0)$ to $(1,1)$.

2. Evaluation of the line integral with variable limits

If the equation of the curve C is in the parametric form $x = \phi(t), y = \psi(t)$, then

(1) $\int_c f(x,y)dx = \int_{t_1}^{t_2} f(\phi(t),\psi(t))\phi'(t)dt$

(2) $\int_c f(x,y)dy = \int_{t_1}^{t_2} f(\phi(t),\psi(t))\psi'(t)dt$

(3) $\int_c f(x,y)ds = \int_{t_1}^{t_2} f(\phi(t),\psi(t))\sqrt{\left(\frac{dx}{dt}\right)^2 + \left(\frac{dy}{dt}\right)^2}dt.$

1. Integrate $(x+y)dx + (y-x)dy$ along $x = 3t+1$ and $y = t+1$ where $0 \le t \le 1$

2.1. Program:

```
kill(all)$
load("vect")$
x:3*t+1$
y:t+1$
F:[x+y,y-x]$
dx:diff(x,t)$
dy:diff(y,t)$
dt:[dx,dy]$
I:(F.dt)$
integrate(I,t,0,1);
```

2. Integrate $2x^2dx + (2xz-y)dy + zdz$ along $x = 2t, y = t$ and $z = 3t$ where $0 \le t \le 1$

2.2. Program:

```
kill(all)$
load("vect")$
x:2*t$
y:t$
z:3*t$
F:[2*x^2,2*x*z-y,z]$
dx:diff(x,t)$
dy:diff(y,t)$
```

```
dz:diff(z,t)$
dt:[dx,dy,dz]$
I:(F.dt)$
integrate(I,t,0,1);
```

3. Integrate $ydx + xdy - z^2dz$ along $x = \sin t, y = \cos t$ and $z = t^2$ where $0 \leq t \leq 1$

2.3. **Program:**

```
kill(all)$
load("vect")$
x:sin(t)$
y:cos(t)$
z:t^2$
F:[y,x,-z^2]$
dx:diff(x,t)$
dy:diff(y,t)$
dz:diff(z,t)$
dt:[dx,dy,dz]$
I:(F.dt)$
integrate(I,t,0,1);
```

2.4. **Exercise.**

(1) Evaluate $\int\limits_c xydx + yzdy + zxdz$ along $x = t, y = t^2$ and $z = t^3$ where $-1 \leq t \leq 1$.

(2) Evaluate $\int\limits_c (3x^2 + 6y)dx - 14yzdy + 20xz^2dz$ along $x = t, y = t^2$ and $z = t^3$ where $0 \leq t \leq 1$.

3. Evaluation of the double integral with constant limits

Double integral can be evaluated byt expressing it in terms of two single integrals called iterated or repeated integral. Double integral over region R may be evaluated by two successive integrations.

If R is described as $f_1(x) \leq y \leq f_2(x)$, i.e. $y_1 \leq y \leq y_2$ and $a \leq x \leq b$, then

(1)
$$\iint\limits_R f(x,y)dxdy = \int\limits_a^b \left[\int\limits_{y_1}^{y_2} f(x,y)dy \right] dx$$

$f(x,y)$ is first integrated with respect to y treating x as constant between the limits y_1 and y_2 and then the resulting function is integrated with respect to x between limits a and b.

(2)

$$\iint\limits_R f(x,y)dxdy = \int\limits_c^d \left[\int\limits_{x_1}^{x_2} f(x,y)dx \right] dy$$

$f(x,y)$ is first integrated with respect to x treating y as constant between the limits x_1 and x_2 and then the resulting function is integrated with respect to y between limits c and d.

(3) If the region R is a rectangle bounded by the lines $x = a, x = b, y = c, y = d$, then

$$\iint\limits_R f(x,y)dxdy = \int\limits_a^b \int\limits_c^d f(x,y)dydx = \int\limits_c^d \int\limits_a^b f(x,y)dxdy$$

The order of integration in immaterial for constant limits. That is for constant limits, it does not matter whether we first integrate with respect to x and then with respect to y of vice versa.

3.1. **Program:** Evaluate $\int\limits_0^a \int\limits_0^b (x^2 + y^2)dxdy$

```
'integrate('integrate(x^2+y^2,x,0,b),y,0,a)=
integrate(integrate(x^2+y^2,x,0,b),y,0,a);
```

3.2. **Program:** Evaluate $\int\limits_0^3 \int\limits_1^2 xy(1 + x + y)dxdy$

```
'integrate('integrate(x*y*(1+x+y),x,1,2),y,0,3)=
integrate(integrate(x*y*(1+x+y),x,1,2),y,0,3);
```

3.3. **Program:** Evaluate $\int\limits_2^3 \int\limits_0^1 xye^x dxdy$

```
'integrate('integrate(x*y*%e^x,x,0,1),y,2,3)=
integrate(integrate(x*y*%e^x,x,0,1),y,2,3);
```

3.4. **Program:** Evaluate $\int\limits_{-3}^{-2} \int\limits_{-5}^{-4} (3w - 2z^3)dwdz$

```
'integrate('integrate(3*w-2*z^3,w,-5,-4),z,-3,-2)=
integrate(integrate(3*w-2*z^3,w,-5,-4),z,-3,-2);
```

3.5. **Exercise.**

(1) Evaluate $\int\limits_0^{\frac{\pi}{2}} \int\limits_0^{\frac{\pi}{2}} \sin(x + y)dxdy$

(2) Evaluate $\int\limits_{1}^{2}\int\limits_{3}^{4}(xy+e^y)dydx$

4. Evaluation of the double integral with variable limits

4.1. Program: Evaluate $\int\limits_{1}^{2}\int\limits_{0}^{3y} ydxdy$

```
'integrate('integrate(y,x,0,3*y),y,1,2)=
integrate(integrate(y,x,0,3*y),y,1,2);
```

4.2. Program: Evaluate $\int\limits_{0}^{1}\int\limits_{0}^{1-x} xydydx$

```
'integrate('integrate(x*y,y,0,1-x),x,0,1)=
integrate(integrate(x*y,y,0,1-x),x,0,1);
```

4.3. Program: Evaluate $\int\limits_{0}^{a}\int\limits_{0}^{\frac{bx}{a}} xdydx$

```
'integrate('integrate(x,y,0,b*x/a),x,0,a)=
integrate(integrate(x,y,0,b*x/a),x,0,a);
```

4.4. Program: Evaluate $\int\limits_{0}^{2a}\int\limits_{0}^{\sqrt{2ax-x^2}}(x^2+y^2)dydx$

```
'integrate('integrate(x^2+y^2,y,0,sqrt(2*a*x-x^2)),x,0,2*a)=
integrate(integrate(x^2+y^2,y,0,sqrt(2*a*x-x^2)),x,0,2*a);
```

4.5. Exercise.

(1) Evaluate $\int\limits_{0}^{2}\int\limits_{x^2}^{2x}(2x+3y)dydx$

(2) Evaluate $\int\limits_{0}^{1}\int\limits_{0}^{1-x}(x^2+y^2)dydx$

5. Evaluation of the triple integral with constant limits

5.1. **Program:** Evaluate $\int_0^1 \int_1^2 \int_1^2 x^2 yz dz dy dx$

```
'integrate('integrate('integrate(x^2*y*z,z,1,2),y,1,2),x,0,1)=
integrate(integrate(integrate(x^2*y*z,z,1,2),y,1,2),x,0,1);
```

5.2. **Program:** Evaluate $\int_1^2 \int_0^1 \int_{-1}^1 (x^2+y^2+z^2) dz dy dx$

```
'integrate('integrate('integrate(x^2+y^2+z^2,z,-1,1),y,0,1),x,1,2)=
integrate(integrate(integrate(x^2+y^2+z^2,z,-1,1),y,0,1),x,1,2);
```

5.3. **Program:** Evaluate $\int_1^2 \int_1^2 \int_1^2 (\frac{x}{y}+\frac{y}{z}+\frac{z}{x}) dz dy dx$

```
'integrate('integrate('integrate(x/y+y/z+z/x,z,1,2),y,1,2),x,1,2)=
integrate(integrate(integrate(x/y+y/z+z/x,z,1,2),y,1,2),x,1,2);
```

5.4. **Program:** Evaluate $\int_0^2 \int_1^3 \int_1^2 xy^2 z dx dy dz$

```
'integrate('integrate('integrate(x*y^2*z,x,1,2),y,1,3),z,0,2)=
integrate(integrate(integrate(x*y^2*z,x,1,2),y,1,3),z,0,2);
```

6. Evaluation of the triple integral with variable limits

6.1. **Program.** Evaluate $\int_0^1 \int_0^3 \int_{x^2}^{\sqrt{x}} (x+y+z) dy dz dx$

```
'integrate('integrate('integrate(x+y+z,y,x^2,sqrt(x)),z,0,3),x,0,1)=
integrate(integrate(integrate(x+y+z,y,x^2,sqrt(x)),z,0,3),x,0,1);
```

6.2. **Program.** Evaluate $\int_{-1}^1 \int_0^z \int_{x-z}^{x+z} (x+y+z) dy dx dz$

```
'integrate('integrate('integrate(x+y+z,y,x-z,x+z),x,0,z),z,-1,1)=
integrate(integrate(integrate(x+y+z,y,x-z,x+z),x,0,z),z,-1,1);
```

6.3. **Program.** Evaluate $\int_0^1 \int_0^{1-z} \int_0^{y^2} x dx dy dz$

```
'integrate('integrate('integrate(x,x,0,y^2),y,0,1-z),z,0,1)=
integrate(integrate(integrate(x,x,0,y^2),y,0,1-z),z,0,1);
```

6.4. **Program.** Evaluate $\int\limits_0^1 \int\limits_0^{\sqrt{1-x^2}} \int\limits_0^{\sqrt{1-x^2-y^2}} xyzdzdydx$

```
'integrate('integrate('integrate(x*y*z,z,0,sqrt(1-x^2-y^2)),y,0,
sqrt(1-x^2)),x,0,1)=integrate(integrate(integrate(x*y*z,z,0,
sqrt(1-x^2-y^2)),y,0,sqrt(1-x^2)),x,0,1);
```

7. Maxima programs for area and volume

1. Find the area of the circle with radius a units

7.1. **Program.**

```
integrate(integrate(1,y,-\sqrt(a^2-x^2),\sqrt(a^2-x^2)),x,-a,a);
```

2. Integrate over the region D $\int\limits_D xydxdy$ where D is the region bounded by the curve $y = \sin x$ and the segment $0 \leq x \leq \pi$

7.2. **Program.**

```
ratsimp(integrate(integrate(x*y,y,0,sin(x)),x,0,%pi));
```

3. Integrate over the region D $\int\limits_D xydxdy$ where D is bounded by $x^2 + y^2 = a^2$ and $x \geq 0, y \geq 0$

7.3. **Program.**

```
'integrate('integrate(x*y,x,0,sqrt(a^2-y^2)),y,0,a)=
integrate(integrate(x*y,x,0,sqrt(a^2-y^2)),y,0,a);
```

4. Find the volume of the sphere with radius a units

7.4. **Program.**

```
8*'integrate('integrate('integrate(1,z,0,sqrt(a^2-x^2-y^2)),y,0,
sqrt(a^2-x^2)),x,0,a)=8*integrate(integrate(integrate(1,z,0,
sqrt(a^2-x^2-y^2)),y,0,sqrt(a^2-x^2)),x,0,a);
```

5. Find the volume common to the cylinder $x^2 + y^2 = a^2$ and $x^2 + z^2 = a^2$

7.5. **Program.**

```
'integrate('integrate('integrate(1,z,-sqrt(a^2-x^2),sqrt(a^2-x^2)),
y,-sqrt(a^2-x^2),sqrt(a^2-x^2)),x,-a,a)=integrate(integrate(integrate(1,
z,-sqrt(a^2-x^2),sqrt(a^2-x^2)),y,-sqrt(a^2-x^2),sqrt(a^2-x^2)),x,-a,a);
```

8. Express a vector as a linear combination of given vectors

8.1. Program.

Express the vector $(1, 7, -4)$ as a linear combination of the vectors $(1, -3, 2)$ and $(2, -1, 1)$.

```
kill(all)$
v1:[1,-3,2]$
v2:[2,-1,1]$
w:[1,7,-4]$
n:length(v1)$
print("To find c1,c2 such that",c1,"*",transpose(v1),"+",
c2,"*",transpose(v2),"=",transpose(w))$
for i:1 thru n do(
expr[i]:v1[i]*c1+v2[i]*c2)$
soln:solve([expr[1]=w[1],expr[2]=w[2],expr[3]=w[3]],[c1,c2]);
c1:ev(c1,soln)$
c2:ev(c2,soln)$
W:c1*v1+c2*v2$
if w=W then
print("The linear combination of vector",w,"=",c1,v1,"+",c2,v2)
else
print("v is not a linear combination of v1,v2")$
```

8.2. Exercise.

(1) Express $(3, 7, -4)$ as a linear combination of the vectors $(1, 2, 3)$, $(2, 3, 7)$ and $(3, 5, 6)$.

(2) Express then vector $(2, -5, 4)$ as a linear combination of the vectors $(1, -3, 2)$ and $(2, -1, 1)$.

8.3. Program.

Program to express $(3, -1, 1, -2)$ as a linear combination of the vectors $(1, 1, 0, -1)$, $(1, 1, -1, 0)$ and $(1, -1, 0, 0)$.

```
kill(all)$
v1:[1,1,0,-1]$
v2:[1,1,-1,0]$
v3:[1,-1,0,0]$
w:[3,-1,1,-2]$
n:length(v1)$
print("To find c1,c2 such that",c1,"*",transpose(v1),"+",
c2,"*",transpose(v2),"+",c3,"*",transpose(v3),"=",transpose(w))$
for i:1 thru n do(
```

```
expr[i]:v1[i]*c1+v2[i]*c2+v3[i]*c3)$
soln:solve([expr[1]=w[1],expr[2]=w[2],expr[3]=w[3],expr[4]=w[4]],[c1,c2,c3]);
c1:ev(c1,soln)$
c2:ev(c2,soln)$
c3:ev(c3,soln)$
W:c1*v1+c2*v2+c3*v3$
if w=W then
print("The linear combination of vector",w,"=",c1,v1,"+",c2,v2,"+",c3,v3)
else
print("v is not a linear combination of v1,v2,v3")$
```

9. Linear dependence and independence of vectors

Let V be a vector space over a field F. A finite set $S = \{v_1, v_2, \ldots, v_n\}$ of vectors in V is linearly independent if the linear combination of distinct vectors in S that produces the zero vector is a trivial linear combination, i.e., if $c_1v_1 + c_2v_2 + \cdots + c_kv_k = 0$, then $c_1 = c_2 = \cdots = c_k = 0$.

The set $S = \{v_1, v_2, \ldots, v_n\}$ of vectors of the vector space V over a field is dependent if and only if determinant of matrix of vectors is zero.

9.1. Program.

Show that the vectors $(1, 1, -1)$, $(2, -3, 5)$ and $(-2, 1, 4)$ of $\mathbb{R}^3$ are linearly independent.

```
kill(all)$
v1:[1,1,-1]$
v2:[2,-3,5]$
v3:[-2,1,4]$
for i:1 thru 3 do(
expr[i]:c1*v1[i]+c2*v2[i]+c3*v3[i])$
soln:solve([expr[1]=0,expr[2]=0,expr[3]=0],[c1,c2,c3]);
c1:ev(c1,soln)$
c2:ev(c2,soln)$
c3:ev(c3,soln)$
if c1=0 and c2=0 and c3=0 then(
print("The vectors ",v1,",",v2,",",v3,"are linearly independent"))
else (
print("The vectors ",v1,",",v2,",",v3,"are linearly dependent"))$
```

9.2. Program.

Show that the vectors $(1, 1, 2)$, $(-3, 1, 0)$ $(1, -1, 1)$ and $(1, 2, -3)$ of $\mathbb{R}^3$ are linearly dependent.

```
kill(all)$
v1:[1,1,2]$
v2:[-3,1,0]$
v3:[1,-1,1]$
v4:[1,2,-3]$
for i:1 thru 3 do(
expr[i]:c1*v1[i] +c2*v2[i] +c3*v3[i]+c4*v4[i] )$
soln:solve([expr[1]=0,expr[2]=0,expr[3]=0],[c1,c2,c3,c4]);
c1:ev(c1,soln)$
c2:ev(c2,soln)$
c3:ev(c3,soln)$
c4:ev(c4,soln)$
if c1=0 and c2=0 and c3=0 and c4=0 then(
print("The vectors ",v1,",",v2,",",v3,",",v4,"are linearly independent"))
else (
print("The vectors ",v1,",",v2,",",v3,",",v4,"are linearly dependent"))$
```

9.3. **Exercise.**

(1) Examine whether the set of vectors $(1,2,1,2)$, $(3,2,3,2)$, $(-1,-3,0,4)$ and $(0,4,-1,-3)$ are linearly dependent.
(2) Examine whether vectors $(1,\sqrt{2},1)$ $(1,-\sqrt{2},1)$ and $(-1,0,1)$ are linearly independent.

10. **Basis and Dimension**

- Let $V(F)$ be a vector space of dimension n, then any set of n linearly independent vector is a basis.
- Let A be any $m \times n$ matrix row equivalent to a row reduced echelon matrix E, then the non-zero rows of E form a basis of the subspace spanned by the rows of A. The number of non-zero rows of E is the dimension of the subspace spanned by the rows of the matrix A.

10.1. **Program.**
Show that the vectors $(1,1,0)$, $(0,1,0)$ and $(0,1,1)$ for a basis of $\mathbb{R}^3$.

```
kill(all)$
A:matrix([1,1,0],[0,1,0],[0,1,1]);
D:determinant(A);
if (D=0) then
disp("Given vectors are Linearly Dependent and hence not a basis.")
else
```

```
disp("Given vectors are Linearly independent and hence form a basis.");
```

10.2. **Program.**
Show that the set of vectors $\{(1,2,3),(3,1,0),(-2,1,3)\}$ is not a basis of $\mathbb{R}^3$. Determine the dimension and basis of the subspace spanned by the given vectors.

```
kill(all)$
A:matrix([1,2,3],[3,1,0],[-2,1,3]);
D:determinant(A);
if (D=0) then
disp("The given vectors are Linearly Dependent and hence not a basis")
else
disp("The given vectors are Linearly Independent and hence a basis")$
e:echelon(A);

/*Execute the above and display the non-zero rows of 'e' as follows:*/

print("The vectors ",e[1],",",e[2],"are the basis for the subspace")$
r:rank(A)$
print("Dimension of the subspace is =",r)$
```

10.3. **Exercise.**
Find the basis and dimension of the subspace spanned by the vectors $(2,-3,1)$, $(3,0,1)$, $(0,2,1)$, $(1,1,1)$ of $\mathbb{R}^3$.

11. Verifying whether a given transformation is linear

Let U, V be two vector spaces over the same field F. The mapping $T : U \to V$ is linear if it satisfies the following

- $T(x+y) = T(x) + T(y); \qquad \forall x, y \in U$
- $T(\alpha x) = \alpha T(x); \qquad \forall x \in U, \alpha \in F$

11.1. **Program.**
Verify whether $T : V_2(\mathbb{R} \to V_2(\mathbb{R})$ defined by $T(x,y) = (x+y, y)$ is a linear transformation.

```
kill(all)$
T(x):=[x[1]+x[2], x[2]];
u[1]:[a,b]$
u[2]:[c,d]$
a1:T(u[1])+T(u[2]);
a2:radcan(T(u[1]+u[2]));
p1:factor(T(k*u[1]));
p2:factor(k*T(u[1]));
```

```
if (a1=a2 and p1=p2) then
print("The given mapping is a linear transformation")
else
print("The given mapping is not a linear transformation")$
```

11.2. **Program.**
Verify whether $T : V_3(\mathbb{R} \to V_3(\mathbb{R})$ defined by $T(x, y, z) = (x^2 + xy, xy, yz)$ is a linear transformation.

```
kill(all)$
T(x):=[x[1]^2+x[1]*x[2],x[1]*x[2],x[2]*x[3]];
u[1]:[a,b,c]$
u[2]:[d,e,f]$
a1:T(u[1])+T(u[2]);
a2:radcan(T(u[1]+u[2]));
p1:factor(T(k*u[1]));
p2:factor(k*T(u[1]));
if (a1=a2 and p1=p2) then
print("The given mapping is a linear transformation")
else
print("The given mapping is not a linear transformation")$
```

11.3. **Exercise.**

(1) Verify whether $T : \mathbb{R}^3 \to \mathbb{R}^2$ defined by $T(x, y, z) = (x + 2y - 3z, 4x - 5y + 6z)$ a linear transformation.
(2) Verify whether $T : V_2(\mathbb{R}) \to V_3(\mathbb{R})$ defined by $T(x, y) = (x + 3, 2y, x + y)$ a linear transformation.

12. To find matrix of linear transformation

Let $T : U \to V$ be a linear transformation where U and V are finite dimensional vector spaces of dimensions m and n respectively. Let $B_1 = \{u_1, u_2, \dots, u_m\}$ and $B_2 = \{v_1, v_2, \dots, v_n\}$ be ordered bases of U and V respectively such that

$$T(u_1) = a_{11}v_1 + a_{21}v_2 + \cdots + a_{n1}v_n$$

$$T(u_2) = a_{12}v_1 + a_{22}v_2 + \cdots + a_{n2}v_n$$

$$\vdots$$

$$T(u_m) = a_{1m}v_1 + a_{2m}v_2 + \cdots + a_{nm}v_n$$

The matrix formed by taking the transpose of the coefficient matrix of the above system of equations is called the matrix of the linear transformation T relative to the bases B_1 and B_2.

12.1. **Program.**
Find the matrix of linear transformation $T : \mathbb{R}^2 \to \mathbb{R}^2$ defined by $T(x, y) = (2x+3y, 4x-5y)$ with respect to the standard basis.

```
kill(all)$
T(x):=[2*x[1]+3*x[2],4*x[1]-5*x[2]];
u[1]:[1,0]$
u[2]:[0,1]$
v[1]:[1,0]$
v[2]:[0,1]$
for i:1 thru 2 do(
eq[i]: v[1][i] * x+v[2][i] * y)$
for k:1 thru 2 do(
soln:solve([eq[1]=T(u[k])[1],eq[2]=T(u[k])[2]],[x,y]),
a[k]:ev(x,soln),
b[k]:ev(y,soln))$
print("The matrix of the linear transformations is")$
M:matrix([a[1],a[2]],[b[1],b[2]]);
```

12.2. **Program.**
Given $T(x, y, z) = (x - y + z, 2x + 3y - \frac{1}{2}z, x + y - 2z)$, find the matrix of T relative to the bases $B_1 = \{(-1, 1, 0), (5, -1, 2), (1, 2, 1)\}$ and $B_2 = \{(1, 1, 0), (0, 0, 1), (1, 5, 2)\}$.

```
kill(all)$
T(x):=[x[1]-x[2]+x[3], 2*x[1]+3*x[2]-(1/2)*x[3], x[1]+x[2]-2*x[3]];
u[1]:[-1,1,0]$
u[2]:[5,-1,2]$
u[3]:[1,2,1]$
v[1]:[1,1,0]$
v[2]:[0,0,1]$
v[3]:[1,5,2]$
for i:1 thru 3 do(
eq[i]: v[1][i] * x+v[2][i] * y+v[3][i] * z)$
for k:1 thru 3 do(
soln:solve([eq[1]=T(u[k])[1],eq[2]=T(u[k])[2],eq[3]=T(u[k])[3]],[x,y,z]),
a[k]:ev(x,soln),
b[k]:ev(y,soln),
c[k]:ev(z,soln))$
print("The matrix of the linear transformations is")$
M:matrix([a[1],a[2],a[3]],[b[1],b[2],b[3]],[c[1],c[2],c[3]]);
```

12.3. **Exercise.**

(1) Find the matrix of linear transformation $T : V_2(\mathbb{R}) \to V_3(\mathbb{R})$ defined by $T(x,y) = (-x+2y, y, -3x+3y)$ relative to the basis $B_1 = \{(1,2),(-2,1)\}$ and $B_2\{(-1,0,2),(1,2,3),(1,-1,-1)\}$.

(2) Find the matrix of linear transformation $T : V_3(\mathbb{R}) \to V_2(\mathbb{R})$ defined by $T(x,y,z) = (x+y, y+z)$ relative to the basis $B_1 = \{(1,1,0),(1,0,1),(1,1,-1)\}$ and $B_2\{(2,-3),(1,4)\}$.

13. To find linear transformation of matrix

Given a matrix $A = (a_{ij})_{n\times m}$, we can associate a linear transformation $T : U \to V$ where U and V are finite dimensional vector spaces of dimensions m and n respectively. Let $B_1 = \{u_1, u_2, \ldots, u_m\}$ and $B_2 = \{v_1, v_2, \ldots, v_n\}$ be ordered bases of U and V respectively. Define the linear transformation $T : U \to V$ by

$$T(u_1) = a_{11}v_1 + a_{21}v_2 + \cdots + a_{n1}v_n$$

$$T(u_2) = a_{12}v_1 + a_{22}v_2 + \cdots + a_{n2}v_n$$

$$\vdots$$

$$T(u_m) = a_{1m}v_1 + a_{2m}v_2 + \cdots + a_{nm}v_n$$

This T can be extended linearly to the entire space V.

13.1. **Program.**

Given the matrix $A = \begin{pmatrix} 2 & 3 \\ 4 & -5 \end{pmatrix}$ of the linear transformation T. Find T with respect to the bases $B_1 = \{(1,-1),(1,1)\}$ and $B_2 = \{(1,0),(0,1)\}$.

```
kill(all)$
A:matrix([2,3],[4,-5]);
u[1]:[1,-1]$
u[2]:[1,1]$
v[1]:[1,0]$
v[2]:[0,1]$
for i:1 thru 2 do(
T[i]:A[1,i] * v[1]+A[2,i] * v[2])$
for j:1 thru 2 do(
eq[j]:u[1][j] * p+u[2][j] * q)$
soln:solve([eq[1]=x,eq[2]=y],[p,q])$
a:ev(p,soln)$
b:ev(q,soln)$
print("The linear transformation is")$
```

```
T:radcan(T[1] * a+T[2] * b)$
print("T(x,y)=",T)$
```

13.2. **Program.**

Given the matrix $A = \begin{pmatrix} -1 & 0 \\ 2 & 0 \\ 1 & 3 \end{pmatrix}$ of the linear transformation T. Find T with respect to the bases $B_1 = \{(1,0),(2,-1)\}$ and $B_2 = \{(1,2,0),(0,-1,0),(1,-1,1)\}$.

```
kill(all)$
A:matrix([-1,0],[2,0],[1,3]);
u[1]:[1,0]$
u[2]:[2,-1]$
v[1]:[1,2,0]$
v[2]:[0,-1,0]$
v[3]:[1,-1,1]$
for i:1 thru 2 do(
T[i]:A[1,i] * v[1]+A[2,i] * v[2]+A[3,i] * v[3])$
for j:1 thru 2 do(
eq[j]:u[1][j] * p+u[2][j] * q)$
soln:solve([eq[1]=x,eq[2]=y],[p,q])$
a:ev(p,soln)$
b:ev(q,soln)$
print("The linear transformation is")$
T:radcan(T[1] * a+T[2] * b)$
print("T(x,y)=",T)$
```

13.3. **Exercise.**

(1) Find the linear transformation T with respect to standard basis, given the matrix of linear transformation $A = \begin{pmatrix} 1 & 0 & 1 \\ 2 & 1 & 1 \\ -1 & 1 & -2 \end{pmatrix}$.

14. Rank and nullity

Rank: Let $T : U \to V$ be a linear transformation. The dimension of the range space $R(T)$ is called the rank of the linear transformation T denoted by $r(T)$.

Nullity: Let $T : U \to V$ be a linear transformation. The dimension of the null space $N(T)$ is called the nullity of the linear transformation T denoted by $n(T)$.

Rank-nullity theorem: Let $T : U \to V$ be a linear transformation and U be a finite dimensional vector space, then $r(T) + n(T) = d[V]$ or rank + nullity =dimension of the domain.

14.1. **Program.**

If linear transformation $T : \mathbb{R}^3 \to \mathbb{R}^3$ is defined by $T(e_1) = (1, -1, 0)$, $T(e_2) = (2, 0, 1)$ and $T(e_3) = (1, 1, 1)$ then find range space, rank and nullity of T and hence verify rank-nullity theorem.

```
kill(all)$
d[V]:3$
print("Dimension of domain space is",d[V])$
M:matrix([1,-1,0],[2,0,1],[1,1,1]);
r:rank(M)$
print("Dimension of the range space R(T) is",r)$
e:echelon(M);
print("Range space R(T) is generated by",{e[1],e[2]})$
nullspace(M);
n:nullity(M)$
print("Dimension of null space N(T) is",n)$
if d[V]=(r+n) then
disp("Rank-nullity theorem is verified")$
```

14.2. **Program.**

If linear transformation $T : \mathbb{R}^3 \to \mathbb{R}^3$ is defined by $T(e_1) = (1, 1, 0)$ $T(e_2) = (0, 1, 1)$ and $T(e_3) = (1, 2, 1)$ then find range space, rank and nullity of T and hence verify rank-nullity theorem.

```
kill(all)$
d[V]:3$
print("Dimension of domain space is",d[V])$
M:matrix([1,1,0],[0,1,1],[1,2,1]);
r:rank(M)$
print("Dimension of the range space R(T) is",r)$
e:echelon(M);
print("Range space R(T) is generated by",{e[1],e[2]})$
nullspace(M);
n:nullity(M)$
print("Dimension of null space N(T) is",n)$
if d[V]=(r+n) then
disp("Rank-nullity theorem is verified")$
```

14.3. **Exercise.**

(1) If linear transformation $T : \mathbb{R}^3 \to \mathbb{R}^3$ is defined by $T(e_1) = (1,1,0)$ $T(e_2) = (1,0,1)$ and $T(e_3) = (0,1,1)$ then find range space, rank and nullity of T and hence verify rank-nullity theorem.

(2) If linear transformation $T : \mathbb{R}^3 \to \mathbb{R}^3$ is defined by $T(e_1) = (1,1,2)$ $T(e_2) = (1,-1,0)$ and $T(e_3) = (0,0,1)$ then find range space, rank and nullity of T and hence verify rank-nullity theorem.

15. Gradient, divergence and curl

Gradient of a scalar point function: Let $\phi(x,y,z)$ be a continuously differential scalar point function. The gradient of ϕ, is denoted by $\nabla\phi$ or gradϕ, defined by

$$\text{grad}\phi = \nabla\phi = \left[\frac{\partial\phi}{\partial x}, \frac{\partial\phi}{\partial y}, \frac{\partial\phi}{\partial z}\right] = \mathbf{i}\frac{\partial\phi}{\partial x} + \mathbf{j}\frac{\partial\phi}{\partial y} + \mathbf{k}\frac{\partial\phi}{\partial z}.$$

Divergence of a vector function: If $F(x,y,z)$ be a continuously differentiable vector function, then divergence of F, denoted by $\nabla \cdot F$ or div$F$ is defined by

$$\text{div}F = \nabla \cdot F = \left(\mathbf{i}\frac{\partial}{\partial x} + \mathbf{j}\frac{\partial}{\partial y} + \mathbf{k}\frac{\partial}{\partial z}\right) \cdot F.$$

If $F = f_1\mathbf{i} + f_2\mathbf{j} + f_3\mathbf{k}$, then

$$\text{div}F = \nabla \cdot F = \mathbf{i}\frac{\partial f_1}{\partial x} + \mathbf{j}\frac{\partial f_2}{\partial y} + \mathbf{k}\frac{\partial f_3}{\partial z}.$$

Curl of a vector function: If $F(x,y,z)$ be a continuously differentiable vector function, then curl of F, denoted by $\nabla \times F$ or curl$F$ is defined by

$$\text{curl}F = \nabla \times F = \left(\mathbf{i} \times \frac{\partial F}{\partial x} + \mathbf{j} \times \frac{\partial F}{\partial y} + \mathbf{k} \times \frac{\partial F}{\partial z}\right).$$

If $F = f_1\mathbf{i} + f_2\mathbf{j} + f_3\mathbf{k}$, then

$$\text{curl}F = \nabla \times F = \mathbf{i}\left(\frac{\partial f_3}{\partial y} - \frac{\partial f_2}{\partial z}\right) + \mathbf{j}\left(\frac{\partial f_1}{\partial z} - \frac{\partial f_3}{\partial x}\right) + \mathbf{k}\left(\frac{\partial f_2}{\partial x} - \frac{\partial f_1}{\partial y}\right).$$

Laplacian operator: The operator denoted by ∇^2 and defined by

$$\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}.$$

is called a Laplacian operator.

15.1. Program for gradient of a function.

To find gradient of $\phi = x^2 + y^2 + z^2$

```
/*to find gradient of x^2+y^2+z^2 */
kill(all)$
phi:x^2+y^2+z^2;
grad_phi:[diff(phi,x,1),diff(phi,y,1),diff(phi,z,1)];
```

```
sum1:0$
a[1]:i$
a[2]:j$
a[3]:k$
for i: 1 thru 3 do (
sum1:sum1+grad_phi[i]*a[i])$
print("Gradient of Phi is ",sum1)$
```

15.2. **Program for divergence of a function.**
To find divergence of vector function $F(x, y) = [x^3y, xy^3]$

```
/*To find divergence of vector function F(x,y)=[x^3*y,x*y^3]*/
F(x,y):=[x^3*y,x*y^3];
divV:diff(F(x,y)[1],x,1)+diff(F(x,y)[2],y,1)$
print("DivF=",divV)$
```

To find divergence of vector function $F(x, y, z) = [x^3y^2, y^3z^2, x^2z^3]$

```
/*To find divergence of vector function
F(x,y,z)=[x^3*y^2,y^3*z^2,x^2*z^3]*/
F(x,y,z):=[x^3*y^2,y^3*z^2,x^2*z^3];
divV:diff(F(x,y,z)[1],x,1)+diff(F(x,y,z)[2],y,1)+diff(F(x,y,z)[3],z,1);
```

15.3. **Program for curl of a function.**
To find the curl of $F(x, y, z) := [x^2yz, xy^2z, xyz^2]$

```
/*To find the curl of F(x,y,z):=[x^2*y*z,x*y^2*z,x*y*z^2]*/
kill(all)$
load(vect)$
F(x,y,z):=[x^2*y*z,x*y^2*z,x*y*z^2];
curl_F:[diff(F(x,y,z)[3],y)-diff(F(x,y,z)[2],z),diff(F(x,y,z)[1],z)
-diff(F(x,y,z)[3],x),diff(F(x,y,z)[2],x)-diff(F(x,y,z)[1],y)];
curlF:express(curl(F(x,y,z)));
ev(curlF,diff);
```

15.4. **Program for Laplacian.**
Find the Laplacian of $F(x, y, z) = e^x \cos y$.

```
kill(all)$
load(vect)$
F(x,y,z):=%e^x*cos(y);
diff(F(x,y,z),x,2)+diff(F(x,y,z),y,2)+diff(F(x,y,z),z,2);
express(laplacian(F(x,y,z)));
ev(%,diff);
```

15.5. **Gradient, divergence, curl, Laplacian in Cartesian coordinates.**

```
kill(all)$load(vect)$load(vect_transform)$
declare([phi,psi],scalar)$declare([F,G],nonscalar)$
scalefactors(cartesian3d)$
express(grad(phi));
express(div(F));
express(curl(G));
express(laplacian(phi));
express(div(phi*F));
express(curl(grad(phi)));
express(curl(F*G));
```

15.6. **Gradient, divergence, curl, Laplacian in cylindrical coordinates.** Maxima code for Gradient, divergence, curl, Laplacian in cylindrical coordinates

```
kill(all)$load(vect)$load(vect_transform)$
declare([phi,psi],scalar)$declare([F,G],nonscalar)$
scalefactors(polarcylindrical)$
express(grad(phi));
express(div(F));
express(curl(G));
express(laplacian(Phi));
express(div(phi*F));
express(curl(grad(phi)));
express(curl(F*G));
```

15.7. **Program.**
To find the gradient, divergence, curl, laplacian for $\phi = r\cos\theta$,
$F = [r, r\cos\theta, \sin\theta]$.

```
kill(all)$load(vect)$load(vect_transform)$
phi:r*cos(theta);F:[r,r*cos(theta),sin(theta)];
scalefactors(polarcylindrical)$
express(grad(phi));ev(%,nouns);
express(div(F));ev(%,nouns);
express(curl(F));ev(%,nouns);
express(laplacian(phi));ev(%,nouns);
express(div(phi*F));ev(%,nouns);
```

15.8. **Gradient, divergence, curl, Laplacian in spherical coordinates.** Maxima code for Gradient, divergence, curl, Laplacian in spherical coordinates

```
kill(all)$load(vect)$load(vect_transform)$
declare([phi,psi],scalar)$declare([F,G],nonscalar)$
scalefactors(spherical)$
express(grad(phi));
express(div(F));
express(curl(G));
express(laplacian(phi));
express(div(phi*F));
express(curl(grad(phi)));
express(curl(F*G));
```

16. Cyclic notations to derive vector identities

Some of the vector identities are listed in the following program.

```
kill(all)$
load(vect)$
declare([phi,psi],scalar)$
declare([F,G],nonscalar)$
f:grad(phi*psi)$
print(grad(phi*psi),"=",ev(vectorsimp(f),expandall))$
g:grad(phi/psi)$
print(grad(phi/psi),"=",ev(vectorsimp(g),expandall))$
h:div(phi*psi)$
print(div(phi*psi),"=",ev(vectorsimp(h),expandall))$
p:div(phi*F)$
print(div(phi*F),"=",ev(vectorsimp(p),expandall))$
q:laplacian(phi*psi)$
print(laplacian(phi*psi),"=",ev(vectorsimp(%),expandall))$
r:div(curl(F))$
print("div(curl(F))=",ev(vectorsimp(r),expandall))$
```

17. Verifying Green's theorem

Statement of Green's theorem in the plane: If $P(x, y)$ and $Q(x, y)$ be two continuous functions having continuous partial derivatives in a region R of the xy-plane, bounded by a simple closed curve C, then

$$\oint (Pdx + Qdy) = \int\int_R \left(\frac{\partial Q}{\partial x} - \frac{\partial P}{\partial y} \right) dxdy.$$

17.1. **Program.**

Verify Green's theorem for the $\oint (x+2y)dx + (x-2y)dy$, where c is the curve consists of the coordinate axis and the line $x = 1, y = 1$.

```
kill(all)$
P(x,y):=x+2*y$
Q(x,y):=x-2*y$
s:diff(Q(x,y),x)-diff(P(x,y),y)$
Z1:integrate(integrate(s,x,0,1),y,0,1);
depends(x,y)$
F:[P(x,y),Q(x,y)]$
x(x):=1$
dx:diff(1,x)$
dy:diff(y,y)$
dt2:[dx,dy]$
H:(F.dt2)$
H1:subst(x=1,H)$
I1:integrate(H1,y,0,1);
diff(0,x)$
diff(y,y)$
dt3:[dx,dy]$
H2:(F.dt3)$
H3:subst(x=0,H2)$
I2:integrate(H3,y,1,0);
depends(y,x)$
y(y):=0$
dx:diff(x,x)$
dy:diff(0,y)$
dt:[dx,dy]$
H4:(F.dt)$
H5:subst(y=0,H4)$
I3:integrate(H5,x,0,1);
depends(y,x)$
dx:diff(x,x)$
dy:diff(1,y)$
dt1:[dx,dy]$
H6:(F.dt1)$
H7:subst(y=1,H6)$
I4:integrate(H7,x,1,0);
```

```
Z:I1+I2+I3+I4;
if Z=Z1 then
disp("Greens theorem is verified")
else
disp("Greens theorem is not verified")$
```

17.2. **Program.**
Verify Green's theorem for $\oint(3x^2-8y^2)dx+2y(2-3x)dy$, where C is the boundary of the rectangular area enclosed by the lines $x=0,\ x=1,\ y=0\ y=2$.

```
/*\int{(3x^2-8y^2)dx+2y(2-3x)dy},
where c is closed curve x=0,x=1,y=0 and y=2*/
kill(all)$
P(x,y):=3*x^2-8*y^2$
Q(x,y):=2*y*(2-3*x)$
s:diff(Q(x,y),x)-diff(P(x,y),y)$
Z1:integrate(integrate(s,x,0,1),y,0,2);
depends(x,y)$
F:[P(x,y),Q(x,y)]$
dx:diff(x,x)$
dy:diff(0,y)$
dt2:[dx,dy]$
H:(F.dt2);
H1:subst(y=0,H);
I1:integrate(H1,x,0,1);

x(x):=1$
dx:diff(1,x)$
dy:diff(y,y)$
dt3:[dx,dy]$
H2:(F.dt3);
H3:subst(x=1,H2)$
I2:integrate(H3,y,0,2);

depends(y,x)$
y(y):=2$
dx:diff(x,x)$
dy:diff(0,y)$
dt:[dx,dy]$
H4:(F.dt)$
```

```
H5:subst(y=2,H4);
I3:integrate(H5,x,1,0);

depends(y,x)$
dx:diff(1,x)$
dy:diff(y,y)$
dt1:[dx,dy]$
H6:(F.dt1)$
H7:subst(x=0,H6)$
I4:integrate(H7,y,2,0);
Z:I1+I2+I3+I4;
if Z=Z1 then
disp("Greens theorem is verified")
else
disp("Greens theorem is not verified")$
```

17.3. **Program.**
Verify Green's theorem for the $\oint(xy+y^2)dx+x^2dy$, where c is the closed curve bounded by $y=x,\ y=x^2$.

```
/*\int{(xy+y^2)dx+x^2dy}, c is closed bounded by y=x and y=x^2*/
kill(all)$
P(x,y):=x*y+y^2$
Q(x,y):=x^2$
s:diff(Q(x,y),x)-diff(P(x,y),y)$
Z1:integrate(integrate(s,y,x^2,x),x,0,1);
depends(y,x)$
F:[P(x,y),Q(x,y)];
y(x):=x^2;
dx:diff(x,x);
dy:diff(y(x),x);
dt2:[dx,dy];
H:(F.dt2);
H1:subst(y=x^2,H);
I1:integrate(H1,x,0,1);
dx:diff(x,x);
dy:diff(y,y);
dt3:[dx,dy];
H2:(F.dt3);
H3:subst(y=x,H2);
```

```
I2:integrate(H3,x,1,0);
Z:I1+I2;
if Z=Z1 then
disp("Greens theorem is verified")
else
disp("Greens theorem is not verified")$
```

17.4. **Program.**

Verify Green's theorem for the $\oint(3x^2-8y^2)dx+(4y-6xy)dy$, where c is the closed curve bounded by $y=x^2,\ y^2=x$.

```
/*\int{(3x^2-8y^2)dx+(4y-6xy)dy}, c is closed bounded by y=x^2 and y^2=x*/
kill(all)$
P(x,y):=3*x^2-8*y^2$
Q(x,y):=4*y-6*x*y$
s:diff(Q(x,y),x)-diff(P(x,y),y)$
Z1:integrate(integrate(s,y,x^2,sqrt(x)),x,0,1);
depends(y,x)$
F:[P(x,y),Q(x,y)];
y(x):=x^2;
dx:diff(x,x);
dy:diff(y(x),x);
dt2:[dx,dy];
H:(F.dt2);
H1:subst(y=x^2,H);
I1:integrate(H1,x,0,1);

y(x):=sqrt(x);
dx:diff(x,x);
dy:diff(y(x),x);
dt3:[dx,dy];
H2:(F.dt3);
H3:subst(y=sqrt(x),H2);
I2:integrate(H3,x,1,0);
Z:I1+I2;
if Z=Z1 then
disp("Greens theorem is verified")
else
disp("Greens theorem is not verified")$
```

18. Verifying Gauss divergence theorem

Statement of Gauss divergence theorem: Let S be the closed boundary surface of a region of volume V. Then for a continuously differentiable vector field $\overrightarrow{F}$ defined on V and on S

$$\iint\limits_{S} \overrightarrow{F}\cdot\hat{n}ds=\iiint\limits_{V} \operatorname{div}\overrightarrow{F}dv,$$

where $\hat{n}$ is the outward drawn unit normal vector at any point of S.

18.1. **Program.**
Verify the divergence theorem for $\overrightarrow{F} = 2xy\hat{i} + yz^2\hat{j} + xz\hat{k}$ over the rectangular parallelepiped $0 \leq x \leq 1, 0 \leq y \leq 2, 0 \leq z \leq 3$.

```
kill(all)$
load("vect")$
F:[2*x*y,y*z^2,x*z];
div(F)$
express(%)$
I:ev(%,diff)$
H:integrate(integrate(integrate(I,x,0,1),y,0,2),z,0,3);
i:[1,0,0]$
j:[0,1,0]$
k:[0,0,1]$
S1:F.-k$
s1:subst(z=0,S1)$
I1:integrate(integrate(s1,x,0,1),y,0,2);
S2:F.-j$
s2:subst(y=0,S2)$
I2:integrate(integrate(s2,x,0,1),z,0,3);
S3:F.-i$
s3:subst(x=0,S3)$
I3:integrate(integrate(s3,y,0,2),z,0,3);
S4:F.k$
s4:subst(z=3,S4)$
I4:integrate(integrate(s4,x,0,1),y,0,2);
S5:F.j$
s5:subst(y=2,S5)$
I5:integrate(integrate(s5,x,0,1),z,0,3);
S6:F.i$
s6:subst(x=1,S6)$
I6:integrate(integrate(s6,y,0,2),z,0,3);
J:I1+I2+I3+I4+I5+I6;
if(H=J) then
disp("Gauss divergence theorem is satisfied")
else
disp("Gauss divergence theorem is not satisfied")$
```

18.2. **Exercise.**
Verify the divergence theorem for $\overrightarrow{F} = 4x\hat{i} + y\hat{j} + z\hat{k}$ over the region bounded by the planes $x = 0, y = 0, z = 0$ and $2x + y + 2z = 6$.

19. Verifying Stokes theorem

Statement Stokes theorem: The line integral of the tangential component of a vector $\overrightarrow{F}$ taken around a simple closed curve C is equal to the surface integral of the normal component of the curl $\overrightarrow{F}$ taken over any surface S having C as its boundary i.e.,

$$\oint_C \overrightarrow{F}.d\overrightarrow{r} = \iint_S \left(\nabla \times \overrightarrow{F}\right) \cdot \hat{n} ds.$$

19.1. **Program.**
Verify the Stokes theorem for $\overrightarrow{F} = (2x - y)\hat{i} - yz^2\hat{j} - y^2z\hat{k}$ over S, the upper half sphere $x^2 + y^2 + z^2 = 1$ and C is its boundary.

```
kill(all)$
load("vect")$
F:[2*x-y,-y*z^2,-y^2*z];
curl(F)$
express(%)$
E:ev(%,diff)$
k:[0,0,1]$
E1:E.k$
assume(x^2<1)$
I:4*integrate(integrate(E1,y,0,sqrt(1-x^2)),x,0,1);
depends([x,y,z],t)$
x:cos(t)$
y:sin(t)$
z:0$
F:[2*x-y,-y*z^2,-y^2*z]$
dx:diff(x,t)$
dy:diff(y,t)$
dz:diff(z,t)$
dr:[dx,dy,dz]$
F1:F.dr$
I1:integrate(F1,t,0,2*%pi);
if(I=I1) then
disp("Stokes theorem is verified")
else
```

```
disp("Stokes theorem is not verified")$
```

19.2. **Exercise.**
Verify the divergence theorem for $\overrightarrow{F} = y^2\hat{i}+xy\hat{j}-xz\hat{k}$ over S, the hemisphere $x^2+y^2+z^2 = a^2$ and $z \geq 0$.