CS 450: Numerical Anlaysis[1]

Boundary Value Problems for Ordinary Differential Equations

University of Illinois at Urbana-Champaign

[1] *These slides have been drafted by Edgar Solomonik as lecture templates and supplementary material for the book "Scientific Computing: An Introductory Survey" by Michael T. Heath (slides).*

Boundary Conditions

- Often we seek to solve a differential equation that satisfies conditions on its values and derivatives on parts of the domain boundary.

- Consider a first order ODE $\boldsymbol{y}'(t) = \boldsymbol{f}(t, \boldsymbol{y})$ with *linear boundary conditions* on domain $t \in [a, b]$:
$$\boldsymbol{B}_a \boldsymbol{y}(a) + \boldsymbol{B}_b \boldsymbol{y}(b) = \boldsymbol{c}$$

Existence of Solutions for Linear ODE BVPs

- The solutions of linear ODE BVP $\boldsymbol{y}'(t) = \boldsymbol{A}(t)\boldsymbol{y}(t) + \boldsymbol{b}(t)$ are linear combinations of solutions to linear homogeneous ODE IVPs $\boldsymbol{y}'(t) = \boldsymbol{A}(t)\boldsymbol{y}(t)$:

- Solution $\boldsymbol{u}(t)$ (and $\boldsymbol{y}(t)$) exists if $\boldsymbol{Q} = \boldsymbol{B}_a\boldsymbol{Y}(a) + \boldsymbol{B}_b\boldsymbol{Y}(b)$ is invertible:

Green's Function Form of Solution for Linear ODE BVPs

- For any given $\boldsymbol{b}(t)$ and $\boldsymbol{c}$, the solution to the BVP can be written in the form:

$$\boldsymbol{y}(t) = \boldsymbol{\Phi}(t)\boldsymbol{c} + \int_a^b \boldsymbol{G}(t,s)\boldsymbol{b}(s)ds$$

$\boldsymbol{\Phi}(t) = \boldsymbol{Y}(t)\boldsymbol{Q}^{-1}$ is the *fundamental matrix* and the *Green's function* is

$$\boldsymbol{G}(t,s) = \boldsymbol{Y}(t)\boldsymbol{Q}^{-1}\boldsymbol{I}(s)\boldsymbol{Y}^{-1}(s), \quad \boldsymbol{I}(s) = \begin{cases} \boldsymbol{B}_a\boldsymbol{Y}(a) & : s < t \\ -\boldsymbol{B}_b\boldsymbol{Y}(b) & : s \geq t \end{cases}$$

Conditioning of Linear ODE BVPs

- For any given $\boldsymbol{b}(t)$ and $\boldsymbol{c}$, the solution to the BVP can be written in the form:

$$\boldsymbol{y}(t) = \boldsymbol{\Phi}(t)\boldsymbol{c} + \int_a^b \boldsymbol{G}(t,s)\boldsymbol{b}(s)ds$$

- The absolute condition number of the BVP is $\kappa = \max\{||\boldsymbol{\Phi}||_\infty, ||\boldsymbol{G}||_\infty\}$:

Shooting Method for ODE BVPs

Demo: *Shooting method*

- For linear ODEs, we construct solutions from IVP solutions in $\boldsymbol{Y}(t)$, which suggests the *shooting method* for solving BVPs by reduction to IVPs:

- *Multiple shooting* employs the shooting method over subdomains:

Finite Difference Methods

- Rather than solve a sequence of IVPs that satisfy the ODEs until they (approximately) satisfy boundary conditions, we can refine an approximation that satisfies the boundary conditions, until it satisfies the ODE:

- Convergence to solution is obtained with decreasing step size h so long as the method is consistent and stable:

Finite Difference Methods

- Lets derive the finite difference method for the ODE BVP defined by

$$u'' + 7(1 + t^2)u = 0$$

with boundary conditions $u(-1) = 3$ and $u(1) = -3$, using a centered difference approximation for u'' on $t_1, \ldots, t_n$, $t_{i+1} - t_i = h$.

Collocation Methods

- *Collocation methods* approximate y by representing it in a basis

$y'(t) = f(t, y)$ ⇒ simplify $f(t,y) = f(t)$

$$\boldsymbol{y}(t) \approx \boldsymbol{v}(t, \boldsymbol{x}) = \sum_{i=1}^{n} x_i \phi_i(t).$$

basis

unknown coefficients

satisfy the ODE at a set of collocation points:

$\forall_i \in [2, n-1]$ $\quad v'(t_i, x) = f(t_i, \underbrace{v(t_i, x)}_{f(t_i)})$

$$\sum_{j=1} x_j \underbrace{\boxed{\varphi_j'(t_i)}}_{A_{ji}} = \underbrace{\boxed{f(t_i)}}_{b_i}$$

$Ax = b$

$t_1 \ldots t_n$

$a \qquad b$

boundary B.C.s

- Choices of basis functions give different families of methods:

• spectral (polynomials)

• localized (hat functions, B-splines)

Solving BVPs by Optimization

- To improve robustness, define and minimize a residual error over the whole domain rather than at collocation points.

$$r(t,x) = v'(t,x) - f(t) = \left[\sum_{i=1}^{n} x_i \varphi_i'(t)\right] - f(t)$$

$$\min_{\substack{x \\ \uparrow \\ \text{coeff}}} \underbrace{\int_a^b \frac{1}{2}\langle r(t,x), r(t,x)\rangle\, dt}_{F(x)}$$

- The first-order optimality conditions of the optimization problem are a system of linear equations $\boldsymbol{Ax} = \boldsymbol{b}$:

$$0 = \frac{\partial F}{\partial x_i} = \int_a^b \left\langle r(t,x), \frac{\partial r}{\partial x_i}(t,x)\right\rangle dt \qquad \varphi_i'(t)$$

$$Ax = b$$

$$= \left[\sum_{j=1}^{n} x_j \underbrace{\int_a^b \varphi_j'(t)\varphi_i'(t)\, dt}_{A_{ij}}\right] - \underbrace{\int_a^b f(t)\varphi_i'(t)\, dt}_{b_i}$$

Weighted Residual

- *Weighted residual methods* work by ensuring the residual is orthogonal with respect to a given set of weight functions:

$\phi_1 \ldots \phi_n$ e.g. orthogonal

$$\int_a^b r(t,x)\,\phi_i(t)\,dt = 0 \quad \forall i \Rightarrow 0 = \sum_{j=1}^n x_j \underbrace{\int_a^b \varphi_j(t)\,\phi_i(t)\,dt}_{A_{ij}} - \underbrace{\int_a^b f(t)\,\phi_i(t)\,dt}_{b_i}$$

test functions

- The Galerkin method is a weighted residual method where $w_i = \phi_i$.

$$0 = \sum_{j=1}^n x_j \int_a^b \varphi_j(t)\,\boxed{\varphi_i(t)}\,dt - \int_a^b f(t)\,\boxed{\varphi_i(t)}\,dt$$

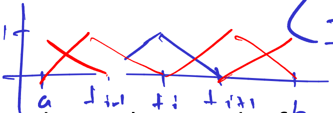
difference from optimization method is lack of derivative of weight functions

Second-Order BVPs: Poisson Equation

In practice, BVPs are at least second order and its advantageous to work in the natural set of variables.

- Consider the *Poisson equation* $u'' = f(t)$ with boundary conditions $u(a) = u(b) = 0$ and define a localized basis of hat functions:

$$\varphi_i(t) = \begin{cases} \frac{t_i - t}{h} & \in [t_{i-1}, t_i] \\ \frac{t - t_i}{h} & \in [t_i, t_{i+1}] \\ 0 & \text{otherwise} \end{cases}$$



- Defining residual equation by analogy to the first order case, we obtain,

$$r(t, x) = \sum_j x_j \varphi_j''(t) - f(t)$$

$$0 = \sum_j x_j \int_a^b \varphi_j''(t)\, \varphi_i(t)\, dt - \int_a^b f(t)\, \varphi_i(t)\, dt$$

Weak Form and the Finite Element Method

- The finite-element method permits a lesser degree of differentiability of basis functions by casting the ODE in *weak form*:

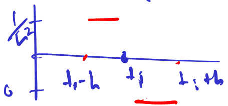
$$0 = \sum_j x_j \int_a^b \varphi_j''(t)\,\varphi_i(t)\,dt - \int_a^b f(t)\,\varphi_i(t)\,dt$$

BC

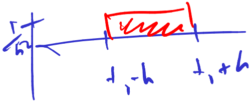
$$\int_a^b \varphi_j''(t)\,\varphi_i(t)\,dt = \varphi_j'(b)\underbrace{\varphi_i(b)}_{0} - \varphi_j'(a)\underbrace{\varphi_i(a)}_{0} - \int_a^b \varphi_j'(t)\,\varphi_i'(t)\,dt$$

$$0 = -\sum_j x_j \underbrace{\int_a^b \varphi_j'(t)\,\varphi_i'(t)\,dt}_{A_{ij}} - \underbrace{\int_a^b f(t)\,\varphi_i(t)\,dt}_{b_i}$$

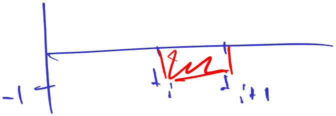
$\varphi_i'(t)$



$[\varphi_i'(t)]^2$



$\varphi_i'(t)\,\varphi_{i+1}'(t)$



$$A_{ii} = 2/h \qquad b_i = \int_a^b f\, e_i(t)\, dt$$

$$A_{i+1,i} = -1/h \qquad b_i = hf$$

$$Ax = -b$$

$$\frac{1}{h}\begin{bmatrix} 2 & -1 & & \\ -1 & \ddots & & \\ & & \ddots & -1 \\ & & -1 & 2 \end{bmatrix} = -\begin{bmatrix} hf \\ \vdots \\ hf \end{bmatrix}$$

Eigenvalue Problems with ODEs

- A typical second-order scalar *ODE BVP eigenvalue problem* is

$$u'' = \lambda f(t, u, u'), \quad \text{with boundary conditions } u(a) = 0, u(b) = 0.$$

These can be solved, e.g. for $f(t, u, u') = g(t)u$ by finite differences:

$u'' = \lambda g(t) u$

matrix eigenvalue problem

$$\frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} = \lambda g(t_i) u_i \quad \forall i$$

$$\frac{-u_{i+1} + 2u_i - u_{i-1}}{h^2 g(t_i)} = \lambda u_i \implies Au = \lambda u$$

$$A[i,:] = \left[0 \ \ldots \ -\frac{1}{h^2 g(t_i)} \ \ \frac{2}{h^2 g(t_i)} \ \ -\frac{1}{h^2 g(t_i)}\right.$$

Using Generalized Matrix Eigenvalue Problems

- Generalized matrix eigenvalue problems arise from more sophisticated ODEs,

$$u'' = \lambda(g(t)u + h(t)u'), \quad \text{with boundary conditions } u(a) = 0, u(b) = 0.$$

$$Au = \lambda Bu$$

$$B^{-1}Au = \lambda u$$

generalized eigenvalue problem
(matrix pencil)