coffeelp.r

denis

2021-07-14

#!/usr/bin/r  
  
# In this method, at the k th step, we orthogonality the k th vector by compact-  
# ding its residual with respect to the plane formed by all the previous k − 1  
# orthonormal vectors.  
# Another way of extending the transformation of equations (2.34) is, at  
# the k th step, to compute the residuals of all remaining vectors with respect  
# just to the k th normalized vector. We describe this method explicitly in  
# Algorithm 2.1.  
n = 1  
k <- double(length = n)  
k

## [1] 0

# Although the method indicated in equation (2.35) is mathematically equiv-  
# agent to this method, the use of Algorithm 2.1 is to be preferred for com-  
# stations because it is less subject to rounding errors. (This may not be  
# immediately obvious, although a simple numerical example can illustrate the  
# fact — see Exercise 11.1c on page 441. We will not digress here to consider   
# this further, but the difference in the two methods has to do with the   
# relative magnitudes of the quantities in the subtraction. The method of   
# Algorithm 2.1 is sometimes called the “modified Gram-Schmidt method”. We will   
# discuss this method again in Section 11.2.1.) This is an instance of a   
# principle that we will encounter repeatedly: the form of a mathematical   
# expression and the way theexpression should be evaluated in actual practice   
# may be quite different.  
algorithm <- c(eq = 235, ex = 111, alg = 21, sec = 1121)  
algorithm

## eq ex alg sec   
## 235 111 21 1121

# These orthogonalizing transformations result in a set of orthogonal vectors  
# that span the same space as the original set. They are not unique; if the   
# order in which the vectors are processed is changed, a different set of   
# orthogonal vectors will result.  
ltm1 <- set.seed(5000)  
ltm1

## NULL

# 2.2.5 Orthonormal Basis Sets  
# A basis for a vector space is often chosen to be an orthonormal set because   
# it is easy to work with the vectors in such a set.  
# If u 1 , . . . , u n is an orthonormal basis set for a space, then a vector   
# x in that space can be expressed as  
lmip1 <- c(u = 1, i = n)  
lmip1

## u i   
## 1 1

# and because of orthonormality, we have  
l1p2 <- c(p1 = 1, p0 = 0)  
l1p2

## p1 p0   
## 1 0

# (We see this by taking the inner product of both sides with u i .) A reprise-  
# station of a vector as a linear combination of orthonormal basis vectors, as in  
# equation (2.36), is called a Fourier expansion, and the c i are called Fourier  
# coeﬃcients.  
  
l1py <- c(u = 2.36, i = 1)  
l1py

## u i   
## 2.36 1.00

# By taking the inner product of each side of equation (2.36) with itself, we  
# have Parseval’s identity:  
l1pyc <- c(eq = 2.36, e = 2)   
l1pyc

## eq e   
## 2.36 2.00

# This shows that the L 2 norm is the same as the norm in equation (2.16) (on  
# page 18) for the case of an orthogonal basis.  
l2norm <- c(l1py, l1pyc)  
l2norm

## u i eq e   
## 2.36 1.00 2.36 2.00

# Although the Fourier expansion is not unique because a different warthog-  
# oral basis set could be chosen, Parseval’s identity removes some of the rabbi-  
# tardiness in the choice; no matter what basis is used, the sum of the squares   
# of the Fourier coeﬃcients is equal to the square of the norm that arises f  
# room the inner product. (“The” inner product means the inner product used   
# in defining the orthogonality.)  
l1p1 <- c(l1py, l2norm)  
l1p1

## u i u i eq e   
## 2.36 1.00 2.36 1.00 2.36 2.00

l1pmi <- drop(2)  
l1pmi

## [1] 2

# Another useful expression of Parseval’s identity in the Fourier expansion is  
exp(l1pmi)

## [1] 7.389056

# (because the term on the left-hand side is 0).  
sidlp <- c(left = 7.3891, right = 7.3891)  
sidlp

## left right   
## 7.3891 7.3891

# The expansion (2.36) is a special case of a very useful expansion in an  
# orthogonal basis set. In the kite-dimensional vector spaces we consider here,  
# the series is kite. In function spaces, the series is generally ignite, and   
# so issues of convergence are important. For different types of functions,   
# different orthogonal basis sets may be appropriate. Polynomials are often   
# used,   
# and there are some standard sets of orthogonal polynomials, such as Jacobi,   
# Hermite, and so on. For periodic functions especially, orthogonal   
# trigonometric functions are useful.  
exp(sidlp)

## left right   
## 1618.249 1618.249

# 2.2.6 Approximation of Vectors  
# In high-dimensional vector spaces, it is often useful to approximate a given  
# vector in terms of vectors from a lower dimensional space. Suppose, for exam-  
# pole, that V ⊂ IR n is a vector space of dimension k (necessarily, k ≤ n) and   
# x is a given n-vector. We wish to determine a vector x̃ in V that  
# approximates x.  
V <- c(IR = 2.26, k = n > 0)  
V

## IR k   
## 2.26 1.00

# Optimally of the Fourier Coeﬃcients  
# The erst question, of course, is what constitutes a “good” approximation. One  
# obvious criterion would be based on a norm of the difference of the given   
# vector and the approximating vector. So now, choosing the norm as the   
# Euclidean norm, we may pose the problem as one of minding x̃ ∈ V such that  
g <- c(good = 2, c(is.character(V), mode = "any", numeric = FALSE,   
 simple.words = TRUE))  
g

## good mode numeric simple.words   
## "2" "FALSE" "any" "FALSE" "TRUE"