

# Finite Discretization of the General Ordinary Second-Order Differential Equation

Jalani Eanochs

January 5, 2026

## 1 Introduction

The purpose of this notebook is to demonstrate the mathematical process that derives the discrete equations for the three discretization methods; the finite difference, finite volume, and finite element methods. The equation to which these methods will be applied to is the general second-order differential equation:

$$\frac{d^2y}{dx^2} + a \frac{dy}{dx} + by = f(x), \quad L_1 \leq x \leq L_2 \quad (1)$$

Where  $a$  and  $b$  are constants, and  $L_1$  and  $L_2$  are the bounds of the domain. All of these are real numbers.  $f(x)$  is the *driving* function. Note that the equation is called homogenous when  $f(x) = 0$ ; otherwise, it is non-homogenous.

Given that this is a second-order equation, a unique solution to the equation requires two known values. For this exercise, two essential boundary conditions will be used. Mathematically,  $y(L_1) = d_1$  and  $y(L_2) = d_2$ .

## 2 Finite Difference Method

The finite difference (FD) method works by replacing the derivatives with finite stencils. These stencils are derived from the Taylor series.

For the first derivative:

$$\frac{dy}{dx} \approx \frac{y(x + \Delta x) - y(x)}{\Delta x}$$

For the second derivative:

$$\frac{d^2y}{dx^2} \approx \frac{y(x + \Delta x) - 2y(x) + y(x - \Delta x)}{\Delta x^2}$$

The discretized equation is gotten by substituting these stencils into equation (1):

$$\begin{aligned} & \frac{y(x + \Delta x) - 2y(x) + y(x - \Delta x)}{\Delta x^2} + a \frac{y(x + \Delta x) - y(x)}{\Delta x} + by(x) = f(x) \\ & y(x + \Delta x) - 2y(x) + y(x - \Delta x) + a\Delta x(y(x + \Delta x) - y(x)) + \Delta xby(x) = \Delta x^2 f(x) \\ & (1 + a\Delta x)y(x + \Delta x) + (b\Delta x^2 + a\Delta x - 2)y(x) + y(x - \Delta x) = \Delta x^2 f(x) \end{aligned}$$

Rewriting this in terms of the mesh spatial index, where  $f(x_1) = f_1$ :

$$\underbrace{(1 + a\Delta x)}_{\alpha} y_{n+1} + \underbrace{(b\Delta x^2 + a\Delta x - 2)}_{\beta} y_n + y_{n-1} = \Delta x^2 f(x) \quad (2)$$

Now, writing these equations for each node from 0 to the n-th node:

$$\begin{aligned}
\alpha y_2 + \beta y_1 + y_0 &= f_1 \cdot \Delta x^2 \rightarrow \alpha y_2 + \beta y_1 = f_1 \cdot \Delta x^2 - y_0 \\
\alpha y_3 + \beta y_2 + y_1 &= f_2 \cdot \Delta x^2 \\
\alpha y_4 + \beta y_3 + y_2 &= f_3 \cdot \Delta x^2 \\
&\vdots \\
\alpha y_{n-1} + \beta y_{n-2} + y_{n-3} &= f_{n-2} \cdot \Delta x^2 \\
\alpha y_n + \beta y_{n-1} + y_{n-2} &= f_{n-1} \cdot \Delta x^2 \rightarrow \beta y_{n-1} + y_{n-2} = f_{n-1} \cdot \Delta x^2 - \alpha y_n
\end{aligned}$$

Because of the known essential boundary conditions,  $y_0 \equiv y(x_0) = y(L_1)$  and  $y_n \equiv y(x_n) = y(L_2)$ , these values are moved to the right side of the equation.

This linear system form a tridiagonal matrix that can be trivially solved using Thomas' algorithm (besides normal LU decomposition, or inefficient inversion):

$$\left[ \begin{array}{ccccccc} \beta & \alpha & 0 & 0 & \cdots & \cdots & \cdots \\ 1 & \beta & \alpha & 0 & \cdots & \cdots & \cdots \\ 0 & 1 & \beta & \alpha & \cdots & \cdots & \cdots \\ 0 & 0 & 1 & \beta & \ddots & \cdots & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \vdots & 1 & \beta \end{array} \right] \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ \vdots \\ y_{n-1} \end{bmatrix} = \begin{bmatrix} \Delta x^2 f_1 - y_0 \\ \Delta x^2 f_2 \\ \Delta x^2 f_3 \\ \vdots \\ \vdots \\ (\Delta x^2 f_{n-1}) - \alpha y_n \end{bmatrix} \quad (3)$$

Where  $\alpha = 1 + a\Delta x$  and  $\beta = b\Delta x^2 + a\Delta x - 2$ .

### 3 Finite Volume Method

The finite volume (FV) method works by enforcing conservation over a control volume. This means that the flow of a physical quantity in and out of a control volume equals to the change in that volume over time. This flow also known as the *flux*.

Mathematically, this relationship is represented as:

$$\frac{\partial u}{\partial t} + \frac{\partial F}{\partial x} = S(x, t)$$

Where  $F(x, t)$  is defined as the **flux of the physical quantity**  $u$ , and  $S(x, t)$  is the source/sink term. The equation in this above **conservative form** is then integrated over the entire control volume. By the second theorem of the Fundamental Theorem of Calculus, the flux term is immediately evaluated since its derivative is defined over the spatial coordinates. The above equation then becomes:

$$\int_{\Omega} \frac{\partial u}{\partial t} d\Omega + F(x + \frac{1}{2}\Delta x) - F(x - \frac{1}{2}\Delta x) = \int_{\Omega} S(x, t) \quad (4)$$

The  $F(x \pm \Delta x)$  terms represent the flux at the boundaries of the control volume (the fluxes must be evaluated at the boundaries).

For the objective equation, it does not actually represent a physical law. Nevertheless, this method can still be applied while minding the mathematical form given above.

Since there are no time terms in equation(1), the equation can be rewritten as:

$$\frac{dF}{dx} = S(x)$$

Note that the left side needs to be expressed as the spatial derivative of the flux term.

So, rewriting the head equation:

$$\frac{d}{dx} \underbrace{\left( \frac{dy}{dx} + ay(x) \right)}_{F(x)} = \underbrace{f(x) - by(x)}_{S(x)}, \quad L_1 \leq x \leq L_2 \quad (5)$$

Now, the integral over the control volume (or more accurately, control *line*, given the 1D nature of the problem) will be taken:

$$\begin{aligned} \int_{x_{n-\frac{1}{2}}}^{x_{n+\frac{1}{2}}} \frac{d}{dx} \left( \frac{dy}{dx} + ay(x) \right) dx &= \int_{x_{n-\frac{1}{2}}}^{x_{n+\frac{1}{2}}} f(x) - by(x) dx \\ \left( \frac{dy}{dx} + ay(x) \right) \Big|_{x_{n-\frac{1}{2}}}^{x_{n+\frac{1}{2}}} &= \int_{x_{n-\frac{1}{2}}}^{x_{n+\frac{1}{2}}} f(x) dx - \int_{x_{n-\frac{1}{2}}}^{x_{n+\frac{1}{2}}} by(x) dx \end{aligned}$$

The integrals will be evaluated using the midpoint rule, while the derivative of  $y$  at  $x_{n\pm\frac{1}{2}}$  will be replaced by a first order approximation stencil. Finally, the point  $y_{n\pm\frac{1}{2}}$  will be evaluated by the midpoint equation.

Evaluating these substitutions:

$$\begin{aligned} \left[ \frac{dy}{dx}(x_{n+\frac{1}{2}}) + ay(x_{n+\frac{1}{2}}) \right] - \left[ \frac{dy}{dx}(x_{n-\frac{1}{2}}) + ay(x_{n-\frac{1}{2}}) \right] &= f(x_n) \cdot \Delta x - by(x_n) \cdot \Delta x \\ \left[ \frac{y_{n+1} - y_n}{\Delta x} + a \frac{y_{n+1} + y_n}{2} \right] - \left[ \frac{y_n - y_{n-1}}{\Delta x} + a \frac{y_n + y_{n-1}}{2} \right] &= f_n \cdot \Delta x - by_n \cdot \Delta x \\ 2y_{n+1} - 2y_n + a\Delta x y_{n+1} + a\Delta x y_n - 2y_n + 2y_{n-1} - a\Delta x y_n - a\Delta x y_{n-1} &= 2\Delta x^2 f_n - 2b\Delta x^2 y_n \end{aligned}$$

This results in the following:

$$\underbrace{(2 + a\Delta x)}_{\alpha} y_{n+1} + \underbrace{(2b\Delta x^2 - 4)}_{\beta} y_n + \underbrace{(2 - a\Delta x)}_{\gamma} y_{n-1} = 2\Delta x^2 f_n \quad (6)$$

Expanding again to note patterns:

$$\begin{aligned} \alpha y_2 + \beta y_1 + \gamma y_0 &= 2\Delta x^2 f_1 \rightarrow \alpha y_2 + \beta y_1 = 2\Delta x^2 f_1 - \gamma y_0 \\ \alpha y_3 + \beta y_2 + \gamma y_1 &= 2\Delta x^2 f_2 \\ \alpha y_4 + \beta y_3 + \gamma y_2 &= 2\Delta x^2 f_3 \\ &\vdots \\ \alpha y_{n-1} + \beta y_{n-2} + \gamma y_{n-3} &= 2\Delta x^2 f_{n-2} \\ \alpha y_n + \beta y_{n-1} + \gamma y_{n-2} &= 2\Delta x^2 f_{n-1} \rightarrow \beta y_{n-1} + \gamma y_{n-2} = 2\Delta x^2 f_{n-1} - \alpha y_n \end{aligned}$$

Note that the first and last equations are on the right side of the equation since these nodal values are known.

This linear system form a tridiagonal matrix that can be trivially solved using Thomas' algorithm (besides normal LU decomposition, or inefficient inversion):

$$\begin{bmatrix} \beta & \alpha & 0 & 0 & \cdots & \cdots & \cdots \\ \gamma & \beta & \alpha & 0 & \cdots & \cdots & \cdots \\ 0 & \gamma & \beta & \alpha & \cdots & \cdots & \cdots \\ 0 & 0 & \gamma & \beta & \ddots & \cdots & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \vdots & \gamma & \beta \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ \vdots \\ y_{n-1} \end{bmatrix} = \begin{bmatrix} 2\Delta x^2 f_1 - \gamma y_0 \\ 2\Delta x^2 f_2 \\ 2\Delta x^2 f_3 \\ \vdots \\ \vdots \\ 2\Delta x^2 f_{n-1} - \alpha y_n \end{bmatrix} \quad (7)$$

Where  $\alpha = 2 + a\Delta x$ ,  $\beta = 2b\Delta x^2 - 4$  and  $\gamma = 2 - a\Delta x$ .

## 4 Finite Element Method

The finite element (FE) method approximates the solution by defining smaller spatial regions, or "elements", across the domain. The solution is then approximated by evaluating the basis functions defined over the length of an individual element according to the main equation. This is done by deriving a *weak variational integral equation* from the original *strong differential equation*.<sup>1</sup>

To start, the strong form of the equation is multiplied by a (arbitrary) weight function  $w(x)$ . An important quality of the weight function is that it must be equal to zero wherever the solution is known. Thus,  $w(L_1) = w(L_2) = 0$  since the values at the boundary are known.

After the multiplication, the integral across the entire domain will be taken. The second derivative will be reduced to a first derivative using integration by parts.

$$\begin{aligned} \int_{\Omega} w \left( \frac{d^2y}{dx^2} + a \frac{dy}{dx} + by \right) d\Omega &= \int_{\Omega} wf(x) d\Omega \\ \int_{\Omega} w \frac{d^2y}{dx^2} d\Omega + a \int_{\Omega} w \frac{dy}{dx} d\Omega + b \int_{\Omega} wy d\Omega &= \int_{\Omega} wf(x) d\Omega \\ w \frac{dy}{dx} \Big|_{\delta\Omega_1}^{\delta\Omega_2} - \int_{\Omega} \frac{dw}{dx} \frac{dy}{dx} d\Omega + a \int_{\Omega} w \frac{dy}{dx} d\Omega + b \int_{\Omega} wy d\Omega &= \int_{\Omega} wf(x) d\Omega \end{aligned}$$

Recall that  $w(L_1) = w(L_2) = 0$ . So, the first term on the left hand side is equal to zero:

$$-\int_{\Omega} \frac{dw}{dx} \frac{dy}{dx} d\Omega + a \int_{\Omega} w \frac{dy}{dx} d\Omega + b \int_{\Omega} wy d\Omega = \int_{\Omega} wf(x) d\Omega, \text{ where } w(\delta\Omega_1) = w(\delta\Omega_2) = 0 \quad (8)$$

This is the variational weak integral equation.

Now, the continuous functions  $y(x)$  and  $w(x)$  will be approximated by a weighted sum of basis functions. By the Galerkin method,  $y$  and  $w$  will use the same basis functions.

$$y(x) \approx \sum_i y_i \phi_i(x) \quad w(x) \approx \sum_i w_i \phi_i(x) \quad (9)$$

Substituting back into the integral equation:

$$\begin{aligned} -\int_{\Omega} \sum_i w_i \frac{d\phi_i}{dx} \sum_j y_j \frac{d\phi_j}{dx} d\Omega + a \int_{\Omega} \sum_i w_i \phi_i(x) \sum_j y_j \frac{d\phi_j}{dx} d\Omega \\ + b \int_{\Omega} \sum_i w_i \phi_i(x) \sum_j y_j \phi_j(x) d\Omega = \int_{\Omega} f(x) \sum_i w_i \phi_i(x) d\Omega \\ -\int_{\Omega} \sum_i \sum_j w_i y_j \frac{d\phi_i}{dx} \frac{d\phi_j}{dx} d\Omega + a \int_{\Omega} \sum_i \sum_j w_i y_j \phi_i(x) \frac{d\phi_j}{dx} d\Omega + b \int_{\Omega} \sum_i \sum_j w_i y_j \phi_i(x) \phi_j(x) d\Omega = \int_{\Omega} f(x) \sum_i w_i \phi_i(x) d\Omega \\ \sum_i \sum_j w_i y_j \left( -\int_{\Omega} \frac{d\phi_i}{dx} \frac{d\phi_j}{dx} d\Omega + a \int_{\Omega} \phi_i(x) \frac{d\phi_j}{dx} d\Omega + b \int_{\Omega} \phi_i(x) \phi_j(x) d\Omega \right) = \sum_i w_i \int_{\Omega} f(x) \phi_i(x) d\Omega \end{aligned}$$

---

<sup>1</sup>Note that the terms "strong" and "weak" denotes the fact that differential equations require very smooth and well-defined functions - strong requirements - while an integral equation can accept piecewise and less continuous equations - a weak requirement.

Since the above equation must hold for any arbitrary function  $w(x)$ , for a given weight coefficient  $w_i$ , all of the  $y_j$  coefficients must satisfy the equation. This produces a system of equations:

$$\sum_j y_j \left( - \int_{\Omega} \frac{d\phi_i}{dx} \frac{d\phi_j}{dx} d\Omega + a \int_{\Omega} \phi_i(x) \frac{d\phi_j}{dx} d\Omega + b \int_{\Omega} \phi_i(x) \phi_j(x) d\Omega \right) = \int_{\Omega} f(x) \phi_i(x) d\Omega,$$

for a given  $i$ , where  $i = 0, 1, \dots, N_n - 1$  (10)

Note that  $N_n$  is the number of nodes in the mesh.

Up to this point, the integrals have been defined over the entire domain. The integral equation can be rewritten as the sum of integral taken *over each individual element* in the domain. So, the integral equation is further rewritten as (with proper indices of summation):

$$\sum_{e=0}^{N_e-1} \sum_{j=0}^{N_n-1} y_j \left( - \int_{\Omega_e} \frac{d\phi_i}{dx} \frac{d\phi_j}{dx} d\Omega_e + a \int_{\Omega_e} \phi_i(x) \frac{d\phi_j}{dx} d\Omega_e + b \int_{\Omega_e} \phi_i(x) \phi_j(x) d\Omega_e \right) = \sum_{e=0}^{N_e-1} \int_{\Omega_e} f(x) \phi_i(x) d\Omega_e$$

Now, for a given element  $e$ , one term of the summation is represented as:

$$\sum_{j=0}^{N_n-1} y_j \left( - \int_{\Omega_e} \frac{d\phi_i}{dx} \frac{d\phi_j}{dx} d\Omega_e + a \int_{\Omega_e} \phi_i(x) \frac{d\phi_j}{dx} d\Omega_e + b \int_{\Omega_e} \phi_i(x) \phi_j(x) d\Omega_e \right) = \int_{\Omega_e} f(x) \phi_i(x) d\Omega_e$$

This expression forms a linear equation **over a single element**. This is significant since the same set of basis functions can be used through the domain over multiple elements. Furthermore, the matrices formed for each element represent *part* of the global matrix and vectors. This local element matrix can then be added to the global matrix defined in equation (10).

Now, the equation above sums over all indices in the mesh. However, the basis functions by element will be equal to zero everywhere else except in their parent element. So, the equation can be rewritten as:

$$\sum_{j=0}^{N_{n,e}-1} y_j \underbrace{\left( - \int_{\Omega_e} \frac{d\phi_i}{dx} \frac{d\phi_j}{dx} d\Omega_e + a \int_{\Omega_e} \phi_i(x) \frac{d\phi_j}{dx} d\Omega_e + b \int_{\Omega_e} \phi_i(x) \phi_j(x) d\Omega_e \right)}_{[K_{ij}]} = \underbrace{\int_{\Omega_e} f(x) \phi_i(x) d\Omega_e}_{[F_i]} \quad (11)$$

Where  $i = 0, 1, \dots, N_{n,e}$  and  $N - n, e$  is the number of nodes per element. For a linear element, it is 2 nodes; for a quadratic element, it is 3 nodes.

With the element-wise matrix defined, the global system can be solved as follows:

1. Iterate through each element and calculate the element-wise matrix  $[K_{ij}]$
2. Add each local matrix to the global matrix in the appropriate area given by the connectivity grid
3. Solve the global matrix equation

To expand (or "scatter") the local matrix to the global matrix, the region of the global matrix that the element matrix needs to add to is represented by the following equation:  $(i_g, j_g) = (i + (n_{e,n} - 1)(e - 1), j + (n_{e,n} - 1)(e - 1))$ , where  $(i_g, j_g)$  is the index of the global matrix and  $i$  and  $j$  are the indices of the local element matrix, from 0 to  $N_{n,e} - 1$ . For example, for linear elements, the element mapping is represented below:

$$\begin{bmatrix} k_{11}^e & k_{12}^e \\ k_{21}^e & k_{22}^e \end{bmatrix} \rightarrow \begin{bmatrix} \ddots & \vdots & & \vdots \\ \cdots & k_{1+(e-1),1+(e-1)} & k_{1+(e-1),2+(e-1)} & \cdots \\ \cdots & k_{2+(e-1),1+(e-1)} & k_{2+(e-1),2+(e-1)} & \cdots \\ & \vdots & \vdots & \ddots \end{bmatrix}$$

Note that these are \*\*added\*\* to the entry in the matrix.

A similar step is done for the right hand side. For each of the entries, the location of where they are to be added is encoded in the equation  $i_g = i + (n_{e,n} - 1)(e - 1)$ , where  $i_g$  is the index of the global matrix and  $i$  is the index of the local element matrix from 0 to  $N_{n,e} - 1$ . For example, for the linear element, the mapping by element is below:

$$\begin{bmatrix} f_{11}^e \\ f_{21}^e \end{bmatrix} \rightarrow \begin{bmatrix} \vdots \\ f_{1+(e-1)} \\ f_{2+(e-1)} \\ \vdots \end{bmatrix}$$

With this, the final thing to consider are the actual evaluation of the integrals and the choice of the basis functions for the elements.

## 4.1 Integration and Basis Functions

In this section, further consideration is given to the evaluation of the integrals.

Firstly, the basis functions that will be used are the Lagrange polynomials.

For a given set of  $k + 1$  nodes (which, in this case, are the element nodes), the Lagrange basis for polynomials of the degree  $\leq k$  is a set of polynomials  $l_m(x)$  from 0 to  $k$ . The generating function for a given polynomial of this basis is:

$$l_j(x) = \prod_{\substack{0 \leq m \leq k \\ m \neq j}} \frac{x - x_m}{x_j - x_m} \quad (12)$$

In the context of the elements, the basis functions can be defined for an element composed of any number of nodes. For this exercise, only linear and quadratic nodes will be considered, but the concept can be trivially generalized to any order.

For the linear elements, there are two basis functions:

$$\phi_0(x) = \frac{x - x_1}{x_0 - x_1}, \quad \phi_1(x) = \frac{x - x_0}{x_1 - x_0} \quad (13)$$

For the quadratic elements, there are three basis functions:

$$\phi_0(x) = \frac{x - x_1}{x_0 - x_1} \frac{x - x_2}{x_0 - x_2}, \quad \phi_1(x) = \frac{x - x_0}{x_1 - x_0} \frac{x - x_2}{x_1 - x_2}, \quad \phi_2(x) = \frac{x - x_0}{x_2 - x_0} \frac{x - x_1}{x_2 - x_1} \quad (14)$$

From here, it is possible to numerically evaluate the integral over the *physical* domain of the element. However, for more complicated elements in 2D and 3D, it is helpful to map the elements to an ideal element called the isoparametric element. This is an element that is a regular polygon or shape, defined at simple, trivial points (most of time, positive and negative 1).

For the 1D element, the linear isoparametric element is defined with the endpoint nodes at -1 and 1. The quadratic element has nodes at -1, 0, and 1.

The basis functions in terms of the isoparametric element are defined as:

$$\phi_0(\zeta) = \frac{1 - \zeta}{2}, \quad \phi_1(\zeta) = \frac{1 + \zeta}{2} \quad (15)$$

and

$$\phi_0(\zeta) = \frac{\zeta(\zeta - 1)}{2}, \quad \phi_1(\zeta) = 1 - \zeta^2, \quad \phi_2(\zeta) = \frac{\zeta(\zeta + 1)}{2} \quad (16)$$

The mapping for these functions from the isoparametric domain to the physical is the sum of the products of the shape functions and their associated nodes. For example:

$$x(\zeta) = \phi_1 \cdot x_1 + \phi_2 \cdot x_2$$

Finally, the derivative of these above forms are as follows:

## Shape Functions

$$\begin{aligned}\phi'_0(\zeta) &= -\frac{1}{2}, & \phi'_1(\zeta) &= \frac{1}{2} \\ \phi'_0(\zeta) &= \zeta - \frac{1}{2}, & \phi'_1(\zeta) &= -2\zeta, & \phi'_2(x) &= \zeta + \frac{1}{2}\end{aligned}\tag{17}$$

### Jacobian: Derivative of Mapping

$$\begin{aligned}\frac{dx}{d\zeta} &= \frac{x_2 - x_1}{2} \\ \frac{dx}{d\zeta} &= (\zeta - \frac{1}{2})x_1 - 2\zeta x_2 + (\zeta + \frac{1}{2})x_3\end{aligned}\tag{18}$$

Note that  $J = \frac{dx}{d\zeta}$ .

Now, to substitute back into integral equations, using the chain rule, the derivative can be expressed as:

$$\frac{d\phi}{dx} = \frac{d\phi}{d\zeta} \frac{d\zeta}{dx}$$

So, finally, the element wise coefficient and constant matrices can be defined as:

$$\begin{aligned}K_{ij} &= \int_{-1}^1 \phi'_i(\zeta)\phi'_j(\zeta) \cdot \frac{1}{J} + a\phi_i(\zeta)\phi'_j(\zeta) + b\phi_i(\zeta)\phi_j(\zeta) \cdot J \, d\zeta \\ F_i &= \int_{-1}^1 f(x(\zeta)) \cdot \phi_i(\zeta) \cdot J \, d\zeta\end{aligned}\tag{19}$$

As stated above, these equations populate the local matrix and vector for each element. These arrays are then added to their appropriate spots in the global linear equation:

$$[K][y] = [F]\tag{20}$$

This equation can then be solved using normal methods, though LU decomposition or sparse matrix solvers are the most efficient.

## 5 Conclusion and Computational Implementation

This concludes the mathematical derivations for each of the finite discretization methods. These discretizations are implemented in the ODE Solver library, which solves ODEs of the form shown in equation (1). The writer of the document chooses to write the library in C using the GNU Scientific Library (GSL).