Class 17. ODEs, Part 4 (2-pt BVPs)

Two-point Boundary Value Problems

- *NRiC* §17.
- BCs specified at two or more points, e.g., start and end.
- For IVP, just integrate away.
- For 2-pt BVP, must make a free choice of unknown BVs at initial point, then integrate away. But solution will almost certainly *not* satisfy other BCs at end.
- Strategy: Use information about how much the other BVs "missed" to iteratively improve initial guess.

  $\Longrightarrow$ Techniques are all iterative (and expensive).

Notation

- Denote standard system as:

$$\frac{dy_i(x)}{dx} = g_i(x, y_1, ..., y_N) \qquad i = 1, ..., N.$$

- At x_1, the solution is supposed to satisfy:

$$B_{1j}(x_1, y_1, ..., y_N) = 0 \qquad j = 1, ..., n_1.$$

- At x_2, it is supposed to satisfy:

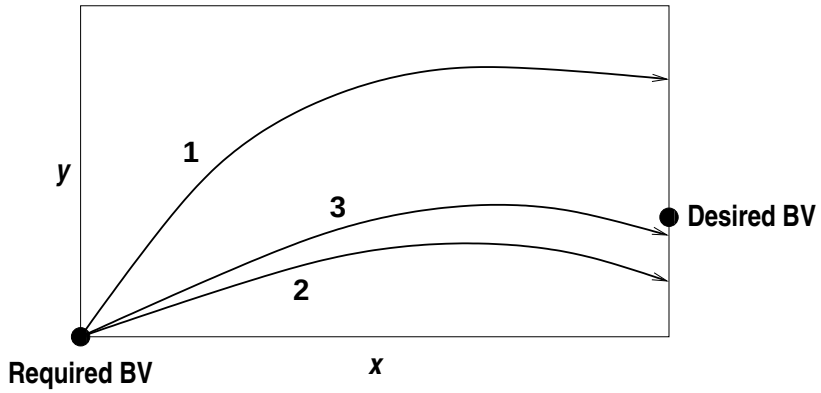
$$B_{2k}(x_2, y_1, ..., y_N) = 0 \qquad k = 1, ..., n_2,$$

  where $n_2 = N - n_1$.

Two Basic Techniques

Shooting method

1. Begin at x_1.
2. Guess values for free BCs (n_2 values).
3. Integrate as IVP to x_2.
4. Adjust n_2 guesses to get closer to BVs at x_2.

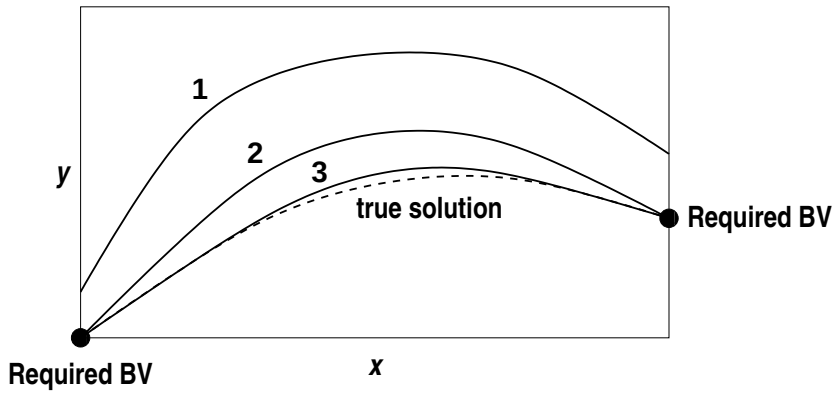


- Heart of technique: system of iteratively improving guesses.

  $\Longrightarrow$ Multi-D root finding.

Relaxation method

1. Replace ODEs by finite-difference equations on mesh from x_1 to x_2.
2. Guess solution on this mesh.
3. Mathematically, FDEs are just algebraic relations between unknowns. Use iterative technique to relax this solution to true solution.



- Relaxation very powerful for smooth solutions, or ODEs that must be solved many times with different parameter values. Also good when ODEs have extraneous solutions, i.e., stiff equations.
- *NRiC*: "Shoot first, relax later."

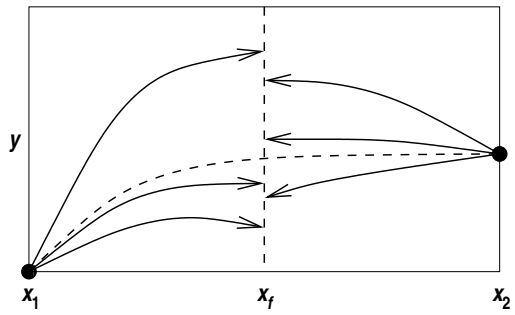
2-pt BVP: Shooting Method

Procedure (*NRiC* §17.1):

1. At x_1, must specify N starting values for y_i, $i = 1, ..., N$.
   - n_1 values given by BC at x_1.
   - $\therefore n_2 = N - n_1$ values can be chosen freely.
2. Represent the free values as a vector $\mathbf{V}$ of dimension n_2 (actually, $\mathbf{V}$ represents schematically any parameter values that specify unknown BVs).
3. Now integrate to x_2.
4. Define "discrepancy vector" $\mathbf{F}$ of dimension n_2, where
   $$F_k = B_{2k}(x_2, \mathbf{y}) \qquad k = 1, ..., n_2.$$
   - We want to find $\mathbf{V}$ that zeroes $\mathbf{F}$.
5. Solve n_2 linear equations:
   $$\mathbf{J}\,\delta\mathbf{V} = -\mathbf{F},$$
   where $J_{ij} \equiv \partial F_i/\partial V_j$ is the Jacobian matrix.
   - This is the globally convergent Newton's method (*NRiC* §9.7).
6. Then $\mathbf{V}_{\rm new} = \mathbf{V}_{\rm old} + \delta\mathbf{V}$.
7. Use $\mathbf{V}_{\rm new}$ to solve ODEs again as IVP, recompute $\mathbf{F}$, and iterate again until $|\mathbf{F}| < \varepsilon$, the convergence criterion.

- Infeasible to compute Jacobian analytically. Instead evaluate differences numerically, i.e.,
  $$\frac{\partial F_i}{\partial V_j} \simeq \frac{F_i(V_1, ..., V_j + \Delta V_j, ..., V_{n_2}) - F_i(V_1, ..., V_j, ..., V_{n_2})}{\Delta V_j},$$
  i.e., **solve IVP n_2 times**, varying each component of $\mathbf{V}$ by $\pm\Delta\mathbf{V}$ each time to build up Jacobian (recall $F_i(V_1, ..., V_{n_2})$ already computed in step 4).
- Overall procedure requires $n_2 + 1$ solutions to ODEs per iteration.
- For linear systems, one iteration is enough.
- For nonlinear systems, many (say M) iterations may be required to converge $\Longrightarrow$ $M \times (n_2 + 1)$ solutions of ODEs!
- $\therefore$ need efficient integrator... (*NRiC* routine `shoot()` uses `odeint()`).

- NOTE: Can also shoot to a fitting point x_f between x_1 and x_2 (*NRiC* §17.2; specify known points at x_1 and x_2, choose the rest, integrate in both directions, and require that $\mathbf{y}(x_f)$ match for both integrations). Useful for singular BC(s) and/or domain point(s); integrate away from these.



2-pt BVP: Relaxation Method

Procedure (*NRiC* §17.3):

1. Replace ODEs

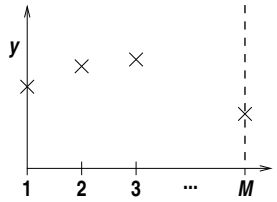
$$\frac{d\mathbf{y}(x)}{dx} = \mathbf{g}(x, \mathbf{y})$$

with finite difference equations (FDEs) on a grid:[1]

$$\frac{\mathbf{y}_k - \mathbf{y}_{k-1}}{x_k - x_{k-1}} = \mathbf{g}\left(\frac{x_k + x_{k-1}}{2}, \frac{\mathbf{y}_k + \mathbf{y}_{k-1}}{2}\right),$$

where $\mathbf{y}_k$ refers to the entire set of dependent variables $y_1, y_2, \ldots, y_N$ at point x_k.

- Here x_k and the components of $\mathbf{y}_k$ are discrete values of independent and dependent variables at "mesh points."



- For M mesh points and N coupled equations, have $M \times N$ $\mathbf{y}_k$ components to solve for. Approximate the set of N first-order ODEs by

$$0 = \mathbf{E}_k \equiv \mathbf{y}_k - \mathbf{y}_{k-1} - (x_k - x_{k-1})\mathbf{g}_k(x_k, x_{k-1}, \mathbf{y}_k, \mathbf{y}_{k-1}), \qquad k = 2, 3, \ldots, M.$$

Here $\mathbf{g}_k$ can be evaluated using information from both points k, $k-1$.

[1] This is not a unique representation. We could, for example, evaluate $\mathbf{g}$ at $(x_k, \mathbf{y}_k)$ and $(x_{k-1}, \mathbf{y}_{k-1})$, then take the average.

- This is $(M-1)N$ equations; get remaining equations from the boundary conditions:

$$\begin{aligned} 0 = \mathbf{E}_1 &\equiv \mathbf{B}(x_1, \mathbf{y}_1) \\ 0 = \mathbf{E}_{M+1} &\equiv \mathbf{C}(x_M, \mathbf{y}_M) \end{aligned}$$

2. The "solution" of the FDE problem consists of a set of variables $y_{j,k}$. Need to guess starting values for all $y_{j,k}$. We then determine increments $\Delta y_{j,k}$ such that $y_{j,k} + \Delta y_{j,k}$ is an improved approximation to the solution.

3. To get increments, expand FDEs in first-order Taylor series about $\mathbf{y}_k$ (N-R method):

$$\mathbf{E}_k(\mathbf{y}_k + \Delta\mathbf{y}_k, \mathbf{y}_{k-1} + \Delta\mathbf{y}_{k-1}) \simeq \mathbf{E}_k(\mathbf{y}_k, \mathbf{y}_{k-1}) + \sum_{n=1}^{N} \frac{\partial \mathbf{E}_k}{\partial y_{n,k-1}} \Delta y_{n,k-1} + \sum_{n=1}^{N} \frac{\partial \mathbf{E}_k}{\partial y_{n,k}} \Delta y_{n,k}.$$

   Similar relations can be obtained for the boundary conditions (see *NRiC* §17.3 for details; the partial derivatives can be computed analytically—it's just tedious! *NRiC* §17.4 gives a worked example).

4. Want $\mathbf{E}(\mathbf{y} + \Delta\mathbf{y}) = 0$. Result is a large $(M \times N) \times (M \times N)$ block-diagonal matrix[2] that can be solved using optimized Gaussian elimination for the $\Delta\mathbf{y}$'s.

5. After applying the new increments, iteratively improve ("relax") solution until the boundary conditions are satisfied and the difference equations between grid points are zeroed to the desired accuracy.

   - Need to solve a matrix equation each iteration.
   - Choice of grid points is an important issue and leads to adaptive mesh strategies in modern solvers.

Example: Stellar Structure

- Numerical methods for 2-pt BVPs largely developed by astronomers seeking to solve equations of stellar structure.
  - Form a system of four coupled ODEs.

1. Consider spherical shell, thickness dr, distance r from origin. Then $dM_r = 4\pi r^2 dr\rho$, or,

$$\frac{dM_r}{dr} = 4\pi r^2 \rho.$$

   It is convenient to transform this equation (and the rest) so that M_r is the independent variable. In this case, just take the reciprocal:

$$\frac{dr}{dM_r} = \frac{1}{4\pi r^2 \rho}.$$

[2]Each interior point supplies a block of N equations coupling $2N$ corrections to the solution variables at points k, $k-1$. The boundary conditions supply smaller blocks, $n_1 \times N$ and $n_2 \times N$.

2. Hydrostatic equilibrium $\Longrightarrow$ net force on shell is zero. $\therefore -\nabla_r P - \rho g = 0$, where g = gravitational acceleration per unit mass = GM_r/r^2, or,

$$\frac{dP}{dr} = -\frac{GM_r}{r^2}\rho.$$

(To derive, note upward force on shell per unit area = $P(r) - P(r + \Delta r) = -\Delta P$ must equal downward force per unit area = $[GM_r/r^2](M_{\rm shell}/4\pi r^2) = (GM_r/r^2)\rho\, dr$.) Transforming,

$$\frac{dP}{dM_r} = -\frac{GM_r}{4\pi r^4}.$$

3. Let ε = energy generation rate/unit mass. Then energy transport rate through shell $\Delta L = L(r + \Delta r) - L(r)$ must equal energy generation rate $4\pi r^2 dr\, \rho\varepsilon$, where L = luminosity, or,

$$\frac{dL_r}{dr} = 4\pi r^2 \rho\varepsilon.$$

Transforming,

$$\frac{dL_r}{dM_r} = \varepsilon.$$

4. Finally, there is an equation describing energy transport. For radiative (and conductive) transport,

$$\frac{dT}{dr} \propto \frac{\kappa\rho L_r}{r^2 T^3},$$

where κ is the mean absorption coefficient (opacity; so higher opacity $\Longrightarrow$ higher T gradient). Transforming,

$$\frac{dT}{dM_r} \propto \frac{\kappa L_r}{r^4 T^3}.$$

   - This equation harder to derive since it depends on radiation transport mechanism and convective stability.

- We have 4 ODEs in 7 unknowns $(r, \rho, P, L_r, T, \varepsilon, \kappa)$.
- Need 3 constitutive relations:
  1. $P = P(\rho, T)$ — equation of state.
  2. $\varepsilon = \varepsilon(\rho, T)$ — nuclear energy generation rate.
  3. $\kappa = \kappa(\rho, T)$ — opacity (for radiative transport; otherwise need an equivalent relation for convection).
- Boundary conditions (need 4):
  - At center ($M_r = 0$): $r = 0$, $L_r = 0$. But $P = P_c = ?$, $T = T_c = ?$
  - At surface ($M_r = M$): $P \simeq 0$, $T \simeq 0$ (also $\rho \simeq 0$). But $r = R = ?$, $L_r = L = ?$
- This is a classic 2-pt BVP.

- First solving techniques based on shooting method.
- Singularity at center $\Longrightarrow$ must fit at an intermediate point: Schwarzschild Scheme (e.g., Schwarzchild, *Structure and Evolution of the Stars*).
- Modern stellar structure codes use relaxation method with adaptive mesh (e.g., P. Eggleton, *MNRAS*, **151**, 351 (1971)).

Polytropes

- Can illustrate technique with calculation of structure of polytropes.
- Assume there is no energy generation anywhere inside ($\varepsilon \equiv 0$), e.g., white dwarf or neutron star.
- Assume EOS of form $P = k\rho^{(n+1)/n}$ (no T dependence).
  - E.g., EOS for monatomic gas (such as degenerate electron gas) is:
    $P \propto \rho^{5/3}$ if non-relativistic ($n = 3/2$);
    $P \propto \rho^{4/3}$ if relativistic ($n = 3$).
- This form is called a polytropic EOS. n is polytropic index.
- It is convenient to rewrite $\rho = \theta^n$. Then the first stellar structure equation becomes

$$\frac{d\mathcal{M}}{d\mathcal{R}} = \mathcal{R}^2\theta^n,$$

  and the second becomes

$$\frac{d\theta}{d\mathcal{R}} = -\frac{\mathcal{M}}{\mathcal{R}^2},$$

  where $\mathcal{R} \equiv r/r_0$, $M = 4\pi r_0^3\mathcal{M}$, and $r_0^2 = (n+1)k/4\pi G$ (E.F.T.S.).
- These are the "Lane-Emden Equations."
- This is a system of 2 ODEs with BC $\mathcal{M} = 0$ at $\mathcal{R} = 0$ and $\theta = 0$ at $\mathcal{R} = R_\star/r_0$.
  - If we have a desired $R_\star$ known in advance (or, equivalently, $M_\star$), then we can set $\theta = \theta_c$ (say) at $\mathcal{R} = 0$ and iterate over different starting θ_c until the outer boundary condition is satisfied.