



---

# COLNEWSC USER MANUAL

---



JANUARY 18, 2024

COLLOCATION SOFTWARE WITH  
SUPERCONVERGENT INTERPOLANTS FOR  
THE NUMERICAL SOLUTION OF  
BOUNDARY VALUE ORDINARY DIFFERENTIAL  
EQUATIONS

Authors: Mark Adams, Paul Muir

Department of Mathematics and Computing Science

Saint Mary's University

Halifax, N.S., CANADA B3H 3C3

Email: [adams.mark.alex@gmail.com](mailto:adams.mark.alex@gmail.com), [muir@smu.ca](mailto:muir@smu.ca)

## Table of Contents

Disclaimer .....	3
Platform Requirements.....	4
Description Of Software .....	5
Installation of Software.....	6
Test Problems .....	7
Description of Output.....	8

# Disclaimer: Open Source Software

## Platform Requirements :

Windows 10 with access to a Linux environment through PuTTY and Winscp.

PuTTY: a free implementation of SSH and Telnet for Windows and Unix platforms, along with an xterm terminal emulator

download: <https://www.putty.org>

WinSCP: a free popular SFTP client and FTP client for Microsoft Windows. Copy file between a local computer and remote servers using FTP, FTPS, SCP, SFTP, WebDAV or S3 file transfer protocols.

download: <https://winscp.net>

## Description Of Software

This is a BETA version of COLNEWSC. Please report bugs to Mark Adams at [adams.mark.alex@gmail.com](mailto:adams.mark.alex@gmail.com) with COLNEWSC in the subject line. COLNEWSC is a software package that uses much of the COLNEW source code and, like COLNEW, is used to find numerical solutions for boundary value ordinary differential equations. COLNEWSC is an update of the collocation solver COLNEW (available at <https://www.netlib.org/ode/colnew.f>). The collocation solution is augmented with a superconvergent (higher accuracy) interpolant based on the use of continuous mono-implicit Runge-Kutta methods. This generally allows COLNEWSC to obtain a numerical solution that meets the user tolerance using a much coarser mesh than would be required by COLNEW, leading to substantial savings in execution time. COLNEWSC requires that all problems be written in first-order system form,  $y'(x) = f(x, y(x))$ , with separated boundary conditions,  $bR(y(a))=0$ ,  $bL(y(b))=0$ . The main feature of the new software package is the use of a superconvergent interpolant as described in Muir and Enright, Superconvergent interpolants for the collocation solution of boundary value ordinary differential equations, 1999. The superconvergent technology is employed for  $K=2,3$ , or 4 collocation points. For  $K=1,5,6,7$ , the COLNEWSC executes identical to COLNEW.

COLNEW is written in Fortran 77 whereas COLNEWSC has been adapted to FORTRAN 95 and uses some of FORTRAN 95 features. The main changes to source code are with the error estimation scheme. Other new features in COLNEWSC include the capability to approximate the analytic derivatives with divided differences instead of providing the corresponding user supplied subroutines. COLNEWSC has a simplified user interface.

For more details: See the documentation in the colnewsc.f90 source code.

Corresponding paper : Submitted and currently under review.

# Installation of Software

These instructions assume the user has experience using PuTTY, WinScp and programming Fortran in a Linux environment.

- 1) Select each link for the files:
  - a) colnewsc.f90
  - b) colnewsc\_wrapper.f90
  - c) dblas.f
  - d) one of the test problems
- 2) Right click on the text for each file and save to a device.
- 2) Download WinSCP and transfer each file to a directory on a Linux account
- 3) Download Putty and logon to the Linux account and directory
- 4) At the command line create the executable with:
  - a) `gfortran -w colnewsc.f90 colnewsc_wrapper.f90 dblas.f test_problem.f90`  
and to save output to a temporary file (see description of output)
  - b) `./a.out > temp`

# Test Problems

Available test problems for download:

- 1) swf\_iii.f90 (see description at Test Set for BVP Solvers website)
- 2) tp20.f90 (see description at Test Set for BVP Solvers website)
- 3) tp21.f90 (see description at Test Set for BVP Solvers website)
- 4) measles.f90 (see description at Test Set for BVP Solvers website)
- 5) tmlpde.f90
- 6) swave.f90 (see description at Test Set for BVP Solvers website)
- 7) tp4.f90 (see description at Test Set for BVP Solvers website)

Instructions for writing a new test problem by editing swf\_iii.f90.

1. Beginning with editing the DEFINE\_FCNC module, if a problem defining parameter exists enter the value for EPS.
2. Enter the number of ODES in both NODE and NSOL. These are the same as the user interface currently expects a first order system.
3. Enter the number of boundary conditions at the lefthand endpoint.
4. Enter a data array of the orders of each differential equation. The are all one since a first order representation is currently required. Change the dimension of the array.
5. Enter the ODEs into the F array in the subroutine FSUB
6. If the analytic derivatives for FSUB are provided then enter into the DF array in the subroutine DFDY (see the tp4.f90 test problem for an example without DFDY and DBCDY provided)
7. Enter the boundary conditions in the G array in the subroutine BCSUB
8. If the analytic derivatives for BCSUB are provided then enter into the array DG in subroutine DBCDY
9. Enter an initial guess in the arrays DMVAL and Z in the subroutine GUESS. This routine is also optional.
10. In the main program, enter the lefthand endpoint for A and the righthand endpoint for B
- 11) See documentation if optional parameters are required.



## Description of Output

The output for COLNEWSC is the almost the same as the output for COLNEW for the default value of the optional parameter TRACE. The number of differential equations, orders of each differential equation, lefthand and righthand endpoints, number of collocation points, the tolerance for each component of the numerical solution, maximum number of subintervals are listed first. Then a sequence of meshes and steps in the Newton iteration are outputted. Assuming convergence of the Newton iteration and if a mesh doubling occurs, a primary meshpoint error estimate to check if the user tolerance is satisfied is computed and printed. If the user tolerance is satisfied, then the primary error estimates for the superconvergent continuous solution approximation for the next sequence of meshes, are computed and printed. If the degree of equidistribution for a mesh is below 0.5 then a mesh redistribution is computed using a secondary “rough” error estimate and no primary estimate can be computed. At the end of the output, solution approximations at 11 evenly spaced points on the problem domain are printed.

