# hwk8.R

*evanjohnston*

*Sat Mar 26 18:20:56 2016*

```
# Evan Johnston
# M348 - hwk 8 - 29 March 2016

# Composite Simpson's Rule
# inputs:
#   a: first endpoint: a
#   b: last endpoint: b
#   n: even + integer: n
#   f: function to integrate
cSimpson<-function(a, b, n, f){

  # create h
  h<- (b - a)/n

  # initialize approximations
  xi0<- f(a) + f(b)
  xi1<- 0
  xi2<- 0

  # loop over the 1 to n-1 iterations
  for (k in 1:(n-1)){

    # set first xk
    x<- a + k*h

    # if k is even
    if (k%%2==0){

      # update xi2
      xi2<- xi2 + f(x)
    }

    # if k is odd
    else{

      # update xi1
      xi1<- xi1 + f(x)
    }
  }
  # create final step approximation
  xi<- h*(xi0+2*xi2+4*xi1)/3

  return (xi)
}

# Composite Trapezoid Rule
# inputs:
```

```r
#   a: first endpoint: a
#   b: last endpoint: b
#   n: even + integer: n
#   f: function to integrate
cTrapezoid<-function(a, b, n, f){

  # create h
  h<- (b - a)/n

  # initialize approximations
  xi0<- f(a)
  xi1<- 0
  xi2<- f(b)

  # loop over the 1 to n-1 iterations
  for (k in 1:(n-1)){

    # set first xk
    x<- a + k*h

    # update xi1
    xi1<- xi1 + f(x)
  }
  # create final step approximation
  xi<- h*(xi0+2*xi1+xi2)/2

  return (xi)
}

# create function to be integrated
g<-function(x){
  return(exp(1)^(2*x)*sin(3*x))
}

# create matrix for output
out<-matrix(rep(0,6),nrow=2)
colnames(out)<-c('n=10', 'n=100', 'n=1000')
rownames(out)<-c('cSimpson', 'cTrapezoid')

# fill matrix
for (i in (1:2)){
  for (j in (1:3)){
    n<-10^j
    if (i==1){
      out[i,j]<-cSimpson(0,2,n,g)
    }
    else{
      out[i,j]<-cTrapezoid(0,2,n,g)
    }
  }
}

out
```

```
##                  n=10     n=100    n=1000
## cSimpson   -14.20223 -14.21398 -14.21398
## cTrapezoid -13.80410 -14.20985 -14.21394
```