

eXtreme Partial Differential Equation Solver

Andrew Sullivan

Department of Physics, Montana State University, Bozeman, MT 59717, USA.

September 24, 2020

This is the documentation of the eXtreme Partial Differential Equation Solver (XPDES) as detailed in arxiv:2009.10614. This includes an overview of each component and the necessary steps to compile and execute the program. Note, you will need the GNU Scientific Library for the linear solvers. For any questions, contact Andrew Sullivan at andrew.sullivan@princeton.edu.

1 Folder Structure

The XPDES project folder is structured as such:

- Code
 - BVP_GENICBC.c: procedurally generated C-file to evaluate boundary conditions at each iteration and set initial guess.
 - BVP_GENsys.c: procedurally generated C-file to evaluate residual and Jacobian at each iteration.
 - BVP_grid.c: Grid adjustment and interpolation.
 - BVP_header.h: Main header file. Provides every BVP function declaration as well as ‘#include’ the procedurally generated field equation and Jacobian function declarations.
 - BVP_LSsolver.c: Modular linear system solver.
 - BVP_main.c: Main Newton-Raphson control loop.
 - BVP_out.c: Output generator to directory ‘Data/BVPout’.
 - BVP.c: Main file to initialize executable variables and call ‘BVP_main()’.
- Documentation
 - MasterDoc.pdf: Master documentation file
 - Linear Solvers.pdf: Description of linear solver algorithms
- Execs
 - BVPEDGB.exe: Main executable for scalar-Gauss-Bonnet gravity.
 - mgen.exe: Executable to generate C header function declarations and Maple field equation exporter code.
 - ibgen.exe: Executable to generate initial and boundary conditions.
 - jgen.exe: Executable to generate system Jacobian.
- Funcs
 - CExport_Funcs.c: C file with procedurally generated function declarations that is #included in the C header file.

- FEout.c: C file with exported field equations for evaluation.
- ICBCout.c: C file with exported initial conditions and boundary conditions for evaluation.
- Gens
 - Maple_Export Folder
 - * This folder contains the Maple field equation calculation and export. Filenames have the general structure “FE_THEORY_export.mw”. Creates: ‘FEout.c’.
 - Export_MapleGen.c: C code to generate C header function declarations and Maple field equation exporter code. Creates: ‘Decl_JSFuncs.c’, ‘Decl_CFuncs.c’, ‘MapleREAD_FEexport.txt’, and ‘MapleREAD_ICBCexport.txt’.
 - Export_ICBC.c: C code to generate C initial conditions and boundary conditions. Creates: ‘BVP_ICBC.c’.
 - Export_Jac.c: Code to procedurally generate the system Jacobian evaluation file. Creates: ‘BVP_sys.c’.
 - MapleREAD_FEexport.txt: Text file with Maple commands to calculate and export the field equations. Executed through a “read” command in the Maple command line.
 - MapleREAD_ICBCexport.txt: Text file with Maple commands to calculate and export initial conditions and boundary conditions. Executed through a “read” command in the Maple command line.
- BVPcompile.sh: Bash compile file
- BVPbatch.sh: Bash executable file

1.1 Compilation Pipeline

To facilitate the variable system of equations, certain files must be procedurally generated to adjust for the number of equations and unknown functions. This requires some back and forth communication between C executables and Maple. Reprocedurally generating all of the C-files is unnecessary as you can simply use the pre-generated files that will be described in the next subsection. For completeness, the step by step process is all done from the terminal using ‘BVPcompileFULL.sh’:

1. Compile full program with ‘BVPcompileFULL.sh’

- (a) The generators must be compiled with the user defined input of the number of field equations ‘nfields’ which will be twice the number of fields. For example, for scalar-Gauss-Bonnet gravity, the 4 metric functions and the scalar field provide 5 fields total, each with a mixed partial derivative field equation associated with it for a total of 10 field equations so nfields = 10. The maximum derivative order must also be supplied. The default is n=2. (Note that the code allows for n=3 but it has not been tested so use at your own risk.)
 - ‘Export_MapleGen.c’ → ‘mgen.exe’
- (b) Generators are executed
 - ‘mgen.exe’ procedurally generates → ‘MapleREAD_FEexport.txt’ and ‘MapleREAD_ICBCexport.txt’ as well as ‘Decl_JSFuncs.c’ and ‘Decl_CFuncs.c’. These are essential as the number of arguments to every declared function will vary depending on the number of field equations.
 - ‘MapleREAD_FEexport.txt’ and ‘MapleREAD_ICBCexport.txt’ are read inside Maple to calculate and export the field equations and the initial and boundary conditions.
 - ‘Decl_JSFuncs.c’ provides the function declarations for the estimation of the number of nonzeros of the Jacobian. They are ‘#included’ into the ‘BVP_header.h’ file.
 - ‘Decl_CFuncs.c’ provides the function declarations for the Newton polynomial calculator ‘NP.calc.c’ (which needs the maximum derivative order), the physical observables, the field equation residuals, the Jacobian partial derivatives, the boundary conditions, and the initial conditions. They are ‘#included’ into the ‘BVP_header.h’ file.

- (c) On the pause prompt...
2. Execute the Maple worksheet in the Maple_Export folder.
 - (a) Maple will “read” ‘MapleREAD_FEexecute.txt’ and procedurally generate \rightarrow ‘Defn_FEout.c’. These are the function declarations for the declared functions above.
 - (b) Maple will “read” ‘MapleREAD_ICBCexecute.txt’ and procedurally generate \rightarrow ‘Defn_ICBCout.c’. These are the function declarations for the declared functions above.
 3. Press enter terminal window to continue compiling the full program through gcc \rightarrow ‘BVPNAME.exe’
 - Automatically, ‘Export_NNZGen.c’ is compiled into ‘nnzgen.exe’ which uses the defined Jacobian structure functions from ‘Decl_JSFuncs.c’ and ‘Defn_JSFuncs.c’ to estimate the number of the nonzeros of the Jacobian when given a grid size so that the algorithm can calculate approximately how much memory to allocate.
 - Automatically, ‘Export_CGen.c’ uses ‘CodeGen_GENNNZ.c’ and is compiled into ‘cgen.exe’ which procedurally generates the initial conditions, boundary conditions, and Jacobian evaluation C-files in the ‘Code’ folder. ‘BVP_GENICBC.c’ and ‘BVP_GENsys.c’ respectively.
 - Then ‘BVPNAME.exe’ is compiled to the ‘Execs’ folder.
 4. Execute program with ‘BVPbatch.sh’ to determine inputs. WARNING: this will also delete and remake the ‘Data’ directory where the solutions are stored.

1.2 Compilation Shortcut

The field equation dependent files that are required upon compilation are:

- ‘/Code/BVP_GENICBC.c’
- ‘/Code/BVP_GENsys.c’
- ‘/Funcs/Decl_CFuncs.c’
- ‘/Funcs/Decl_JSFuncs.c’
- ‘/Funcs/Defn_FEout.c’
- ‘/Funcs/Defn_ICBCout.c’
- ‘/Funcs/Defn_JSout.c’

To compile, copy the above files from the labeled folder in the ‘XPDES’ directory into their respective location (either ‘/Code’ or ‘/Funcs’) and from the command line execute ‘BVPcompile.sh’. This will compile and place the ‘BVPNAME.exe’ executable in the ‘/Execs’ folder.