```
//  file: derivative_test.cpp
//
//  Program to study the error in differentiation rules
//
//  Programmer:  Dick Furnstahl  furnstahl.1@osu.edu
//
//  Revision history:
//      01/14/04  original version, translated from derivative_test.c
//      01/14/07  modified to be consistent with derivative_test_simple.cpp
//
//  Notes:
//   * Based on the discussion of differentiation in Chap. 8
//      of "Computational Physics" by Landau and Paez.
//   * Use the adaptive gsl_diff_central function as well.
//      Output from this with e^(-x) at x=1 is:
//  gsl_diff_central(1) = -3.6787944117560983e-01 +/- 6.208817e-04
//   actual relative error: 1.13284386e-11
//
//*****************************************************************
// include files
#include <iostream>             // note that .h is omitted
#include <iomanip>              // note that .h is omitted
#include <fstream>              // note that .h is omitted
using namespace std;            // we need this when .h is omitted
#include <gsl/gsl_math.h>
#include <gsl/gsl_diff.h>

// function prototypes
double test_function (double x, void *params_ptr);
double test_function_derivative (double x, void *params_ptr);

double forward_diff (double x, double h,
                     double (*f) (double x, void *params_ptr),
                     void *params_ptr);
double central_diff (double x, double h,
                     double (*f) (double x, void *params_ptr),
                     void *params_ptr);
double extrap_diff (double x, double h,
                    double (*f) (double x, void *params_ptr),
                    void *params_ptr);

//************************* main program **************************
int
main (void)
{
  void *params_ptr;             // void pointer passed to functions

  const double hmin = 1.e-10;   // minimum mesh size
  double x = 1.;                // find the derivative at x
  double alpha = 1.;            // a parameter for the function
  double diff_cd, diff_fd;      // central, forward difference
  double diff_extrap;           // extrapolated derivative
  double diff_gsl_cd;           // gsl adaptive central derivative
  gsl_function My_F;            // gsl_function type
  double abserr;                // absolute error

  ofstream out ("derivative_test.dat");        // open the output file

  params_ptr = &alpha;          // double to pass to function

  // exact answer for test
  double answer = test_function_derivative (x, params_ptr);

  My_F.function = &test_function;       // set up the gsl function
  My_F.params = params_ptr;
  gsl_diff_central (&My_F, x, &diff_gsl_cd, &abserr);   // gsl calculation

  cout << "gsl_diff_central(" << x << ")=" << scientific
    << setprecision (16) << diff_gsl_cd << " +/– "
```

```
    << setprecision (6) << abserr << endl;
  cout << " actual relative error: " << setprecision (8)
    << fabs((diff_gsl_cd - answer)/answer) << endl;

  double h = 0.1;               // initialize mesh spacing
  while (h >= hmin)
  {
    diff_fd = forward_diff (x, h, &test_function, params_ptr);
    diff_cd = central_diff (x, h, &test_function, params_ptr);
    diff_extrap = extrap_diff (x, h, &test_function, params_ptr);

    // print relative errors to output file
    out << scientific << setprecision (8)
      << log10 (h) << "  "
      << log10 (fabs ((diff_fd - answer) / answer)) << "  "
      << log10 (fabs ((diff_cd - answer) / answer)) << "  "
      << log10 (fabs ((diff_extrap - answer) / answer)) << endl;

    h /= 2.;            // reduce mesh by 2
  }

  out.close ();         // close the output stream
  return (0);           // successful completion
}

//************************* funct **************************
double
test_function (double x, void *params_ptr)
{
  double alpha;
  alpha = *(double *) params_ptr;

  return (exp (-alpha * x));
}

//************************* funct_deriv ********************
double
test_function_derivative (double x, void *params_ptr)
{
  double alpha = *(double *) params_ptr;

  return (-alpha * exp (-alpha * x));
}

//************************* forward_diff ********************
double
forward_diff (double x, double h,
              double (*f) (double x, void *params_ptr), void *params_ptr)
{
  return ( f(x + h, params_ptr) - f(x, params_ptr) ) / h;
}

//************************* central_diff ********************
double
central_diff (double x, double h,
              double (*f) (double x, void *params_ptr), void *params_ptr)
{
  return ( f(x + h/2., params_ptr) - f(x - h/2., params_ptr) ) / h;
}

//************************* extrap_diff ********************
double
extrap_diff (double x, double h,
             double (*f) (double x, void *params_ptr), void *params_ptr)
{
  return ( 8.*(f(x + h/4., params_ptr) - f(x - h/4., params_ptr))
         - (f(x + h/2., params_ptr) - f(x - h/2., params_ptr)) )
         / (3.*h);
}
```