Homotopy Path-Following with EasyHomotopy: Solving Nonlinear Equations for Economic Models*

Karl Schmedders

KSM-MEDS, Northwestern University (e-mail: k-schmedders@kellogg.nwu.edu)

Summary. HOMPACK is a large collection of FORTRAN subroutines for solving various nonlinear systems of equations by homotopy methods. This note accompanies the program EasyHomotopy for solving nonlinear equations in economics, which is a user-friendly adaptation of a small subset of the HOMPACK routines. The purpose of this note is to give an overview of EasyHomotopy and to enable any economist who wants to use homotopy path-following, to do so quickly and without having to know HOMPACK in much detail. As a matter of fact, a new user needs to change only two files of EasyHomotopy in order to solve a desired system of equations.

1 HOMPACK

HOMPACK is a large collection of FORTRAN subroutines for solving various nonlinear systems of equations by homotopy methods. There are subroutines for solving fixed-point problems $f(x) = x$, subroutines for finding zeros of functions, that is, for solving $f(x) = 0$, and there are subroutines for homotopy path-following, that is, for tracking solutions to $H(x, \eta) = 0$ as η changes from 0 to 1. In addition, there are special subroutines for both dense and sparse Jacobian matrices and for the different types of algorithms that can be used for path-following. This comprehensive software package exists both in FORTRAN77 and in FORTRAN90, and so it is also found under the names HOMPACK77 and HOMPACK90, respectively. Watson et al. (1987) give a detailed description of HOMPACK.

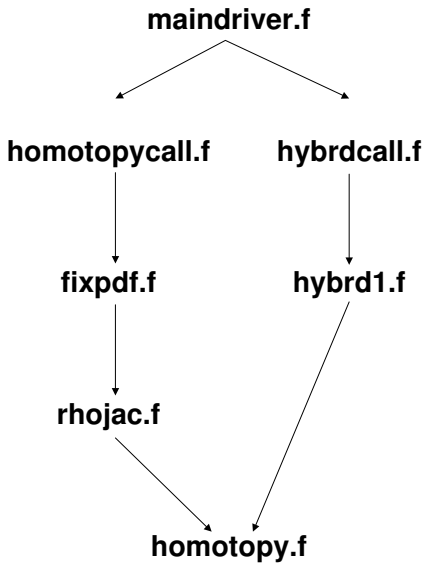
*I am grateful to Ken Judd, Annette Krauss, and Sevin Yeltekin for helpful comments.

While HOMPACK offers a tremendous selection of powerful solution procedures, the vast number of subroutines can make its first use a daunting task. In order to significantly reduce the set-up cost of starting to use HOMPACK I have selected a small set of subroutines from HOMPACK77 that are sufficient to do homotopy path-following and have embedded them into a framework of a few other subroutines that is supposed to minimize the necessary knowledge of HOMPACK for a new user. For lack of a better name this new package is called EasyHomotopy.

2 EasyHomotopy

The purpose of EasyHomotopy is to numerically solve a nonlinear system of equations, $f(x) = 0$, by homotopy path-following. The user must specify homotopy equations $H(x, \eta) = 0$ so that $H(x, 1) = f(x)$ and with a starting point x^0 such that $H(x^0, 0) = 0$. For an excellent introduction to homotopy theory see Garcia and Zangwill (1981). Eaves and Schmedders (1999) provide an introduction to using homotopies in economics.

EasyHomotopy consists of seven FORTRAN files. The flow chart below illustrates the calling sequences between the main program in the first file and the subroutines in the other six files in order to track the homotopy path implicitly defined by $H(x, \eta) = 0$.



The main program **maindriver** specifies the starting point for the homotopy and then calls the subroutine **homotopycall**. This subroutine calculates the computational error at the starting point, sets the necessary parameters for HOMPACK and then calls the HOMPACK driver **fixpdf.** This subroutine is the driver within the many subroutines in the file fixpdf.f, it organizes all the components of the path-following algorithm. The only file I have taken out of fixpdf.f and stored separately for easier access is **rhojac**. In rhojac the Jacobian of the function H is calculated by a finite-difference method. For this purpose rhojac calls the subroutine **homotopy** which contains the calculations for the equations that are to be solved. After the path-following algorithm has successfully tracked the homotopy path from $\eta = 0$ to $\eta = 1$ it returns with its solution to $H(x, 1) = 0$ to homotopycall. This subroutine calculates the numerical error at the final homotopy solution and then returns with the solution to $H(x, 1) = 0$ to maindriver. The main program contains a user-controlled indicator variable indicating whether the program should terminate or should attempt to reduce the computational error at the solution found by the path-following routine. In the latter case the main program calls **hybrdcall** which in turn calls **hybrd1**. The purpose of hybrd1 is to take the solution found by HOMPACK as a starting point for solving the desired system $f(x) = 0$. hybrd1 is a nonlinear equation solver based on a modification of Powell's hybrid method. The file hybrd1.f contains hybrd1 and all subroutines necessary for calculations. Once hybrd has found a solution it returns to hybrdcall. This subroutine calculates the numerical error at the final solution.

In the following I explain what a user must do in order to use EasyHomotopy for solving a desired system of equations. The package is set up in such a way that the user only needs to adapt two programs for his purposes, namely maindriver and homotopy. In order to make any changes easy, the variables in these two subroutines are clearly differentiated into those that are needed for any application, and those that are specific to the economic model and so need to be replaced accordingly. Both routines contain comments to help new users with the programming of the code for their own specific problems. In addition, I explain the purpose of homotopycall, rhojac, and hybrdcall, because a user may occasionally want to change some of the parameters in these subroutines. In order to illustrate the description of the various program parts I am going to refer to two specific examples. Example 1 deals with the simple example of computing an equilibrium for a Cournot duopoly. Example 2 refers to the computation of equilibria for finance economies with transaction fees, see Schmedders (2001, Section 3.1).

maindriver

The main program maindriver is one of the two routines that a user needs to adapt for solving his own system of equations. In maindriver the user must provide the following information.

1. The user must set the parameter N, which is the number of equations to be solved.

2. The user must declare the variables for all economic parameters, declare these variables as common

variables if necessary, and initialize them.

3. The user must calculate a starting point for the homotopy and store it in the array $XVEC$ so that it will be reported to homotopycall.

4. The user must set the indicator variable IND. If he wants to use the hybrd solver after using the path-following routine (possibly with a loose stopping criterion) then IND must be set to 1. If the user wants to use only the path-following routine then IND must be set to a different value.

The program then calls homotopycall and maybe, depending on the value of IND, subsequently hybrdcall. The typical user will be interested in the solution of the system of equations and so may want to print various numbers either on the screen or into an output file at the end of maindriver.

Example 1. Consider a Cournot duopoly with linear demand. The inverse demand function is denoted by $p = a - b(x_1 + x_2)$ with x_i being the production quantity of firm $i = 1, 2$. The two firms face constant marginal cost of production c_1 and c_2, respectively. The firms' reaction functions are

$$x_1(x_2) = \frac{a - c_1 - bx_2}{b} \quad \text{and} \quad x_2(x_1) = \frac{a - c_2 - bx_1}{b}.$$

Of course, one could easily determine the Cournot equilibrium analytically by solving a linear system of two equations in two unknowns. But this simple example will nicely illustrate how to use EasyHomotopy to compute a solution to a system of equations.

Computing a Cournot equilibrium requires solving the following equations.

$$\begin{aligned} x_1 - \frac{a - c_1 - bx_2}{b} &= 0 \\ x_2 - \frac{a - c_2 - bx_1}{b} &= 0 \end{aligned}$$

Define a homotopy function with a homotopy parameter η as follows.

$$\begin{aligned} x_1 - \eta\frac{a - c_1 - bx_2}{b} &= 0 \\ x_2 - \eta\frac{a - c_2 - bx_1}{b} &= 0 \end{aligned}$$

The corresponding main program maindriver consists of the following code.

```
      PROGRAM MAINDRIVER
      IMPLICIT NONE
C ECONOMIC MODEL: COURNOT MODEL WITH LINEAR DEMAND
```

```
C NUMBER OF EQUATIONS AND VARIABLES
 INTEGER N
 PARAMETER (N=2)
C VARIABLES NEEDED FOR CALLS TO SUBROUTINES
 DOUBLE PRECISION HVEC(N),ETA,XVEC(N)
 INTEGER HYBRDFLAG,IND
C VARIABLES NEEDED FOR ECONOMIC MODEL (USER-SPECIFIC)
 DOUBLE PRECISION A,B,C(2)
C THE HOMOTOPY PARAMETER IS A COMMON VARIABLE
 COMMON/HOMOTOPYPARAMETER/ETA
C COMMON VARIABLES USED IN THE SUBROUTINE HOMOTOPY (USER-SPECIFIC)
 COMMON/DPVAR/A,B,C
************************************************************
C DATA FOR ECONOMIC MODEL
 A=10.0D00
 B= 1.0D00
 C(1)=1.0D00
 C(2)=2.0D00
************************************************************
C STARTING POINT FOR HOMOTOPY
 ETA=0.0D00
 XVEC(1)=0.0D00
 XVEC(2)=0.0D00
************************************************************
C IND=1 MEANS HYBRD IS CALLED AFTER HOMOTOPY
C OTHERWISE HYBRD IS NOT CALLED
 IND=1
C CALL TO HOMPACK AND HYBRD
 CALL HOMPACKCALL(N,XVEC)
 IF (IND .EQ. 1) THEN
 ETA=1.0D00
 CALL HYBRDCALL(N,XVEC)
```

```
 ENDIF
*********************************************************
C PRINTING OF RESULTS
 WRITE(6,100) XVEC(1),XVEC(2)
 100 FORMAT(/' COURNOT QUANTITIES X(1),X(2):',/,3F14.5)
 PAUSE
 END
```

First, the user must specify the number of homotopy equations in the parameter N. Here $N = 2$. Second, the user must declare the variables that are specific to the economic model, here A, B, and C, and subsequently make them into common variables so that the subroutine homotopy has access to these variables. In the main part of the program the user needs to specify values for the economic parameters (here $A = 10, B = 1, C(1) = 1, C(2) = 2$) and provide the starting point for the homotopy. The homotopy parameter is always set to $ETA = 0$ and in this simple model the starting point equals $x_1 = x_2 = 0$, which is stored in the variable $XVEC$. Finally, the user needs to decide whether he wants to just do homotopy path-following or whether he also wants to use the hybrd solver to reduce the numerical error at the optimal solution. For this purpose the user must set the indicator variable IND to the appropriate value.

Example 2. Exogenous parameters that characterize the finance economy with transaction fee (see Schmedders (2001) for a detailed description of the model) are the security payoffs $ASSET$, the endowments $ENDOW$ and $EZERO$, the probabilities $PROB$ of the states in the second period, agents' coefficients of risk aversion $GAMMA$, and finally the transaction fee $TRAFEE$ for and the strike price $STRIKE$ of the option. All these variables are declared and also used as common variables, because the subroutine homotopy needs all these variables for the calculations of the homotopy function. In addition, the implemented homotopy needs starting values for the asset prices, which are stored in the common variable $QSTART$. Moreover, the homotopy has been defined in such a way, that the starting point is very simple, namely the first three values are just equal to the respective values of $QSTART$, and all other variables equal zero.

homotopy

The subroutine homotopy contains the homotopy function H, that is, this subroutine calculates the vector $H(x, \eta)$. Once the homotopy has found a solution for $H(x, 1) = f(x) = 0$ the solver hybrd1 (if used) also calls this subroutine. The user needs to do the following programming in homotopy.

1. All relevant parameters have been set in maindriver and declared as common variables. The identical common declaration must be provided in homotopy in order to have access to these variable values.

2. The user must provide the calculations of $H(x, \eta)$ in the vector H.

In order to program the calculations for the vector H it may be helpful to "translate" the array X of endogenous variables into variables with meaningful names.

Example 1 cont'd. The code for the subroutine homotopy is as follows.

```
 SUBROUTINE HOMOTOPY(N,X,H,HYBRDFLAG)
C EVALUATE HOMOTOPY(X) AND RETURN IN THE VECTOR H.
C ECONOMIC MODEL: COURNOT MODEL WITH LINEAR DEMAND
 IMPLICIT NONE
C VARIABLES THAT ARE ALWAYS NEEDED FOR HOMOTOPY
 INTEGER N,HYBRDFLAG
 DOUBLE PRECISION X(N),H(N),ETA
C VARIABLES THAT ARE SPECIFIC TO THE ECONOMIC MODEL (USER SPECIFIC)
 DOUBLE PRECISION A,B,C(2)
C COMMON VARIABLE ETA (THE HOMOTOPY PARAMETER) IS ALWAYS NEEDED FOR THE HOMOTOPY
 COMMON/HOMOTOPYPARAMETER/ETA
C COMMON VARIABLES SPECIFIC TO THE ECONOMIC MODEL (USER SPECIFIC)
 COMMON/DPVAR/A,B,C
*********************************************************
C HOMOTOPY EQUATIONS
 H(1)=X(1)- ETA* ( (A-C(1))-B*X(2) )/(2.0*B)
 H(2)=X(2)- ETA* ( (A-C(2))-B*X(1) )/(2.0*B)
 END
```

Again the user needs to declare the variables $A, B, C(2)$ needed for the Cournot model and also specify them as common variables just as he did in the main program, so that the subroutine homotopy has access to the values of these variables. In the main part of the subroutine the user must calculate the value of homotopy function, that is, the values of $H(1)$ and $H(2)$ for the two homotopy equations that were defined

above.

Example 2 cont'd. At the beginning of the subroutine I translate the endogenous variable array X into the asset price variables Q, the portfolio variables $THETA$ and the variables $ALLON$ and $ALSHO$ that are used for the definition of the option variables via the change of variable trick of Garcia and Zangwill (1981). Then the calculations for the homotopy term $H(x, \eta)$ follow and the term is stored in the vector H. The homotopy equations consists of the market-clearing equations and the first-order conditions of the two agents.

homotopycall

The user does not need to edit this subroutine. It sets some parameters for the homotopy path-following. Most importantly, the value $IFLAG = -2$ indicates to fixpdf to perform homotopy path-following. (fixpdf has other capabilities, see the long description in the file fixpdf.f.) The parameter $ARCRE$ determines an error tolerance for the following of the homotopy path, and the parameter $ANSRE$ specifies an error tolerance for the final solution at $\eta = 1$. If a user is only interested in the final solution then these values can be set relatively high (say to the order of 10^{-5}), and then a final call to hybrd (set IND=1 in maindriver) ensures a low numerical error at the final solution. But if a user is interested in following the path very closely, then the parameter $ARCRE$ should be sufficiently small. And if the user does not want to use hybrd, then the parameter $ANSRE$ should also be set sufficiently small.

Before homotopycall calls fixpdf it calculates the numerical error at the starting point and prints it on the screen. If the user does not want this information printed or calculated then he should delete the corresponding program statements. After fixpdf returns with a solution, its properties are printed on the screen. A "FLAG" of 1 indicates that HOMPACK tracked the path successfully until $\eta = 1$. All other values indicate problems, for a description of other values of FLAG the user should read fixpdf.f.

rhojac

This subroutine calculates the Jacobian of the homotopy function by a two-sided finite-difference formula. The user does not need to edit this subroutine unless he decides to calculate the Jacobian differently.

hybrdcall

The user does not need to edit this subroutine. It sets some parameters for hybrd1 and then calls this solver. Afterwards it calculates the maximum numerical error at the final solution and reports it on the

screen. (Note, that this error is not the relative error users should typically be interested in when reporting numerical errors.) This report is followed by a "PAUSE" command that a user may want to delete.

fixpdf.f and hybrd1.f

Both fixpdf.f and hybrd1.f were downloaded from www.netlib.org. I did not make any changes to hybrd1.f. With the exception of changing a single parameter, separating the subroutine rhojac and adapting it to finite-difference calculations, and adding a missing subroutine I made no changes to fixpdf. The only parameter I changed in fixpdf was $LIMITD$, which is the maximum number of steps fixpdf takes along the homotopy path. If a user finds that the homotopy does not reach $\eta = 1$ but instead terminates with $FLAG = 3$, then the maximum number of steps along the homotopy path has been reached. A first approach to resolve this issue may be to increase $LIMITD$ appropriately.

3 A Word of Caution

A user cannot expect to compute a solution to any system of equations by just simply defining any homotopy function and applying EasyHomotopy to solve the defined system. The theory of solving nonlinear systems of equations with homotopy methods is nontrivial (see Garcia and Zangwill (1981), Eaves and Schmedders (1999)) and the development of a theoretically sound homotopy algorithm requires great care. A new user should first carefully develop a homotopy with a reasonable starting point before he starts to implement it with EasyHomotopy.

References

[1] Eaves, B.C., Schmedders, K.: General equilibrium models and homotopy methods. Journal of Economic Dynamics and Control **23**, 1249–1279 (1999)

[2] Garcia, C., Zangwill, W.: Pathways to solutions, fixed points, and equilibria. Englewood Cliffs: Prentice Hall 1981

[3] Schmedders, K.: Monopolistic security design in finance economies. Economic Theory **18**, 37–72 (2001)

[4] Watson, L.T., S.C. Billups, and A.P. Morgan: HOMPACK: A suite of codes for globally convergent homotopy algorithms. ACM Transactions on Mathematical Software **13**, 281–310 (1987)