Review for 575B Final: Definitions

Carlos Chiquete

July 11, 2007

1 Stable and Asymptotically Stable Solutions, [Ascher & Petzold, p.25]

If $\mathbf{y}(t)$ and $\hat{\mathbf{y}}(t)$ both satisfy our problem, $\mathbf{y}' = \mathbf{f}(t, \mathbf{y})$, then a solution is **stable** if

$$|\mathbf{y}(0) - \hat{\mathbf{y}}(0)| \leq \delta$$

and,

$$|\mathbf{y}(t) - \hat{\mathbf{y}}(t)| \leq \epsilon, \quad \forall t \geq 0$$

A solution is **asymptotically stable**,

$$|\mathbf{y}(t) - \hat{\mathbf{y}}(t)| \to 0, \quad t \to \infty$$

2 Conditions of stability of $\mathbf{y}' = \mathbf{A}\mathbf{y}$, [A & P, p.27]

For simple ODE systems of the form $y' = Ay$, we know the solution is like the following $\mathbf{y}(t) = e^{\mathbf{A}t}\mathbf{y}(0)$. We can write the the simililarity transformation if A is diagonalizable,

$$\mathbf{A} = \mathbf{T}\mathbf{\Lambda}\mathbf{T}^{-1}$$

Then we can write the change of variable, $w = T^{-1}y$, and we can write our decoupled system, $w' = \mathbf{\Lambda}w$. This leads to m equation of the sort, $w_i = \lambda_i w_i$. So we can write our stability condition as,

- **Stable** if $Re(\lambda) \leq 0$
- **Asymptotically Stable** if $Re(\lambda) < 0$

We assume the eigenvalues are simple (1×1 Jordan block). We use that if $|\mathbf{w}_n| \leq |\mathbf{w}_{n-1}|$ if and only if all real parts of eigenvalues are less that or equal to 0.

3 Conditions for stability of higher order linear systems of equations, [A & P, p.29]

We can write systems of this sort as,

$$a_k u + a_{k-1}u' + \cdots + a_0 u^{(k)} = \sum_{j=0}^{k} a_j \frac{d^{k-j}}{dt^{k-j}} u = 0$$

We rewrite this problem as a system of k equations with constant coefficients, $u' = \mathbf{A}u$. We can show the characteristic equation of the resultant matrix $\mathbf{A}$ is,

$$\phi(z) = \sum_{j=0}^{k} a_j z^{k-j}$$

If the roots $(z*)$ are of ϕ are simple we have

- **Stability** if $Re(z^*) \leq 0$
- **Asymptotic Stability** if $Re(z^*) < 0$.

4 Local Truncation Error, [A & P, p.40-41]

The difference operator is defined as the case where, (and $\mathbf{y}(0) = \mathbf{c}$,

$$\mathcal{N}_h \mathbf{y}_h(t_n) = 0$$

For instance, the Forward euler method difference operator is defined on a mesh function $\mathbf{u}$ as

$$\mathcal{N}_h \mathbf{u}(t_n) = \frac{\mathbf{u}(t_n) - \mathbf{u}(t_{n-1})}{h_n} - \mathbf{f}(t_{n-1}, \mathbf{u}(t_{n-1}))$$

and so the numerical method is given when $\mathcal{N}_h \mathbf{y}_h(t_n) = 0$.

The local truncation error $\mathbf{d}_n$ is the residual of the difference operator $\mathcal{N}_{h_n}$ when it is applied to the exact solution $\mathbf{y}(t_n)$,

$$\mathbf{d}_n = \mathcal{N}_{h_n} \mathbf{y}(t_n)$$

A method is consistent (or accurate) of order p if $\mathbf{d}_n = O(h^p)$.

5 Local Error, [A & P, p.43]

Local error refers to the error between the numerical solution at step n, $\mathbf{y}(t)$ and the solution of the initial value problem, $\bar{\mathbf{y}}_n$,

$$\bar{\mathbf{y}}'(t) = f(t, \bar{\mathbf{y}}(t))$$
$$\bar{\mathbf{y}}(t_{n-1}) = \mathbf{y}_{n-1}$$

We note that $\bar{\mathbf{y}}$ is the exact solution to the IVP above. The local error is thus,

$$\mathbf{l}_n = \bar{\mathbf{y}}(t_n) - \mathbf{y}_n$$

Moreover, it can be shown that the numerical solution exists and we can write,

$$|\mathbf{d}_n| = |\mathcal{N}_h \bar{\mathbf{y}}(t_n)| + O(h^{p+1})$$

and so the local error and the truncation error are related through,

$$h_n |\mathcal{N}_h \bar{\mathbf{y}}(t_n)| = |\mathbf{l}_n| [1 + O(h_n)]$$

6 Definition of Convergence,, [A & P, p.40]

Let h be the maximum step size used in our method. The method is said to be *convergent of order p* if the *global error* $\mathbf{e}_n$, where $\mathbf{e}_n = \mathbf{y}_n - \mathbf{y}(t_n)$ and $\mathbf{e}_0 = 0$ satisfies,

$$\mathbf{e}_n = O(h^p), \quad \text{for} \quad n = 1, 2, \ldots N$$

7 0-stability, [A & P, p.41]

A difference method is 0-stable if there exists positive h_0 and K, such that for any mesh functions $\mathbf{x}_h$ and $\mathbf{z}_h$ with $h \leq h_0$,

$$|\mathbf{x}_h - \mathbf{z}_h| \leq K \left[|\mathbf{x}_0 - \mathbf{z}_0| + \max_{1 \leq j \leq N} |\mathcal{N}_{h_n}\mathbf{x}_h(t_j) - \mathcal{N}_{h_n}\mathbf{z}_h(t_j)| \right], \quad 1 \leq n \leq N$$

This means that the difference operator $\aleph_h$ is invertible and the inverse is bounded by K. This bound measures the effect on the numerical solution of small perturbations in the data.

8 Region of Stability and A-Stability, [A & P, p.44-45]

The region of stability for a discretization method is defined with respect to the test equation, $\mathbf{y}' = \lambda \mathbf{y}$. So the condition for the region of stability is if $Re(\lambda) < 0$, then we must have decay of the solution, i.e.,

$$|\mathbf{y}_n| \leq |\mathbf{y}_{n-1}|, \quad n = 1, 2, \ldots$$

The region of stability is therefore the area of the complex plane of $h\lambda$ where we meet the condition above.

A-Stability is the condition where the region of stability is the entire of left half plane of $z = h\lambda$. The problem is that this method does not distinguish from $Re(\lambda) \ll -1$ and $-1 \ll Re(\lambda) \leq 0, \quad |Im(\lambda)| \gg 1$. This gives the need for the stiff decay conditions.

9 Stiff Equation, [A & P, p.50]

We have the following on pg. 50 in Ascher and Petzold.

> A stiff equation or Initial Value Problem is stiff in some interval $[0, b]$ if the step size needed to maintain stability of the forward euler method [or another explicit method] is much smaller than the step size required to represent the solution accurately.

Also depends on:

- accuracy criterion
- length of integration interval of integration
- region of stability of the method

For a test equation on the interval $[0, b]$, a problem is *stiff* if

$$bRe(\lambda) \ll -1.$$

More generally for the eigenvalues of the Jacobian matrix at $\mathbf{y}$ of $\mathbf{f}$, $\mathbf{f}_\mathbf{y}(t, \mathbf{y}(t))$,

$$b \min_j Re(\lambda_j) \ll -1.$$

10 Newton's Iterations, [A & P, p.55]

It is the generalization of the scalar Newton's Method. We produce the expansion of a function $g(x)$ at x^ν (our current guess),

$$0 = g(x) = g(x^\nu) + g'(x^\nu)(x - x^\nu) + O[(x - x^\nu)2]$$

We discard the higher order terms of the difference, define $x \approx x^{\nu+1}$, and solve the obtain the following linear homogenous equation for the next iterate in terms of the previous guess

$$0 = g(x^\nu) + g'(x^\nu)(x^{\nu+1} - x^\nu)$$

So in a system of m equations, $\mathbf{g}(\mathbf{x}) = 0$,

$$\mathbf{x}^{\nu+1} = \mathbf{x}^\nu - \left(\frac{\partial \mathbf{g}}{\partial \mathbf{x}}(\mathbf{x}^\nu)\right)^{-1} \mathbf{g}(\mathbf{x}^\nu), \quad \nu = 0, 1, \ldots$$

In practice we do not compute the matrix inverse and instead solve the linear system,

$$\left(\frac{\partial \mathbf{g}}{\partial \mathbf{x}}\right) \delta = -\mathbf{g}(\mathbf{x}^\nu)$$

and the new iterate is $\mathbf{x}^{\nu+1} = \mathbf{x}^\nu + \delta$. For example, for the Backward Euler method,

$$\mathbf{g}(\mathbf{y}_n) = \mathbf{y}_n - \mathbf{y}_{n-1} - h\mathbf{f}(t_n, \mathbf{y}_n)$$

Then the expression for the next guess is,

$$\mathbf{y}_n^{\nu+1} = \mathbf{y}_n^\nu - \left(I - h\frac{\partial \mathbf{f}}{\partial \mathbf{y}}\right)^{-1} \mathbf{g}(\mathbf{y}_n^\nu), \quad \text{for} \quad \nu = 0, 1, \ldots$$

11 Inhomogeneous test equation (3.28) and method with stiff decay, [A & P, p.56-59]

The inhomogenous test equation is given in the following,

$$y' = \lambda[y - g(t)]$$

where $g(t)$ is bounded and otherwise arbitrary. Therefore a discretization method has stiff decay if for $t_n > 0$ fixed,

$$|y_n - g(t_n)| \to 0 \quad \text{as} \quad h_n Re(\lambda) \to -\infty$$

Backward euler has stiff decay since

$$y_n - g(t_n) = \frac{1}{1 - h_n\lambda}[y_{n-1} - g(t_n)]$$

12 Rough problems, [A & P, p.61-62]

In general $\mathbf{y}' = \mathbf{f}(t, \mathbf{y}(t))$ can have only k bounded derivatives at the solution $\mathbf{y}(t)$,

$$\sup_{0 \le t \le b} \left| \frac{d^j}{dt^j}\mathbf{f}(t, \mathbf{y}(t)) \right| \le M, \quad j = 0, 1, \ldots k$$

Therefore since $\mathbf{y}' = \mathbf{f}(t, \mathbf{y}(t))$,

$$\| \mathbf{y}' \| \le M, \quad j = 1, \ldots, k+1$$

We expect that if there exists a point of discontinuity, we can 'restart' the problem at that time with a new initial condition equal to the last value we calculated. So if there exists a discontinuity at $\bar{t} \in [0, b]$ we can integrate our solution for the interval $[0, \bar{t}]$,

$$\mathbf{y}' = \mathbf{f}(t, \mathbf{y}(t)), \quad 0 < t < \bar{t}; \quad \mathbf{y}(0) = \mathbf{c}$$

And define a new problem on $[\bar{t} < t < b]$

$$\mathbf{z}' = \mathbf{f}(t, \mathbf{z}(t)), \quad \bar{t} < t < b; \quad \mathbf{z}(\bar{t}) = \mathbf{y}(\bar{\mathbf{t}})$$

If $\bar{t}$ occurs between some step in $[t_{\bar{n}-1}, t_{\bar{n}}]$, then we expect an additional error of order $O(h_{\bar{n}}$.

13 Taylor Methods, [A & P, p.73-74]

Taylor Series methods involve calculating derivatives up to some degree and use these to calculate the next iterate explicitly for the ODE $y' = f(t, y)$. We have the expansion,

$$y_n = y_{n-1} + hy'_{n-1} + \frac{h2}{2}y''_{n-1} + \dots \frac{h^p}{p!}y^{(p)}_{n-1}$$

We evaluate all derivatives at t_{n-1} and y_{n-1}, i.e.

$$\begin{aligned} y'_{n-1} &= f, \\ y''_{n-1} &= f_t + f_y f \\ y'''_{n-1} &= f_{tt} + 2f_{ty}f + f_y f_t + f_{yy}f2 + f2_y f \end{aligned}$$

The local truncation error is therefore

$$d_n = \frac{y(t_n) - y_n}{h} = \frac{h^p y^{(p+1)}(t_n)}{(p+1)!} + O(h^{p+1})$$

14 General Formulation of Runge Kutta Methods, [A & P, p.80,82]

For the problem, $\mathbf{y}' = \mathbf{f}(t, \mathbf{y}(t))$, we can write any Runge Kutta formulation of s stages, as the following,

$$\mathbf{Y}_i = \mathbf{y}_{n-1} + h\sum_{j=1}^{s} a_i j\mathbf{f}(t_{n-1} + c_j h, \mathbf{Y}_j)$$

and

$$y_n = y_{n=1} + h\sum_{i=1}^{s} b_i\mathbf{f}(t_{n-1} + c_i h, \mathbf{Y}_i)$$

Or we can also write the equivalent formulation.

$$\mathbf{K}_i = \mathbf{f}(t_{n-1} + c_i h, \mathbf{y}_{n-1} + h\sum_{j=1}^{s} a_{ij}\mathbf{K}_j)$$

and,

$$\mathbf{y}_n = \mathbf{y}_{n-1} + h\sum_{i=1}^{s} b_i\mathbf{K}_i$$

These can be written in the scheme

$$\begin{array}{c|c} \mathbf{c} & \mathbf{A} \\ & \mathbf{b}^T \end{array}$$

We always choose

$$c_i = \sum_{j=1}^{s} a_{ij}, \quad i = 1, \dots, s \quad \text{and} \quad \sum_{j=1}^{s} b_i = 1$$

A general Runge Kutta method is explicit if $\mathbf{A}$ is lower triangular and has a zero-diagonal.

15 Pairs of Methods (EMBEDDED), [A & P, p.91]

We use the pairs scheme to produce more accurate step sizes. If the solution found through the first of the pair (order p) is y_n and the solution found through the higher order (order p+1) is $\hat{y}_n$, then we require that the new step size $\tilde{h}$ satisfy

$$\left(\frac{\tilde{h}}{h}\right)^{p+1} |\hat{y}_n - y_n| \approx \text{frac}\mathbf{ETOL}$$

where the h is a rejected step size since $|\hat{y} - y| > \mathbf{ETOL}$. $|\hat{y}_n - y_n|$ is an estimate of the local error. Then the pairs are used to estimate the local error. The pair of methods used the same evaluations of $\mathbf{f}$, but add with different $\mathbf{b}^T$. The scheme is summarized,

$$\begin{array}{c|c} \mathbf{c} & \mathbf{A} \\ & \mathbf{b}^T \\ & \hat{\mathbf{b}}^T \end{array}$$

16 Step Doubling, [A & P, p.93]

Step Doubling aims to achieve the same estimation of the local error as in embedded methods. It uses that calculating the solution using a step h and $2h$. We know the local error for a p-order method is given by some function t_n and y_n, i.e.,

$$l_n = \psi(t_n, y(t_n))h^{p+1} + O(h^{p+2})$$

Now imagine starting for y_{n-2} and going two steps of size h, and one step of size $2h$. We make the assumption that the error across 2 steps of size h is twice the error of one step (true if $h \to 0$).

$$2l_n(h) = 2\psi h^{p+1} + O(h^{p+1})$$
$$l_n(2h) = \psi(2h)^{p+1} + O(h^{p+1})$$

Then we can estimate the local error at the n-th step, since we know $|\hat{y} - y_n|$,

$$|\hat{y} - y_n| \approx 2h^{p+1}(2^p - 1)|\psi| + O(h^{p+2})$$

thus,

$$2|l_n| \approx \frac{|\hat{y} - y_n|}{2^p + 1}$$

17 Collocation methods

We want a degree s polynomial $\phi(t_i)$ to satisfy the following:

$$\phi(t_{n-1}) = y_{n-1} \text{ and } \phi'(t_i) = f(t_i, \phi(t_i)),\ i = 1, 2, \ldots, s$$

where $t_i = t_{n-1} + c_i h$ are the collocation points. We also require that $\phi(t_n) = y_n$. Then the derivative of ϕ (degree $s-1$) is given as

$$\phi'(t + \tau h) = \sum_{j=1}^{s} L_j(t_{n-1} + \tau h)K_j$$

where $L_j(t_{n-1} + \tau h) = \Pi_{i=1, i\neq j}^{s} \frac{\tau - c_i}{c_j - c_i}$. So integrating gives,

$$K_i = \phi(t_i) = \phi(t_{n-1}) + h\sum_{j=1}^{s}\left(\int_0^{c_i} L_j(r)dr\right)K_j$$

and

$$\phi(t_n) = \phi(t_{n-1}) + h\sum_{j=1}^{s}\left(\int_0 1 L_j(r)dr\right)K_j$$

So the idea is that

$$y_n = y_{n-1} + h\sum_{i=1}^{s} b_i K_i$$

where $K_i = f(t_i, \phi(t_i)) = f(t_i, y_{n-1} + \sum_{j=1}^{s} a_{ij}K_j)$. So for these methods,

$$a_{ij} = \int_0^{c_i} L_j(r)dr$$

and

$$b_j = \int_0 1 L_j(r)dr$$

Radau

- These methods have stiff decay and they can attain order $2s-1$.
- They consider the right end of the interval.

Gauss

- These methods are the most accurate and they achieve the maximum order of $2s$ and are A-stable.
- They do not consider the boundaries of the interval.

Lobatto

- These methods can attain order $2s-2$ and is A-stable.
- They consider both ends of the interval.

18 Linear Multistep Methods

The general scheme of these methods is as follows:

$$\sum_{j=0}^{k}\alpha_j \mathbf{y}_{n-j} = h\sum_{j=0}^{k}\beta_j \mathbf{f}_{n-j}$$

For scaling we require $\alpha_0 = 1$ and $\alpha_1 = -1$.

Adams-Bashforth

These methods interpolate $\mathbf{f}$ at the k previous points but not including t_n. So the methods are explicit, i.e. $\beta_0 = 0$. The following drops the vector notation (since it can be easily generalized).

$$y_n = y_{n-1} + \sum_{j=1}^{k}\beta_j f_{n-j}$$

Adams-Moulton

These are simply the implicit version of the A-B methods. So $\beta_0 = 1$,

$$y_n = y_{n-1} + \sum_{j=0}^{k} \beta_j f_{n-j}$$

These methods interpolate a polynomial at different points of f_{n-j} for $j = 0 \ldots k$. They are then integrated to get the next value.

BDF

These methods interpolate the solution vector y_{n-j} for $j = 0, \ldots, k$ and we require that the interpolating polynomial derivative be equal to f_n at $t = t_n$. i.e.

$$y_n = \sum_{j=1}^{k} \alpha_j y_{n-j} + h\beta_0 f_n$$

Characteristic Polynomial of the MSL methods

The characteristic polynomial of the MSL methods are define as,

$$\rho(\xi) = \sum_{j=0}^{k} \alpha_j \xi^{k-j} \text{ and } \sigma(\xi) = \sum_{j=0}^{k} \beta_j \xi^{k-j}$$

It is a fact that if $\rho(1) = 0$ and $\rho'(1) = \sigma(1)$, the method is consistent, i.e. $p \geq 1$.

Derive order conditions (Section 5.2.1)

Define the linear operator $\mathcal{L}_h y(t)$,

$$\mathcal{L}_h y(t) = \sum_{j=0}^{k} [\alpha_j y(t - jh) - h\beta_j y'(t - jh)]$$

then the local truncation error is given as,

$$d_n = h^{-1} \mathcal{L}_h y(t)$$

Then we expand y and y' into its Taylor expansions about t. So for $d_n = O(h^p)$, then collecting the terms in terms of derivatives of $y(t)$, we must have,

$$C_0 = C_1 = \cdots = C_p = 0 \text{ and so } C_{p+1} \neq 0$$

Then we have the coefficients

$$C_0 = \sum_{j=0}^{k} \alpha_j \text{ and } C_i = (-1)^i \left[\sum_{j=1}^{k} j^i \alpha_j + \frac{1}{(i-1)!} \sum_{j=0}^{k} j^{i-1} \beta_j \right], \text{ for } j = 1, 2, \ldots$$

Root conditions

- A difference equation is **stable** iff all roots lie within the unit circle ($|\xi_i| \leq 1$) and if $|\xi_i| = 1$, then ξ_i is a simple root.
- A difference equation is **asymptotically stable** iff all roots satisfy $|\xi_i| < 1$.

Strong and weak stability

0-stability

Firstly we need the concept of 0-stability, i.e. iff all roots ξ_i of the characteristic polynomial $\rho(\xi)$ satisfy, $|\xi| \leq 1$ and of $|\xi_i| = 1$, then ξ_i is a simple root ($1 \leq i \leq k$).

As a consequence

- if the root condition is met,
- the method is accurate to order p
- and the method is accurate to order p,

then the method is convergent to order p.

Strong stability

A method is strongly stable is all the roots of $\rho(\xi) = 0$ are inside the unit circle except for $\xi = 1$.

Weak stability

If a method is 0-stable but not strongly stable it is weakly stable.

Predictor Corrector methods

A predictor corrector method is an explicit MSL dubbed "general linear methods".

- P: Adams-Bashforth method of order k, predicts/estimates y_n0.
- E: Evaluates $f(t_n, y_n0)$.
- C: Corrects the prediction for y_n, by using a Adams-Moulton method of order k.

The preceding is a method denoted PEC, but we can iterate up to ν times, to form a $P(EC)^\nu$. Adding a step at the end to evaluate a new $f(t_n, y_n^\nu)$ that becomes the new $f(t_{n-1}, y_{n-1})$ at the next step so the notation becomes $P(EC)^\nu E$.

Absolute stability of MSL methods

Using the test equation $y' = f(t, y)$ and applying it to our method produces the following,

$$\sum_{i=0}^{k} \alpha_j y_{n-j} = h\lambda \sum_{i=0}^{k} \beta_j y_{n-j}$$

Positing that $y_n = \epsilon^n$, then we can get,

$$\sum_{i=0}^{k} \alpha_j \xi^{k-j} = h\lambda \sum_{i=0}^{k} \beta_j \xi^{k-j}$$

or that $\rho(\xi) = h\lambda\sigma(\xi) = z\sigma(\xi)$. Then to test stability we require $\xi = e^{i\theta}$, for $\theta \in (0, 2\pi)$ and get the boundary of the region of stability, i.e.

$$z = \frac{\rho(e^{i\theta})}{\sigma(e^{i\theta})}$$

Difference equations

A difference equation of the sort

$$a_k y_{n-k} + \cdots + a_0 y_n = q_n,\ n = k, k+1, \ldots$$

then the solution is given by $y_n = x_n + v_n$ where v_n is the inhomogeneous part. Then if we posit that the inhomogenous is given by $x_n = \xi^n$, then we must satisfy,

$$\phi(\xi) = \sum_{j=0}^{k} a_j \xi^{n-j} = 0$$

then the solution is given by the following where the ξ_j are the roots of $\phi(\xi)$.

$$y_n = \sum_{j=0}^{k} c_j \xi_j^n + v_n$$

if ξ_i is the a double root, then mode must be proportional to $n\xi_i^n$ or if it's a triple root $n(n-1)\xi_i^n$ and so on. We use the root condition to verify the stability,

- A difference equation is **stable** iff all roots lie within the unit circle ($|\xi_i| \leq 1$) and if $|\xi_i| = 1$, then ξ_i is a simple root.
- A difference equation is **asymptotically stable** iff all roots satisfy $|\xi_i| < 1$.

19 BVP's

General BC's

These are BC's in the form for a problem $\mathbf{y}' = f(t, \mathbf{y}),\ 0 < t < b$,

$$\mathbf{g}(\mathbf{y}(0), \mathbf{y}(b)) = \mathbf{0}$$

Linear BC's

These are BC's where we can write,

$$B_0 \mathbf{y}(0) + B_b \mathbf{y}(b) = b$$

where **b** is given.

Separated BC's

These are linear separated BC's where if a row of B_0 is non-zero, the corresponding row of B_b is zero, i.e. there no element of **g** involves both ends simultaneously.

Stability for BVP's

The stability for linear BVP's ($\mathbf{y}'(t) = A(t)\mathbf{y} + \mathbf{q}(t)$) is given when the stability constant κ is of moderate size (i.e. $\kappa \approx ||A(t)||b$),

$$\kappa = \max(||G||_\infty, ||\phi||_\infty)$$

and so the solution is bounded,

$$||y||_\infty = \max_{0 \le t \le b} |y(t)| \le \kappa \left(|b| + \int_0^b |q(s)| ds \right)$$

Example, for $A(t) = \begin{pmatrix} \lambda & 0 \\ 0 & -\lambda \end{pmatrix}$, the IVP is unstable since one eigenvalue is positive, but in this case the BVP is stable.

Dichotomy

For separated BC's so that $B_0\phi(0) = P = \begin{pmatrix} I_k & 0 \\ 0 & 0 \end{pmatrix}$. And so $B_b\phi(b) = I - P$. If there exists a constant K such that,

$$||\phi(t)P\phi^{-1}(s)|| \le K, \ s \le t \text{ and } ||\phi(t)(I-P)\phi^{-1}(s)|| \le K, \ s > t$$

then we have dichotomy.

Exponential dichotomy occurs if there exists positive α and β,

$$||\phi(t)P\phi^{-1}(s)|| \le Ke^{\alpha(s-t)}, \ s \le t \text{ and } ||\phi(t)(I-P)\phi^{-1}(s)|| \le Ke^{\beta(t-s)}, \ s > t$$

Reformulation tricks

- Adding an equation for an unknown constant a, i.e. $a' = 0$.
- If $\mathbf{y}(0)$ are unknown then we can formulate $\mathbf{y}' = f(t, \mathbf{y}), \ y(0) = a$ by taking $a' = 0$ and so our BC becomes $g(a(b), y(b)) = 0$ and $y(0) = a(0)$. This allows us to write our non-separated conditions into a separated manner.

Shooting/Multiple Shooting

Shooting takes the expertise we have from IVP and attempts to solve an BVP by guessing at the initial conditions that will cause $\mathbf{g}(\mathbf{y}(0) = \mathbf{c}, \mathbf{y}(b)) = 0$. To test $\mathbf{y}(b)$ we need to solve the IVP and get $\mathbf{y}(t; \mathbf{c})$. We iterate this with newton iterations on the function $h(c) = g(c, y(b; c))$.

Multiple shooting subdivides the interval into N intervals and uses shooting on these intervals to "patch" together the final solution.

Finite Difference methods

We use a symmetric implicit one-step method to produce a system of $m(N+1)$ algebraic possible non-linear equations. We can use implicit midpoint

$$y_n = y_{n-1} + hf(t_{n-1/2}, \frac{1}{2}(y_n + y_{n-1}))$$

or trapezoid

$$y_n = y_{n-1} + h\left(f(t_n, y_n) + f(t_{n-1}, y_{n-1})\right)$$

in this method. These methods are used because in a BVP times flows in both directions. The resultant equation can be solved using Newton's iterations.

Quasi-linearization

We directly linearize and truncate the non-linear differential system, such that the ν-th iteration satisfies a linear BVP.

Continuation (Homotopy)

We embed our problem into a family of related problems by some parameter μ. We solve choose a range of μ_0 such that the associated problem can be easily solved. We then change the parameter μ by $\mu + \Delta\mu$, and define our initial iterate $y_\pi 0$ as the first guess for the perturbed problem. We continue up to $\mu = \mu_1$ which solves our problem. This method can become rather expensive to execute.

Finite element method

The **weak formulation** involves the operator $-(au')' + bu = Lu = f$. We approximate the solution u by a function $u_m = \varphi_0 + \sum_{j=1}^{m} \gamma_j \varphi_j$, $c \leq x \leq d$. The boundary conditions depend on the problem. We consider the defect, i.e.

$$d_m(x) = Lu_m - f(x)$$

We want this defect to be zero since that implies $u = u_m$, so we consider that the expansion function must be perpendicular to the zero function (approximately) so we arrive at the Galerkin equations,

$$(d_m(x), \varphi_k) = 0, \;\; k = 1, 2, \ldots, m$$

We therefore multiply the defect by the k-th approximation function and integrate over the interval,

$$0 = (Lu_m, \varphi_k) - (f, \varphi_k)$$

We can perform an integration by parts to get something like,

$$\sum_{l=1}^{m} a_{kl}\gamma_l = (f, \varphi_k) - a_{k0} + b_{k0}$$

where $a_{kl} = \int_c^d (a\varphi_l'\varphi_k' + b\varphi_l\varphi_k)dt$ and b_{k0} might be an additional boundary term.

The **variational approach** to find the conditions above for Dirichlet conditions on $0 < x < 1$ involves defining the following functional,

$$\mathcal{J}(v) = \int_0 1(a(v')2 + bv2 - 2fv)dt$$

We want to minimize it using a function $u = v + w$,

$$\mathcal{J}(u) = \min_{w=\varphi_0+v, v\in H_0, \varphi_0 \in H} \mathcal{J}(w)$$

The next step is to assume u is the minimizer and then evaluating $J(u + \epsilon v)$, for real ϵ. We can eventually draw the conclusion that $\int_0 1(au'v' + buv - fv)dt = 0$. Anyway, if we have the minimizer, then we know that $w = \varphi_0 + \sum \gamma_j \varphi_j$, this leads to the condition that,

$$\mathcal{J}_m(\gamma) = \mathcal{J}(\varphi_0 + \sum \gamma_j \varphi_j) = \int_0 1(blah!)dt$$

We then take the derivative of $\mathcal{J}_m(\gamma)$ with respect to γ_k, and recover the conditions we found previously.

Neummann conditions

We simply include $k = 0$ as well as $k = m + 1$. So that the natural boundary conditions occur at the end by themselves.