

**Q1) Chech whether the given set or vector is an orthogonal set or not. Is the set orthonormal ? If not convert it to an orthonormal set by normalizing vector.**

**1)**

x=c(2,-1);y=c(3,6)

> z=sum(x\*y)

> z

[1] 0

> if(z==0)

+ {print('vectors are orthogonal')

+ ###Check vectors are orthonormal or not

+ normx=sqrt(sum(x\*x));normx

+ normy=sqrt(sum(y\*y));normy

+ if(normx==1&&normy==1){print('vectors are orthonormal')}else

+ {print('vectors are not orthonormal')

+ orthonormx=x/normx

+ orthonormx

+ orthonormy=y/normy

+ orthonormy

+ }

+ }else

+ {print('vectors are not orthogonal')}

[1] "vectors are orthogonal"

[1] "vectors are not orthonormal"

### convert vectors into orthonormal

[1] 0.4472136 0.8944272

> > **2)**

u=c(-1,0,1);v=c(2,0,2);w=c(0,5,0)

> #

> z1=sum(u\*v);z1

[1] 0

> z2=sum(v\*w);z2

[1] 0

> z3=sum(u\*w);z3

[1] 0

> if(z1==0&&z2==0&&z3==0)

+ {print('vectors are orthogonal')

+ ###Check vectors are orthonormal or not

+ normu=sqrt(sum(u\*u));normu

+ normv=sqrt(sum(v\*v));normv

+ normw=sqrt(sum(w\*w));normw

+ if(normu==1&&normv==1&&normw==1){print('vectors are orthonormal')}else

+ {print('vectors are not orthonormal')

+ orthonormu=u/normu;orthonormu

+ orthonormv=v/normv;orthonormv

+ orthonormw=w/normw;orthonormw

+ }

+ }else

+ {print('vectors are not orthogonal')

+ print('and hence vectors are not orthonormal')

+ }

[1] "vectors are orthogonal"

[1] "vectors are not orthonormal"

### convert vectors into orthonormal

[1] -0.7071068 0.0000000 0.7071068

[1] 0.7071068 0.0000000 0.7071068

[1] 0 1 0

***3)***

u=c(1,0,1);v=c(-1,-5,1);w=c(2,0,-2)

> #

> z1=sum(u\*v);z1

[1] 0

> z2=sum(v\*w);z2

[1] -4

> z3=sum(u\*w);z3

[1] 0

> if(z1==0&&z2==0&&z3==0)

+ {print('vectors are orthogonal')

+ ###Check vectors are orthonormal or not

+ normu=sqrt(sum(u\*u));normu

+ normv=sqrt(sum(v\*v));normv

+ normw=sqrt(sum(w\*w));normw

+ if(normu==1&&normv==1&&normw==1){print('vectors are orthonormal')}else

+ {print('vectors are not orthonormal')

+ orthonormu=u/normu;orthonormu

+ orthonormv=v/normv;orthonormv

+ orthonormw=w/normw;orthonormw

+ }

+ }else

+ {print('vectors are not orthogonal')

+ print('and hence vectors are not orthonormal')

+ }

[1] "vectors are not orthogonal"

[1] "and hence vectors are not orthonormal"

>

***4)***

x=c(4,-7);y=c(6,-1)

> z=sum(x\*y)

> z

[1] 31

> if(z==0)

+ {print('vectors are orthogonal')

+ ###Check vectors are orthonormal or not

+ normx=sqrt(sum(x\*x));normx

+ normy=sqrt(sum(y\*y));normy

+ if(normx==1&&normy==1){print('vectors are orthonormal')}else

+ {print('vectors are not orthonormal')

+ orthonormx=x/normx

+ orthonormx

+ orthonormy=y/normy

+ orthonormy

+ }

+ }else

+ {print('vectors are not orthogonal')

+ print('and hence vectors are also not orthonormal')

+ }

[1] "vectors are not orthogonal"

[1] "and hence vectors are also not orthonormal"

***5)***

u=c(1/5,1/5,1/5);v=c(-1/2,1/2,0);w=c(1/3,1/3,-1/3)

> #

> z1=sum(u\*v);z1

[1] 0

> z2=sum(v\*w);z2

[1] 0

> z3=sum(u\*w);z3

[1] 0.06666667

> if(z1==0&&z2==0&&z3==0)

+ {print('vectors are orthogonal')

+ ###Check vectors are orthonormal or not

+ normu=sqrt(sum(u\*u));normu

+ normv=sqrt(sum(v\*v));normv

+ normw=sqrt(sum(w\*w));normw

+ if(normu==1&&normv==1&&normw==1){print('vectors are orthonormal')}else

+ {print('vectors are not orthonormal')

+ orthonormu=u/normu;orthonormu

+ orthonormv=v/normv;orthonormv

+ orthonormw=w/normw;orthonormw

+ }

+ }else

+ {print('vectors are not orthogonal')

+ print('and hence vectors are not orthonormal')

+ }

[1] "vectors are not orthogonal"

[1] "and hence vectors are not orthonormal"

>

**Q2 Find projection of u along v**

**1)**

u=c(2,1,-3);v=c(3,0,6)

> iuv=sum(u\*v)

> iuv

[1] -12

> ivv=sum(v\*v)

> ivv

[1] 45

> projuv=(iuv/ivv)\*v

> projuv

[1] -0.8 0.0 -1.6

**2)**

u=c(4,2,0);v=c(-6,2,-6)

> iuv=sum(u\*v)

> iuv

[1] -20

> ivv=sum(v\*v)

> ivv

[1] 76

> projuv=(iuv/ivv)\*v

> projuv

[1] 1.5789474 -0.5263158 1.5789474

**3)**

u=c(-1,0,1);v=c(1,0,1)

> iuv=sum(u\*v)

> iuv

[1] 0

> ivv=sum(v\*v)

> ivv

[1] 2

> projuv=(iuv/ivv)\*v

> projuv

[1] 0 0 0

**Q3 . Find fourier Coefficient of u along v**

**1)**

u=c(-3,2,-1);v=c(4,0,-9)

> ip1=sum(v\*u);ip1

[1] -3

> ip2=sum(v\*v);ip2

[1] 97

> ## Fourier coeficient of u along v

> k=(ip1/ip2);k

[1] -0.03092784

> **2)**

u=c(1,0,1);v=c(0,1,-1)

> ip1=sum(v\*u);ip1

[1] -1

> ip2=sum(v\*v);ip2

[1] 2

> ## Fourier coeficient of u along v

> k=(ip1/ip2);k

[1] -0.5

> **3)**

u=c(-1,5,3);v=c(2,1,6)

> ip1=sum(v\*u);ip1

[1] 21

> ip2=sum(v\*v);ip2

[1] 41

> ## Fourier coeficient of u along v

> k=(ip1/ip2);k

[1] 0.5121951

>

**Q4 . Use Gram chmidt orthogonalization Process to transform the given basis into an orthonormal basis**

**1)**

u1=c(1,-3);u1

[1] 1 -3

> u2=c(2,2);u2

[1] 2 2

> v1=u1;v1

[1] 1 -3

> ip1=sum(u2\*v1);ip1

[1] -4

> ipv1=sum(v1\*v1);ipv1

[1] 10

> v2=u2-(ip1/ipv1)\*v1;v2

[1] 2.4 0.8

>

**2)**

u1=c(1,0);u1

[1] 1 0

> u2=c(3,-5);u2

[1] 3 -5

> v1=u1;v1

[1] 1 0

> ip1=sum(u2\*v1);ip1

[1] 3

> ipv1=sum(v1\*v1);ipv1

[1] 1

> v2=u2-(ip1/ipv1)\*v1;v2

[1] 0 -5

>

**3)**

u1=c(1,0,0);u1

[1] 1 0 0

> u2=c(3,7,-2);u2

[1] 3 7 -2

> u3=c(0,4,2);u3

[1] 0 4 2

> v1=u1;v1

[1] 1 0 0

> ip1=sum(u2\*v1);ip1

[1] 3

> ipv1=sum(v1\*v1);ipv1

[1] 1

> v2=u2-(ip1/ipv1)\*v1;v2

[1] 0 7 -2

> ipv2=sum(v2\*v2);ipv2

[1] 53

> ip2=sum(u3\*v1);ip2

[1] 0

> ip3=sum(u3\*v2);ip3

[1] 24

> v3=u3-(ip2/ipv1)\*v1-(ip3/ipv2)\*v2;v3

[1] 0.0000000 0.8301887 2.9056604

>