Numerical Methods in Engineering and Applied Science

Lecture 14. Boundary value problems for ODEs.

A boundary value problem consists of a differential equation for which we seek a solution taking values imposed at the limits of the resolution domain. Indeed, the initial value problem is a special case of the boundary value problem, in which the Cauchy–Lipschitz theorem provides an answer to the question of existence and uniqueness of the solution. In general, a boundary value problem can have zero, one, or more than one solution.

Example. Consider the following problem

$$\begin{aligned} &\frac{\mathrm{d}^2 y(x)}{\mathrm{d}x^2} + y(x) = 0, \quad 0 < x < b, \\ &y(0) = 0, \quad y(b) = \beta_2. \end{aligned} \tag{1}$$

The general solution that satisfies $y(0) = 0$ is $y(x) = c\sin(x)$, where c is an arbitrary constant. Then in the case $b = n\pi$, the problem (1) has no solution if $\beta_2 \neq 0$ and it has an infinity of solutions if $\beta_2 = 0$.

In what follows, we consider only the ordinary differential equations written in the form

$$\boldsymbol{y}' = \boldsymbol{f}(x, \boldsymbol{y}), \quad a < x < b, \tag{2}$$

where $\boldsymbol{y}(x) = (y_1(x), ..., y_m(x))^T$ and $\boldsymbol{f}(x) = (f_1(x, \boldsymbol{y}), ..., f_m(x, \boldsymbol{y}))^T$. The boundary conditions of interest are

$$\boldsymbol{g}(\boldsymbol{y}(a), \boldsymbol{y}(b)) = \mathbf{0}, \tag{3}$$

where $\boldsymbol{g} = (g_1, ..., g_m)^T$ is a nonlinear function (in general) of its arguments. The initial condition is a special case of (3):

$$\boldsymbol{y}(a) - \boldsymbol{\alpha} = \mathbf{0}, \tag{4}$$

Another important case is the linear condition

$$B_a \boldsymbol{y}(a) + B_b \boldsymbol{y}(b) = \boldsymbol{\beta}, \tag{5}$$

where $B_a,\ B_b \in \mathbb{R}^{m \times m}$ et $\boldsymbol{\beta} \in \mathbb{R}^m$.

If $rank(B_a, B_b) = m$ but $rank(B_a) < m$ or $rank(B_b) < m$, the boundary conditions are *partially separated*. If, for example $rank(B_b) = q < m$, the problem can be transformed in such a way that

$$\begin{aligned} &B_{a1}\boldsymbol{y}(a) = \boldsymbol{\beta}_1, \\ &B_{a2}\boldsymbol{y}(a) + B_{b2}\boldsymbol{y}(b) = \boldsymbol{\beta}_2, \end{aligned} \tag{6}$$

where $B_{a1} \in \mathbb{R}^{(m-q)\times m}$, B_{a2} and $B_{b2} \in \mathbb{R}^{q\times m}$, $\boldsymbol{\beta}_1 \in \mathbb{R}^{m-q}$ and $\boldsymbol{\beta}_2 \in \mathbb{R}^{q}$.
The boundary conditions are *separated* if (7) reduces to

$$\begin{aligned} &B_{a1}\boldsymbol{y}(a) = \boldsymbol{\beta}_1, \\ &B_{b2}\boldsymbol{y}(b) = \boldsymbol{\beta}_2. \end{aligned} \tag{7}$$

A problem with non-separated or partially separated boundary conditions can be reduced to a problem with separated boundary conditions. We add q équations

$$\boldsymbol{z}' = \boldsymbol{0}$$

We obtain a system of $m+q$ equations with separated boundary conditions,

$$\boldsymbol{Y}' = \boldsymbol{0}, \quad \text{where} \quad \boldsymbol{Y} = \begin{pmatrix} \boldsymbol{y} \\ \boldsymbol{z} \end{pmatrix} \text{ et } \boldsymbol{F} = \begin{pmatrix} \boldsymbol{f} \\ \boldsymbol{0} \end{pmatrix},$$
$$\begin{pmatrix} B_{a1} & 0 \\ B_{a2} & -I \end{pmatrix} \boldsymbol{Y}(a) = \begin{pmatrix} \boldsymbol{\beta}_1 \\ 0 \end{pmatrix}, \quad (B_{b2} I)\boldsymbol{Y}(b) = \boldsymbol{\beta}_2.$$

Consider the problem

$$\begin{cases} \boldsymbol{y}' = \boldsymbol{f}(x, \boldsymbol{y}), \quad a < x < b, \\ \boldsymbol{g}(\boldsymbol{y}(a), \boldsymbol{y}(b)) = 0. \end{cases} \tag{8}$$

We can associate to it an IVP,

$$\begin{cases} \boldsymbol{w}' = \boldsymbol{f}(x, \boldsymbol{w}), \quad a < x < b, \\ \boldsymbol{w}(a) = \boldsymbol{s}. \end{cases} \tag{9}$$

where $\boldsymbol{s}$ is a parameter. Assuming that the solution of (20) exists, then $\boldsymbol{w}(x; \boldsymbol{s})$ is unique for each $\boldsymbol{s}$.

We see that $\boldsymbol{w}(x;\boldsymbol{s})$ is the solution of the boundary value problem (18) if we can choose $\boldsymbol{s}$ such that

$$\boldsymbol{\phi}(\boldsymbol{s}) \equiv \boldsymbol{g}(\boldsymbol{s}, \boldsymbol{w}(b, \boldsymbol{s})) = \mathbf{0}. \tag{10}$$

Theorem. Suppose that $\boldsymbol{f}(x, \boldsymbol{y})$ is continuous on

$$D = \{(x, \boldsymbol{y}) \ : \ a \leq x \leq b, |\boldsymbol{y} < \infty|\}$$

and Lipschitz continuous with respect to $\boldsymbol{y}$. Then there exist as many solutions of the boundary value problem (18) as distinct roots $\boldsymbol{s}^*$ of $\boldsymbol{\phi}(\boldsymbol{s}^*) = \mathbf{0}$. For each $\boldsymbol{s}^*$, the solution of the boundary value problem is

$$\boldsymbol{y}(x) = \boldsymbol{w}(x; \boldsymbol{s}^*).$$

In general, $\boldsymbol{y}$ is not unique.

Definition. A solution $\boldsymbol{y}$ of the boundary value problem is *locally unique* if there exists $\rho > 0$ such that among functions

$$F = \left\{ \boldsymbol{z} \; ; \; \boldsymbol{z} \in C[a,b], \; \sup_{a \le x \le b} |\boldsymbol{z}(x) - \boldsymbol{y}(x)| \le \rho \right\} \tag{11}$$

$\boldsymbol{y}$ is the only solution to the boundary value problem.

We will be interested in the local uniqueness of solutions with linear boundary conditions,

$$\begin{cases} \boldsymbol{y}' = \boldsymbol{f}(x, \boldsymbol{y}), \quad a < x < b, \\ B_a \boldsymbol{y}(a) + B_b \boldsymbol{y}(b) = \boldsymbol{\beta}. \end{cases} \tag{12}$$

To establish local uniqueness, we consider the variational problem

$$\begin{cases} \boldsymbol{z}' = A(x) \boldsymbol{z}, \quad a < x < b, \\ B_a \boldsymbol{z}(a) + B_b \boldsymbol{z}(b) = \boldsymbol{0}. \end{cases} \tag{13}$$

where

$$A(x) = A(x; \boldsymbol{y}(x)) = \left(\frac{\partial \boldsymbol{f}_i(x, \boldsymbol{y}(x))}{\partial y_j} \right)$$

is the Jacobian. If this (linear) variational problem has only one solution $\boldsymbol{z}(x) \equiv \mathbf{0}$, we say that the nonlinear problem (12) has an *isolated solution*.

Theorem. Suppose that $\boldsymbol{f}$ is smooth enough for the variational problem to be well-posed. If $\boldsymbol{y}(x)$ is an isolated solution of the boundary value problem (12), it is locally unique.

We now consider the linear boundary value problem,

$$\begin{cases} \boldsymbol{y}' = A(x)\boldsymbol{y} + \boldsymbol{q}(x), \quad a < x < b, \\ B_a \boldsymbol{y}(a) + B_b \boldsymbol{y}(b) = \boldsymbol{\beta}. \end{cases} \tag{14}$$

We assume that $A(x)$ and $\boldsymbol{q}(x)$ are continuous.

Theorem. The problem (14) has a unique solution $\boldsymbol{y}(x)$ iff the matrix

$$Q = B_a Y(a) + B_b Y(b) \tag{15}$$

is not singular.

The matrix Y is a fundamental solution, i.e., solution of the initial value problem

$$\begin{aligned} \frac{\mathrm{d}Y(x;t)}{\mathrm{d}x} &= A(x)Y(x;t), \\ Y(t,t) &= I, \end{aligned} \tag{16}$$

where $t \in [a,b]$ is a parameter. Then the solution to the boundary value problem is

$$\begin{aligned} \boldsymbol{y}(x) = Y(x)Q^{-1} &\left(\boldsymbol{\beta} - B_b Y(b) \int_a^b Y^{-1}(t)\boldsymbol{q}(t)\mathrm{d}t\right) \\ &+ Y(x) \int_a^x Y^{-1}(t)\boldsymbol{q}(t)\mathrm{d}t. \end{aligned} \tag{17}$$

There are two groups of methods for the solution of ordinary differential equations with boundary conditions specified for two different points:

- *Shooting methods*, where the problem is reduced to the solution of several initial value problems;
- *Global methods*, where the approximation is done simultaneously in the whole domain.

Shooting methods.

Consider the problem

$$\begin{cases} \boldsymbol{y}' = \boldsymbol{f}(x, \boldsymbol{y}), \quad a < x < b, \\ \boldsymbol{g}(\boldsymbol{y}(a), \boldsymbol{y}(b)) = 0. \end{cases} \tag{18}$$

To obtain a solution of (18), we solve

$$\boldsymbol{g}(\boldsymbol{s}, \boldsymbol{w}(b, \boldsymbol{s})) = \mathbf{0} \tag{19}$$

using a method for solving systems of non-linear equations. The values of $\boldsymbol{w}(b, \boldsymbol{s})$ are obtained with one of the numerical methods of solution of initial value problems applied to

$$\begin{cases} \boldsymbol{w}' = \boldsymbol{f}(x, \boldsymbol{w}), \quad a < x < b, \\ \boldsymbol{w}(a) = \boldsymbol{s}. \end{cases} \tag{20}$$

Superposition method. It is a shooting method for linear problems

$$\begin{cases} \boldsymbol{y}' = A(x)\boldsymbol{y} + \boldsymbol{q}(x), \quad a < x < b, \\ B_a\boldsymbol{y}(a) + B_b\boldsymbol{y}(b) = \boldsymbol{\beta}. \end{cases} \tag{21}$$

Thanks to linearity, we can represent the solution of (21) as a linear combination of the solutions of the associated initial value problems.

The general solution of (21) is

$$\boldsymbol{y}(x) = Y(x)\boldsymbol{s} + \boldsymbol{v}(x), \tag{22}$$

where $Y(x)$ is the fundamental solution, $\boldsymbol{s}$ is vector of parameters and $\boldsymbol{v}$ is a particular solution:

$$\begin{cases} Y' = A(x)Y, \quad a < x < b, \\ Y(a) = I. \end{cases} \tag{23}$$

and

$$\begin{cases} \boldsymbol{v}' = A(x)\boldsymbol{v} + \boldsymbol{q}(x), \quad a < x < b, \\ \boldsymbol{v}(a) = \boldsymbol{\alpha}, \end{cases} \tag{24}$$

with any $\boldsymbol{\alpha}$ (e.g. $\boldsymbol{\alpha} = \mathbf{0}$).
We use the boundary condition of (21) to calculate $\boldsymbol{s}$,

$$\boldsymbol{\beta} = B_a(I\boldsymbol{s} + \boldsymbol{v}(a)) + B_b(Y(b)\boldsymbol{s} + \boldsymbol{v}(b)) \quad \text{or}$$

$$Q\boldsymbol{s} = \hat{\boldsymbol{\beta}}, \quad \text{with} \quad Q := B_a + B_bY(b), \;\; \hat{\boldsymbol{\beta}} := \boldsymbol{\beta} - B_a\boldsymbol{v}(a) - B_b\boldsymbol{v}(b).$$

Algorithm (shooting / superposition method).
Input : Points $a = x_0 < x_1 < ... < x_J = b$, matrices $A(x)$, B_a and B_b.
Output : $\boldsymbol{y}(x_j)$, $j = 0, 1, ..., J$.

1. Solve (23) and (24) numerically to obtain $Y(b)$ and $\boldsymbol{v}(b)$ (with arbitrary $\boldsymbol{\alpha}$).
2. Calculate Q and $\hat{\boldsymbol{\beta}}$.
3. Solve $Q\boldsymbol{s} = \hat{\boldsymbol{\beta}}$.
4. $\boldsymbol{y}(a) := \boldsymbol{s} + \boldsymbol{\alpha}$.
5. Solve the initial value problem with $\boldsymbol{y}(a)$ as initial condition.

Suppose that the method we use to solve initial value problems is adaptive and that it ensures error less than *tol*. The error of the shooting method in this case can be estimated as

$$||\boldsymbol{y}_i - \boldsymbol{y}(x_i)|| \leq K \; tol \; \kappa, \tag{25}$$

where $K \sim 1$ and κ is the *condition number* of the boundary value problem. To calculate the conditioning number, we consider the differential operator

$$L\boldsymbol{y} := \boldsymbol{y}'(x) - A(x)\boldsymbol{y}(x), \quad a < x < b \tag{26}$$

and the boundary condition operator

$$B\boldsymbol{y} := B_a\boldsymbol{y}(a) + B_b\boldsymbol{y}(b). \tag{27}$$

The boundary value problem takes the form

$$\begin{cases} L\boldsymbol{y} = \boldsymbol{q}, \\ B\boldsymbol{y} = \boldsymbol{\beta}. \end{cases} \tag{28}$$

We define the fundamental solution

$$\begin{cases} L\Phi = 0, \\ B\Phi = I. \end{cases} \tag{29}$$

We use it to rewrite the solution of the boundary value problem in the form

$$\boldsymbol{y}(x) = \Phi(x)\boldsymbol{\beta} + \int_a^b G(x,t)\boldsymbol{q}(t)\mathrm{d}t \tag{30}$$

where G is the Green function

$$G(x,t) = \begin{cases} \Phi(x)B_a\Phi(a)\Phi^{-1}(t), & t \leq x, \\ -\Phi(x)B_b\Phi(b)\Phi^{-1}(t), & t > x. \end{cases} \tag{31}$$

We obtain

$$||\boldsymbol{y}||_\infty = \kappa(|\boldsymbol{\beta}| + \int_a^b |\boldsymbol{q}(t)|\mathrm{d}t), \tag{32}$$

where

$$\kappa = \max(||G||_\infty, ||\Phi||_\infty). \tag{33}$$

We now consider the stability of the superposition method. We assume that the problem is well-conditioned ($\kappa \sim 1$). It turns out that the method can be unstable in the case where the fundamental solution has modes which vary rapidly but not their linear combination which gives the solution of the boundary value problem. Then even if we find the approximate value $\hat{\boldsymbol{s}}$ of $\boldsymbol{s} = \boldsymbol{y}(a) - \boldsymbol{\alpha}$ with good precision, the error will be amplified as

$$||\hat{\boldsymbol{y}} - \boldsymbol{y}|| \sim ||Y||\ ||\hat{\boldsymbol{s}} - \boldsymbol{s}|| \approx \epsilon_m e^{||A||_\infty (b-a)}. \tag{34}$$

Multiple shooting method. We can decrease the error (34) if we decrease the size of the integration domain. We subdivide the domain into intervals $[\xi_i, \xi_{i+1}]$, where $a = \xi_1 < \xi_2 < ... < \xi_{N+1} = b$. The solution in each subdomain is

$$\boldsymbol{y}(x) = Y_i(x)\boldsymbol{s}_i + \boldsymbol{v}_i(x), \quad \xi_i \leq x \leq \xi_{i+1}, \tag{35}$$

where

$$\begin{cases} Y_i' = A(x)Y_i, & \xi_i < x < \xi_{i+1}, \\ Y_i(\xi_i) = I. \end{cases} \tag{36}$$

and

$$\begin{cases} \boldsymbol{v}_i' = A(x)\boldsymbol{v}_i + \boldsymbol{q}(x), & \xi_i < x < \xi_{i+1}, \\ \boldsymbol{v}_i(\xi_i) = \mathbf{0}, \end{cases} \tag{37}$$

We have mN parameters

$$\boldsymbol{s}^T = (\boldsymbol{s}_1^T, \boldsymbol{s}_2^T, ..., \boldsymbol{s}_N^T) \tag{38}$$

to be determined so that the solution $\boldsymbol{y}$ is continuous and so that it satisfies the boundary conditions,

$$Y_i(\xi_{i+1})\boldsymbol{s}_i + \boldsymbol{v}(\xi_{i+1}) = Y_{i+1}(\xi_{i+1})\boldsymbol{s}_{i+1} + \boldsymbol{v}_{i+1}(\xi_{i+1}), \tag{39}$$

$$B_a(Y_1(a)\boldsymbol{s}_1 + \boldsymbol{v}_1(a)) + B_b(Y_N(b)\boldsymbol{s}_N + \boldsymbol{v}_N(b)) = \boldsymbol{\beta}. \tag{40}$$

We obtain a linear system $M\boldsymbol{s} = \hat{\boldsymbol{\beta}}$,

$$
\begin{pmatrix}
-Y_1(\xi_2) & I & & & \\
& -Y_2(\xi_3) & I & & \\
& & \cdots & & \\
& & & -Y_{N-1}(\xi_N) & I \\
B_a & & & & B_b Y_N(b)
\end{pmatrix}
\begin{pmatrix}
\boldsymbol{s}_1 \\ \boldsymbol{s}_2 \\ \vdots \\ \boldsymbol{s}_{N-1} \\ \boldsymbol{s}_N
\end{pmatrix}
=
\begin{pmatrix}
\boldsymbol{v}_1(\xi_2) \\ \boldsymbol{v}_2(\xi_3) \\ \vdots \\ \boldsymbol{v}_{N-1}(\xi_N) \\ \boldsymbol{\beta} - B_b \boldsymbol{v}_N(b)
\end{pmatrix}
\tag{41}
$$

The solution to the boundary value problem is

$$
\boldsymbol{y}(x_i) = \boldsymbol{s}_i.
$$

Algorithm (multiple shooting / superposition method).
Input: Points $a = \xi_1 < \xi_2 < ... < \xi_{N+1} = b$, matrices $A(x)$, B_a, B_b.
Output: $\boldsymbol{y}(\xi_i)$, $i = 1, ..., N$.

1. Solve the initial value problems to get $Y_i(x_{i+1})$ and $\boldsymbol{v}(x_{i+1})$, $i = 1, ..., N$.
2. Build M and $\hat{\boldsymbol{\beta}}$.
3. Solve $M\boldsymbol{s} = \hat{\boldsymbol{\beta}}$.

We will show that

$$||\hat{\boldsymbol{s}} - \boldsymbol{s}|| \leq KNJ\epsilon_m \max_{1 \leq i \leq N} ||Y_i||_\infty, \tag{42}$$

where $K \sim 1$ and J is the number of steps used for the solution of initial value problems. We choose N so that the error is less than the given tolerance.

Now consider the nonlinear problem,

$$\begin{cases} \boldsymbol{y}' = \boldsymbol{f}(x, \boldsymbol{y}), \quad a < x < b, \\ \boldsymbol{g}(\boldsymbol{y}(a), \boldsymbol{y}(b)) = 0. \end{cases} \tag{43}$$

An additional difficulty for the simple shooting method is that the solution to the initial value problem associated with may cease to exist on $[a, b]$ for certain values of $\boldsymbol{s}$. This problem is relieved if the multiple shot method is used.

The connection and boundary conditions are

$$\begin{cases} \boldsymbol{y}_i(\xi_{i+1}; \boldsymbol{s}_i) = \boldsymbol{s}_{i+1}, \quad 1 \leq i \leq N-1, \\ \boldsymbol{g}(\boldsymbol{s}_1, \boldsymbol{y}_N(b; \boldsymbol{s}_N)) = 0. \end{cases} \tag{44}$$

We use the Newton–Raphson method to solve

$$\boldsymbol{F}(\boldsymbol{s}) = 0, \quad \text{where} \quad \boldsymbol{F}(\boldsymbol{s}) = \begin{pmatrix} \boldsymbol{s}_2 - \boldsymbol{y}_1(\xi_2; \boldsymbol{s}_1) \\ \boldsymbol{s}_3 - \boldsymbol{y}_2(\xi_3; \boldsymbol{s}_2) \\ \dots \\ \boldsymbol{s}_N - \boldsymbol{y}_{N-1}(\xi_N; \boldsymbol{s}_{N-1}) \\ \boldsymbol{g}(\boldsymbol{s}_1, \boldsymbol{y}_N(b; \boldsymbol{s}_N)) \end{pmatrix}. \tag{45}$$

The Jacobian $\boldsymbol{F}$ is

$$\begin{pmatrix} -Y_1(\xi_2) & I & & & \\ & -Y_2(\xi_3) & I & & \\ & & \dots & & \\ & & & -Y_{N-1}(\xi_N) & I \\ B_a & & & & B_b Y_N(b) \end{pmatrix} \tag{46}$$

where $Y_i(x) = Y_i(x; \xi_i, \boldsymbol{s}_i)$ is the solution of

$$\begin{cases} Y_i' = A_i(x) Y_i, & \xi_i < x < \xi_{i+1}, \\ Y_i(\xi_i) = I \end{cases} \tag{47}$$

with

$$A_i = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{y}}(x, \boldsymbol{y}_i(x; \boldsymbol{s}_i)) \tag{48}$$

and

$$B_a = \frac{\partial \boldsymbol{g}(\boldsymbol{u}, \boldsymbol{v})}{\boldsymbol{u}}, \quad B_b = \frac{\partial \boldsymbol{g}(\boldsymbol{u}, \boldsymbol{v})}{\boldsymbol{v}} \quad \text{à} \quad \boldsymbol{u} = \boldsymbol{s}_1, \quad \boldsymbol{v} = \boldsymbol{y}_N(b, \boldsymbol{s}_N). \tag{49}$$

One can use the fact that the Jacobian matrix is sparse to reduce the computational cost.

Global methods. Finite difference methods.

One can approximate

$$\boldsymbol{y}' = A(x)\boldsymbol{y} + \boldsymbol{q}(x), \quad a < x < b \tag{50}$$

using the *trapezoidal* formula

$$\frac{\boldsymbol{y}_{i+1} - \boldsymbol{y}_i}{h_i} = \frac{1}{2}[A(x_{i+1})\boldsymbol{y}_{i+1} + A(x_i)\boldsymbol{y}_i] + \frac{1}{2}[\boldsymbol{q}(x_{i+1}) + \boldsymbol{q}(x_i)], \quad 1 \leq i \leq N. \tag{51}$$

of the *midpoint* formula

$$\frac{\boldsymbol{y}_{i+1} - \boldsymbol{y}_i}{h_i} = \frac{1}{2}A(x_{i+1/2})(\boldsymbol{y}_{i+1} + \boldsymbol{y}_i) + \boldsymbol{q}(x_{i+1/2}), \quad 1 \leq i \leq N. \tag{52}$$

With the boundary conditions

$$B_a\boldsymbol{y}(x_1) + B_b\boldsymbol{y}(x_{N+1}) = \boldsymbol{\beta}$$

we obtain the system

$$
\begin{pmatrix}
S_1 & R_1 & & & \\
& S_2 & R_2 & & \\
& & \cdots & & \\
& & & S_N & R_N \\
B_a & & & & B_b
\end{pmatrix}
\begin{pmatrix}
\boldsymbol{y}_1 \\ \boldsymbol{y}_2 \\ \vdots \\ \boldsymbol{y}_N \\ \boldsymbol{y}_{N+1}
\end{pmatrix}
=
\begin{pmatrix}
\boldsymbol{q}_1 \\ \boldsymbol{q}_2 \\ \vdots \\ \boldsymbol{q}_N \\ \boldsymbol{\beta}
\end{pmatrix}
\tag{53}
$$

where

$$
S_i = -\frac{1}{h_i}I - \frac{1}{2}A(x_i), \quad R_i = \frac{1}{h_i}I - \frac{1}{2}A(x_{i+1}), \quad \boldsymbol{q}_i = \frac{1}{2}[\boldsymbol{q}(x_{i+1}) + \boldsymbol{q}(x_i)] \tag{54}
$$

for the trapezoidal method and

$$
S_i = -\frac{1}{h_i}I - \frac{1}{2}A(x_{i+1/2}), \quad R_i = \frac{1}{h_i}I - \frac{1}{2}A(x_{i+1/2}), \quad \boldsymbol{q}_i = \boldsymbol{q}(x_{i+1/2}) \tag{55}
$$

for the midpoint method.

To study the properties of these methods, we define the operator

$$L_h \boldsymbol{y}_i = \frac{\boldsymbol{y}_{i+1} - \boldsymbol{y}_i}{h_i} - \Psi(\boldsymbol{y}_i, \boldsymbol{y}_{i+1}; x_i, h_i) \tag{56}$$

For example, for the midpoint method

$$\Psi(\boldsymbol{y}_i, \boldsymbol{y}_{i+1}; x_i, h_i) = \frac{1}{2} A(x_{i+1/2})(\boldsymbol{y}_{i+1} + \boldsymbol{y}_i). \tag{57}$$

The consistency error is defined as

$$\boldsymbol{\tau}_i = L_h \boldsymbol{y}(x_i) - \boldsymbol{q}_i. \tag{58}$$

The method is *consistent of order* p if $\max_{1 \leq i \leq N} |\boldsymbol{\tau}_i| = \mathcal{O}(h^p)$.
The method is *stable* if there are constants $K = \mathcal{O}(\kappa)$ and h_0 such that for all $h < h_0$,

$$|\boldsymbol{v}_i| \leq K \max[|B_a \boldsymbol{v}_1 + B_b \boldsymbol{v}_{N+1}|, \max_{1 \leq j \leq N} |L_h \boldsymbol{v}_j|], \quad 1 \leq i \leq N+1. \tag{59}$$

Indeed, it can be shown that $K \leq \kappa(b - a + 1) + \mathcal{O}(h)$ for the above methods.

The method is *convergent* if

$$\max_{1 \leq i \leq N+1} |\boldsymbol{y}_i - \boldsymbol{y}(x_i)| \to 0 \tag{60}$$

in the limit of $h \to 0$.

Consider the global error

$$\boldsymbol{e}_i = \boldsymbol{y}(x_i) - \boldsymbol{y}_i. \tag{61}$$

It is related with the consistency error,

$$L_h \boldsymbol{e} = \boldsymbol{\tau}_i \tag{62}$$

and the boundary conditions

$$B_a \boldsymbol{e}_1 + B_b \boldsymbol{e}_{N+1} = \boldsymbol{0}. \tag{63}$$

Stability gives

$$|\boldsymbol{e}_i| \leq const\ K\ h^p \tag{64}$$

The trapezoidal formula for nonlinear problems takes the form

$$\begin{cases} \dfrac{\boldsymbol{y}_{i+1} - \boldsymbol{y}_i}{h_i} = \dfrac{1}{2}[\boldsymbol{f}(x_{i+1}, \boldsymbol{y}_{i+1}) + \boldsymbol{f}(x_i, \boldsymbol{y}_i)], \quad 1 \leq i \leq N, \\ \boldsymbol{g}(\boldsymbol{y}_1, \boldsymbol{y}_{N+1}) = \boldsymbol{0}. \end{cases} \tag{65}$$

To solve this system we can use the Newton–Raphson method written in the form

$$\begin{cases} \dfrac{\boldsymbol{w}_{i+1} - \boldsymbol{w}_i}{h_i} - \dfrac{1}{2}[A(x_{i+1})\boldsymbol{w}_{i+1} + A(x_i)\boldsymbol{w}_i] \\ \quad = -\dfrac{\boldsymbol{y}_{i+1}^{[m]} - \boldsymbol{y}_i^{[m]}}{h_i} + \dfrac{1}{2}[\boldsymbol{f}(x_{i+1}, \boldsymbol{y}_{i+1}^{[m]}) + \boldsymbol{f}(x_i, \boldsymbol{y}_i^{[m]})], \quad 1 \leq i \leq N, \\ B_a \boldsymbol{w}_1 + B_b \boldsymbol{w}_{N+1} = -\boldsymbol{g}(\boldsymbol{y}_1^{[m]}, \boldsymbol{y}_{N+1}^{[m]}). \end{cases} \tag{66}$$

and

$$\boldsymbol{y}_i^{[m+1]} = \boldsymbol{y}_i^{[m]} + \boldsymbol{w}_i. \tag{67}$$