# Numerical methods to solve boundary value problems for ODEs

In this note we provide a brief overview of the most common numerical methods to approximate the solution of a boundary value problem for an ODE or a system of ODEs. In particular,

- Shooting method;
- Methods based on finite-differences or collocation;
- Methods based on weighted residuals (e.g., Galerkin, least squares).

When applying these methods to a boundary value problem, we will always assume that the problem has at least one solution[1].

## Shooting method

The shooting method is a method for solving a boundary value problem by reducing it an to initial value problem which is then solved multiple times until the boundary condition is met. To describe the method let us first consider the following two-point boundary value problem for a second-order nonlinear ODE with Dirichlet boundary conditions

$$\begin{cases} \dfrac{d^2y}{dt^2} = f\left(\dfrac{dy}{dt}, y, t\right) \qquad t \in [0,T] \\ y(0) = \alpha \\ y(T) = \beta \end{cases} \tag{1}$$

We have seen in previous lecture that this problem can have an infinite number of solutions (e.g., the pendulum problem), a unique solution, or no solution at all. The shooting method replaces the boundary condition $y(T) = \beta$ in the system (1) with the initial condition $dy(0)/dt = v$, for an unknown "slope" $v$, and attempts to find $v$ by using an iterative root-finding algorithm or an optimization method so that the quantity

$$E(v) = y(T; v) - \beta \tag{2}$$

(or $E(v)^2$ in the case of optimization) is equal to zero[2]. In equation (2) $y(T; v)$ represents the solution (flow) to the initial value problem

$$\begin{cases} \dfrac{dy}{dt} = z \\ \dfrac{dz}{dt} = f(z, y, t) \\ y(0) = \alpha \\ z(0) = v \end{cases} \tag{4}$$

at time $T$. The notation $y(T; v)$ emphasizes that the solution of (4) depends on $v$. Recall, in fact, that if the right hand side of the system (4), i.e., $(z, f(y, z, t))$ is of class $C^k$ then the solution $(y(t; \alpha, v), z(t; \alpha, v))$ is of class $C^k$ in the initial condition $(\alpha, v)$. This implies the following lemma.

[1] Recall that a boundary value problem can have a unique solution, an infinite number of solution or no solution at all.

[2] Note that the shooting method is essentially a *control problem* of the form

$$\min_{v\in\mathbb{R}} |y(T; v) - \beta| \quad \text{subject to (4).} \tag{3}$$

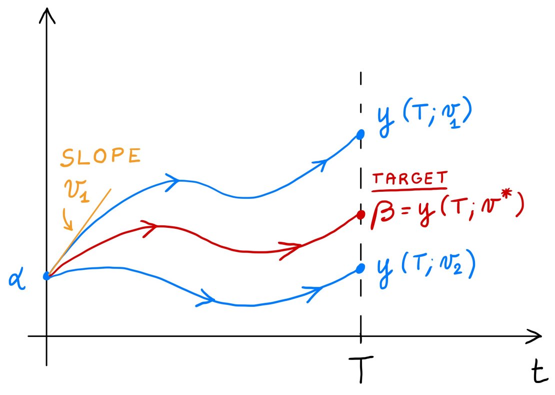

Figure 1: Sketch of the shooting method to solve the two-point boundary value problem (1). Basically, we look for the slope $v$ of the solution at initial time ("shoot" with elevation $v$), so that we "hit" the target $y(T) = \beta$ at time $t = T$.

**Lemma 1.** If the initial value problem (4) is well-posed, i.e., if the right-hand-side is Lipshitz and $T$ is within the domain of existence of the solution, then the error function (2) is at least continuous in $v$. Moreover, if $f$ is of class $C^k$ then the error function (2) is of class $C^k$ (differentiable $k$ times with continuous derivative).

In Figure 1 we provide a sketch of the shooting method for the initial value problem (4) corresponding to the boundary value problem (1).

With the error function (2) available, we can construct an iterative procedure that generates a sequence of slopes $\{v_0, v_1, v_2, \ldots\}$ with the property

$$\lim_{k\to\infty} E(v_k) = 0 \qquad \text{(rootfinding methods)}. \tag{5}$$

To generate a sequence $\{v_k\}$ satisfying (5) we can use any rootfinding method for scalar equations, such as the bisection method, the secant method or the Newton's method. Note that the function $E(v)$ is not explicitly known, but it is certainly continuous (or even smoother depending on the regularity of $f$), and it can be sampled at any point $v$ we like. To compute such samples, we just need to integrate (4) forward in time up to $t = T$ for the initial condition $z(0) = v$ and then evaluate (2).

**Basic rootfinding algorithms.** Let us briefly describe three basic rootfinding methods we can use to generate a sequence $\{v_0, v_1, v_2, \ldots\}$ with the property (5). These methods are sketched in Figure 2.

- **Bisection method:** Given any two initial guesses $v_0$ and $v_1$ we solve the initial value problem (4) to obtain
$$E(v_0) = y(T; v_0) - \beta, \qquad \text{and} \qquad E(v_1) = y(T; v_1) - \beta. \tag{6}$$
If the sign of the product $E(v_0)E(v_1)$ is strictly positive, then we cannot claim that there exists a zero $v^*$ of the function $E(v)$ in the interval $[v_0, v_1]$ (although there maybe actually one). On the other hand, if the sign of the product $E(v_0)E(v_1)$ is strictly negative then there exists a point $v^*$ within the interval $[v_0, v_1]$ such that $E(v^*) = 0$. To find $v^*$ we split the interval $[v_0, v_1]$ in half (hence the name "bisection method"), and evaluate $E(v)$ at $v_2 = (v_1 + v_0)/2$. At this point we proceed as before, i.e., if $E(v_0)E(v_2) > 0$ then we forget about the interval $[v_0, v_2]$ and split $[v_2, v_1]$ in half, evaluate $E(v)$

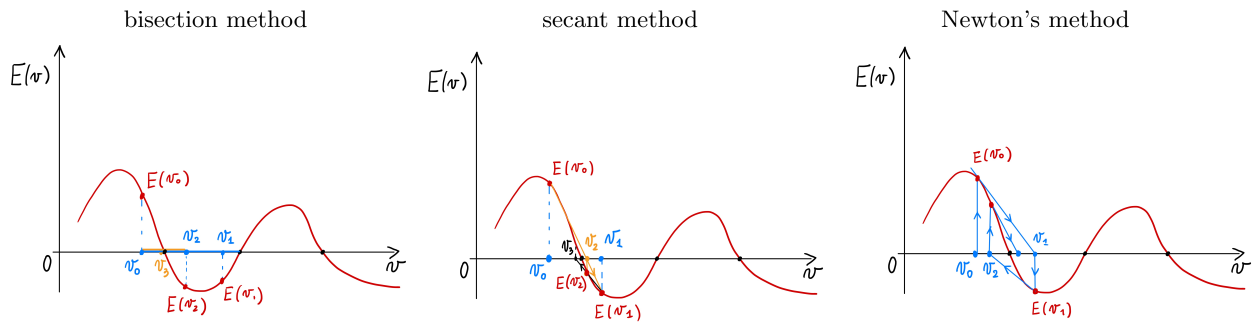

Figure 2: Sketch of the most common rootfinding methods applied to equation (2).

at $(v_2 + v_1)/2$ and so on so forth. On the other hand, if $E(v_0)E(v_2) < 0$ then we forget about the interval $[v_2, v_1]$ and split $[v_0, v_2]$ in half, evaluate $E(v)$ at $(v_2 + v_0)/2$, etc. The bisection procedure until either the function value $E(v_k)$ or the difference between two subsequent iterates $|v_{k+1} - v_k|$ or both is smaller than a prescribed tolerance. For more details on the bisection method see [2, §6.2.1]. Convergence of the bisection method is, on average, linear with the interation number.

- **Secant method:** Similarly to the bisection method, we start with two initial guesses $v_0$ and $v_1$ (hopefully close enough to the zero $v^*$ we are interested in), and evaluate $E(v_0)$ and $E(v_1)$. Then we construct the line passing through $(v_0, E(v_0))$ and $(v_1, E(v_1))$ and extrapolate such a line onto the $x$ axis (see Figure 2). By doing this iteratively, we obtain the sequence (see [2, p. 254]

$$v_{k+1} = v_k - \frac{v_k - v_{k-1}}{E(v_k) - E(v_{k-1})} E(v_k) \qquad k = 1, 2, \dots \tag{7}$$

The convergence order of the secant method is $(\sqrt{5} + 1)/2$.

- **Newton method:** To determine a zero of (2) with the Newton method, we start from an initial guess $v_0$, and compute $E(v_0)$ and also $E'(v_0)$, i.e., the first derivative of the error function (2) evaluated at $v_0$. This allows us to initialize the iterative formula [2, p. 255]

$$v_{k+1} = v_k - \frac{E(v_k)}{E'(v_k)}. \tag{8}$$

The first derivative of $E(v)$ is usually not available, but it can be estimated numerically, e.g., with finite differences, based on samples of $v$ that are sufficiently close. A better (more accurate) approach relies on deriving an evolution equation for $dy(t;v)/dv$, solve such such equation, and evaluate the solution at final time to obtain

$$E'(v) = \frac{\partial y(T;v)}{\partial v}. \tag{9}$$

The evolution equation for $dy(t;v)/dv$ can be determined by differentiating (4) with respect to $v$. This yields the linear initial value problem

$$\begin{cases} \dfrac{d^2\eta}{dt^2} = \dfrac{\partial f}{\partial y'}\dfrac{d\eta}{dt} + \dfrac{\partial f}{\partial y}\eta \\ \eta(0;v) = 0 \\ \dfrac{d\eta(0;v)}{dt} = 1 \end{cases} \tag{10}$$

where

$$\eta(t;v) = \frac{\partial y(t;v)}{\partial v}. \tag{11}$$

Note, in fact, that if differentiate the ODE in (1) with respect to $v$ we obtain

$$\frac{d^2y}{dt^2} = f\left(\frac{dy}{dt}, y, t\right) \qquad \Rightarrow \qquad \frac{d^2}{dt^2}\left(\frac{dy}{dv}\right) = \frac{\partial f}{\partial y'}\frac{d}{dt}\left(\frac{\partial y}{\partial v}\right) + \frac{\partial f}{\partial y}\frac{\partial y}{\partial v}. \tag{12}$$

The system (10) depends on the solution of (4), i.e., and it can be solved only if the solution to (4) is available[3]. In summary, to solve (1) with the Newton method we proceed as follows:

1. Choose $v_0$.
2. Solve the initial value problem

$$\begin{cases} \dfrac{d^2y}{dt^2} = f\left(\dfrac{dy}{dt}, y, t\right) \\ y(0; v) = \alpha \\ \dfrac{dy(0; v)}{dt} = v \\ \dfrac{d^2\eta}{dt^2} = \dfrac{\partial f}{\partial y'}\dfrac{d\eta}{dt} + \dfrac{\partial f}{\partial y}\eta \\ \eta(0; v) = 0 \\ \dfrac{d\eta(0; v)}{dt} = 1 \end{cases} \tag{13}$$

3. Evaluate

$$y(T; v_0) \qquad \text{and} \qquad \eta(T; v_0) = \frac{\partial y(T; v_0)}{\partial v} \tag{14}$$

at final time $T$.

4. Update the initial guess $v_0$ as

$$v_1 = v_0 - \frac{y(T; v_0) - \beta}{\eta(T; v_0)}. \tag{15}$$

5. Go to point 2. and repeat the calculation with the updated initial condition $dy/dt(0; v) = v_1$.

**Example (pendulum equations):** Consider the two-point boundary value problem for the pendulum equation

$$\begin{cases} \dfrac{d^2\theta}{dt^2} = -\sin(\theta) \\ \theta(0) = \alpha \\ \theta(T) = \beta \end{cases} \tag{16}$$

The system of equations (13) corresponding to the pendulum BVP (16) is

[3] If the ODE is linear then (10) can be solved independently of (4).

$$\begin{cases} \dfrac{d^2\theta}{dt^2} = -\sin(\theta) \\ \theta(0) = \alpha \\ \dfrac{d\theta(0)}{dt} = v_k \\ \dfrac{d^2\eta}{dt^2} = \cos(\theta)\eta \\ \eta(0; v) = 0 \\ \dfrac{d\eta(0; v)}{dt} = 1 \end{cases} \tag{17}$$

By solving this system and updating $v_k$ according to

$$v_{k+1} = v_k - \frac{y(T; v_k) - \beta}{\eta(T; v_k)}, \tag{18}$$

for a properly chosen $v_0$, we eventually converge to one of the solutions of (16).

**Optimization methods.** A different class of techniques that can be used in the shooting method relies on *optimization*. In the optimization setting, we seek for a minimizer, e.g., of the function

$$C(v) = (y(T; v) - \beta)^2, \tag{19}$$

at or nearby $C(v) = 0$. For example, we can use a descent method [2, p. 305] to minimize (19), e.g. the classical gradient descent scheme

$$v_{v+1} = v_k - \gamma_k C'(v_k), \tag{20}$$

where

$$\gamma_k = \left| \frac{v_k - v_{k-1}}{C'(v_k) - C'(v_{k-1})} \right|, \qquad C'(v_k) = 2E(v_k)E'(v_k). \tag{21}$$

The function $C'(v_k)$ is not known analytically, but needs to be evaluated as in the Newtown's method. In particular, each step of gradient descent requires the evaluation of both $E(v_k)$ and $E'(v_k)$.

It can be shown that the shooting method can be *extremely sensitive* to the choice of the initial condition $v_0$. Indeed the set of initial conditions for which the method converges (i.e., the basin of attraction of a solution) is often concentrated in a small neighborhood of the solution.

**Shooting method for higher-order ODEs and system of ODEs.** A two-point boundary value problem for a system of $n$-dimensional ODEs can be written in the abstract form

$$\begin{cases} \dfrac{d\boldsymbol{y}}{dt} = \boldsymbol{f}(\boldsymbol{y}, t) \qquad t \in [0, T] \\ \boldsymbol{g}(\boldsymbol{y}(0), \boldsymbol{y}(T)) = \boldsymbol{0} \end{cases} \tag{22}$$

where $\boldsymbol{g} \in \mathbb{R}^n$ is a nonlinear function. Every two-point boundary value problem we considered so far can be written in this form, upon definition of appropriate phase variables and boundary function $\boldsymbol{g}$. Let us provide an example of a boundary value problem involving a fourth-order ODE.

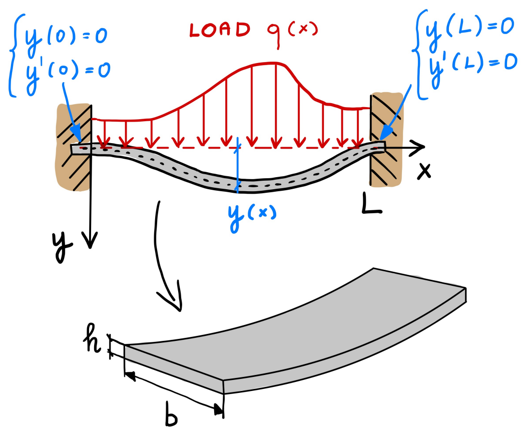

Figure 3: Sketch of the fully clamped Euler-Bernoulli beam modeled by the two-point boundary value problem (22).

**Euler-Bernoulli beam equations.** An example of a boundary value problem in the form (22) is the equation describing the displacement of a fully clamped Euler-Bernoulli beam subject to a load $q(x)$

$$\begin{cases} EI\dfrac{d^4y}{dx^4} = q(x) \\ y(0) = 0 \\ y(L) = 0 \\ \dfrac{dy(0)}{dx} = 0 \\ \dfrac{dy(L)}{dx} = 0 \end{cases} \tag{23}$$

Here, $E$ is the modulus of elasticity[4] of the beam, and $I$ is the flexural moment of inertia. For a square section of thickness $h$ and width $b$ (see Figure 3) we have

$$I = \frac{bh^3}{12}. \tag{24}$$

Upon definition of

$$z_0(x) = y(x), \qquad z_1(x) = \frac{dz_0(x)}{dx} \qquad z_2(x) = \frac{dz_1(x)}{dx}, \qquad z_3(x) = \frac{dz_2(x)}{dx} \tag{25}$$

we can rewrite (23) as

$$\begin{cases} \dfrac{dz_3}{dx} = \dfrac{q(x)}{EI}, \qquad \dfrac{dz_2}{dx} = z_3(x), \qquad \dfrac{dz_1}{dx} = z_2(x), \qquad \dfrac{dz_0}{dx} = z_1(x), \\ z_0(0) = 0 \\ z_0(L) = 0 \\ z_1(0) = 0 \\ z_1(L) = 0 \end{cases} \tag{26}$$

[4]For stainless steel we have $E \simeq 200$ GPa.

i.e., as a system of four first-order ODEs with four simple boundary condition conditions involving $z_0$ and $z_1$ at $x = 0$ and $x = L$. To solve (26) with the shooting method, e.g., by using Newton's iterations, we proceed as follows. We first replace the boundary value problem with the initial value problem

$$\begin{cases} \dfrac{dz_3}{dx} = \dfrac{q(x)}{EI}, \qquad \dfrac{dz_2}{dx} = z_3(x), \qquad \dfrac{dz_1}{dx} = z_2(x), \qquad \dfrac{dz_0}{dx} = z_1(x), \\ z_0(0) = 0 \\ z_1(0) = 0 \\ z_2(0) = v_1 \\ z_3(0) = v_2 \end{cases} \tag{27}$$

depending on two unknown parameters $v_1$ and $v_2$. To determine these parameters, we define the vector-valued error function

$$\boldsymbol{E}(\boldsymbol{v}) = \begin{bmatrix} z_0(T;\boldsymbol{v}) - 0 \\ z_1(T;\boldsymbol{v}) - 0 \end{bmatrix}, \qquad \boldsymbol{v} = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}. \tag{28}$$

Clearly, the dependence of $z_0(T;\boldsymbol{v})$ and $z_1(T;\boldsymbol{v})$ on $\boldsymbol{v}$ is affine (flow map generated by a linear system driven by $q(x)$). This allows us to avoid Newton's iterations and solve the linear system

$$\boldsymbol{E}(\boldsymbol{v}) = \boldsymbol{0} \tag{29}$$

for $\boldsymbol{v}$. Note that $z_0(T;\boldsymbol{v})$ and $z_1(T;\boldsymbol{v})$ involve integrals of $q(x)$, and therefore the solution to (29) is not the trivial vector $\boldsymbol{v} = 0$.

More generally, if the system is nonlinear, then we can use Newton's iterations to compute the solution to the problem. To this end, suppose we are given a boundary value problem for a fourth-order system of the form

$$\begin{cases} \dfrac{dz_3}{dx} = f(z_2, z_1, z_0), \qquad \dfrac{dz_2}{dx} = z_3(x), \qquad \dfrac{dz_1}{dx} = z_2(x), \qquad \dfrac{dz_0}{dx} = z_1(x), \\ z_0(0) = 0 \\ z_0(L) = 0 \\ z_1(0) = 0 \\ z_1(L) = 0 \end{cases} \tag{30}$$

We rewrite this system as an initial value problem with unknown $v_1$ and $v_2$

$$\begin{cases} \dfrac{dz_3}{dx} = f(z_2, z_1, z_0), \qquad \dfrac{dz_2}{dx} = z_3(x), \qquad \dfrac{dz_1}{dx} = z_2(x), \qquad \dfrac{dz_0}{dx} = z_1(x), \\ z_0(0) = 0 \\ z_1(0) = 0 \\ z_2(0) = v_1 \\ z_3(0) = v_2 \end{cases} \tag{31}$$

Given an initial guess $\boldsymbol{v}_0 = \begin{bmatrix} v_{01} & v_{02} \end{bmatrix}^T$ we construct the sequence of iterates $\{\boldsymbol{v}_1, \boldsymbol{v}_2, \dots\}$ satisfying

$$\boldsymbol{v}_{k+1} = \boldsymbol{v}_k - \boldsymbol{J}^{-1}(\boldsymbol{v}_k)\boldsymbol{E}(\boldsymbol{v}_k), \tag{32}$$

where $\boldsymbol{J}(\boldsymbol{v}_k)$ is the Jacobian of the error function (28)

$$\boldsymbol{J}(\boldsymbol{v}) = \begin{bmatrix} \dfrac{\partial E_1(\boldsymbol{v})}{\partial v_1} & \dfrac{\partial E_1(\boldsymbol{v})}{\partial v_2} \\ \dfrac{\partial E_2(\boldsymbol{v})}{\partial v_1} & \dfrac{\partial E_2(\boldsymbol{v})}{\partial v_2} \end{bmatrix} \tag{33}$$

evaluated at $\boldsymbol{v}_k$. As before, it is possible to derive evolution equations for the components of the Jacobian.

**Remark:** The Jacobian of the error function (28) for the Euler-Bernoulli beam model (27) has the form

$$\boldsymbol{J}(\boldsymbol{v}) = \begin{bmatrix} \dfrac{\partial z_0(L;\boldsymbol{v})}{\partial v_1} & \dfrac{\partial z_0(L;\boldsymbol{v})}{\partial v_2} \\ \dfrac{\partial z_1(L;\boldsymbol{v})}{\partial v_1} & \dfrac{\partial z_1(L;\boldsymbol{v})}{\partial v_2} \end{bmatrix}. \tag{34}$$

As shown hereafter, $\boldsymbol{J}(\boldsymbol{v})$ does not depend on $\boldsymbol{v}$. This means that with just one Newton's iteration we can compute the correct initial condition for the system, and therefore solve the problem with the shooting method. The evolution equations for the components of the Jacobian (34) are obtained by differentiating the system (27) with respect to $v_1$ and $v_2$. This yields

$$\frac{d}{dx}\frac{\partial z_3}{\partial v_i} = 0 \quad \text{and} \quad \frac{d}{dx}\frac{\partial z_j}{\partial v_i} = \frac{\partial z_{j+1}}{\partial v_i} \qquad j = 0,1,2 \qquad i = 1,2, \tag{35}$$

with initial conditions

$$\frac{\partial z_3}{\partial v_1} = 0, \qquad \frac{\partial z_3}{\partial v_2} = 1, \qquad \frac{\partial z_2}{\partial v_1} = 1, \qquad \frac{\partial z_2}{\partial v_2} = 0, \tag{36}$$

and

$$\frac{\partial z_1}{\partial v_1} = 0, \qquad \frac{\partial z_1}{\partial v_2} = 0, \qquad \frac{\partial z_0}{\partial v_1} = 0, \qquad \frac{\partial z_0}{\partial v_2} = 0. \tag{37}$$

Clearly, the solution to the system (35)-(37) does not depend on $\boldsymbol{v}$ and therefore the Jacobian (34) does not depend on $\boldsymbol{v}$. This is just another way to say that we can solve the shooting problem for the linear Euler-Bernoulli beam by just one Newton iteration as

$$\boldsymbol{v}_1 = \boldsymbol{v}_0 - \boldsymbol{J}^{-1}\boldsymbol{E}(\boldsymbol{v}_0), \tag{38}$$

where $\boldsymbol{v}_0$ is any initial guess, $E(\boldsymbol{v}_0)$ is defined in (28) and $\boldsymbol{J}$ is the Jacobian (34).

**Finite difference methods for BVP**

To solve a two-point boundary value problem with finite difference methods we simply discretize its solution on a grid and replace the derivatives appearing in the ODE and the boundary conditions with appropriate finite-difference formulas. To illustrate this process let us first consider the simple prototype problem

$$\begin{cases} \dfrac{d^2y(x)}{dx^2} = f(x) \qquad x \in [0,1] \\ y(0) = \alpha \\ y(1) = \beta \end{cases} \tag{39}$$

Let $\{x_0, \ldots, x_{N+1}\}$ be $N+2$ evenly-spaced grid points in the interval $[0,1]$, i.e.,

$$x_j = j\Delta x, \qquad \Delta x = \frac{1}{N+1}, \qquad j = 0, \ldots, N+1. \tag{40}$$

We approximate the second derivative $d^2y/dx^2$ in (39), e.g., by using the second-order finite difference formula

$$\frac{d^2y(x_j)}{dx^2} \simeq \frac{y_{j-1} - 2y_j + y_{j+1}}{\Delta x^2}, \qquad u_j = u(x_j). \tag{41}$$

A substitution of (41) into (39) yields the system of equations[5]

$$\begin{cases} \dfrac{u_{j-1} - 2u_j + u_{j+1}}{\Delta x^2} = f_j \qquad j = 1, \ldots, N \\ u_0 = \alpha \\ u_{N+1} = \beta \end{cases} \tag{42}$$

where we defined $f_j = f(x_j)$. The system (42) can be written compactly as

$$\boldsymbol{D}^2_{\text{FD}} \boldsymbol{u} = \boldsymbol{f}, \tag{43}$$

where

$$\boldsymbol{D}^2_{\text{FD}} = \frac{1}{\Delta x^2} \begin{bmatrix} -2 & 1 & 0 & 0 & \cdots & \cdots & 0 \\ 1 & -2 & 1 & 0 & \cdots & \cdots & 0 \\ 0 & 1 & -2 & 1 & \cdots & \cdots & 0 \\ \vdots & & \ddots & \ddots & \ddots & & \vdots \\ \vdots & & & \ddots & \ddots & \ddots & \vdots \\ \vdots & & & & 1 & -2 & 1 \\ 0 & \cdots & \cdots & \cdots & 0 & 1 & -2 \end{bmatrix}, \qquad \boldsymbol{u} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ \vdots \\ \vdots \\ u_{N-1} \\ u_N \end{bmatrix}, \qquad \boldsymbol{f} = \begin{bmatrix} f_1 - \alpha/\Delta x^2 \\ f_2 \\ f_3 \\ \vdots \\ \vdots \\ f_{N-1} \\ f_N - \beta/\Delta x^2 \end{bmatrix}. \tag{44}$$

The differentiation matrix $\boldsymbol{D}^2_{\text{FD}}$ corresponding to the second-order finite difference approximation is tridiagonal, diagonally dominant and negative definite. The eigenvalues of $\boldsymbol{D}^2_{\text{FD}}$ are

$$\lambda_k = \frac{2}{\Delta x^2} \left( \cos(k\pi\Delta x) - 1 \right). \tag{45}$$

Clearly, $\lambda_k < 0$ for all $k = 1, \ldots, N$. This implies that matrix $\boldsymbol{D}^2_{\text{FD}}$ is invertible[6] and therefore the system (43) has a unique solution.

[5] In equation (42) $u_j$ represents the numerical approximation of $y_j = y(x_j)$.

[6] Recall that the determinant of a matrix is the product of the eigenvalues.

To prove that (45) are indeed the eigenvalues of $\boldsymbol{D}^2_{\text{FD}}$, consider the eigenvalue problem

$$\boldsymbol{D}^2_{\text{FD}}\boldsymbol{u} = \lambda \boldsymbol{u}, \tag{46}$$

i.e.,

$$\frac{u_{j-1} - 2u_j + u_{j+1}}{\Delta x^2} = \lambda u_j \quad \Rightarrow \quad u_{j+1} = (2 + \Delta x^2 \lambda) u_j - u_{j-1} \quad \text{with} \quad u_0 = u_{N+1} = 0 \tag{47}$$

Setting $2Q = (2 + \Delta x^2\lambda)$ and rescaling all equations so that $u_1 = 1$ yields

$$\begin{cases} u_0 = 0 \\ u_1 = 1 \\ u_{j+1} = 2Q u_j - u_{j-1} \end{cases} \tag{48}$$

Equation (48) is the three-term recurrence relation satisfied by the Chebyshev polynomials of the second kind $U_j(Q)$ in the variable $Q$. Setting $u_{j+1} = U_j(Q)$ and using the boundary condition $u_{N+1} = 0$ we have that $U_N(Q) = 0$. Hence, $Q$ must be a root of the $N$-th degree Chebyshev polynomial of the second kind. Such roots are

$$Q_k = \cos\left(\frac{k\pi}{N+1}\right) = \cos(k\pi\Delta x), \qquad k = 1\ldots, N \tag{49}$$

Recalling that $2Q_k = (2 + \Delta x^2 \lambda_k)$, we obtain

$$\lambda_k = \frac{2}{\Delta x^2}\left(\cos(k\pi\Delta x) - 1\right). \tag{50}$$

**Convergence Analysis.** We now perform convergence analysis of the finite difference approximation (43). To this end, we need to show that the error between the analytical solution of (39) and the numerical solution goes to zero as we send $\Delta x$ to zero, i.e., as we consider more and more evenly-spaced grid points in the interval $[0, 1]$. Let $y_j = y(x_j)$ and $u_j$ be, respectively, the analytical solution and the finite-differences solution of (39). We define the error

$$\boldsymbol{e} = \boldsymbol{y} - \boldsymbol{u}, \tag{51}$$

where

$$\boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}, \qquad \boldsymbol{u} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_N \end{bmatrix}. \tag{52}$$

By applying $\boldsymbol{D}^2_{\text{FD}}$ to $\boldsymbol{e}$ we obtain

$$\boldsymbol{D}^2_{\text{FD}}\boldsymbol{e} = \boldsymbol{\tau}, \tag{53}$$

where

$$\tau_j = \frac{y_{j-1} - 2y_j + 2y_{j+1}}{\Delta x^2} - f_j \qquad j = 1, \ldots, N \tag{54}$$

is the *local truncation error* (LTE) associated with the finite-difference approximation[7]. At this point we recall that the matrix $\boldsymbol{D}^2_{\text{FD}}$ is symmetric and invertible. This allows us to express the error $\boldsymbol{e}$ in equation

[7]By using Taylor series we obtain

$$\begin{aligned} \tau_j &= \frac{y_{j-1} - 2y_j + y_{j+1}}{\Delta x^2} - f_j \\ &= \frac{d^2 y(x_j)}{dx^2} - f_j + \frac{\Delta x^2}{12}\frac{d^4 y(x_j)}{dx^4} + \cdots \\ &= \frac{\Delta x^2}{12}\frac{d^4 y(x_j)}{dx^4} + \cdots. \end{aligned} \tag{55}$$

Therefore the local truncation error goes to zero as $\Delta x^2$.

(53) explicitly in terms of the truncation error $\boldsymbol{\tau}$ as

$$\boldsymbol{e} = \left[\boldsymbol{D}^2_{\text{FD}}\right]^{-1}\boldsymbol{\tau}. \tag{56}$$

By taking the 2-norm of this expression we obtain

$$\|\boldsymbol{e}\|_2 \leq \left\|\left[\boldsymbol{D}^2_{\text{FD}}\right]^{-1}\right\|_2 \|\boldsymbol{\tau}\|_2\,, \tag{57}$$

where the matrix 2-norm $\left\|\left[\boldsymbol{D}^2_{\text{FD}}\right]^{-1}\right\|_2$ induced by the vector 2-norm coincides with the largest singular value of the matrix $\left[\boldsymbol{D}^2_{\text{FD}}\right]^{-1}$. Recall that the inverse of a symmetric matrix is symmetric. This implies that the square root of the singular values of $\left[\boldsymbol{D}^2_{\text{FD}}\right]^{-1}$ matrix coincide with the absolute values of the eigenvalues[8] of $\left[\boldsymbol{D}^2_{\text{FD}}\right]^{-1}$. Moreover the eigenvalues of the inverse matrix are the inverses of the eigenvalues of the matrix. This proves the following equality

$$\left\|\left[\boldsymbol{D}^2_{\text{FD}}\right]^{-1}\right\|_2 = \max_{k=1,\dots,N}\left|\frac{1}{\lambda_k}\right| = \frac{1}{\displaystyle\min_{k=1,\dots,N}|\lambda_k|}, \tag{59}$$

where $\lambda_k$ are the eigenvalues of $\boldsymbol{D}^2_{\text{FD}}$. By using equation (45) we see that

$$\min_{k=1,\dots,N}|\lambda_k| = |\lambda_1| = \frac{2}{\Delta x^2}\left|\cos(\pi\Delta x)-1\right|. \tag{60}$$

Moreover, for sufficiently small $\Delta x$ we can expand (60) in a Taylor series to obtain

$$|\lambda_1| = \frac{2}{\Delta x^2}\left|1-\frac{\pi^2\Delta x^2}{2}+\frac{\pi^4\Delta x^4}{24}\cdots - 1\right| = \pi^2\left|1-\frac{\pi^2}{12}\Delta x^2+\cdots\right|. \tag{61}$$

Therefore, in the limit $\Delta x \to 0$ we have

$$\|\boldsymbol{e}\|_2 \leq \frac{1}{\pi^2}\|\boldsymbol{\tau}\|_2 = \frac{\Delta x^2}{12\pi^2}\sqrt{\sum_{k=1}^{N}\left[\frac{d^4y(x_k)}{dx^4}\right]^2} = \frac{\Delta x^{3/2}}{12\pi^2}\sqrt{\sum_{k=1}^{N}\Delta x\left[\frac{d^2f(x_k)}{dx^2}\right]^2}, \tag{62}$$

where we used equation (55) for the 2-norm of the local truncation error. Note that the quantity under the square root at the right hand side of (62) converges to the integral of the square of the second derivative of $f(x)$ in the limit $N \to \infty$, i.e.,

$$\lim_{N\to\infty}\sum_{k=1}^{N}\Delta x\left[\frac{d^2f(x_k)}{dx^2}\right]^2 = \int_0^1\left[\frac{d^2f(x)}{dx^2}\right]^2 dx. \tag{63}$$

Assuming that such an integral is finite, i.e., that the second derivative of $f$ is square integrable in $[0,1]$, we conclude that the second-order finite difference approximation (42) of the boundary value problem (39) is convergent with order $3/2$ in $\Delta x$. Similarly, in the uniform norm we obtain

$$\|\boldsymbol{e}\|_\infty \leq \|\boldsymbol{e}\|_2 \leq \frac{1}{\pi^2}\|\boldsymbol{\tau}\|_2 \leq \sqrt{N}\,\|\boldsymbol{\tau}\|_\infty \simeq \frac{\sqrt{N}}{12\pi^2(N+1)^2}\left\|\frac{d^2f(x)}{dx^2}\right\|_\infty. \tag{64}$$

Clearly, in the limit $N \to \infty$ we have that $\|\boldsymbol{e}\|_\infty$ goes to zero as $1/N^{3/2}$.

[8]Recall that for any symmetric matrix

$$\|\boldsymbol{A}\|_2 = \max_j\sqrt{\lambda_j(\boldsymbol{A}^T\boldsymbol{A})} = \max_j\sqrt{\lambda_j(\boldsymbol{A}^2)} = \max|\lambda_j(\boldsymbol{A})|. \tag{58}$$

**Neumann boundary conditions.** Consider the boundary value problem

$$\begin{cases} \dfrac{d^2y(x)}{dx^2} = f(x) \qquad x \in [0,1] \\ \dfrac{dy(0)}{dx} = \alpha \\ y(1) = \beta \end{cases} \tag{65}$$

How do we impose the Neumann boundary condition $dy(0)/dx = \alpha$ in a finite difference setting? The simplest way is to use forward finite differences, e.g.,

$$\frac{dy(0)}{dx} \simeq \frac{-3y_0 + 4y_1 - y_2}{2\Delta x} = \alpha. \tag{66}$$

In this way, we can write the fully discrete finite difference system as

$$\frac{1}{\Delta x^2}\begin{bmatrix} -3\Delta x/2 & 2\Delta x & -\Delta x/2 & 0 & \cdots & \cdots & 0 \\ 1 & -2 & 1 & 0 & \cdots & \cdots & 0 \\ 0 & 1 & -2 & 1 & \cdots & \cdots & 0 \\ \vdots & & \ddots & \ddots & \ddots & & \vdots \\ \vdots & & & \ddots & \ddots & \ddots & \vdots \\ \vdots & & & & 1 & -2 & 1 \\ 0 & \cdots & \cdots & \cdots & 0 & 1 & -2 \end{bmatrix} \begin{bmatrix} u_0 \\ u_1 \\ u_2 \\ \vdots \\ \vdots \\ u_{N-1} \\ u_N \end{bmatrix} = \begin{bmatrix} \alpha \\ f_1 \\ f_2 \\ \vdots \\ \vdots \\ f_{N-1} \\ f_N - \beta/\Delta x^2 \end{bmatrix}. \tag{67}$$

**Nonlinear boundary value problem.** Consider the nonlinear boundary value problem

$$\begin{cases} \dfrac{d^2y(x)}{dx^2} = f\left(\dfrac{dy}{dx}, y, x\right) \qquad x \in [0,1] \\ y(0) = \alpha \\ y(1) = \beta \end{cases} \tag{68}$$

A second-order finite difference approximation of (68) is

$$\begin{cases} \dfrac{u_{j-1} - 2u_j + u_{j+1}}{\Delta x^2} = f\left(\dfrac{u_{j+1} - u_{j-1}}{2\Delta x}, u_j, x_j\right) \qquad j = 1, \dots, N \\ u_0 = \alpha \\ u_{N+1} = \beta \end{cases} \tag{69}$$

This is a system of $N$ nonlinear equations in $N$ unknowns $\{u_1, \dots u_N\}$ which can be solved, e.g., with the Newton's method.

We conclude this section by emphasizing that we could have used also higher-order finite difference formulas to solve the problem (68) or (39). For instance, we could have used a fourth-order formula based on a stencil with 5 points, with forward and backward representation at the left and the right boundary, respectively, to accommodate Dirichlet or Neumann boundary conditions.

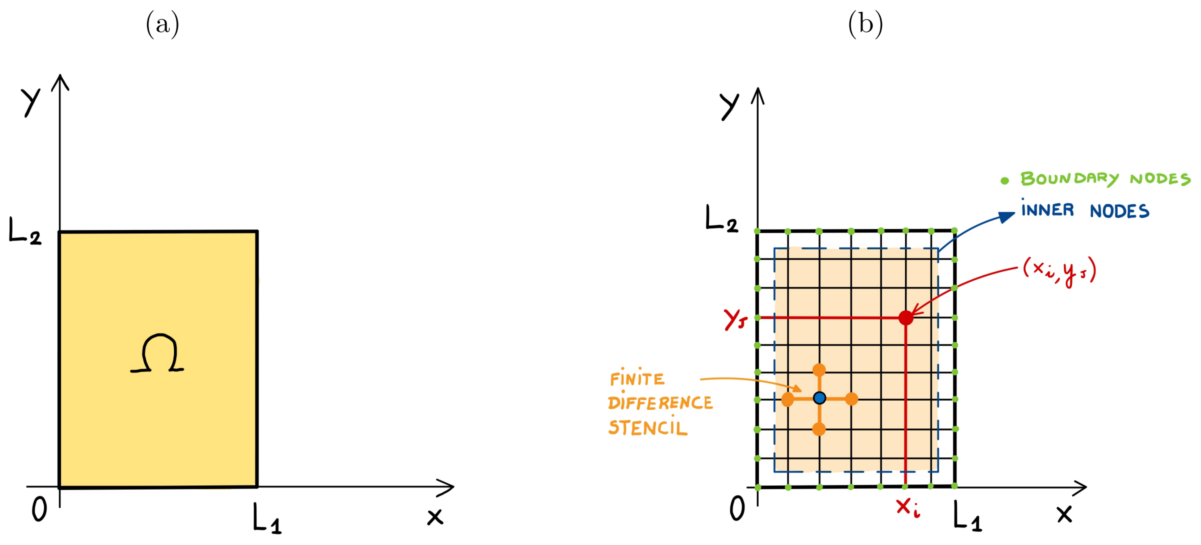

Figure 4: (a) Sketch of the spatial domain for the BVP (70). (b) Sketch of the two-dimensional grid and stencil used to approximate the Laplacian $\nabla^2 U = \partial^2 U/\partial x^2 + \partial^2 U/\partial y^2$.

**Boundary value problems for PDEs.** In this section we briefly discuss how to solve boundary value problems for PDEs using finite differences. To this end, consider the Poisson's equation in two-dimensions[9] with homogeneous Dirichlet boundary conditions

$$\begin{cases} \dfrac{\partial^2 U(x,y)}{\partial x^2} + \dfrac{\partial^2 U(x,y)}{\partial y^2} = f(x,y) \qquad (x,y) \in \Omega \\ U(x,y) = 0 \qquad (x,y) \in \partial\Omega \quad (\partial\Omega \text{ is the boundary of } \Omega) \end{cases} \tag{70}$$

For simplicity, here we consider a rectangular domain $\Omega = [0, L_1] \times [0, L_2]$ (see Figure 4(a)). The boundary of $\Omega$, i.e., $\partial\Omega$ is the union of four line segments.

We discretize $\Omega$ in terms of the two-dimensional grid (Figure 4(b))

$$(x_i, y_j) = \begin{cases} x_i = i\Delta x \qquad \Delta x = \dfrac{L_1}{N+1} \qquad i = 0, \dots, N+1, \\ y_j = j\Delta y \qquad \Delta y = \dfrac{L_2}{M+1} \qquad j = 0, \dots, M+1. \end{cases} \tag{71}$$

By using second-order centered finite differences, we approximate the partial derivatives $\partial^2 U/\partial x^2$ and $\partial^2 U/\partial y^2$ at the node $(x_i, y_j)$ as

$$\frac{\partial^2 U(x_i, y_j)}{\partial x^2} \simeq \frac{U_{i-1,j} - 2U_{i,j} + U_{i+1,j}}{\Delta x^2}, \tag{72}$$

$$\frac{\partial^2 U(x_i, y_j)}{\partial y^2} \simeq \frac{U_{i,j-1} - 2U_{i,j} + U_{i,j+1}}{\Delta y^2}, \tag{73}$$

where we denoted by $U_{i,j} = U(x_i, y_j, t)$. A substitution of (72)-(73) into (70) yields

$$\frac{u_{i-1,j} - 2u_{i,j} + u_{i+1,j}}{\Delta x^2} + \frac{u_{i,j-1} - 2u_{i,j} + u_{i,j+1}}{\Delta y^2} = f(x_i, y_j), \tag{74}$$

[9]Existence and uniqueness of weak solutions to the boundary value problem (70) can be established by using the Lax-Milgram lemma.

with boundary conditions

$$\begin{cases} u_{0,j} = u_{N+1,j} = 0 \qquad j = 0,\ldots,M, \\ u_{i,0} = u_{i,M+1} = 0 \qquad i = 0,\ldots,N. \end{cases} \tag{75}$$

The unknowns are the $N \times N$ entries of the matrix $u_{i,j}$ for for $i = 1,\ldots,N$ and $j = 1,\ldots,M$, i.e., the solution at the *inner nodes*. The linear system (70) can be written in a matrix-vector form. There is no unique way to do so, as there is no unique way to vectorize the matrix of unknowns $u_{i,j}$, where $i, j$ identify inner nodes, i.e., $(i = 1,\ldots N,\ j = 1,\ldots,M)$. For instance we may decide to reshape the $u_{i,j}$ as

$$\boldsymbol{u} = \left[\begin{bmatrix} u_{1,1} & u_{2,1} & \cdots & u_{N,1}\end{bmatrix} \begin{bmatrix} u_{1,2} & u_{2,2} & \cdots & u_{N,2}\end{bmatrix} \cdots \begin{bmatrix} u_{1,M} & u_{2,M} & \cdots & u_{N,M}\end{bmatrix}\right]^T. \tag{76}$$

With this ordering it can be shown that (74) can be written as

$$\left(\boldsymbol{D}_x^2 + \boldsymbol{D}_y^2\right)\boldsymbol{u} = \boldsymbol{f}, \tag{77}$$

where

$$\boldsymbol{f} = \left[\begin{bmatrix} f_{1,1} & f_{2,1} & \cdots & f_{N,1}\end{bmatrix} \begin{bmatrix} f_{1,2} & f_{2,2} & \cdots & f_{N,2}\end{bmatrix} \cdots \begin{bmatrix} f_{1,M} & f_{2,M} & \cdots & f_{N,M}\end{bmatrix}\right]^T, \tag{78}$$

$$\boldsymbol{D}_x^2 = \frac{1}{\Delta x^2}\begin{bmatrix} \boldsymbol{T} & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{T} & \cdots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{T} \end{bmatrix}, \qquad \boldsymbol{D}_y^2 = \frac{1}{\Delta y^2}\begin{bmatrix} -2\boldsymbol{I}_N & \boldsymbol{I}_N & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{I}_N & -2\boldsymbol{I}_N & \boldsymbol{I}_N & \cdots & \boldsymbol{0} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \boldsymbol{I}_N & -2\boldsymbol{I}_N & \boldsymbol{I}_N \\ \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{I}_N & -2\boldsymbol{I}_N \end{bmatrix}, \tag{79}$$

and the $N \times N$ matrix $\boldsymbol{T}$ is defined as

$$\boldsymbol{T} = \begin{bmatrix} -2 & 1 & 0 & \cdots & 0 \\ 1 & -2 & 1 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & 1 & -2 & 1 \\ 0 & \cdots & 0 & 1 & -2 \end{bmatrix}. \tag{80}$$

Note that both matrices $\boldsymbol{D}_x^2$ and $\boldsymbol{D}_y^2$ are block matrices with $M \times M$ blocks each one of which is $N \times N$.

## Method of weighted residuals

The method of weighted residuals for BVP is based on the so-called weak (or variational) formulation of the problem. To describe the method, consider the following prototype boundary value problem

$$\begin{cases} -\dfrac{d}{dx}\left(a(x)\dfrac{dy(x)}{dx}\right) + b(x)y(x) = f(x) \qquad x \in [0,1] \\ y(0) = 0 \\ y(1) = 0 \end{cases} \tag{81}$$

where $a(x)$ is a strictly positive function, i.e., $a(x) > 0$ for all $x \in [0,1]$. Multiply the differential equation in (81) by a test function $v(x)$ and integrate over $[0,1]$ to obtain

$$-\int_0^1 \frac{d}{dx}\left(a(x)\frac{dy(x)}{dx}\right)v(x)dx + \int_0^1 b(x)y(x)v(x)dx = \int_0^1 f(x)v(x)dx. \tag{82}$$

By integrating the first term by parts and assuming that $v(x)$ satisfies the boundary condition in (81) we obtain

$$\int_0^1 a(x)\frac{dy(x)}{dx}\frac{dv(x)}{dx}dx + \int_0^1 b(x)y(x)v(x)dx = \int_0^1 f(x)v(x)dx. \tag{83}$$

Clearly, if $y(x)$ is of class $C^2([0,1])$ and it satisfies (81) (strong solution) then $y(x)$ is also a solution to the following *weak formulation* of the BVP:

*Find $y \in H_0^1([0,1])$ such that* (83) *is satisfied for all $v \in H_0^1([0,1])$.*

Here, $H_0^1([0,1])$ denotes the Sobolev space square integrable functions vanishing at $x=0$ and $x=1$, with square integrable first-order derivatives, i.e.,

$$H_0^1([0,1]) = \{v \in L^2([0,1]) \quad \text{such that} \quad \frac{dv}{dx} \in L^2([0,1]) \quad \text{and} \quad v(0)=v(1)=0\}. \tag{84}$$

Note that the weak formulation (83) involves only the first derivative of $y(x)$. Therefore a weak solution to the BVP (81), i.e., a solution to (83), may not satisfy (81). This means that the weak formulation of a BVP might have a solution even when the strong formulation does not.

With the weak formulation (83) available, we look for a finite-dimensional approximation of $y(x)$. To this end, suppose that $y(x)$ can be accurately represented in a finite-dimensional subspace $V_N \subset H_0^1([0,1])$, i.e.,

$$y_N(x) = \sum_{k=1}^N a_k\varphi_k(x), \qquad \varphi_k \in H_0^1([0,1]). \tag{85}$$

A substitution of (85) into (83) yields a residual[10] $R_N(x)$

$$\int_0^1 a(x)\frac{dy_N(x)}{dx}\frac{dv(x)}{dx}dx + \int_0^1 b(x)y_N(x)v(x)dx = \int_0^1 f(x)v(x)dx + \int_0^1 R_N(x)v(x)dx. \tag{86}$$

Depending on the way we handle the residual we can have different classes of methods:

- **Galerkin method:** We set the residual orthogonal to the span of $\{\varphi_1,\dots,\varphi_N\}$, i.e.,

$$\int_0^1 R_N(x)\varphi_j(x)dx = 0 \qquad j=1,\dots,N. \tag{87}$$

  This yields a system of $N$ equations in the $N$ unknowns $\{a_1,\dots,a_N\}$ (see equation (85)). A variation of the Galerkin method is the Petrov-Galerkin method in which we require $R_N(x)$ to be orthogonal to a linear space spanned by a set of basis functions other than $\{\varphi_k\}$.

- **Collocation method:** We set the residual $R_N(x)$ equal to zero at a set of collocation nodes $\{x_1,\dots,x_N\}$, i.e.,

$$R_N(x_j) = 0 \qquad j=1,\dots,N. \tag{88}$$

  In this way, the differential equation is satisfied exactly at the collocation nodes.

- **Least-squares method:** We minimize the $L^2$ norm of the residual $R_N(x)$ over the parameters $\{a_1,\dots,a_N\}$ in equation (85))

$$\min_{a_1,\dots,a_N}\int_0^1 R_N^2(x)dx \tag{89}$$

[10]The residual in (86) arises because $y_N(x)$ does not satisfy (in general) the differential equation.

As an example, let us apply these methods to the BVP (39), hereafter rewritten for convenience

$$\begin{cases} \dfrac{d^2y}{dt^2} = f(x) \\ y(0) = \alpha \\ y(1) = \beta \end{cases} \tag{90}$$

To this end, we look for a representation of the solution in the form

$$y_N(x) = (1-x)\alpha + x\beta + \sum_{k=1}^{N} a_k \varphi_k(x) \tag{91}$$

where

$$\varphi_k(x) = x(1-x)T_k(2x-1) \qquad x \in [0,1] \tag{92}$$

and $T_k(x)$ are, e.g., Chebyshev polynomials of the first kind or Legendre polynomials. The linear functions $(1-x)$ and $x$ are called *boundary modes* in finite-element analysis, while $\varphi_k(x)$ are called *interior modes*[11]. A substitution of (91) into (90) yields

$$\frac{d^2y_N(x)}{dx^2} = f(x) + R_N(x). \tag{94}$$

Clearly, the boundary conditions are automatically satisfied by $y_N(x)$ (91)-(92).

**Galerkin method**

In the Galerkin method we impose that the residual $R_N(x)$ is orthogonal (in the $L^2$ sense) to the span of $\{\varphi_1, \ldots, \varphi_N\}$. To impose such orthogonality, we first multiply (94) by $\varphi_j(x)$ and integrate over $[0,1]$ to obtain

$$\int_0^1 \frac{d^2y_N(x)}{dx^2}\varphi_j(x)dx = \int_0^1 f(x)\varphi_j(x)dx + \int_0^1 R_N(x)\varphi_j(x)dx \qquad j = 1, \ldots, N. \tag{95}$$

Setting

$$\int_0^1 R_N(x)\varphi_j(x)dx = 0 \qquad j = 1, \ldots, N \tag{96}$$

yields the system of equations

$$-\int_0^1 \frac{dy_N(x)}{dx}\frac{d\varphi_j(x)}{dx}dx = \int_0^1 f(x)\varphi_j(x)dx \qquad j = 1, \ldots, N. \tag{97}$$

where we integrated by parts the first term in (95). Substituting (91) and

$$\frac{dy_N(x)}{dx} = \beta - \alpha + \sum_{k=1}^{N} a_k \frac{d\varphi_k(x)}{dx} \tag{98}$$

[11]In the *spectral method* the basis fuctions $\varphi_k(x)$ are chosen to be Lagrange characteristic polynomials at Gauss-Lobatto nodes in $[0,1]$. This yields the following expansion of the solution

$$y_N(x) = \varphi_0(x)\alpha + \varphi_{N+1}(x)\beta + \sum_{k=1}^{N} a_k\varphi_k(x), \tag{93}$$

where $\varphi_0(x)$ and $\varphi_{N+1}(x)$ are the boundary modes and $\varphi_k(x)$ are the interior modes. The integrals in (95) can be computed using Gauss-Lobatto quadrature.

into (97) yields

$$-(\beta-\alpha)\int_0^1 \frac{d\varphi_j(x)}{dx}dx - \sum_{k=1}^{N} a_k \underbrace{\int_0^1 \frac{d\varphi_j(x)}{dx}\frac{d\varphi_k(x)}{dx}dx}_{\text{stiffness matrix } S_{jk}} = \int_0^1 f(x)\varphi_j(x)dx \qquad j=1,\dots,N. \tag{99}$$

i.e.,

$$\sum_{k=1}^{N} S_{jk}a_k = -\int_0^1 f(x)\varphi_j(x)dx - (\beta-\alpha)\int_0^1 \frac{d\varphi_j(x)}{dx}dx \qquad j=1,\dots,N. \tag{100}$$

Upon definition of

$$\boldsymbol{a} = \begin{bmatrix} a_1 \\ \vdots \\ a_N \end{bmatrix}, \qquad \boldsymbol{h} = \begin{bmatrix} \int_0^1 f(x)\varphi_1(x)dx \\ \vdots \\ \int_0^1 f(x)\varphi_N(x)dx \end{bmatrix}, \qquad \boldsymbol{b} = (\beta-\alpha)\begin{bmatrix} \int_0^1 \frac{d\varphi_1(x)}{dx}dx \\ \vdots \\ \int_0^1 \frac{d\varphi_N(x)}{dx}dx \end{bmatrix}, \tag{101}$$

we can write the system (100) in a matrix-vector form as

$$\boldsymbol{S}\boldsymbol{a} = -(\boldsymbol{h}+\boldsymbol{b}) \tag{102}$$

Inverting the (positive definite) stiffness matrix $\boldsymbol{S}$ yields the solution $\boldsymbol{a} = -\boldsymbol{S}^{-1}(\boldsymbol{f}+\boldsymbol{b})$ which can be then substituted back into (91).

## Collocation method

In the collocation method we impose that the residual $R_N(x)$ is equal to zero at $N$ distinct (interior) nodes $\{x_0,\dots,x_N\}$. A substitution of (85) into (81) yields (94). By imposing $R_N(x_i)=0$ in (94) we obtain

$$\sum_{k=1}^{N} a_k \frac{d^2\varphi_k(x_j)}{dx^2} = \underbrace{f(x_j) - \alpha\frac{d^2\varphi_0(x_j)}{dx^2} - \beta\frac{d^2\varphi_{N+1}(x_j)}{dx^2}}_{g_j} \qquad j=2,\dots,N-1 \tag{103}$$

Upon definition of the differentiation matrix $D^2_{jk} = d^2\varphi_k(x_j)/dx^2$ we can write the linear system (103) as

$$\boldsymbol{D}^2_{\text{Coll}}\boldsymbol{a} = \boldsymbol{g}, \tag{104}$$

which resembles very much the system (43). Indeed the finite-difference methods is a particular type of collocation method. Another important example of collocation method is the so-called *spectral collocation method* [1] in which the collocation nodes $\{x_k\}$ are Gauss-Lobatto quadrature points and $\varphi_k(x)$ are Lagrange characteristic polynomials corresponding to the grid. Specifically, suppose that the collocation points are $\{x_0,\dots,x_N\}$. In this case, (103) can be written as

$$\begin{bmatrix} D^2_{1,2} & \cdots & D^2_{1,N-1} \\ \vdots & \ddots & \vdots \\ D^2_{N-1,2} & \cdots & D^2_{N-1,N-1} \end{bmatrix}\begin{bmatrix} u_1 \\ \vdots \\ u_{N-1} \end{bmatrix} = \begin{bmatrix} f(x_1) \\ \vdots \\ f(x_{N-1}) \end{bmatrix} - \alpha\begin{bmatrix} D_{1,0} \\ \vdots \\ D_{N-1,0} \end{bmatrix} - \beta\begin{bmatrix} D_{1,N} \\ \vdots \\ D_{N-1,N} \end{bmatrix} \tag{105}$$

In the special case of the Gauss-Chebyshev-Lobatto method the matrix at the left hand side is obtained by the removing the first and last row and column from the second-order differentiation matrix (122). Convergence analysis of spectral collocation methods for BVP follows the analysis we have done for second order finite differences. In particular, the eigenvalues of second-order spectral collocation matrices are discussed in [4].

**Least squares method**

Finally, we set up the BVP (39) as a least-squares problem. To this end, we consider the residual

$$R_N(x) = \sum_{k=1}^{N} a_k \frac{d^2\varphi_k(x)}{dx^2} - f(x) \tag{106}$$

and its $L^2$ norm

$$\|R_N(x)\|^2_{L^2([0,1]} = \int_0^1 \left( \sum_{k=1}^{N} a_k \frac{d^2\varphi_k(x)}{dx^2} - f(x) \right)^2 dx. \tag{107}$$

By expanding integrand we see that minimization of (107) is essentially a quadratic programming problem with two linear constraints (boundary conditions) which can be solved, for example, using high-performance solvers such as OSQP [3] (`https://osqp.org/`).

## Appendix A: polynomial approximation theory

In this section we review some of the results of polynomial approximation theory we discussed in the course note 1. To this end, let us first notice that collocation, Galerkin, and least squares methods are based on functional series expansions of the form

$$y_N = \sum_{k=0}^{N} a_k \varphi_k(x). \tag{108}$$

For simplicity we consider $x \in [-1,1]$ (any bounded interval can be rescaled to $[-1,1]$). The functions $\varphi_k(x)$ here are orthogonal polynomials relative to a weight function $w(x)$, i.e.,

$$\int_{-1}^{1} \varphi_k(x)\varphi_j(x)w(x)dx = \delta_{kj} \left\|\varphi_k\right\|^2_{L^2_w([-1,1])}. \tag{109}$$

or Lagrange polynomials associated to a properly chosen set of notes, e.g., zeros of orthogonal polynomials. Polynomial approximation theory theory is thoroughly discussed in [1, Chapter 6]. Hereafter we briefly summarize some of the main results.

**Theorem 1.** Let $\{\varphi_k(x)\}$ in (108) be a set polynomials orthogonal in $[-1,1]$ relative to the weight function $w(x)$. Then for any $y(x) \in H^p_w([-1,1])$ $p \geq 0$, there exists a constant $C$, independent of $N$, such that

$$\left\|y(x) - y_N(x)\right\|_{L^2_w([-1,1])} \leq CN^{-p} \left\|y(x)\right\|_{H^p_w([-1,1])} \tag{110}$$

Theorem 1 demonstrates that the error between the function $y(x)$ and the approximation (108) decays spectrally with the number of basis functions. If the function $y(x)$ is infinitely smooth, then the error decays exponentially fast with the number of basis functions. Similar results can be obtained the approximation of the derivatives of $y(x)$ (see [1, Theorem 6.2 and Theorem 6.3]).

The next theorem summarizes the approximation properties of *spectral collocation* representations, i.e., series expansions of the form (108) where $a_k = y(x_k)$ and $\varphi_k(x)$ are Lagrangian characteristic polynomials associated with a set of Gauss or Gauss-Lobatto nodes[12] $\{x_0, \ldots, x_N\}$ in $[-1,1]$.

**Theorem 2.** Let $\varphi_k(x)$ in (108) be Lagrange characteristic polynomials associated with a set of Gauss or Gauss-Lobatto nodes in $[-1,1]$. Suppose that such Gauss or Gauss Lobatto nodes are defined by a polynomial orthogonal in $[-1,1]$ relative to the weight function $w(x)$. Then for any $y(x) \in H^p_w([-1,1])$ $p \geq 1$, there exists a constant $C$, independent of $N$, such that (110) holds.

Theorem 2 demonstrates that contrary to finite difference methods, the error of spectral collocation methods does not decay as a fixed power of $1/N$ but rather as a power that depends on the smoothness of the function we are approximating. For infinitely differentiable function functions, the error decreases exponentially fast with $N$. Hence, spectral collocation methods are, in a certain sense, *methods of infinite order* when applied to smooth problems.

**Gauss-Chebyshev-Lobatto spectral collocation method for BVP.** Let briefly review the main ingredients of the Gauss-Lobatto Chebyshev expansion, and its usage for boundary value problems. For more details we refer to [1]. We first recall that the Chebyshev polynomials of the first kind are defined as

$$T_k(x) = \cos(k \arccos(x)) \qquad x \in [-1,1] \qquad \text{(trigonometric representation)}. \tag{111}$$

[12]Recall that *Gauss nodes* in $[-1,1]$ are the zeros of an orthogonal polynomial $P_{N+1}(x)$ of degree $N+1$ defined in $[-1,1]$. Orthogonality is relative to some weight function $w(x)$. If $w(x) = 1$ then $P_k(x)$ are Lagrange polynomials. On the other hand, *Gauss-Lobatto nodes* are zeros of the polynomial $(1-x^2)dP_N(x)/dx$.

It can be shown that $T_k(x)$ (like any other orthogonal polynomial) satisfies a three-term recurrence relation

$$\begin{aligned} T_0(x) &= 1 \\ T_1(x) &= x \\ T_{n+1}(x) &= 2x\,T_n(x) - T_{n-1}(x). \end{aligned} \tag{112}$$

which gives

$$T_2(x) = 2x^2 - 1, \qquad T_3(x) = 4x^3 - 3x \qquad T_4(x) = 8x^4 - 8x^2 + 1, \ldots. \tag{113}$$

The Gauss-Chebyshev-Lobatto nodes are zeros of the polynomial

$$Q_{N+1}(x) = (1 - x^2)\frac{dT_N(x)}{dx}, \tag{114}$$

i.e., $x_0 = -1$, $x_N = 1$ and all maxima and minima of $T_N(x)$. By differentiating (111) with respect to $x$ we obtain

$$\frac{dT_N(x)}{dx} = \frac{\sin(N\arccos(x))}{\sqrt{1-x^2}}. \tag{115}$$

Hence $Q_{N+1}(x) = 0$ implies that

$$x_j = \cos\left(\frac{k\pi}{N}\right) \qquad j = 0, \ldots, N \qquad \text{(Gauss-Chebyshev-Lobatto points)}. \tag{116}$$

These points are obtained by dividing half unit circle in evenly spaced parts and projecting them onto the $x$-axis. It can be shown that the Lagrange characteristic polynomials associated with the Gauss-Chebyshev-Lobatto nodes are

$$\varphi_j(x) = \frac{(-1)^{N+j+1}(1-x^2)}{d_j N^2 (x - x_j)}\frac{dT_N(x)}{dx} = \frac{(-1)^{N+j+1}\sqrt{(1-x^2)}}{d_j N^2(x-x_j)}\sin(N\arccos(x)), \tag{117}$$

where $x_j$ is given in (116) and

$$d_0 = d_N = 2 \qquad d_1 = d_2 = \cdots = d_{N-1} = 1. \tag{118}$$

A substitution of (117) into (108) yields the series expansion[13]

$$y_N(x) = \sum_{k=0}^{N} y(x_k)\varphi_k(x). \tag{119}$$

At this point we can differentiate (119) with respect to $x$ and evaluate the derivative at $x = x_j$. This yields the expressions

$$\frac{dy_N(x_j)}{dx} = \sum_{k=0}^{N} y(x_k)\underbrace{\frac{d\varphi_k(x_j)}{dx}}_{D^1_{jk}}, \qquad \frac{d^2 y_N(x_j)}{dx^2} = \sum_{k=0}^{N} y(x_k)\underbrace{\frac{d^2\varphi_k(x_j)}{dx^2}}_{D^2_{jk}}, \tag{120}$$

where $\boldsymbol{D}^1$ and $\boldsymbol{D}^2$ are, respectively, first- and second-order Gauss-Chebyshev-Lobatto differentiation matrices. A direct calculation shows that

$$D^1_{ij} = \begin{cases} -\dfrac{2N^2+1}{6} & i = j = 0 \\ \dfrac{d_i}{d_j}\dfrac{(-1)^{i+j}}{x_i - x_j} & i \neq j \\ -\dfrac{x_i}{2(1-x_i^2)} & i = j \\ \dfrac{2N^2+1}{6} & i = j = N \end{cases} \qquad \text{(first-order differentiation matrix)} \tag{121}$$

[13]Note that the series expansion (119), is the Lagrange interpolant of $y(x)$ through the Gauss-Chebyshev-Lobatto points (116).

where $d_i$ are defined in (118), and

$$D^2_{ij} = \begin{cases} \dfrac{(-1)^{i+j}}{d_j}\dfrac{x_i^2 + x_i x_j - 2}{\left(1-x_i^2\right)\left(x_i - x_j\right)^2} & 1 \le i \le N-1, \quad 0 \le j \le N, \qquad j \neq i \\ -\dfrac{(N^2-1)(1-x_i^2)+3}{3\left(1-x_i^2\right)^2} & 1 \le i = j \le N-1 \\ \dfrac{2(-1)^j}{3d_j}\left[\dfrac{(2N^2+1)(1-x_j)-6}{(1-x_j)^2}\right] & i = 0, \quad 1 \le j \le N \\ \\ \dfrac{2(-1)^{N+j}}{3d_j}\left[\dfrac{(2N^2+1)(1+x_j)-6}{(1+x_j)^2}\right] & i = N, \quad 0 \le j \le N-1 \\ \dfrac{(N^4-1)}{15} & i = j = 0, N \end{cases} \tag{122}$$

The matrix $\boldsymbol{D}^2$ can be also approximated by a product of two matrices $\boldsymbol{D}^1$, i.e.,

$$\boldsymbol{D}^2 \simeq \boldsymbol{D}^1\boldsymbol{D}^1, \tag{123}$$

although $\boldsymbol{D}^2$ is obviously more accurate than $\boldsymbol{D}^1\boldsymbol{D}^1$.

*Example:* Chebyshev-Gauss-Lobatto nodes are defined in $[-1,1]$. If we are given a VBP on the interval $[a,b]$ then we can transform it to $[-1,1]$ by using the following elementary coordinate transformation

$$x = \frac{b-a}{2}z + \frac{b+a}{2} \qquad z \in [-1,1]. \tag{124}$$

This yields the following transformation for the derivatives

$$y(x) = y\left(\frac{b-a}{2}z + \frac{b+a}{2}\right) \qquad \Rightarrow \qquad \frac{dy}{dx} = \frac{dy}{dz}\frac{dz}{dx} = \frac{dy}{dz}\frac{2}{b-a}, \tag{125}$$

and

$$\frac{d^2y}{dx^2} = \frac{d^2y}{dz^2}\left(\frac{2}{b-a}\right)^2. \tag{126}$$

The last equation implies that the differentiation matrix for a function defined in $[a,b]$ is simply a re-scaled version of the differentiation matrix in $[-1,1]$, the rescaling factor being some power of $2/(b-a)$. Substituting (119) into (81) (mapped from $x \in [0,1]$ to $z \in [-1,1]$ using the simple transformation $z = 2x-1$) and setting the residual equal to zero at the nodes (116) yields the system of equations

$$\begin{cases} 2\displaystyle\sum_{k=0}^{N} D^2_{jk}u_k = f\left(\frac{z_j+1}{2}\right) & j = 1,\dots,N-1 \\ u_0 = \alpha \\ u_N = \beta \end{cases} \tag{127}$$

where $z_j$ are the Chebyshev-Gauss-Lobatto nodes (116). This system allows us to compute the numerical solution to the BVP (81) using the Chebyshev-Gauss-Lobatto spectral method.

## References

[1] J. S. Hesthaven, S. Gottlieb, and D. Gottlieb. *Spectral methods for time-dependent problems*, volume 21 of *Cambridge Monographs on Applied and Computational Mathematics*. Cambridge University Press, Cambridge, 2007.

[2] A. Quarteroni, R. Sacco, and F. Salieri. *Numerical mathematics*. Springer, 2007.

[3] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd. OSQP: an operator splitting solver for quadratic programs. *Mathematical Programming Computation*, 12(4):637–672, 2020.

[4] J. A. C. Weideman and L. N. Trefethen. The eigenvalues of second-order spectral differentiation matrices. *SIAM Journal on Numerical Analysis*, 25(6):1279–1298, 1988.