

```

RealVectorSum[u_, v_] := u + v

RealVectorScalarProduct[a_, u_] := a * u

FunctionSum[f_, g_] :=
Module[{fRule, fDomain, gRule, gDomain, sumRule, sumDomain, sum},
  fRule = f[[1]];
  fDomain = f[[2]];
  gRule = g[[1]];
  gDomain = g[[2]];
  sumRule[x_] := fRule[x] + gRule[x];
  sumDomain = fDomain;
  sum = {sumRule, sumDomain};
  sum
]

FunctionScalarProduct[a_, f_] :=
Module[{fRule, fDomain, productRule, productDomain, product},
  fRule = f[[1]];
  fDomain = f[[2]];
  productRule[x_] := a * fRule[x];
  productDomain = fDomain;
  product = {productRule, fDomain};
  product
]

DotProduct[u_, v_] := Sum[u[[i]] * v[[i]], {i, 1, Length[u]}]

FunctionProduct[f_, g_] := Module[{fFormula, gFormula, domain, a, b},
  fFormula = f[[1]];
  gFormula = g[[1]];
  domain = f[[2]];
  a = domain[[1]];
  b = domain[[2]];
  Integrate[fFormula[x] * gFormula[x], {x, a, b}]
]

VectorLength[u_] := Sqrt[InnerProduct[u, u]]

ProjectionVector[v_, uList_] := Module[{sum, i},
  sum = ScalarProduct[InnerProduct[v, uList[[1]]], uList[[1]];
  For[i = 2, i ≤ Length[uList], i++,
    sum = VectorSum[sum, ScalarProduct[InnerProduct[v, uList[[i]]], uList[[i]]];
  ];
  sum
]

```

```

NextVector[v_, uList_] := Module[
  {zeroVector, P, nextVector, differenceVector, scaleFactor, differenceLength},
  zeroVector = ScalarProduct[0, v];
  P = ProjectionVector[v, uList];
  differenceVector = VectorSum[v, ScalarProduct[-1, P]];
  differenceLength = VectorLength[differenceVector];
  If[differenceLength < 10^(-100),
    nextVector = zeroVector,
    nextVector = ScalarProduct[(differenceLength)^(-1), differenceVector]
  ];
  nextVector
]

```

```

OrthonormalBasis[vList_] :=
Module[{zeroVector, uList, i, u, basis, vectorLength, f},
  zeroVector = ScalarProduct[0, vList[[1]]];
  uList = {zeroVector};
  For[i = 1, i ≤ Length[vList], i++,
    u[i] = NextVector[vList[[i]], uList];
    uList = Append[uList, u[i]];
  ];
  basis = {};
  For[i = 1, i ≤ Length[uList], i++,
    vectorLength = VectorLength[uList[[i]]];
    If[vectorLength > 0,
      basis = Append[basis, ScalarProduct[vectorLength^(-1), uList[[i]]]
    ];
  ];
  basis
]

```