Chapter 5

Methods for ordinary differential equations

5.1 Initial-value problems

Initial-value problems (IVP) are those for which the solution is entirely known at some time, say $t = 0$, and the question is to solve the ODE

$$y'(t) = f(t, y(t)), \qquad y(0) = y_0,$$

for other times, say $t > 0$. We will consider a scalar y, but considering systems of ODE is a straightforward extension for what we do in this chapter. We'll treat both theoretical questions of existence and uniqueness, as well as practical questions concerning numerical solvers.

We speak of t as being time, because that's usually the physical context in which IVP arise, but it could very well be a space variable.

Does a solution exist, is it unique, and does it tend to infinity (blow up) in finite time? These questions are not merely pedantic. As we now show with two examples, things can go wrong very quickly if we posit the wrong ODE.

Example 12. *Consider*

$$y' = \sqrt{y}, \qquad y(0) = 0.$$

By separation of variables, we find

$$\int \frac{dy}{\sqrt{y}} = \int dt \qquad \Rightarrow \qquad y(t) = \frac{(t+C)^2}{4}.$$

Imposing the initial condition yields $C = 0$, hence $y(t) = t^2/4$. However, $y(t) = 0$ is clearly another solution, so we have non-uniqueness. In fact, there is an infinite number of solutions, corresponding to $y(t) = 0$ for $0 \leq t \leq t^*$ for some t^*, which then takes off along the parabola the parabola $y(t) = (t-t^*)^2/4$ for times $t \geq t^*$.

Example 13. *Consider*

$$y' = y^2, \qquad y(0) = 1.$$

By separation of variables, we find

$$\int \frac{dy}{y^2} = \int dt \qquad \Rightarrow \qquad y(t) = \frac{-1}{t+C}.$$

The initial condition gives $C = -1$, hence $y(t) = 1/(1-t)$. It blows up at time $t = 1$, because $y(t)$ has a vertical asymptote. We say that the solution exists locally in any interval to the left of $t = 1$, but we don't have global existence.

Blowup and non-uniqueness are generally, although not always[1], unrealistic in applications. A theory of existence and uniqueness, including global existence (non blowup), is desired to guide us in formulating valid models for physical phenomena.

The basic result is the following.

Theorem 8. *(Picard) For given T, C, and y_0, consider the box B in (t, y) space, given by*

$$B = [0, T] \times [y_0 - C, y_0 + C].$$

Assume that

- *$f(t, y)$ is continuous over B;*
- *$|f(t, y)| \leq K$ when $(t, y) \in B$; (boundedness)*
- *$|f(t, u) - f(t, v)| \leq L|u - v|$ when $(t, u), (t, v) \in B$. (Lipschitz continuity).*

[1] In nonlinear optics for instance, laser pulses may "collapse". This situation is somewhat realistically modeled by an ODE that blows up, although not for times arbitrarily close to the singularity.

Assume furthermore that $C \geq \frac{K}{L}(e^{LT} - 1)$. *Then there exists a unique* $y \in C^1[0, T]$, *such that*

$$y'(t) = f(t, y(t)), \qquad y(0) = y_0,$$

and such that $|y(t) - y_0| \leq C$. *In short, the solution exists, is unique, and stays in the box for times* $0 \leq t \leq T$.

Proof. The technique is called Picard's iteration. See p.311 in Suli-Mayers. □

The ODE $y' = \sqrt{y}$ does not satisfy the assumptions of the theorem above because the square root is not a Lipschitz function. The ODE $y' = y^2$ does not satisfy the assumptions of the theorem above because y^2 is not bounded by a constant K.

5.2 Numerical methods for Initial-Value Problems

Here is an overview of some of the most popular numerical methods for solving ODEs. Let $t_n = nh$, and denote by y_n the approximation of $y(t_n)$.

1. Forward Euler (a.k.a. explicit Euler).

$$y_{n+1} = y_n + hf(t_n, y_n).$$

This formula comes from approximating the derivative y' at $t = t_n$ by a forward difference. It allows to march in time from the knowledge of y_n, to get y_{n+1}.

2. Backward Euler (a.k.a. implicit Euler).

$$y_{n+1} = y_n + hf(t_{n+1}, y_{n+1}).$$

This time we use a backward difference for approximating the derivative at $t = t_{n+1}$. The unknown y_{n+1} appears implicitly in this equation, hence the name implicit. It still needs to be solved for as a function of y_n, using (for instance) Newton's method. The strategy is still to march in time, but at every step there is a nonlinear equation to solve.

3. Trapezoidal (a.k.a. midpoint) rule (implicit).

$$y_{n+1} = y_n + \frac{h}{2}\left[f(t_n, y_n) + f(t_{n+1}, y_{n+1})\right].$$

   In this equation $\frac{y_{n+1}-y_n}{h}$ has the interpretation of a centered difference about the midpoint $t_{n+\frac{1}{2}} = \frac{t_n+t_{n+1}}{2}$, but since $f(t_{n+\frac{1}{2}}, y_{n+\frac{1}{2}})$ is not accessible ($y_{n+\frac{1}{2}}$ is not part of what we wish to solve for), we replace it by the average $\frac{1}{2}\left[f(t_n, y_n) + f(t_{n+1}, y_{n+1})\right]$. This gives a more balanced estimate of the slope $\frac{y_{n+1}-y_n}{h}$. It is an implicit method: y_{n+1} needs to be solved for.

4. Improved Euler, Runge-Kutta 2 (explicit).

$$\tilde{y}_{n+1} = y_n + hf(t_n, y_n),$$

$$y_{n+1} = y_n + \frac{h}{2}\left[f(t_n, y_n) + f(t_{n+1}, \tilde{y}_{n+1})\right].$$

   This is the simplest of "predictor-corrector" methods. It is like the midpoint rule, except that we use a guess $\tilde{y}_{n+1}$ for the unknown value of y_{n+1} in the right-hand side, and this guess comes from the explicit Euler method. Now y_{n+1} only appears in the left-hand side, so this is an explicit method.

5. Runge-Kutta 4 (explicit).

$$y_{n+1} = y_n + h[k_1 + 2k_2 + 2k_3 + k_4],$$

   where the slopes $k_1, \ldots, k_4$ are given in succession by

$$k_1 = f(t_n, y_n), \qquad k_2 = f(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_1),$$

$$k_3 = f(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_2), \qquad k_4 = f(t_n + h, y_n + hk_3).$$

6. There are also methods that involve not just the past value y_n, but a larger chunk of history y_{n-1}, y_{n-2},etc. These methods are called multistep. They are in general less flexible than the one-step methods described so far, in that they require a constant step h as we march in time.

Two features of a numerical method are important when choosing a numerical method:

- Is it *convergent*, i.e., does the computed solution tend to the true solution as $h \to 0$, and at which rate?
- Is it *stable*, i.e., if we solve with different initial conditions y_0 and $\tilde{y}_0$, are the computed solutions close in the sense that $|y_n - \tilde{y}_n| \le C|y_0 - \tilde{y}_0|$, with $n = O(1/h)$, and C independent of h?

5.2.1 Convergence

To understand convergence better in the case of one-step methods, let us write the numerical scheme as

$$y_{n+1} = \Psi(t_n, y_n, h),$$

and introduce the *local error*

$$e_{n+1}(h) = \Psi(t_n, y(t_n), h) - y(t_{n+1}),$$

as well as the *global error*

$$E_n(h) = y_n - y(t_n).$$

The expression of the local error can be explained as "trying the exact solution in the numerical scheme" — although the exact solution is unknown. It is a lot easier to approach the convergence question via local errors than global errors. It would not be practical, for instance, to "try an approximate solution in the exact ODE".

Convergence is the study of the global error. *Consistency* is the study of the local error. A numerical method is called consistent if the local error decays sufficiently fast as $h \to 0$ that there is hope that the global error would be small as well. The particular rate at which the local error decays is related to the notion of order of an ODE solver.

Definition 6. *(Consistency) Ψ is consistent if, for any $n \ge 0$,*

$$\lim_{h \to 0} \frac{e_n(h)}{h} = 0$$

Definition 7. *(Order)* Ψ *is of order* p *if* $e_n(h) = O(h^{p+1})$.

The basic convergence theorem for one-step solvers, that we will not prove, is that if the local error is $O(h^{p+1})$, then the global error is $O(h^p)$. This convergence result is only true as stated for one-step methods; for multi-step methods we would also need an assumption of stability (discussed below). Intuitively, the local errors compound over the $O(1/h)$ time steps necessary to reach a given fixed time t, hence the loss of one power of h. Of course the local errors don't exactly add up; but they do up to a multiplicative constant. It is the behavior of the global error that dictates the notion of order of the numerical scheme.

It is a good exercise to show, using elementary Taylor expansions, that the explicit and implicit Euler methods are of order 1, and that the midpoint rule and improved Euler methods are of order 2. It turns out that Runge-Kutta 4 is of order 4, but it is not much fun to prove that.

5.2.2 Stability

Consistency and convergence do not tell the whole story. They are helpful in the limit $h \to 0$, but don't say much about the behavior of a solver in the interesting regime when h is small, but not so small as to make the method computationally inefficient.

The single most important consideration in the regime of moderately small h is perhaps stability. The criterion stated above for stability ($|y_n - \tilde{y}_n| \leq C|y_0 - \tilde{y}_0|$) is too complex to handle as such. The important ideas already appear if we study the representative setting of linear stability. The linearization of the ODE $y' = f(t, y)$ about a point y_0 is obtained from writing the Taylor expansion

$$\frac{d}{dt}(y(t) - y_0) = f(t, y(t)) = f(t, y_0) + \frac{\partial f}{\partial y}(t, y_0)(y(t) - y_0) + o(|y(t) - y_0|).$$

The culprit for explosive (exponential) growth or decay is the linear term $\frac{\partial f}{\partial y}(t, y_0)(y(t) - y_0)$. (Indeed, if the other two terms are neglected, we can write the solution, locally, as an exponential.) For practical purposes it is sufficient to check stability for the linear equation $y' = \lambda y$, keeping in mind that λ is a number representative of the derivative $\frac{\partial f}{\partial y}(t, y_0)$ of f at y_0 in the y variable.

Definition 8. *(Linear stability[2]) Suppose $y' = \lambda y$ for some $\lambda \in \mathbb{C}$. Then the numerical method Ψ is linearly stable if $y_n \to 0$ as $n \to \infty$.*

Of course linear stability depends on the value of λ. Stability for the original equation $y' = \lambda y$ is guaranteed if $\mathrm{Re}(\lambda) < 0$ (because the solution is $y(0)e^{\lambda t}$), and the question is that of showing whether a numerical method Ψ is stable under the same condition or not.

If a numerical method is stable in the above sense for a certain range of values of λ, then it is possible to show that it will be stable for the ODE $y' = f(t, y)$ as long as $\frac{\partial f}{\partial y}$ is in that range of λ (and f is smooth enough). We won't prove this theorem here.

Let us consider a few examples

Example 14. *For the forward Euler method applied to $y' = \lambda y$, we get*

$$y_{n+1} = y_n + h\lambda y_n = (1 + h\lambda)y_n.$$

The iterates y_n tend to zero provided $|1 + h\lambda| < 1$, where the $| \cdot |$ denote the complex modulus. Write $h\lambda = x + iy$, so that the inequality becomes $(1 + x)^2 + y^2 < 1$. This is the equation of a disc in the complex $(h\lambda)$-plane, with center at -1, and radius 1. If $h\lambda$ sits inside this disk, the method is stable, and otherwise it isn't. We say that the forward Euler method is conditionally stable: typically, we require both Re$(\lambda) < 0$ and a small step size h in order to guarantee stability.

Example 15. *The backward Euler method applied to $y' = \lambda y$ gives*

$$y_{n+1} = y_n + h\lambda y_{n+1},$$

or in other words

$$y_{n+1} = \frac{y_n}{1 - h\lambda}.$$

The iterates y_n tend to zero provided $|1 - h\lambda| > 1$. In terms of $h\lambda = x + iy$, this becomes $(x - 1)^2 + y^2 > 1$. This condition is satisfied whenever $h\lambda$ is outside the disc in the complex $(h\lambda)$-plane with center at $+1$, and radius 1. In that case the method is stable, and otherwise it isn't. We say that the backward Euler method is unconditionally stable: the stability zone for the ODE (Re$(\lambda) < 0$) is always included in the stability zone of the numerical

[2] Sometimes called A-stability in some texts.

method, regardless of h. In fact the zone of stability for the backward Euler method is larger than the left complex half-plane, so there exist choices of (large) time steps for which the numerical method is stable although the ODE isn't.

Example 16. *The linearized stability analysis of the midpoint method gives*

$$y_{n+1} = y_n + h\lambda(\frac{y_n}{2} + \frac{y_{n+1}}{2}),$$

hence

$$y_{n+1} = \left(\frac{1 + h\lambda/2}{1 - h\lambda/2}\right) y_n.$$

The method is stable provided

$$|\frac{1 + h\lambda/2}{1 - h\lambda/2}| < 1.$$

In terms of $h\lambda = x + iy$, we can simplify to get

$$(1 + \frac{x}{2})^2 + (\frac{y}{2})^2 < (1 - \frac{x}{2})^2 + (\frac{y}{2})^2,$$

which is true if and only if $x < 0$. So the stability region is Re$(h\lambda) < 0$, the same as that of the ODE. As a result, the method is unconditionally stable.

It is a general rule that explicit methods have conditional stability (stability only happens when the time step is small enough, if it does at all), whereas implicit methods are unconditionally stable.

The stability regions for the Runge-Kutta methods are plotted on page 351 of Suli-Mayers.

Stability can also be studied for systems of ODEs $y'(t) = f(t, y(t))$ where both y and f are vectors. The interesting object is now the Jacobian matrix

$$A = \nabla_y f(t, y_0).$$

(it is a matrix because the gradient of a vector function is a "vector of vectors", i.e., a matrix.) The linearized problem is now

$$y'(t) = Ay(t),$$

which has the solution $y(0)e^{tA}$, where e^{tA} is a matrix exponential. Stability will happen when the eigenvalues λ of A *all* obey $\text{Re}(\lambda) < 0$. So the story is the same as in the scalar case, except that λ is now *any* eigenvalue of the Jacobian matrix. So we need to make sure that $h\lambda$ is in the stability zone of the ODE solver in the complex plane, for each λ an eigenvalue of the Jacobian matrix.

Problems for which λ has a very large, negative real part are called *stiff*. Physically they are very stable, but they pose numerical problems for explicit methods since the region of stability does not extend very far along the negative real axis. Implicit methods are hands down the best for stiff problems.

5.2.3 Miscellaneous

Another interesting class of methods, and a way to naturally design high-order methods, is *deferred correction*. Assume that time is split into a uniform grid $t_j = jh$, and that some low-order method has given us the samples $y_1, \ldots, y_{n+1}$ for some (small) m. Denote by $\pi_n(t)$ the n-th order interpolation polynomial passing through the (t_j, y_j). Consider the error ("defect")

$$\delta(t) = y(t) - \pi_n(t).$$

It obeys the equation

$$\delta'(t) = f(t, y(t)) - \pi_n'(t), \qquad \delta(0) = 0.$$

We do not have access to $y(t)$ in the argument of f, but we can replace it by our best guess $\pi_n(t) + \delta(t)$. This way we can compute an approximate defect

$$\tilde{\delta}'(t) = f(t, \pi_n(t) + \tilde{\delta}(t)) - \pi_n'(t), \qquad \tilde{\delta}(0) = 0.$$

using the same (or another) low-order method. Once $\tilde{\delta}(t)$ is computed, add it back to the interpolant to get

$$\tilde{y}(t) = \pi_n(t) + \tilde{\delta}(t).$$

This procedure can be repeated a few times to get the desired accuracy. (This version of deferred correction is actually quite recent and due to Dutt, Greengard, and Rokhlin, 2000).

5.3 Boundary-value problems

Boundary-value problems (BVP) are ODE where some feature of the solution is specified at two ends of an interval. The number of initial or boundary conditions matches the order of the highest derivative in the equation, hence such ODE are generally second-order scalar equations. The simplest examples are

$$-u''(x) = f(x), \qquad x \in [0,1], \qquad u(0) = a, u(1) = b \qquad \text{(Dirichlet)}$$

$$-u''(x) = f(x), \qquad x \in [0,1], \qquad u'(0) = a, u'(1) = b \qquad \text{(Neumann)}$$

$$-u''(x) = f(x), \qquad x \in [0,1], \qquad u(0) = u(1) \qquad \text{(periodic)}$$

We don't really mean to study the equation $-u'' = f$ for its own sake (you can find the general solution by integrating f twice and fixing the two constants from the boundary conditions), but its study serves two purposes:

- It is the simplest ODE that models problems in continuous elasticity: here u is the displacement of a vertical elastic bar, or rod, that sags under its own weight. The right-hand-side f is the gravitational force as a function of x, and u' is the "elongation", or strain of the bar. A condition where u is specified means that the bar is fixed at that end, while the condition $u' = 0$ would mean that that end is free. The condition $u(0) = u(1)$ means that it's an elastic band. By solving the ODE we are finding the displacement u generated by the force f.

- It is the simplest boundary-value problem to treat numerically, and contains many of the important features of such problems. It needs to be understood before moving on to any other example.

Alternative problems of this kind are

$$-u''(x) + \alpha(x)u(x) = f(x), \qquad u(0) = a, u(1) = b,$$

for instance, the solution of which does not generally obey an explicit formula.

A first intuitive method for solving BVP is the *shooting method*. Consider again $u''(x) + \alpha(x)u(x) = f(x), \quad u(0) = a, u(1) = b$. We cannot in general march in time from 0 to 1 (the variable x is more often a spatial variable), but we can guess what the derivative should have been for the solution to end up near b at $x = 1$. Introduce a parameter s, and write

$$-u''(x; s) + \alpha(x)u(x; s) = f(x), \qquad u(0; t) = a, u'(0; s) = s.$$

Of course in general we won't reach b at $x = 1$, but we can refine our estimate of s so that we get closer to it. The problem becomes that of solving for s in $u(1; s) = b$, where $u(1; s)$ is defined implicitly from the ODE. This equation that can be viewed as a root-finding or fixed-point problem and solved using any of the methods we've seen previously. The secant method only requires evaluations of $u(1; s)$, which involves solving the ODE. To set up Newton's method, we need the derivative $\frac{\partial u}{\partial t}(1; s)$ of the solution as a function of its parameter s. Exercise: find the value of this derivative by differentiating the ODE in the t parameter, and obtain an ODE for $\frac{pdu}{\partial t}(1; s)$ itself.

The shooting method is easy and intuitive (and there isn't much more to say about it), but it has the disadvantage of being restricted to ODE. In contrast, we'll now investigate finite difference methods, which can also be applied to partial differential equations (PDE).

Let's start by considering the problem $-u'' = f$, $u(0) = u(1) = 0$. Consider the grid $x_j = jh$, $j = 0, \ldots, N$, $h = 1/N$. This grid has $N + 1$ points. The usual 3-point stencil for the centered second difference gives rise to

$$-\frac{U_{j+1} - 2U_j + U_{j-1}}{h^2} = f(x_j). \tag{5.1}$$

In this context, capital U denotes the numerical solution, while lowercase u denotes the exact solution. For the boundary conditions, we simply set $U_0 = U_N = 0$. The rest of the U_j for $j = 1, \ldots N - 1$ are unknowns and can be solved for from the linear system $KU = F$ generated by (5.1). The resulting matrix K (of size $N - 1$ by $N - 1$) is

$$K = \frac{1}{h^2}\begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{pmatrix}$$

The zero elements are not shown. The right-hand side is here $F_j = f(x_j)$. In Matlab, one then solves U as $K \setminus F$. Had the boundary conditions been

$U_0 = a$ and $U_N = b$ instead, it is a good exercise to check that this does not change K, but that the right-hand side gets modified as

$$F = \begin{pmatrix} f(x_1) + \frac{a}{h^2} \\ f(x_2) \\ \vdots \\ f(x_{N-2}) \\ f(x_{N-1}) + \frac{b}{h^2} \end{pmatrix}.$$

Of course, the matrix K should be invertible for this strategy to make sense. We will see below that this is the case. It is also important that K^{-1} be bounded for convergence of the numerical scheme.

Definition 9. *The local truncation error (LTE) of a numerical scheme $KU = F$, is the error made when evaluating the numerical scheme with the exact solution $u(x_j)$ in place of the numerical solution U_j. It is the quantity τ in*

$$Ku = F + \tau.$$

The local truncation is directly obtained from the truncation error of the finite-difference scheme, for instance

$$-\frac{u(x_{j+1}) - 2u(x_j) + u(x_{j-1})}{h^2} = f(x_j) + O(h^2),$$

so the LTE is $O(h^2)$.

Definition 10. *The (actual) error of a numerical scheme $KU = F$, is the vector of differences $e_j = u(x_j) - U_j$.*

In order to obtain the error e from the LTE, one writes

$$Ku = F + \tau, \qquad KU = F,$$

and subtract those two equations. This gives $K(u - U) = (F - F) + \tau$, or $Ke = \tau$. If K is invertible, this gives

$$e = K^{-1}\tau.$$

The next few sections introduce the tools needed to control how large e can get from the knowledge that it is τ "magnified" by K^{-1}.

5.3.1 Matrix norms and eigenvalues

This section is mostly a linear algebra refresher.

Definition 11. *The spectral norm, or 2-norm, of (any rectangular, real) matrix A is*

$$\|A\|_2 = \max_{x \neq 0} \frac{\|Ax\|_2}{\|x\|_2},$$

where the vector 2-norm is $\|x\|_2 = \sqrt{\sum_i x_i^2}$. In other words, the matrix 2-norm is the maximum stretch factor for the length of a vector after applying the matrix to it.

Remark that, by definition,

$$\|Ax\|_2 \leq \|A\|_2 \|x\|_2,$$

for any vector x, so the matrix 2-norm is a very useful tool to write all sorts of inequalities involving matrices.

We can now characterize the 2-norm of a *symmetric* matrix as a function of its eigenvalues. The eigenvalue decomposition of a matrix A is, in matrix form, the equation $AV = V\Lambda$, where V contains the eigenvectors as columns, and Λ contains the eigenvalues of the diagonal (and zero elsewhere.) For those (non-defective) matrices for which there is a full count of eigenvectors, we also have

$$A = V \Lambda V^{-1}.$$

Symmetric matrices have a full set of orthogonal eigenvectors, so we can further write $V^T V = I$, $V V^T = I$ (i.e. V is unitary, a.k.a. a rotation), hence

$$A = V \Lambda V^T.$$

In terms of vectors, this becomes

$$A = \sum_i v_i \lambda_i v_i^T.$$

Note that λ_i are automatically real when $A = A^T$.

Theorem 9. *Let $A = A^T$. Then*

$$\|A\|_2 = \max_i |\lambda_i(A)|.$$

Proof. First observe that the vector 2-norm is invariant under rotations: if $V^TV = I$ and $VV^T = I$, then $\|Vx\|_2 = \|x\|_2$. This follows from the definition:

$$\|Vx\|_2^2 = (Vx)^T Vx = x^T V^T Vx = x^T x = \|x\|_2^2.$$

Now fix x and consider now the ratio

$$\begin{aligned}\frac{\|Ax\|_2^2}{\|x\|_2^2} &= \frac{\|V\Lambda V^T x\|_2^2}{\|x\|_2^2}\\ &= \frac{\|\Lambda V^T x\|_2^2}{\|x\|_2^2} \qquad \text{(unitary invariance)}\\ &= \frac{\|\Lambda y\|_2^2}{\|Vy\|_2^2} \qquad \text{(change variables)}\\ &= \frac{\|\Lambda y\|_2^2}{\|y\|_2^2} \qquad \text{(unitary invariance again)}\\ &= \frac{\sum_i \lambda_i^2 y_i^2}{\sum_i y_i^2}.\end{aligned}$$

This quantity is maximized when y is concentrated to have nonzero component where λ_i is the largest (in absolute value): $y_j = 1$ when $|\lambda_j| = \max_n |\lambda_n|$, and zero otherwise. In that case,

$$\frac{\|Ax\|_2^2}{\|x\|_2^2} = \max_n \lambda_n^2,$$

the desired conclusion. □

Note: if the matrix is not symmetric, its 2-norm is on general not its largest eigenvalue (in absolute value). Instead, the 2-norm is the largest singular value (not material for this class, but very important concept.)

One very useful property of eigenvalues is that if $A = V\Lambda V^T$ is invertible, then

$$A^{-1} = V\Lambda^{-1}V^T$$

(which can be checked directly), and more generally

$$f(A) = Vf(\Lambda)V^T,$$

where the function f is applied componentwise to the λ_i. The eigenvectors of the function of a matrix are unchanged, and the eigenvalues are the function of the original eigenvalues.

If a matrix A is not invertible, things usually go wrong when trying to solve the linear system $Ax = b$.

Definition 12. *The nullspace of (any rectangular, real) matrix A is the space Null(A) of all vectors v such that $Av = 0$. In other words, it is the eigenspace corresponding to the eigenvalue zero.*

Null(A) always contains the zero vector. The following conditions are equivalent to characterize singular matrices:

- A is singular (non-invertible);
- Null(A) contains some nonzero vector;
- 0 is an eigenvalue of A;
- $\det(A) = 0$;
- The rows/columns are linearly dependent;
- (Zero is a pivot in the row echelon reduction of A.)

We now present a version of the inversion theorem for *symmetric* matrices. If the matrix is not symmetric, the statement looks quite different.

Theorem 10. *Let $A = A^T$. Consider the system $Ax = b$.*

- *If A is invertible, then the solution is unique and $x = A^{-1}b$.*
- *If Null(A) = span($v_1, \ldots v_m$) $\neq 0$, then*
  - *If b has a component along any of the v_j (i.e., $v_j^T b \neq 0$ for some j), then the system has no solution.*
  - *If all $v_j^T b = 0$, $j = 1, \ldots, m$, then there exists an infinite number of solution to the system. If x_0 is a solution, then so is $x_0 + \sum_{j=1}^m c_j v_j$ for arbitrary coefficients c_j.*

In terms of eigenvectors v_j, if the matrix is invertible, the solution of $Ax = b$ is

$$x = \sum_{j=1}^{N} v_j \frac{1}{\lambda_j} v_j^T b.$$

If A is not invertible, but $v_j^T b = 0$ for all the eigenvectors v_j, $j = 1, \ldots, m$ corresponding to the zero eigenvalue (as in the theorem above), then we still have

$$x = \sum_{j=m+1}^{N} v_j \frac{1}{\lambda_j} v_j^T b + \sum_{j=1}^{m} c_j v_j,$$

where the first sum only runs from $m+1$ to N, and the coefficients c_j are arbitrary. (Apply the matrix A to this equation to see that they disappear.)

If $v_j^T b \neq 0$, then the operation $\frac{1}{\lambda_j} v_j^T b$ would result in an infinity when $\lambda_j = 0$ — a crude reason to see why the system has no solution in that case.

5.3.2 Properties of the matrix K

We are now equipped to study the matrix K and its inverse. Recall that we are interested in controlling $e = K^{-1}\tau$ to get the error from the LTE. From the results in the previous section, we know that

$$\|e\|_2 = \|K^{-1}\tau\|_2 \leq \|K^{-1}\|_2 \|\tau\|_2.$$

Since K is a symmetric matrix,

$$\|K^{-1}\|_2 = \max_j |\lambda_j(K^{-1})| = \frac{1}{\min_j |\lambda_j(K)|}.$$

So it remains for us to understand the eigenvalues of K, and specifically to show that the minimum eigenvalue (in absolute value) does not get too small. What we mean by this is that there should exist a number $c > 0$ such that

$$c < \min_j |\lambda_j(K)|,$$

independently of the grid spacing h (recall that the size of K and its entries depend on h.) Only in that case will we be able to conclude that $\|e\|$ is of the same order of magnitude as $\|\tau\|$.

Note in passing that if $\tau = O(h^2)$ then $\|\tau\|_2 = \sqrt{\sum_i \tau_i^2}$ will be $O(\sqrt{\frac{1}{h}h^4}) = O(h^{3/2})$, which is not very appealing. Instead, it is common to modify the vector 2-norm as

$$\|\tau\|_{2,h} = \sqrt{h \sum_i \tau_i^2},$$

so as to restore $\|\tau\|_{2,h} = O(h^2)$ (note how h times the sum resembles an integral quadrature.) In that case, we also expect $\|\tau\|_{2,h} = O(h^2)$. The reasoning with matrix 2-norms does not change one bit form this different choice of normalization.

Let us now study the eigenvalues and eigenvectors of K. The best way to guess them is to notice that $KU = F$ is a discretization of $-u'' = f$ with Dirichlet boundary conditions. The eigenvalue problem

$$-v'' = \lambda v, \qquad v(0) = v(1) = 0,$$

has a solution in terms of sines: for each $n \geq 1$, we have the pair

$$v_n(x) = \sin(n\pi x), \qquad \lambda = n^2\pi^2.$$

This analogy is very fruitful: the eigenvectors of K are precisely the v_n sampled at $x_j = jh$,

$$v_j^{(n)} = \sin(n\pi jh), \qquad j = 1, \ldots, N-1, \qquad n = 1, \ldots, N-1.$$

(here n is the label index, and j is the component index.) It is straightforward and a little tedious to check from trigonometric formulas that $v^{(n)}$ defined by this formula are indeed eigenvectors, with eigenvalues

$$\lambda_n = \frac{4}{h^2}\sin^2(\frac{\pi n h}{2}), \qquad n = 1, \ldots, N-1.$$

A Taylor expansion for small h shows that the minimum eigenvalue is $\lambda_1 = \pi^2 + O(h^2)$, and this $O(h^2)$ is positive, so that $\lambda_1 \geq \pi^2$.

Note in passing that since K is symmetric, the eigenvectors are automatically orthogonal, hence there is a way to normalize them so that $V^TV = I$ and $VV^T = I$. Applying the matrix V^T is called the discrete sine transform (DST), and applying the matrix V is called the inverse discrete sine transform (IDST).

The formula for the eigenvalues has two important consequences:

- The matrix K is invertible, now that it is clear that all its eigenvalues are positive. So setting up the discrete problem as $KU = F$ makes sense.

- Returning to the error bound, we can now conclude that $\|e\|_{2,h} \leq \frac{1}{\pi^2}\|\tau\|_{2,h} = O(h^2)$, establishing once and for all that the finite-difference method is second-order accurate for $-u'' = f$ with Dirichlet boundary conditions.

The reasoning of first establishing consistency (small LTE) and showing a stability result transferring at the level of the actual error is very common in numerical analysis. *Consistency + Stability = Convergence.*

5.3.3 Other boundary conditions and other equations

The boundary conditions (BC) play a crucial role in the description of the problem: if the BC changes, the solution changes completely, and so can the LTE and the error. For instance, let's return to the Neumann problem

$$-u''(x) = f(x), \qquad x \in [0,1], \qquad u'(0) = a, u'(1) = b \qquad \text{(Neumann)}$$

The discretization of the interior equations is the same as previously, but at least one extra equation needs to be added for the BC. For instance, at zero, we may write a forward difference

$$\frac{U_1 - U_0}{h} = a.$$

and similarly, a backward difference at $x = 1$. Each BC adds one row and one column to the original matrix K, and results in a different system of equations. The choice above leads to a first-order LTE, only $O(h)$, even though the LTE for the interior equations is $O(h^2)$. This is enough to spoil the error itself: we don't have $\|e\|_{2,h} = O(h^2)$ in general, as a result of the low-accuracy boundary conditions.

A more accurate way to discretize a Neumann boundary condition is to introduce a "ghost" node U_{-1}, and write

$$\frac{U_1 - U_{-1}}{2h} = a.$$

This new unknown is linked to the others by writing the equation one more time at $j = 0$,

$$\frac{= U_1 + 2U_0 - U_{-1}}{h^2} = f(x_0).$$

(Previously, we had only evaluated the equation at $j = 1, \ldots, N-1$.) This results in *two* additional rows and columns per BC. The same treatment should be applied at $x = 1$.

Note that these additional "boundary" rows can be scaled so that the resulting matrix looks symmetric, or at least more balanced. For instance, by rescaling the $\frac{U_1-U_{-1}}{2h} = a$ by $2/h$ (including the right-hand side) we get the resulting matrix

$$T = \begin{pmatrix} -1 & 0 & 1 & & \\ -1 & 2 & -1 & & \\ & -1 & 2 & -1 & \\ & & \ddots & \ddots & \ddots \end{pmatrix}$$

The eigenvalues depend on this choice of normalization!

Notice that, unlike the Dirichlet problem, the Neumann problem has a nullspace. The vector identically equal to 1 is in the nullspace of either of the 2 discretizations we have presented. As a result, there exists a solution only if the admissibility condition $1^T f = \sum_i f(x_i) = 0$ is satisfied (see a theorem in the previous section). This is also a feature of the non-discretized BVP: it has a solution if and only if $\int_0^1 f(x)\,dx = 0$.

The periodic problem is also very interesting:

$$-u''(x) = f(x), \qquad x \in [0,1], \qquad u(0) = u(1) \qquad \text{(periodic)}$$

The boundary condition $U_0 = U_N$ modifies K by adding two elements in the bottom-left and top-right:

$$C = \frac{1}{h^2}\begin{pmatrix} 2 & -1 & & & -1 \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ -1 & & & -1 & 2 \end{pmatrix}$$

This new matrix C is circulant and singular. We can check that the vector 1 is in its nullspace, so the BVP has a solution if and only if $1^T f = \sum_i f(x_i) = 0$.

The eigenvectors of C are the Fourier components

$$v_j^{(n)} = w^{jn} = e^{2\pi i jn/N}, \qquad w = e^{2\pi i/N}.$$

Since C is symmetric, these $v^{(n)}$ are orthogonal (and orthonormal when divided by $\sqrt{N}$). To deal with vectors and matrices that have complex entries, don't forget that the transposes come with a complex conjugate, so the dot product is

$$x^* y = \overline{x}^T y = \sum_i \overline{x}_i y_i.$$

The norm is $\|x\|_2 = \sqrt{\sum_i |x_i|^2}$. The orthogonality relations for the matrix of eigenvectors are now $V^*V = I$ and $VV^* = I$, where $*$ is transpose conjugate.