NE 155, Classes 22-24 S15
1-D Finite Difference and Volume Methods for 1D
March 13-18, 2015

Much of this can be found in Duderstadt and Hamilton Chp. 5 Section II.B.

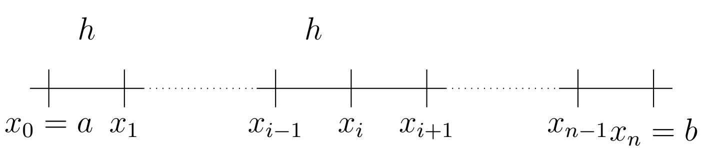
Finite-Difference Method

<u>Problem:</u> Consider the following second order ODE:

$$f''(x) = p(x)f'(x) + q(x)f(x) + r(x)$$

defined on a segment $[a, b]$ with $f(a) = \alpha$ and $f(b) = \beta$ (a boundary value problem).

Now, let's spatially discretize the equation:



where $x_0 = a$, $x_n = b$, and h is the mesh spacing. There are $n + 1$ points and n mesh cells.

We can use **central difference** to approximate the derivatives on this grid. Let's use the $O(h^2)$ versions:

$$f'(x_i) = \frac{f(x_i + h) - f(x_i - h)}{2h} - \frac{h^2}{6}f'''(\mu) \; ,$$
$$f''(x_i) = \frac{f(x_i + h) - 2f(x_i) + f(x_i - h)}{h^2} + \frac{h^2}{12}f^{(4)}(\mu) \; ,$$

We will also define $p_i = p(x_i)$, $q_i = q(x_i)$, $r_i = r(x_i)$.

Substituting into the original equation, we get:

$$\frac{f_{i+1}-2f_i+f_{i-1}}{h^2} = p_i\frac{f_{i+1}-f_{i-1}}{2h} + q_i f_i + r_i \qquad i = 1, 2, \ldots, n-1\,,$$
$$\big(\frac{-h}{2}p_i - 1\big)f_{i-1} + \big(2+h^2q_i\big)f_i + \big(\frac{h}{2}p_i - 1\big)f_{i+1} = -h^2r_i \qquad i = 1, 2, \ldots, n-1\,.$$

We only have $n-1$ equations, but because the boundaries are fixed that is all we need. This is clear when we look at the $i=1$ and $i=n-1$ cases:

$$\big(2+h^2q_1\big)f_1 + \big(\frac{h}{2}p_1 - 1\big)f_2 = -h^2r_1 + \underbrace{\big(\frac{h}{2}p_1 + 1\big)\underbrace{\alpha}_{f_0}}_{bc_L}\,,$$

$$\big(\frac{-h}{2}p_{n-1} - 1\big)f_{n-2} + \big(2+h^2q_{n-1}\big)f_{n-1} = -h^2r_{n-1} + \underbrace{\big(\frac{-h}{2}p_{n-1} + 1\big)\underbrace{\beta}_{f_n}}_{bc_R}\,.$$

Thus, we can write this as a matrix equation:

$$\begin{pmatrix} \big(2+h^2q_1\big) & \big(\frac{h}{2}p_1-1\big) & & \cdots & 0 \\ \big(\frac{-h}{2}p_2-1\big) & \big(2+h^2q_2\big) & \big(\frac{h}{2}p_2-1\big) & & \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ & & \big(\frac{-h}{2}p_{n-2}-1\big) & \big(2+h^2q_{n-2}\big) & \big(\frac{h}{2}p_{n-2}-1\big) \\ 0 & \cdots & & \big(\frac{-h}{2}p_{n-1}-1\big) & \big(2+h^2q_{n-1}\big) \end{pmatrix} \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_{n-2} \\ f_{n-1} \end{pmatrix} = \begin{pmatrix} -h^2r_1 + bc_L \\ -h^2r_2 \\ \vdots \\ -h^2r_{n-1} \\ -h^2r_{n-1} + bc_R \end{pmatrix}.$$

Note: the *numerical solution* to the PDE is an **approximation** to the *exact solution* that is obtained using a discrete representation of the PDE at the grid points x_i in the discrete spatial mesh. Let us denote this numerical solution as F such that

$$F_j \approx f(x_j)\,.$$

Thus, the numerical solution is a collection of finite values

$$F = [F_1, F_2, \ldots, F_{n-1}]\,,$$

and we have boundary values F_0 and F_n.

DE

What does this look like if we apply it to the steady-state, 1-D diffusion equation?

$$-D\frac{d^2\phi(x)}{dx^2} + \Sigma_a\phi(x) = S \; ,$$

$$\frac{d^2\phi}{dx^2} - \frac{1}{L^2}\phi(x) = \frac{-S}{D} \; .$$

Let $\phi(a) = \phi(b) = 0$ (vacuum bcs). Then we get

$$\frac{\phi_{i+1} - 2\phi_i + \phi_{i-1}}{h^2} - \frac{1}{L^2}\phi_i = \frac{-S_{0,i}}{D} \qquad i = 1, 2, \ldots, n-1 \; ,$$

$$-\phi_{i-1} + \big(2 + \frac{h^2}{L^2}\big)\phi_i - \phi_{i+1} = h^2\frac{S_{0,i}}{D} \qquad i = 1, 2, \ldots, n-1 \; .$$

One problem with this formulation is that we only have values at the cell edges: edge centered.

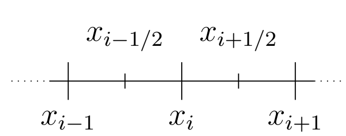
This really only works well when we have homogeneous media. Why might that be?
If we have *material discontinuities* it's going to be difficult to enforce flux continuity between cells.

Note: L and D are not functions of space; we wouldn't know which cell's values to assign when using edge-centered values.

What could we do instead? cell-centered, which leads us to our next topic.

Finite Volume Method

Rather than pointwise approximations on a grid, FVM approximates an average integral value on a reference volume.



Consider the elliptic equation $f''(x) = r(x)$ on a control volume $V_i = [x_{i-1/2}, x_{i+1/2}]$, then

$$\int_{x_{i-1/2}}^{x_{i+1/2}} f''(x)dx = \int_{x_{i-1/2}}^{x_{i+1/2}} r(x)dx \; .$$

We can evaluate the lhs analytically and the rhs using some integration rule. For simplicity, let's use the midpoint rule:

———-

Recall: from open Newton Cotes with Lagrange polynomials and $n = 0$; it only uses one point:

$$\int_a^b f(x)dx = \int_{x_{i-1/2}}^{x_{i+1/2}} f(x)dx = 2hf(x_i) + \frac{h^3}{3}f''(\xi) \; ,$$

where $x_{-1} < \xi < x_1$ and $h = (x_{i+1} - x_{i-1})/2$.

———-

$$f'(x_{i+1/2}) - f'(x_{i-1/2}) = (x_{i+1/2} - x_{i+1/2})r(x_i) \; .$$

We can now use differencing schemes to represent the derivatives on the left. Given the definition of h above, we also know $h = x_{i+1/2} - x_{i-1/2}$, giving

$$\frac{f_{i+1} - 2f_i + f_{i-1}}{h} = hr(x_i) \; .$$

This

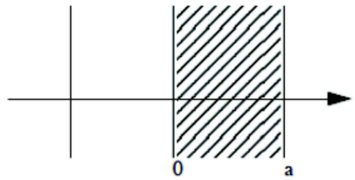
- Applies to the integral form of conservation laws (like the DE).
- Handles discontinuities in solution (because we're using cell-integrated values).
- Works well for heterogeneous systems because each cell can be a different material.
- There exists a theory for convergence, accuracy, and stability.

DE

What does this look like if we apply it to the steady-state, 1-D diffusion equation?

$$-\frac{d}{dx}D(x)\frac{d\phi(x)}{dx} + \Sigma_a(x)\phi(x) = S(x) \; .$$

Let's also assume we have an equilibrium (reflecting) condition at the centerline ($x_0 = 0$) and

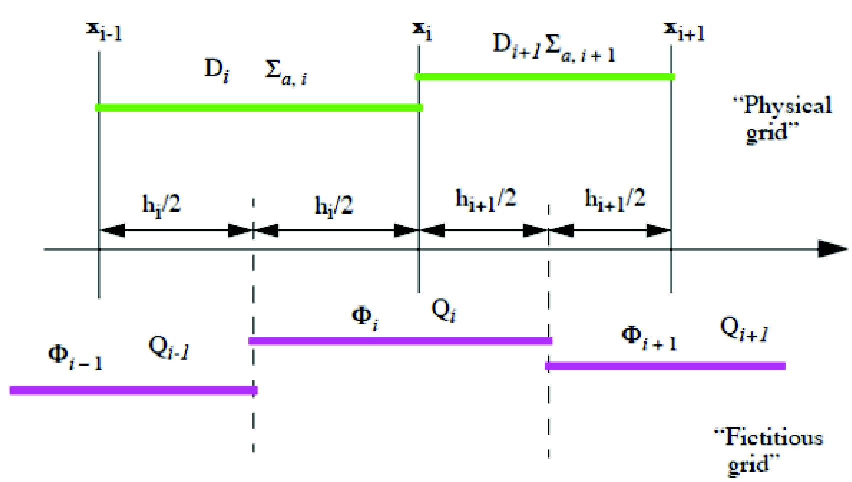


vacuum on the right ($x_n = a$):

$$\frac{d}{dx}\phi(x)\Big|_{x=0} = 0 \qquad \text{zero net current,}$$
$$\phi(\tilde{a}) = 0 \qquad \tilde{a} = a + 2D\,.$$

We again have a spatial mesh, and material discontinuities will coincide with cell edges, x_i. Thus, we assume the cross section and diffusion coefficient are constant in each cell and the unknown fluxes and known sources are defined at the cell edges:

$$D(x) = D_i\,, \qquad x_{i-1} \le x \le x_i\,,$$
$$\Sigma_a(x) = \Sigma_{a,i}\,, \qquad x_{i-1} \le x \le x_i\,,$$
$$h_i \equiv x_i - x_{i-1}\,,$$
$$\phi(x_i) = \phi_i\,,$$
$$S(x_i) = S_i\,.$$



We further assume that the fluxes and sources are constant over the interval centered around x_i:

$$\phi(x) = \phi_i \qquad \text{for } \big(x_i - \frac{h_i}{2}\big) \le x \le \big(x_i + \frac{h_{i+1}}{2}\big) \ ,$$
$$S(x) = S_i \qquad \text{for } \big(x_i - \frac{h_i}{2}\big) \le x \le \big(x_i + \frac{h_{i+1}}{2}\big) \ .$$

Now, we integrate the differential equation over each cell, $\big(x_i - \frac{h_i}{2}\big) \le x \le \big(x_i + \frac{h_{i+1}}{2}\big)$:

$$\int_{(x_i-\frac{h_i}{2})}^{(x_i+\frac{h_{i+1}}{2})} \left(-\frac{d}{dx}D(x)\frac{d\phi(x)}{dx}\right)dx + \int_{(x_i-\frac{h_i}{2})}^{(x_i+\frac{h_{i+1}}{2})} \Sigma_a(x)\phi(x)dx = \int_{(x_i-\frac{h_i}{2})}^{(x_i+\frac{h_{i+1}}{2})} S(x)dx \ .$$

Term by term:

$$\int_{(x_i-\frac{h_i}{2})}^{(x_i+\frac{h_{i+1}}{2})} S(x)dx \approx S_i\big(\frac{h_i + h_{i+1}}{2}\big) \qquad \text{defined at cell edge,}$$

$$\int_{(x_i-\frac{h_i}{2})}^{(x_i+\frac{h_{i+1}}{2})} \Sigma_a(x)\phi(x)dx = \int_{(x_i-\frac{h_i}{2})}^{(x_i)} \Sigma_a(x)\phi(x)dx + \int_{(x_i)}^{(x_i+\frac{h_{i+1}}{2})} \Sigma_a(x)\phi(x)dx$$

$$= \left(\frac{\Sigma_{a,i}h_i + \Sigma_{a,i+1}h_{i+1}}{2}\right)\phi_i \qquad \text{defined at cell center,}$$

$$\int_{(x_i-\frac{h_i}{2})}^{(x_i+\frac{h_{i+1}}{2})} \left(-\frac{d}{dx}D(x)\frac{d\phi(x)}{dx}\right)dx = -\left[D(x)\frac{d\phi(x)}{dx}\right]_{(x_i-\frac{h_i}{2})}^{(x_i+\frac{h_{i+1}}{2})} \qquad \text{defined at cell edge,}$$

$$-D(x)\frac{d\phi(x)}{dx}\Big|_{(x_i+\frac{h_{i+1}}{2})} \cong -D_{i+1}\left(\frac{\phi_{i+1} - \phi_i}{h_{i+1}}\right) ,$$

$$-D(x)\frac{d\phi(x)}{dx}\Big|_{(x_i-\frac{h_i}{2})} \cong -D_i\left(\frac{\phi_i - \phi_{i-1}}{h_i}\right) .$$

Collecting all of the terms:

$$-D_{i+1}\left(\frac{\phi_{i+1} - \phi_i}{h_{i+1}}\right) + D_i\left(\frac{\phi_i - \phi_{i-1}}{h_i}\right) + \left(\frac{\Sigma_{a,i}h_i + \Sigma_{a,i+1}h_{i+1}}{2}\right)\phi_i = S_i\big(\frac{h_i + h_{i+1}}{2}\big) \ .$$

We can express this in matrix form, but we'll use some abbreviations to make it more compact:

$$\begin{aligned} h_{ii} &= \frac{h_i + h_{i+1}}{2} \,, \\ \Sigma_{a,ii} &= \frac{\Sigma_{a,i} h_i + \Sigma_{a,i+1} h_{i+1}}{h_i + h_{i+1}} \,. \end{aligned}$$

Divide through by h_{ii}, then

$$a_{i,i-1}\phi_{i-1} + a_{i,i}\phi_i + a_{i,i+1}\phi_{i+1} = S_i \qquad \text{for } i = 1, 2, \ldots, n-1$$

where

$$\begin{aligned} a_{i,i-1} &= \frac{-D_i}{h_i h_{ii}} \,, \\ a_{i,i} &= \frac{D_i}{h_i h_{ii}} + \frac{D_{i+1}}{h_{i+1} h_{ii}} + \Sigma_{a,ii} \,, \\ a_{i,i+1} &= \frac{-D_{i+1}}{h_{i+1} h_{ii}} \,. \end{aligned}$$

Now we have a set of $n-1$ linear algebraic equations with $n+1$ unknowns. Next up: boundary conditions.

Boundary Conditions

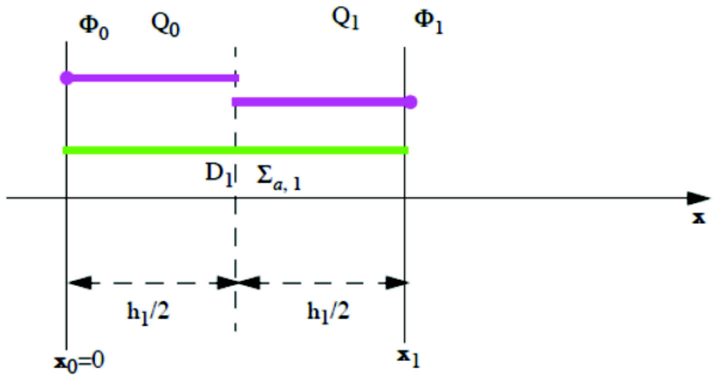
If we assume $x_n = \tilde{a}$, then the **vacuum condition** becomes

$$\phi_n = 0 \,,$$

and the equation for $i = n-1$ becomes

$$a_{n-1,n-2}\phi_{n-2} + a_{n-1,n-1}\phi_{n-1} + a_{n-1,n} \underbrace{\phi_n}_{0} = S_{n-1} \,.$$

Next we'll worry about the **reflecting** or zero current condition. The first step is to integrate over $[0, h_1/2]$.



$$\int_0^{\frac{h_1}{2}} \left(-\frac{d}{dx}D(x)\frac{d\phi(x)}{dx}\right)dx + \int_0^{\frac{h_1}{2}} \Sigma_a(x)\phi(x)dx = \int_0^{\frac{h_1}{2}} S(x)dx \; ,$$
$$-D(x)\frac{d\phi(x)}{dx}\Big|_{\frac{h_1}{2}} + D(x)\frac{d\phi(x)}{dx}\Big|_0 + \Sigma_{a,1}\phi_0\frac{h_1}{2} = S_0\frac{h_1}{2} \; .$$

Now we can apply the boundary condition $\frac{d\phi(x)}{dx}\big|_0 = 0$ to get:

$$-D(x)\frac{d\phi(x)}{dx}\Big|_{\frac{h_1}{2}} + \Sigma_{a,1}\phi_0\frac{h_1}{2} = S_0\frac{h_1}{2} \; .$$

And the first equation ($i = 0$) becomes

$$a^*_{00}\phi_0 + a^*_{01}\phi_1 = S_0 \; ,$$

where we redefine the as to be (I've added the * to indicate that these have different definitions than the rest of the terms):

$$a^*_{00} = \frac{2D_1}{h_1^2} + \Sigma_{a,1} \; ,$$
$$a^*_{01} = -\frac{2D_1}{h_1^2} \; .$$

We now have n equations and n unknowns

$$\underbrace{\begin{pmatrix} a^*_{00} & a^*_{01} & 0 & 0 & \cdots & 0 \\ a_{10} & a_{11} & a_{12} & 0 & \cdots & 0 \\ 0 & a_{21} & a_{22} & a_{23} & & \vdots \\ \vdots & & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & a_{n-3,n-3} & a_{n-2,n-2} & a_{n-2,n-1} \\ 0 & \cdots & 0 & 0 & a_{n-1,n-2} & a_{n-1,n-1} \end{pmatrix}}_{\mathbf{A}} \underbrace{\begin{pmatrix} \phi_0 \\ \phi_1 \\ \phi_2 \\ \vdots \\ \phi_{n-2} \\ \phi_{n-1} \end{pmatrix}}_{\vec{\phi}} = \underbrace{\begin{pmatrix} S_0 \\ S_1 \\ S_2 \\ \vdots \\ S_{n-2} \\ S_{n-1} \end{pmatrix}}_{\vec{S}} .$$

Homogeneous, Uniform Mesh

If we end up having a homogeneous system with a uniform mesh, then we can make some simplifications:

$$\begin{aligned}
&h_i = h \, , \\
&D_i = D \, , \\
&\Sigma_{a,i} = \Sigma_a \, , \quad \text{and then} \\
&\frac{-D}{h^2}\phi_{i-1} + \left(\frac{2D}{h^2} + \Sigma_a\right)\phi_i - \frac{D}{h^2}\phi_{i+1} = S_i \qquad \text{for } i = 1, \ldots, n-2 \, , \\
&\left(\frac{2D}{h^2} + \Sigma_a\right)\phi_0 - \frac{D}{h^2}\phi_1 = S_0 \qquad \text{for } i = 0 \, , \\
&\frac{-D}{h^2}\phi_{n-2} + \left(\frac{2D}{h^2} + \Sigma_a\right)\phi_{n-1} = S_{n-1} \qquad \text{for } i = n-1 \, .
\end{aligned}$$

Solution Methods

Both the FDM and FVM result in tridiagonal systems. Recall that formally solving these systems looks like

$$\vec{\phi} = \mathbf{A}^{-1}\vec{S} \, .$$

There are a few ways that we can solve these.

Directly

Tridiagonal systems can be solved directly using Gaussian elimination.

This is not a bad option in our case because **A** is diagonally dominant

———-

Recall: each diagonal element is greater than the sums of the absolute values of the off-diagonal elements in the same row):

$$a_{ii} \geq |a_{i,i-1}| + |a_{i,i+1}| \,.$$

———-

The general algorithm to solve a tridiagonal system is short and easy, it's called the **Thomas Algorithm**. We covered it back with general linear solution methods.

Would you like to go through that a little more slowly or move to the next thing?

Slowly: let's start by writing our system this way:

$$\begin{pmatrix} B_0 & -C_0 & 0 & 0 & \cdots & 0 \\ -A_1 & B_1 & -C_1 & 0 & \cdots & 0 \\ 0 & -A_2 & B_2 & -C_2 & & \vdots \\ \vdots & & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & -A_{n-2} & B_{n-2} & -C_{n-2} \\ 0 & \cdots & 0 & 0 & -A_{n-1} & B_{n-1} \end{pmatrix} \begin{pmatrix} \phi_0 \\ \phi_1 \\ \phi_2 \\ \vdots \\ \phi_{n-2} \\ \phi_{n-1} \end{pmatrix} = \begin{pmatrix} S_0 \\ S_1 \\ S_2 \\ \vdots \\ S_{n-2} \\ S_{n-1} \end{pmatrix} .$$

Recall that $\phi_n = 0$.

To develop the algorithm, let's look at a 3×3 system of equations.

$$\begin{aligned} B_0\phi_0 - C_0\phi_1 \qquad\quad &= S_0 \\ -A_1\phi_0 + B_1\phi_1 - C_1\phi_2 &= S_1 \\ -A_2\phi_1 + B_2\phi_2 &= S_2 \end{aligned}$$

Now the process:

1. Define

$$u_0 = B_0 \qquad v_0 = S_0 \,,$$

and write the first equation as

$$u_0\phi_0 - C_0\phi_1 = v_0 \ .$$

2. multiply by A_1/u_0

$$A_1\phi_0 - \frac{A_1C_0}{u_0}\phi_1 = \frac{A_1v_0}{u_0} \ .$$

Now add this to the second equation (where the ϕ_0 term subtracts out)

$$\left(B_1 - \frac{A_1C_0}{u_0}\right)\phi_1 - C_1\phi_2 = S_1 + \frac{A_1v_0}{u_0} \ ,$$

and define

$$u_1 = B_1 - \frac{A_1C_0}{u_0} \ , \quad v_1 = S_1 + \frac{A_1v_0}{u_0}$$

to re-write as

$$u_1\phi_1 - C_1\phi_2 = v_1 \ .$$

3. Guess what's next? Multiply this equation by A_2/u_1:

$$A_2\phi_1 - \frac{A_2C_1}{u_1}\phi_2 = \frac{A_2v_1}{u_1} \ ,$$

add to the third equation

$$\left(B_2 - \frac{A_2C_1}{u_1}\right)\phi_2 = S_2 + \frac{A_2v_1}{u_1} \ ,$$

and define

$$u_2 = B_2 - \frac{A_2C_1}{u_1} \ , \quad v_2 = S_2 + \frac{A_2v_1}{u_1}$$

to re-write as

$$u_2\phi_2 = v_2 \ .$$

All together we now have an **upper triangular system**:

$$\begin{aligned} u_0\phi_0 - C_0\phi_1 \qquad\quad &= v_0 \\ u_1\phi_1 - C_1\phi_2 &= v_1 \\ u_2\phi_2 &= v_2 \end{aligned}$$

that we can solve with **backward substitution**.

Here's the compact form of the (Thomas) algorithm:

1. $$u_0 = B_0 \qquad v_0 = S_0$$

2. for $i = 1, \ldots, n-1$
$$u_i = B_i - \frac{A_i C_{i-1}}{u_{i-1}} \qquad v_i = S_i + \frac{A_i v_{i-1}}{u_{i-1}}$$

3. Backward sub starting with $i = n-1$
$$\phi_{n-1} = \frac{v_{n-1}}{u_{n-1}}$$

4. Then for $i = n-2, \ldots, 1$
$$\phi_i = \frac{1}{u_i}\big(v_i + C_i \phi_{i+1}\big)$$

That was pretty easy, but we typically try to *avoid direct inversion of matrices* because it might be expensive in time and/or memory, be sensitive to round-off error, exhibit instability, etc.

We often use an iterative method instead.

Iterative Methods

Recall: produce a sequence of vectors, $\vec{\phi}^{(1)}, \vec{\phi}^{(2)}, \ldots$ based on the prescription

$$\vec{\phi}^{(k+1)} = F(\vec{\phi}^{(k)}, \vec{S}) , \quad \text{where } \lim_{k\to\infty} \vec{\phi}^{(k)} = \vec{\phi}\,.$$

$$\begin{aligned}
(\mathbf{A}+\mathbf{B})\vec{\phi} &= \mathbf{B}\vec{x} + \vec{S} \\
\mathbf{C}\vec{\phi} &= \mathbf{B}\vec{\phi} + \vec{S} \qquad \text{where } \mathbf{C} = \mathbf{A}+\mathbf{B} \\
\vec{\phi} &= \mathbf{C}^{-1}\mathbf{B}\vec{\phi} + \mathbf{C}^{-1}\vec{S} \\
\vec{\phi} &= \mathbf{P}\vec{\phi} + \tilde{\vec{S}} \qquad \text{assuming regular } \mathbf{C}
\end{aligned}$$

And the **fixed-point** iterative process is:

$$\vec{\phi}^{(0)} = \text{ arbitrary },$$
$$\vec{\phi}^{(k+1)} = \mathbf{P}\vec{\phi}^{(k)} + \tilde{\vec{S}} .$$

How we split $\mathbf{A}$ determines what method we're doing.

Jacobi

Let $\mathbf{D} = diag(\mathbf{A})$, then

$$\mathbf{D}\vec{\phi}^{k+1} = (\mathbf{D} - \mathbf{A})\vec{\phi}^{(k)} + \vec{S}$$
$$\vec{\phi}^{k+1} = \mathbf{D}^{-1}(\mathbf{D} - \mathbf{A})\vec{\phi}^{(k)} + \mathbf{D}^{-1}\vec{S}$$

In our original syntax, $\mathbf{P}_J = \mathbf{I} - \mathbf{D}^{-1}\mathbf{A}$ and $\tilde{\vec{b}} = \mathbf{D}^{-1}\vec{S}$.

The algorithm for this method is, for $i = 1, \ldots, n$:

$$\phi_i^{(k+1)} = \frac{1}{a_{ii}}(b_i - \sum_{j=1}^{i-1} a_{ij}\phi_j^{(k)} - \sum_{j=i+1}^{n} a_{ij}\phi_j^{(k)}) .$$

We can apply Gauss Seidel and SOR in exactly the same way.