Boundary Value Problems and Eigenvalue Problems

Boundary Value Problems and Eigenvalue Problems

Frame 1

In previous studies, you might have focused on solving differential equations (DEs) where all necessary conditions (like position and velocity) are specified at a single starting point. These are called Initial Value Problems (IVPs).

Today, we will extend our knowledge to solve problems where conditions are specified at the *boundaries* of an interval. These are called **Boundary Value Problems (BVPs)**.

Our goal is twofold:

- Learn how to solve BVPs for differential equations.
- Apply this knowledge to find special solutions called **eigenfunctions** and their corresponding **eigenvalues** for certain DEs. (This explains the section title!)

Go to Frame 2.

Frame 2

What is the key difference between an Initial Value Problem (IVP) and a Boundary Value Problem (BVP) for a differential equation defined on an interval $[a, b]$? _____

Go to Frame 3 for the answer.

Frame 3

The key difference lies in where the conditions are specified:

- **IVP**: All conditions (e.g., $y(a)$, $y'(a)$) are given at the *same starting point* a.
- **BVP**: Conditions are split between the *endpoints* (boundaries) a and b.

We will typically consider second-order DEs of the general form:

$$y''(x) = f(x, y, y')$$

where we are looking for the function $y(x)$ on the interval $x \in [a, b]$. To solve this uniquely, we need boundary conditions (BCs).

Go to Frame 4.

Frame 4

There are several common types of boundary conditions. The simplest are **Dirichlet boundary conditions**, where the value of the function $y(x)$ is specified at both ends:

$$y(a) = \alpha, \quad y(b) = \beta$$

Here, α and β are given constants.

What physical quantity is specified at the boundaries in Dirichlet conditions? [a] The slope of the function [b] The value of the function [c] A mix of value and slope

Go to Frame 5.

Frame 5

Your answer was [a — b — c]. The correct answer is [b], the value of the function itself.

Another type is the **von Neumann boundary conditions**, where the derivative $y'(x)$ is specified at the boundaries:

$$y'(a) = \alpha, \quad y'(b) = \beta$$

What physical quantity is often related to the derivative of a position function $y(x)$? _____

Go to Frame 6.

Frame 6

The derivative $y'(x)$ often represents a rate of change, like velocity if x is time, or the slope of the function if x is a spatial coordinate. So, Neumann conditions specify the slope or rate of change at the boundaries.

We can also have **mixed boundary conditions**, which involve a linear combination of the function and its derivative at each boundary:

$$A_1 y(a) + A_2 y'(a) = \alpha, \quad B_1 y(b) + B_2 y'(b) = \beta$$

where A_1, A_2, B_1, B_2 are constants.

Go to Frame 7.

Frame 7

Finally, a special case often encountered in physics (e.g., for systems with repeating structures) is **periodic boundary conditions**:

$$y(a) = y(b) \quad \text{and} \quad y'(a) = y'(b)$$

This forces the function and its slope to have the same values at the start and end of the interval.

Now that we know the types of problems, how do we solve them? One common technique is the "Shooting Method".

Go to Frame 8.

Frame 8

The Shooting Method

A useful tool for solving BVPs is the **Shooting Method**. The main idea is clever: we convert the BVP back into an IVP (which we often know how to solve numerically, e.g., using Runge-Kutta methods) and then use iteration or adjustment to meet the boundary condition at the far end.

Let's start with a *linear* second-order DE with Dirichlet boundary conditions:

$$y'' + P(x)y' + Q(x)y = R(x)$$

$$y(a) = \alpha, \quad y(b) = \beta$$

We assume a solution exists (a reasonable physical assumption usually).

Go to Frame 9.

Frame 9

In the shooting method, we replace this single BVP with *two* auxiliary IVPs. The trick is how we choose the initial conditions.

Let $u(x)$ be the solution to the first IVP:

$$u'' + P(x)u' + Q(x)u = R(x)$$

with initial conditions:

$$u(a) = \alpha, \quad u'(a) = 0$$

Does $u(x)$ satisfy the boundary condition given for the original problem at $x = a$? [Yes — No]

Go to Frame 10.

Frame 10

Yes, $u(a) = \alpha$ satisfies the first boundary condition of the original BVP. However, we just *guessed* the initial slope $u'(a) = 0$. This guess is unlikely to make $u(x)$ satisfy the condition $y(b) = \beta$ at the other end.

Now, let $v(x)$ be the solution to the second IVP:

$$v'' + P(x)v' + Q(x)v = 0$$

with initial conditions:

$$v(a) = 0, \quad v'(a) = 1$$

Notice two things: 1. Does $v(a)$ satisfy the $y(a) = \alpha$ condition? [Yes (if $\alpha = 0$) — No (generally)] 2. Why is the differential equation for $v(x)$ homogeneous (right-hand side is 0), unlike the equation for $u(x)$?

Go to Frame 11.

Frame 11

1. No, $v(a) = 0$ does not generally satisfy $y(a) = \alpha$, unless α happens to be 0. 2. The DE for $v(x)$ is chosen to be homogeneous. This is part of the strategy, as we'll see.

We can solve these two IVPs numerically (e.g., using standard Runge-Kutta methods) to find the values of $u(x)$ and $v(x)$ across the interval $[a, b]$.

Now, consider a trial solution $Y(x)$ for the original BVP, constructed as a linear combination of $u(x)$ and $v(x)$:

$$Y(x) = u(x) + \vartheta v(x)$$

where ϑ is a constant we need to determine.

Does this $Y(x)$ satisfy the original differential equation $Y'' + PY' + QY = R$? (Hint: The original DE is linear). [Yes — No] What is $Y(a)$? $Y(a) =$ _____What is $Y'(a)$? $Y'(a) =$ _____

Go to Frame 12.

Frame 12

Yes, because the original DE is linear, a linear combination of a particular solution (u) and a homogeneous solution (v) is also a solution. $Y'' + PY' + QY = (u'' + Pu' + Qu) + \vartheta(v'' + Pv' + Qv) = R + \vartheta(0) = R$.

Now let's check the initial conditions: $Y(a) = u(a) + \vartheta v(a) = \alpha + \vartheta(0) = \alpha$. $Y'(a) = u'(a) + \vartheta v'(a) = 0 + \vartheta(1) = \vartheta$.

So, our constructed solution $Y(x)$ automatically satisfies the condition at $x = a$, $Y(a) = \alpha$. The parameter ϑ represents the initial slope (or "shooting angle") $Y'(a)$.

How do we find the correct value ϑ_0 for ϑ? _____

Go to Frame 13.

Frame 13

We find the correct value ϑ_0 by enforcing the boundary condition at the *other* end, $x = b$. We require:

$$Y(b) = \beta$$

Substituting our combined solution:

$$u(b) + \vartheta_0 v(b) = \beta$$

Since we found $u(b)$ and $v(b)$ by solving the IVPs numerically, we can solve this algebraic equation for ϑ_0.

What is the formula for ϑ_0? (Assume $v(b) \neq 0$) $\vartheta_0 =$ _____

Go to Frame 14.

Frame 14

Solving for ϑ_0 gives:

$$\vartheta_0 = \frac{\beta - u(b)}{v(b)}$$

For a linear BVP with Dirichlet conditions, we can find the exact required initial slope ϑ_0 in one go! We solve two IVPs, calculate ϑ_0 using the formula above, and the final solution is $Y(x) = u(x) + \vartheta_0 v(x)$. We "hit the target" β at $x = b$ with a single, calculated shot.

Go to Frame 15.

Frame 15

What if the boundary conditions are more complicated, like Neumann ($y'(b) = \beta$) or mixed? Or what if the DE is non-linear?

The shooting method still works, but usually requires iteration. Consider the non-linear BVP:

$$y''(x) = f(x, y, y'), \quad y(a) = \alpha, \quad y(b) = \beta$$

We convert this to an IVP by guessing the initial slope: Let $y_1 = y$ and $y_2 = y'$. The system is:

$$y_1' = y_2$$

$$y_2' = f(x, y_1, y_2)$$

With initial conditions:

$$y_1(a) = \alpha$$

$$y_2(a) = \vartheta \quad \text{(Our guess for the initial slope } y'(a)\text{)}$$

We solve this IVP numerically from $x = a$ to $x = b$. Let the resulting value at $x = b$ be $y_1(b)|_\vartheta$.

Our goal is to make $y_1(b)|_\vartheta = \beta$. How can we frame this as a mathematical problem involving ϑ? _____

Go to Frame 16.

Frame 16

We want to find the value of ϑ such that the solution hits the target β. We can define a function $F(\vartheta)$ which represents the "miss distance":

$$F(\vartheta) = y_1(b)|_\vartheta - \beta$$

We are looking for the value of ϑ that makes $F(\vartheta) = 0$.

This is a **root-finding problem**. We need to find the root ϑ_0 of the function $F(\vartheta)$.

How can we find this root? (Think about numerical methods). _____

Go to Frame 17.

Frame 17

We can use standard numerical root-finding algorithms, such as:

- Bisection Method (if we can bracket the root)
- Secant Method
- Newton-Raphson Method (if we can also compute $dF/d\vartheta$)

Each step in these methods requires evaluating $F(\vartheta)$ for one or more values of ϑ. This means we have to: 1. Choose a trial value for $\vartheta = y'(a)$. 2. Solve the IVP from $x = a$ to $x = b$ numerically. 3. Find the resulting $y(b)|_\vartheta$. 4. Calculate $F(\vartheta) = y(b)|_\vartheta - \beta$. 5. Use the root-finding algorithm to choose the next trial ϑ. 6. Repeat until $|F(\vartheta)|$ is sufficiently small.

This iterative process is the essence of the shooting method for non-linear problems or more complex boundary conditions.

Go to Frame 18.

Frame 18

Eigenvalue Problems

A special type of BVP involves finding **eigenvalues** (λ) and corresponding **eigenfunctions** ($y(x)$). A typical form involves a parameter λ in the DE and often homogeneous boundary conditions (BCs set to zero):

$$y''(x) = f(x, y, y', \lambda), \quad y(a) = 0, \quad y(b) = 0$$

For example, the Sturm-Liouville equation:

$$y'' = -Q(x)y - \lambda V(x)y$$

Can we use the shooting method here? What quantity do we need to adjust or "shoot" for? _____

Go to Frame 19.

Frame 19

Yes, we can adapt the shooting method. Here, the unknown parameter we need to find is the eigenvalue λ.

We can set up an IVP. Since the equation is typically linear in y for eigenvalue problems, and the BCs are homogeneous ($y(a) = 0$), we can fix the initial slope without loss of generality (due to linearity, the solution is determined only up to a multiplicative constant). A common choice is:

$$y(a) = 0, \quad y'(a) = 1$$

We then solve the IVP $y'' = f(x, y, y', \lambda)$ numerically from a to b for a *chosen value* of λ. Let the result at $x = b$ be $y(b)|_\lambda$.

What condition must $y(b)|_\lambda$ satisfy if λ is a true eigenvalue? _____

Go to Frame 20.

Frame 20

For λ to be an eigenvalue, the solution $y(x)$ must satisfy the boundary condition at $x = b$. Since the condition is $y(b) = 0$, we must have:

$$y(b)|_\lambda = 0$$

So, similar to the non-linear BVP case, we are looking for the roots of a function, but this time the function depends on λ:

$$F(\lambda) = y(b)|_\lambda$$

We need to find the values of λ for which $F(\lambda) = 0$. These values are the eigenvalues.

Typically, there will be multiple (often infinitely many, but discretely spaced) eigenvalues $\lambda_1, \lambda_2, ...$ that satisfy this condition. The corresponding solutions $y(x)|_{\lambda_n}$ are the eigenfunctions. Root-finding methods are used again to find these specific λ values.

Go to Frame 21.

Frame 21

The Finite Difference Method

An alternative to the shooting method is the **Finite Difference Method**. Instead of converting the BVP to IVPs, this method discretizes the entire problem directly.

The core idea is to replace the derivatives in the DE with algebraic approximations based on the function values at discrete points on a grid.

Let's divide the interval $[a, b]$ into N equal subintervals, each of width $h = (b - a)/N$. The grid points are $x_j = a + jh$ for $j = 0, 1, ..., N$. We denote the approximate value of the solution at these points as $Y_j \approx y(x_j)$.

How can we approximate the second derivative $y''(x_j)$ using values at neighboring grid points Y_{j-1}, Y_j, Y_{j+1}? (Recall the central difference formula). $y''(x_j) \approx$ _____

Go to Frame 22.

Frame 22

The standard second-order central difference approximation for the second derivative is:

$$y''(x_j) \approx \frac{Y_{j+1} - 2Y_j + Y_{j-1}}{h^2}$$

Similarly, the central difference for the first derivative is:

$$y'(x_j) \approx \frac{Y_{j+1} - Y_{j-1}}{2h}$$

We substitute these approximations into the original DE, $y'' = f(x, y, y')$. For example, if $y'' = f(x, y, y')$, the discretized equation at point x_j becomes:

$$\frac{Y_{j+1} - 2Y_j + Y_{j-1}}{h^2} = f\left(x_j, Y_j, \frac{Y_{j+1} - Y_{j-1}}{2h}\right)$$

This equation relates the unknown values Y_{j-1}, Y_j, Y_{j+1}.

How many such equations do we get if we write one for each *internal* grid point $j = 1, 2, ..., N - 1$? _

Go to Frame 23.

Frame 23

We get $N - 1$ algebraic equations. We also have $N + 1$ grid points (x_0 to x_N) and thus potentially $N + 1$ unknown values (Y_0 to Y_N).

How do we get enough equations to solve for the unknowns? Where do the other two necessary pieces of information come from? _____

Go to Frame 24.

Frame 24

The other two pieces of information come from the **boundary conditions**. For simple Dirichlet conditions, $y(a) = \alpha$ and $y(b) = \beta$, these translate directly to:

$$Y_0 = \alpha$$

$$Y_N = \beta$$

These values are known. So, we have $N-1$ equations (from the discretized DE at internal points $j = 1...N-1$) for the $N - 1$ unknown internal values $Y_1, Y_2, ..., Y_{N-1}$.

What form does this system of $N - 1$ algebraic equations often take, especially if the original DE is linear? _____

Go to Frame 25.

Frame 25

The system of $N-1$ equations for the $N-1$ unknowns $Y_1, ..., Y_{N-1}$ often forms a **linear system** if the original DE was linear. Furthermore, because the finite difference formulas typically only link adjacent points $(j-1, j, j+1)$, the resulting matrix for this linear system is often **tridiagonal**.

For example, the equation from Frame 22, if f is linear like $f = -Py' - Qy - R$, becomes:

$$\frac{Y_{j+1} - 2Y_j + Y_{j-1}}{h^2} = -P(x_j)\left(\frac{Y_{j+1} - Y_{j-1}}{2h}\right) - Q(x_j)Y_j - R(x_j)$$

Rearranging this puts all the Y terms on one side, resulting in an equation of the form $c_{j-1}Y_{j-1} + c_jY_j + c_{j+1}Y_{j+1} = d_j$. This structure leads to a tridiagonal matrix system when written for all j.

Let $\vec{Y}$ be the vector $[Y_1, Y_2, ..., Y_{N-1}]^T$. The linear system can be written as:

$$\mathbf{A}\vec{Y} = \vec{V}$$

where $\mathbf{A}$ is an $(N-1) \times (N-1)$ tridiagonal matrix, and $\vec{V}$ is a vector containing terms involving $R(x_j)$ and the known boundary values $Y_0 = \alpha$ and $Y_N = \beta$.

Go to Frame 26.

Frame 26

If the original DE $y'' = f(x, y, y')$ is *non-linear*, the discretized system

$$\frac{Y_{j+1} - 2Y_j + Y_{j-1}}{h^2} = f\left(x_j, Y_j, \frac{Y_{j+1} - Y_{j-1}}{2h}\right)$$

results in a system of *non-linear* algebraic equations for $Y_1, ..., Y_{N-1}$.

This system can still be written in vector form, often like:

$$\mathbf{A}\vec{Y} = \vec{V}(\vec{Y})$$

where $\mathbf{A}$ might represent the linear part (from the Y'' term) and $\vec{V}(\vec{Y})$ contains the non-linear function evaluations $f(...)$ and boundary terms.

How can we solve such a non-linear system? (Recall methods discussed earlier or known from numerical analysis). _____

Go to Frame 27.

Frame 27

We can solve the non-linear system $\mathbf{A}\vec{Y} = \vec{V}(\vec{Y})$ (or more generally $\vec{F}(\vec{Y}) = \vec{0}$) using iterative methods:

1. **Picard Iteration (Simple Iteration):** Guess $\vec{Y}^{(0)}$, then iterate $\mathbf{A}\vec{Y}^{(n+1)} = \vec{V}(\vec{Y}^{(n)})$. At each step, solve a linear system for $\vec{Y}^{(n+1)}$. Convergence is not guaranteed or might be slow. Regularization (modifying the iteration, see slide 14) can sometimes help.

2. **Newton-Raphson Method (Multidimensional):** This is often more robust. Frame the problem as finding the root of $\vec{F}(\vec{Y}) = \mathbf{A}\vec{Y} - \vec{V}(\vec{Y}) = \vec{0}$. The iteration involves solving the linear system $\mathbf{J}(\vec{Y}^{(n)})\Delta\vec{Y} = -\vec{F}(\vec{Y}^{(n)})$ for the update $\Delta\vec{Y}$, where $\mathbf{J}$ is the Jacobian matrix $\partial\vec{F}/\partial\vec{Y}$. Then $\vec{Y}^{(n+1)} = \vec{Y}^{(n)} + \Delta\vec{Y}$.

Why is the Newton-Raphson method often preferred despite the need to compute the Jacobian? _____

Go to Frame 28.

Frame 28

Newton-Raphson typically converges much faster (quadratically) than simple Picard iteration (linearly), especially when close to the solution. Even though computing or approximating the Jacobian $\mathbf{J}$ and solving the linear system $\mathbf{J}\Delta\vec{Y} = -\vec{F}$ at each step is more work than the Picard step, the faster convergence often makes it more efficient overall.

Furthermore, for finite difference methods applied to second-order DEs, the Jacobian $\mathbf{J}$ is often tridiagonal or banded, allowing the linear system solve step to be done very efficiently using specialized algorithms (like the Thomas algorithm or 'scipy.linalg.solve$_b$*anded*').

Go to Frame 29.

Frame 29

Finite Differences for Eigenvalue Problems

Let's revisit the eigenvalue problem, e.g., $y'' = -Q(x)y - \lambda V(x)y$ with $y(a) = y(b) = 0$. Using the central difference approximation for y'':

$$\frac{Y_{j+1} - 2Y_j + Y_{j-1}}{h^2} = -Q(x_j)Y_j - \lambda V(x_j)Y_j$$

Rearranging this for $j = 1...N-1$ (and using $Y_0 = Y_N = 0$) leads to a matrix equation.

What kind of standard mathematical problem does this resemble? _____

Go to Frame 30.

Frame 30

This discretizes into a **matrix eigenvalue problem**. The equations can be rearranged into the form:

$$\mathbf{Z}\vec{Y} = \Lambda\mathbf{W}\vec{Y}$$

where $\Lambda = \lambda h^2$ (related to the eigenvalue we seek), $\mathbf{Z}$ is a tridiagonal matrix incorporating terms from the discretization of y'' and the $Q(x)y$ term, and $\mathbf{W}$ is often a diagonal matrix with entries $V(x_j)$.

This is a generalized matrix eigenvalue problem. Solving it numerically (using standard linear algebra libraries) yields a set of approximate eigenvalues Λ_k (from which we get λ_k) and the corresponding eigenvectors $\vec{Y}_k$.

What does the eigenvector $\vec{Y}_k$ represent physically? _____

Go to Frame 31.

Frame 31

The eigenvector $\vec{Y}_k$ represents the discrete values (the function sampled at the grid points $x_1, ..., x_{N-1}$) of the k-th eigenfunction corresponding to the eigenvalue λ_k.

The finite difference method converts the differential eigenvalue problem into a matrix eigenvalue problem. Standard numerical linear algebra techniques can then be used to find approximations to the eigenvalues and eigenfunctions. Note that since the matrices are finite ($N-1 \times N-1$), we will only find $N-1$ eigenvalues/eigenvectors, approximating the lowest ones of the original infinite set from the continuous DE.

This concludes the overview of methods for BVPs and eigenvalue problems.

End of Section.