A Simple Users Guide of BVPh 1.0

The `BVPh` 1.0 is a Mathematica package for highly nonlinear boundary-value problems with multiple solutions and singularity, governed by nth-order nonlinear ordinary different equations (ODEs)

$$\mathcal{F}[u, z] = 0,$$

or eigen-value problems governed by the nonlinear ODEs

$$\mathcal{F}[u, z, \lambda] = 0,$$

subject to n linear multipoint boundary conditions, where $\mathcal{F}$ is a nonlinear operator, $u(z)$ is a continuous function, z is an independent variable in either a finite domain $z \in [a, b]$ or an infinite domain $z \in [a, +\infty)$, λ is an eigenvalue, respectively. This package is valid even for some nonlinear partial difference equations (PDEs) related to steady or unsteady boundary-layer flows.

For details, please refer to Liao's book: *Liao, S.J., Homotopy Analysis Method in Nonlinear Different Equations. Springer (2011).*

Key modules

`BVPh` The module `BVPh[k_,m_]` gives the kth to mth-order homotopy approximations of a nonlinear boundary-value problem (when `TypeEQ = 1`) or a nonlinear eigenvalue problem (when `TypeEQ = 2`), as defined above. It is the basic module. For example, `BVPh[1,10]` gives the 1st to 10th-order homotopy-approximations. Thereafter, `BVPh[11,15]` further gives the 11th to 15th-order homotopy-approximations.

`iter` The module `iter[k_,m_,r_]` provides us homotopy-approximations of the kth to mth iteration by means of the rth-order iteration formula (7.51). It calls the basic module `BVPh`. For example, `iter[1,10,3]` gives homotopy-approximations of the 1st to 10th iteration by the 3rd-order iteration formula. Furthermore, `iter[11,20,3]` gives the homotopy-approximations of the 11th to 20th iterations. For eigenvalue problems, the initial guess of eigenvalue is determined by an algebraic equation. Thus, if there are more than one initial guesses of eigenvalue, it is required to choose one among them by inputting an integer, such as 1 or 2, corresponding to the 1st or the 2nd initial guess of the eigenvalue, respectively. If the convergence-control parameter c_0 is unknown at the beginning of iteration, curves of squared residual of governing equation at the up-to 3rd-order approximations versus c_0 are given at the 1st iteration, in order to choose a proper value of c_0. This value of c_0 will be renewed after `Nupdate` times iterations. The iteration stops when the squared residual of governing equation is less than a critical value `ErrReq`, whose default is 10^{-20}.

`GetErr` The module `GetErr[k_]` gives the squared residual of the governing equation at the kth-order homotopy-approximation gained by the module `BVPh`, or at the kth-iteration homotopy-approximation obtained by the module `iter`. Note that, `error[k]` provides the residual of governing equation at the kth-order homotopy-approximation gained by `BVPh`, or at the kth-iteration homotopy-approximation obtained by `iter`, and `Err[k]` gives the averaged squared residual of the governing equation at the kth-order homotopy-approximation gained by `BVPh`, and `ERR[k]` gives the averaged squared residual of the governing equation at the kth-iteration homotopy-approximation obtained by `iter`, respectively.

`GetMin1D` The module `GetMin1D[f_,x_,a_,b_,Npoint_]` searches for the local minimums of a real function $f(x)$ in the interval $x \in [a, b]$ by means of dividing the interval $[a, b]$ into `Npoint` equal parts. If `Npoint` is large enough, it gives all local minimums with the corresponding position x. In general, `Npoint = 20` is suggested. This module is often used to search for the optimal convergence-control parameter c_0, or multiple solutions of a nonlinear boundary-value problem. It calls the module `GetMin1D0[f_,x_,a_,b_,Npoint_]`.

`GetMin2D` Using `GetMin2D[f_,x_,a_,b_,y_,c_,d_,Npoint_]`, we can search for the local minimums of a real function $f(x, y)$ in the interval $a \leq x \leq b$, $c \leq y \leq d$ by dividing it into `Npoint` $\times$ `Npoint` equal parts. If `Npoint` is large enough, it gives all local minimums with the corresponding position (x, y). In general, `Npoint = 20` is suggested. This module is often used to search for multiple solutions of a nonlinear boundary-value problem. It calls the module `GetMin2D0[f_,x_,a_,b_,c_,d_,Npoint_]`.

`hp` The module `hp[f_,m_,n_]` gives the $[m, n]$ homotopy-Padé approximation of the homotopy-approximation `f`, where `f[0],f[1],f[2]` denotes the zeroth, first and 2nd-order homotopy-approximation of `f`. For details about the homotopy-Padé approximation, please refer to Chapter 2.

Control parameters

`TypeEQ` A control parameter for the type of governing equation: `TypeEQ = 1` corresponds to a nonlinear boundary-value equation, `TypeEQ = 2` corresponds to a nonlinear eigenvalue problem, respectively.

`TypeL` A control parameter for the type of base functions: `TypeL = 1` corresponds to polynomial (7.52) or Chebyshev polynomial (7.53), and `TypeL = 2` corresponds to a trigonometric approximation or a hybrid-base approximation described in §7.2.3, respectively. It is valid only for boundary-value/eigenvalue problems in a finite interval $z \in [0, a]$, where $a > 0$ is a constant.

`ApproxQ` A control parameter for approximation of solutions. When `ApproxQ = 1`, the right-hand side term of all high-order deformation equations is approximated by Chebyshev polynomial (7.53), or the hybrid-base approximation described in § 7.2.3. When `ApproxQ = 0`, there is no such kind of approximation. When the module `iter` is employed, `ApproxQ = 1` is automatically assigned. For the `BVPh` (version 1.0), this parameter is valid only for boundary-value/eigenvalue problems in a finite interval $z \in [0, a]$, where $a > 0$ is a constant.

`TypeBase` A control parameter for the type of base functions to approximate solutions: `TypeBase = 0` corresponds to the Chebyshev polynomial (7.53), `TypeBase = 1` corresponds to the hybrid-base (7.58) with the odd Fourier approximation (7.63), `TypeBase = 2` corresponds to the hybrid-base (7.58) with the even Fourier approximation (7.60), respectively. This parameter is valid only when `ApproxQ = 1` for boundary-value/eigenvalue problems in a finite interval $z \in [0, a]$.

`Ntruncated` A control parameter to determine the positive integer $N_t > 0$, corresponding to the number of truncated terms in (7.53), (7.60) and (7.63). The larger the number N_t, the better the approximations, but the more CPU times. It is valid only when `ApproxQ = 1` for boundary-value/eigenvalue problems in a finite interval $z \in [0, a]$. The default is 10.

`NtermMax` A positive integer used in the module `truncated`, which ignores all polynomial terms whose order is higher than `NtermMax`. The default is 90.

`ErrReq` A critical value of squared residual of governing equation to stop the iteration of the module `iter`. The default is 10^{-10}.

`NgetErr` A positive integer used in the module `BVPh`. The squared residual of governing equation is calculated when the order of approximation divided by `NgetErr` is an integer. The default is 2.

`Nupdate` A critical value of the times of iteration to update the convergence-control parameter c_0. The default is 10.

`Nintegral` Number of discrete points with equal space, which are used to numerically calculate the integral of a function. It is used in the module `int`. The default is 50.

`ComplexQ` A control parameter for complex variables. `ComplexQ = 1` corresponds to the existence of complex variables in governing equations and/or boundary conditions. `ComplexQ = 0` corresponds to the nonexistence of such kind of complex variables. The default is 0.

`c0L` A real number to determine the interval of the convergence-control parameter c_0 for plotting curves of the squared residual of the governing equation versus c_0 in the module `iter`. The default is -2, corresponding to the interval $-2 \le c_0 \le 0$. The value of `c0L` can be positive, such as `c0L = 1`, corresponding to the interval $0 \le c_0 \le 1$.

Input

`f[z_,u_,lambda_]` The governing equation, corresponding to $\mathcal{F}[z, u] = 0$ for nonlinear boundary-value problems in either a finite interval $z \in [0, a]$ or an infinite interval $z \in [b, +\infty)$, or corresponding to $\mathcal{F}[z, u, \lambda] = 0$ for nonlinear eigenvalue problems in a finite interval $z \in [0, a]$, where $a > 0, b \ge 0$ are bounded constants. Note that, `lambda` denotes the eigenvalue for nonlinear eigenvalue problems, but has no meanings at all for nonlinear boundary-value problems.

`BC[k_,z_,u_,lambda_]` The kth boundary condition, corresponding to either $\mathcal{B}_k[z, u] = 0$ for nonlinear boundary-value problems, or $\mathcal{B}_k[z, u, \lambda] = 0$ for nonlinear eigenvalue problems, respectively, where $0 \le k \le n$. Note that, `lambda` denotes the eigenvalue for nonlinear eigenvalue problems, but has no meanings at all for nonlinear boundary-value problems.

`u[0]` The initial guess $u_0(z)$.

`L[f_]` The auxiliary linear operator. For boundary-value/eigenvalue problems in a finite interval $z \in [0, a]$, the auxiliary linear operator (7.37) is automatically chosen when `TypeL=1`, and the auxiliary linear operator (7.40) or (7.41) is used otherwise. For boundary-value problems in an infinite interval $z \in [b, +\infty)$, where $b \ge 0$ is a bounded constant, one may choose either (7.42) for exponentially decaying solutions or (7.44) for algebraically decaying solutions, respectively. In any cases, the auxiliary linear operator must be clearly defined and properly chosen.

`H[z_]` The auxiliary linear operator. The default is `H[z_]:=1`.

`OrderEQ` The order of the boundary-value equation $\mathcal{F}[z,u] = 0$ or the eigenvalue equation $\mathcal{F}[z,u,\lambda] = 0$.

`zL` The left end-point of the interval $z \in [a,b]$, corresponding to $z = a$. For example, `zL=1` corresponds to $a = 1$. For boundary-value/eigenvalue problems in a finite interval $z \in [0,a]$, `zL = 0` is automatically assigned. The default is 0.

`zR` The right end-point of the interval $z \in [a,b]$, corresponding to $z = b$. For example, `zR=1` corresponds to $b = 1$. For boundary-value problems in an infinite interval, `zR = infinity` must be used for `BVPh` (version 1.0).

Output

`U[k]` The kth-order homotopy-approximation of $u(z)$ given by the basic module `BVPh`.

`V[k]` The kth-iteration homotopy-approximation of $u(z)$ given by the iteration module `iter`.

`Uz[k]` The kth-order homotopy-approximation of $u'(z)$ given by the basic module `BVPh`.

`Vz[k]` The kth-iteration homotopy-approximation of $u'(z)$ given by the iteration module `iter`.

`Lambda[k]` The kth-order homotopy-approximation of the eigenvalue λ given by the basic module `BVPh`.

`LAMBDA[k]` The kth-iteration homotopy-approximation of the eigenvalue λ given by the iteration module `iter`.

`error[k]` The residual of governing equation given by either the kth-order homotopy-approximation (obtained by the basic module `BVPh`) or the kth-iteration homotopy-approximation (obtained by the iteration module `iter`).

`Err[k]` The averaged squared residual of governing equation given by the kth-order homotopy-approximation (obtained by the basic module `BVPh`).

`ERR[k]` The averaged squared residual of governing equation given by the kth-iteration homotopy-approximation (obtained by the iteration module `iter`).

Global variables

All control parameters and output variables mentioned above are global. Except theses, the following variables and parameters are also global.

`z` The independent variable z.

`u[k]` The solution $u_k(z)$ of the kth-order deformation equation.

`lambda[k]` A constant variable, corresponding to λ_k.

`delta[k]` A function dependent upon z, corresponding to the right-hand side term $\delta_k(z)$ in the high-order deformation equation.

`L` The auxiliary linear operator $\mathcal{L}$.

`Linv` The inverse operator of $\mathcal{L}$, corresponding to $\mathcal{L}^{-1}$.

`nIter` The number of iteration, used in the module `iter`.

`sNum` A positive integer, which determines which initial guess λ_0 of eigenvalue is chosen when there exist multiple solutions of λ_0.