Operator splitting

Shev MacNamara and Gilbert Strang

Abstract Operator splitting is a numerical method of computing the solution to a differential equation. The splitting method separates the original equation into two parts over a time step, separately computes the solution to each part, and then combines the two separate solutions to form a solution to the original equation. A canonical example is splitting of diffusion terms and convection terms in a convection-diffusion partial differential equation. Related applications of splitting for reaction-diffusion partial differential equations in chemistry and in biology are emphasized here. The splitting idea generalizes in a natural way to equations with more than two operators. In all cases, the computational advantage is that it is faster to compute the solution of the split terms separately, than to compute the solution directly when they are treated together. However, this comes at the cost of an error introduced by the splitting, so strategies have been devised to control this error. This chapter introduces splitting methods and surveys recent developments in the area. An interesting perspective on absorbing boundary conditions in wave equations comes via Toeplitz-plus-Hankel splitting. One recent development, balanced splitting, deserves and receives special mention: it is a new splitting method that correctly captures steady state behavior.

1 Introduction

It has been said that there are only ten big ideas in numerical analysis; all the rest are merely variations on those themes. One example of those big ideas is multi-scale computational approaches. A multi-scale motif reappears in numerous places

Shev MacNamara
School of Mathematics, University of New South Wales, Sydney NSW 2052, Australia. e-mail: s.macnamara@unsw.edu.au

Gilbert Strang
Department of Mathematics, MIT, Cambridge, MA, 02139. e-mail: gs@math.mit.edu

including: multigrid for solving linear systems [51], wavelets for image processing [11], and in Multi-level Monte Carlo for the solution of stochastic differential equations [20]. Another of those big ideas could surely be *splitting* [42, 4, 57]: start with a complicated problem, split it into simpler constituent parts that can each be solved separately, and combine those separate solutions in a controlled way to solve the original overall problem. Often we solve the separate parts sequentially. The output of the first subproblem is the input to the next subproblem (within the time step).

Like all great ideas, splitting is a theme that continues to resurface in many places. Splitting principles have taken a number of generic forms:

- Split linear from nonlinear.
- Split x-direction from y-direction (dimensional splitting).
- Split terms corresponding to different physical processes. For example, split convection from diffusion in ODEs or in PDEs.
- Split a large domain into smaller pieces. For example, domain decomposition helps to solve large PDEs in parallel.
- Split objective functions in optimization.
- Split resolvents when solving linear systems: Instead of working directly with $(\lambda \boldsymbol{I} - (\boldsymbol{A} + \boldsymbol{B}))^{-1}$, we iterate between working separately with each of $(\lambda \boldsymbol{I} - \boldsymbol{A})^{-1}$ and $(\lambda \boldsymbol{I} - \boldsymbol{B})^{-1}$.

Not surprisingly, a principle as fundamental as splitting finds applications in many areas. Here is a non-exhaustive list:

- A recent application of splitting is to low-rank approximation [36].
- Balanced splitting has been developed to preserve the steady state [53].
- Splitting of reaction terms from diffusion terms in reaction-diffusion PDEs is a common application of splitting in biology. Now splitting is also finding applications in stochastic, particle-based methods, such as for master equations [19, 32, 38, 37, 18, 26, 25], including analysis of sample path approaches [16].
- Splitting stochastic differential equations, by applying the component operators in a random sequence determined by coin flipping is, on average, accurate. This has applications in finance [45]. Though in a different sense, splitting also finds application in Monte Carlo estimation of expectations [2].
- Maxwell's equations for electromagnetic waves can be solved on staggered grids via Yee's method, which is closely related to splitting [55, 34].
- Motivated in part by the need for accurate oil reservoir simulations, Alternating Direction Implicit methods and Douglas-Rachford splittings have by now found wide applications [62, 47, 14].
- Split-Bregman methods are a success for compressed sensing and for image processing [21].
- Navier-Stokes equations in fluid mechanics are often approximated numerically by splitting the equations into three parts: (i) a nonlinear convection term, $u \cdot (\nabla u)$, is treated explicitly, (ii) diffusion, Δu, is treated implicitly, and (iii) continuity is imposed via Poisson's equation, $\mathrm{div} u = 0$. Chorin's splitting method is a well-known example of this approach [7, 55].

- Split the problem of finding a point in the intersection of two or more sets into alternating projections onto the individual sets [4].

This chapter places emphasis on applications to partial differential equations (PDEs) involving reaction, diffusion and convection. *Balanced splitting*, which has found application in models of combustion, receives special attention. Computer simulation of *combustion* is important to understand how efficiently or how cleanly fuels burn. It is common to use *operator splitting* to solve the model equations. However, in practice this was observed to lead to an unacceptable error at steady state. The new method of balanced splitting was developed to correct this [53]. This balanced method might be more widely applicable because operator splitting is used in many areas. Often the steady state is important. In reaction-diffusion models, in biology for example, it is very common to split reaction terms from diffusion terms, and the steady state is almost always of interest. As we will see in the next section, the most obvious splitting scheme is only first order accurate but a symmetrized version achieves second order accuracy. Do such schemes yield the correct steady state? The answer is no, not usually. Balanced splitting corrects this.

Outline: The rest of this chapter is organized as follows. We begin with the simplest possible example of splitting. First order accurate and second order accurate splitting methods come naturally. Higher order splitting methods, and reasons why they are not always adopted, are then discussed. Next, we observe that splitting does not capture the correct *steady state*. This motivates the introduction of balanced splitting: a new splitting method that does preserve the steady state. All these ideas are illustrated by examples drawn from reaction-diffusion PDEs such as arise in mathematical biology, and from convection-diffusion-reaction PDEs such as in models of combustion. We aim especially to bring out some recent developments in these areas [59, 53]. Finally, we investigate a very special Toeplitz-plus-Hankel splitting, that sheds light on the reflections at the boundary in a wave equation.

2 Splitting for ordinary differential equations

The best example to start with is the linear ordinary differential equation (ODE)

$$\frac{\mathrm{d}u}{\mathrm{d}t} = (\boldsymbol{A}+\boldsymbol{B})u. \tag{1}$$

The solution is well known to students of undergraduate differential equation courses [58]:

$$u(h) = e^{h(\boldsymbol{A}+\boldsymbol{B})}u(0),$$

at time h. We are interested in splitting methods that will compute that solution for us, at least approximately. If we could simply directly compute $e^{h(\boldsymbol{A}+\boldsymbol{B})}$, then we would have solved our ODE (1), and we would have no need for a splitting approximation. However, in applications it often happens that $e^{h(\boldsymbol{A}+\boldsymbol{B})}$ is relatively difficult to compute directly, whilst there are readily available methods to compute each of

$e^{h\boldsymbol{A}}$ and $e^{h\boldsymbol{B}}$ separately. For example, (1) may arise as a method-of-lines approximation to a PDE in which $\boldsymbol{A}$ is a finite difference approximation to diffusion, and $\boldsymbol{B}$ is a finite difference approximation to convection. (This example of an convection-diffusion PDE, together with explicit matrices, is coming next.) In that case it is natural to make the approximation

First order splitting $$e^{h(\boldsymbol{A}+\boldsymbol{B})} \approx e^{h\boldsymbol{A}} e^{h\boldsymbol{B}}. \quad (2)$$

We call this approximate method of computing the solution *splitting*. This chapter stays with examples where $\boldsymbol{A}$ and $\boldsymbol{B}$ are matrices, although the same ideas apply in more general settings where $\boldsymbol{A}$ and $\boldsymbol{B}$ are operators; hence the common terminology *operator splitting*.

To begin thinking about a splitting method we need to see the matrix that appears in the ODE as the *sum of two matrices*. That sum is immediately obvious in (1) by the way it was deliberately written. However, had we instead been given the ODE $\mathrm{d}u/\mathrm{d}t = \boldsymbol{M}u$, then before we could apply a splitting method, we would first need to identify $\boldsymbol{A}$ and $\boldsymbol{B}$ that add up to $\boldsymbol{M}$. Given $\boldsymbol{M}$, identifying a good choice for $\boldsymbol{A}$ and thus for $\boldsymbol{B} = \boldsymbol{M} - \boldsymbol{A}$ is not trivial, and the choice is critical for good splitting approximations.

When the matrices *commute*, the approximation is exact. [1] That is, if the commutator $[\boldsymbol{A}, \boldsymbol{B}] \equiv \boldsymbol{AB} - \boldsymbol{BA} = \boldsymbol{0}$, then $e^{h(\boldsymbol{A}+\boldsymbol{B})} = e^{h\boldsymbol{A}} e^{h\boldsymbol{B}}$. Otherwise the approximation of (2), $e^{h(\boldsymbol{A}+\boldsymbol{B})} \approx e^{h\boldsymbol{A}} e^{h\boldsymbol{B}}$, is only first order accurate: the Taylor series of $e^{h\boldsymbol{A}} e^{h\boldsymbol{B}}$ agrees with the Taylor series of $e^{h(\boldsymbol{A}+\boldsymbol{B})}$ up to first order, but the second order terms differ. The Taylor series is

$$e^{h(\boldsymbol{A}+\boldsymbol{B})} = \boldsymbol{I} + h(\boldsymbol{A}+\boldsymbol{B}) + \frac{1}{2}h^2(\boldsymbol{A}+\boldsymbol{B})^2 + \dots$$

If we expand $(\boldsymbol{A}+\boldsymbol{B})^2 = \boldsymbol{A}^2 + \boldsymbol{AB} + \boldsymbol{BA} + \boldsymbol{B}^2$ then we see the reason we are limited to first order accuracy. In all terms in the Taylor series for our simple splitting approximation $e^{h\boldsymbol{A}} e^{h\boldsymbol{B}}$, $\boldsymbol{A}$ always comes before $\boldsymbol{B}$, so we can not match the term $\boldsymbol{BA}$ appearing in the correct series for $e^{h(\boldsymbol{A}+\boldsymbol{B})}$.

The previous observation suggests that symmetry might help and indeed it does. The symmetric Strang splitting [54, 57]

Second order splitting $$e^{h(\boldsymbol{A}+\boldsymbol{B})} \approx e^{\frac{1}{2}h\boldsymbol{A}} e^{h\boldsymbol{B}} e^{\frac{1}{2}h\boldsymbol{A}} \quad (3)$$

[1] An exercise in Golub and Van Loan [22] shows that $[\boldsymbol{A}, \boldsymbol{B}] = \boldsymbol{0}$ if and only if $e^{h(\boldsymbol{A}+\boldsymbol{B})} = e^{h\boldsymbol{A}} e^{h\boldsymbol{B}}$ for all h.

agrees with this Taylor series up to second order so it is a more accurate approximation than (2). Splitting methods have grown to have a rich history with many wonderful contributors. Marchuk is another of the important pioneers in the field. He found independently the second-order accurate splitting that we develop and extend in this chapter [40, 41].

When we numerically solve the ODE (1) on a time interval $[0\ \ T]$, we usually do not compute $u(T) = e^{T(\boldsymbol{A}+\boldsymbol{B})}u(0)$ in one big step time step T, nor do we approximate it by $e^{\frac{1}{2}T\boldsymbol{A}}e^{T\boldsymbol{B}}e^{\frac{1}{2}T\boldsymbol{A}}u(0)$. Our approximations are accurate for small time steps $h > 0$ but not for large times T. Therefore, instead, we take very many small steps h that add up to T. The first step is $v_1 = e^{\frac{1}{2}h\boldsymbol{A}}e^{h\boldsymbol{B}}e^{\frac{1}{2}h\boldsymbol{A}}u(0)$, which is our approximation to the solution of (1) at time h. The next step is computed from the previous step, recursively, by $v_{i+1} = e^{\frac{1}{2}h\boldsymbol{A}}e^{h\boldsymbol{B}}e^{\frac{1}{2}h\boldsymbol{A}}v_i$, so that v_i is our approximation to the exact solution $u(ih)$ at time ih for $i = 1, 2, \ldots$. After N steps, so that $Nh = T$, we arrive at the desired approximation $v_N \approx u(T)$.

Notice that a few Strang steps in succession

$$(e^{\frac{1}{2}h\boldsymbol{A}}e^{h\boldsymbol{B}}e^{\frac{1}{2}h\boldsymbol{A}})\,(e^{\frac{1}{2}h\boldsymbol{A}}e^{h\boldsymbol{B}}e^{\frac{1}{2}h\boldsymbol{A}}) = (e^{\frac{1}{2}h\boldsymbol{A}}e^{h\boldsymbol{B}})\quad \underbrace{e^{h\boldsymbol{A}}e^{h\boldsymbol{B}}}_{\text{first order step}}\quad (e^{\frac{1}{2}h\boldsymbol{A}})$$

is the same as a first order step in the middle, with a special half-step at the start and a different half-step at the end. This observation helps to reduce the overall work required to achieve second order accuracy when taking many steps in a row. This was noticed in the original paper [54] but perhaps it is not exploited as often as it could be.

Here is a more explicit example of first order and of second order accuracy. Notice the difference between the *local error* over one small time step h, and the *global error* over the whole time interval (those local errors grow or decay). We are interested in how fast the error decays as the time step h becomes smaller. The power of h is the key number. In general, the local error is one power of h more accurate than the global error from $1/h$ steps. Comparing Taylor series shows that

$$\text{local error} \qquad e^{\frac{1}{2}h\boldsymbol{A}}e^{h\boldsymbol{B}}e^{\frac{1}{2}h\boldsymbol{A}} - e^{h(\boldsymbol{A}+\boldsymbol{B})} = \boldsymbol{C}h^3 + \mathcal{O}(h^4).$$

where the constant is $\boldsymbol{C} = \frac{1}{24}([[\boldsymbol{A},\boldsymbol{B}],\boldsymbol{A}] + 2[[\boldsymbol{A},\boldsymbol{B}],\boldsymbol{B}])$. That is, for a single small time step h, the symmetric Strang splitting (3) has a local error that decays like h^3. As usual, the error gets smaller as we reduce the time step h: if we reduce the time step h from 1 to 0.1, then we expect the error to reduce by a factor of $0.1^3 = 0.001$. However, to compute the solution at the final time T, we take T/h steps, thus more steps with smaller h, and the global error is (number of steps) $\times$ (error at each step) $= (1/h)h^3 = h^2$, so we say the method is second order accurate.

We have examined accuracy by directly comparing the Taylor series of the exact solution and of the approximation. Another approach is via the Baker-Campbell-Hausdorff (BCH) formula:

$$\boldsymbol{C}(h\boldsymbol{A}, h\boldsymbol{B}) = h\boldsymbol{A} + h\boldsymbol{B} + \frac{1}{2}[h\boldsymbol{A}, h\boldsymbol{B}] + \cdots,$$

which, given $\boldsymbol{A}$ and $\boldsymbol{B}$, is an infinite series for the matrix $\boldsymbol{C}$ such that $e^{\boldsymbol{C}} = e^{h\boldsymbol{A}} e^{h\boldsymbol{B}}$. In nonlinear problems we want approximations that are *symplectic*. Then area in phase space is conserved, and approximate solutions to nearby problems remain close. The beautiful book of Hairer, Lubich and Wanner [24] discusses the BCH formula and its connection to splitting, and when splitting methods are symplectic for nonlinear equations. Strang splitting is symplectic.

2.1 Gaining an order of accuracy by taking an average

The nonsymmetric splitting $e^{h(\boldsymbol{A}+\boldsymbol{B})} \approx e^{h\boldsymbol{A}} e^{h\boldsymbol{B}}$ is only first order accurate. Of course, applying the operations the other way around, as in $e^{h\boldsymbol{B}} e^{h\boldsymbol{A}}$ is still only first order accurate. However, taking the *average* of these two first order approximations recovers a certain satisfying symmetry

$$e^{h(\boldsymbol{A}+\boldsymbol{B})} \approx \frac{e^{h\boldsymbol{A}} e^{h\boldsymbol{B}} + e^{h\boldsymbol{B}} e^{h\boldsymbol{A}}}{2}.$$

Symmetry is often associated with higher order methods. Indeed this symmetric average is *second order* accurate. That is, we gain one order of accuracy by taking an average. Whilst this observation is for averages in a very simple setting, we conjecture that it is closely related to the good experience reported in the setting of finance, where a stochastic differential equation is solved with good accuracy in the weak sense even if the order of operations is randomly determined by 'coin flipping' [45].

2.2 Higher order methods

Naturally, we wonder about achieving higher accuracy with splitting methods. Perhaps third-order splitting schemes or even higher order splitting schemes, are possible. Indeed they are, at least in theory. However, they are more complicated to implement: third order or higher order splitting schemes require either substeps that go backwards in time or forward in 'complex time' [3, 65, 24, 34, 13, 12]. For diffusion equations, going backwards in time raises serious issues of numerical stability. For reaction-diffusion equations, second-order splitting is still the most popular.

More generally, it is a meta-theorem of numerical analysis that second order methods often achieve the right balance between accuracy and complexity. First order methods are not accurate enough. Third order and higher order methods are accurate, but they have their own problems associated with stability or with being

too complicated to implement. Dahlquist and Henrici were amongst the pioneers to uncover these themes [5, 9, 10, 31].

2.3 Convection and diffusion

Until now, the discussion has been concerned with ODEs: time but no space. However, a big application of splitting is to PDEs: space and time.

An example is a PDE in one space dimension that models convection and diffusion. The continuous, exact solution $u(x,t)$ is to be approximated by finite differences. We compute a discrete approximation on a regular grid in space $x = \ldots, -2\Delta x, -\Delta x, 0, \Delta x, 2\Delta x, \ldots$ One part of our PDE is convection $\mathrm{d}u/\mathrm{d}t = \mathrm{d}u/\mathrm{d}x$. Convection is often represented by a one-sided finite difference matrix. For example, the finite difference approximation $\mathrm{d}u/\mathrm{d}x \approx (u(x+\Delta x) - u(x))/\Delta x$ comes via the matrix

$$\boldsymbol{P} = \frac{1}{\Delta x}\begin{bmatrix} \text{-1} & 1 & & \\ & \text{-1} & 1 & \\ & & \ddots & \ddots \end{bmatrix}.$$

Or we can approximate convection by a centered difference matrix:

$$\boldsymbol{Q} = \frac{1}{2\Delta x}\begin{bmatrix} 0 & 1 & & \\ \text{-1} & 0 & 1 & \\ & \ddots & \ddots & \ddots \end{bmatrix}.$$

Here we think of $\mathrm{d}u/\mathrm{d}x \approx (u(x+h) - u(x-h))/2\Delta x \approx \boldsymbol{Q}u$. Another part of our PDE is diffusion: $\mathrm{d}u/\mathrm{d}t = \mathrm{d}^2u/\mathrm{d}x^2$. The second spatial difference is often represented by the matrix

$$\boldsymbol{D} = \frac{1}{\Delta x^2}\begin{bmatrix} \text{-2} & 1 & & \\ 1 & \text{-2} & 1 & \\ & \ddots & \ddots & \ddots \end{bmatrix}.$$

We will come back to this matrix at the end of the chapter in (8), where we change signs to $\boldsymbol{K} = -\boldsymbol{D}$ so that $\boldsymbol{K}$ is positive definite and $\boldsymbol{K}$ models $-\mathrm{d}^2/\mathrm{d}x^2$. In solving a simple linear PDE with convection and diffusion terms

$$\frac{\partial u}{\partial t} = \underbrace{\frac{\partial u}{\partial x}}_{\text{convection}} + \underbrace{\frac{\partial^2 u}{\partial x^2}}_{\text{diffusion}}$$

with finite differences we may thus arrive at the ODE

$$\frac{\mathrm{d}u}{\mathrm{d}t} = (\boldsymbol{P} + \boldsymbol{D})u. \tag{4}$$

Here in the ODE we think of $u(t)$ as a column vector, with components storing the values of the solution on the spatial grid $[\ldots, u(-\Delta x), u(0), u(\Delta x), \ldots]^T$. (We are abusing notation slightly by using the same u in the PDE and in its ODE approximation.) This is a discrete-in-space and continuous-in-time, or semi-discrete approximation. We recognize it as the same ODE that we introduced at the very beginning (1), here with the particular choice $\boldsymbol{A} = \boldsymbol{D}$ and $\boldsymbol{B} = \boldsymbol{P}$. The solution to this semi-discrete approximation is the same $u(t) = e^{h(\boldsymbol{P}+\boldsymbol{D})}u(0)$, and it is natural to consider approximating this solution by splitting into convection $\boldsymbol{P}$ and diffusion terms $\boldsymbol{D}$.

In applying the splitting method (2), we somehow compute the approximation $e^{h\boldsymbol{P}}e^{h\boldsymbol{D}}u(0)$. Conceivably, we might choose to compute each of the matrix exponentials, $e^{h\boldsymbol{P}}$ and $e^{h\boldsymbol{D}}$, in full, and then multiply these full matrices by the vector $u(0)$. In practice that is usually not a good idea. One reason is that the matrices are often sparse, as in the examples of $\boldsymbol{D}$ and $\boldsymbol{P}$ here, and we want to take advantage of that sparsity, whereas the matrix exponential is typically full. Moreover, computing the matrix exponential is a classical problem of numerical analysis with many challenges [43].

Usually we only want the solution vector $u(t)$, not a whole matrix. For this purpose, ODE-solvers, such as Runge-Kutta methods or Adams-Bashforth methods, are a good choice [5, 31]. The point of this chapter is merely to observe that we can still apply splitting methods. We proceed in two stages. First stage: starting from $u(0)$, solve $\mathrm{d}u/\mathrm{d}t = \boldsymbol{D}u$ from time $t = 0$ to $t = h$ for the solution $w_{1/2}$. Second stage: starting from this $w_{1/2}$, solve $\mathrm{d}u/\mathrm{d}t = \boldsymbol{P}u$ from time $t = 0$ to $t = h$ for the solution $w_{2/2}$. Thus we have carried out a first order splitting: if our ODE-solvers were exact at each of the two stages, then $w_{2/2} = e^{h\boldsymbol{A}}e^{h\boldsymbol{B}}u(0)$. Often, we treat the diffusion term implicitly [1]. Hundsdorfer and Verwer discuss the numerical solution of convection-diffusion-reaction problems, noting additional issues when applying splitting methods to boundary value problems [30].

2.4 A reaction-diffusion PDE: splitting linear from nonlinear

A typical example in mathematical biology is a reaction-diffusion PDE in the form

$$\frac{\partial}{\partial t}\begin{bmatrix} u \\ v \end{bmatrix} = \underbrace{\left(\nabla \cdot \begin{bmatrix} D_u & 0 \\ 0 & D_v \end{bmatrix} \nabla\right)}_{\text{diffusion}} \begin{bmatrix} u \\ v \end{bmatrix} + \underbrace{\begin{bmatrix} f(u,v) \\ g(u,v) \end{bmatrix}}_{\text{reaction}}. \tag{5}$$

Here D_u and D_v are positive diffusion constants for the concentrations of the two species u and v, respectively. Commonly these are modeled as genuinely constant – not spatially varying – and in the special case that $D_u = D_v = 1$, then our diffusion operator simplifies to $\nabla \cdot \nabla$, which many authors would denote by ∇^2, or in one space dimension by $\partial^2/\partial x^2$. The reactions are modeled by nonlinear functions f

and g . For example, in a Gierer-Meinhardt model, $f(u,v) = a - bu + u^2/v^2$, and $g(u,v) = u^2 - v$ [44, 39].

Diffusion is linear. Reactions are nonlinear. We split these two terms and solve them separately. We solve the linear diffusion implicitly. The nonlinear reactions are solved explicitly. By analysis of our linear test problem (1) we found the accuracy of splitting approximations to be second order accurate, in the case of symmetric Strang splitting. However, the questions of accuracy and of stability concerning splitting approximations to the more general form

$$\frac{\mathrm{d}u}{\mathrm{d}t} = \boldsymbol{A}u + f(u)$$

where $\boldsymbol{A}$ is still linear but now f is nonlinear, such as arise in reaction-diffusion PDEs, has no such simple answer [52, 1].

2.5 Stability of splitting methods

We have seen that splitting can be accurate. Now we wonder about stability. Together, *stability and consistency imply convergence*. That is both a real theorem for many fundamental examples, and also a meta-theorem of numerical analysis [33]. Indeed it is sometimes suggested (together with its converse) as The Fundamental Theorem of Numerical Analysis [60, 63, 8, 55].

In this context those three keywords have special meanings. *Accuracy* means that the computed approximation is close to the exact solution. A method is *stable* if over many steps, the local errors only grow slowly in a controlled way and do not come to dominate the solution. (A mathematical problem is well-conditioned if small perturbations in the input only result in correspondingly small perturbations in the output. Stability for numerical algorithms is analogous to the idea of conditioning for mathematical problems.) The method is *consistent* if, over a single time step h, the numerical approximation is more and more accurate as h becomes smaller. For instance, we saw that the local error of a single step with symmetric Strang splitting (3), scales like h^3 as $h \to 0$, so that method is consistent. For a consistent method, we hope that the global error, after many time steps, also goes to zero as $h \to 0$: if this happens then the numerical approximation is converging to the true solution. Usually, finding a direct proof of convergence is a formidable task, whereas showing consistency and showing stability separately is more attainable. Then the theorem provides the desired assurance of convergence.

When we start thinking about the question of stability of splitting methods, typically we assume that the eigenvalues of $\boldsymbol{A}$ and of $\boldsymbol{B}$ all lie in the left half, $\mathrm{Re}(\lambda) \leq 0$, of the complex plane. Separately, each system is assumed stable.

It is natural to wonder if the eigenvalues of $(\boldsymbol{A} + \boldsymbol{B})$ also lie in the left half plane. This is not true in general. Turing patterns[2] in mathematical biology are a famous

[2] See, for example, Rauch's notes on Turing instability [48].

instance of this [44, 39]. Typically we linearise at a steady state. For example, if $\boldsymbol{J}$ is the Jacobian matrix of the reaction terms in (5) at the steady state, then we study the linear equation $\mathrm{d}\boldsymbol{u}/\mathrm{d}t = (\boldsymbol{J}+\boldsymbol{D})\boldsymbol{u}$, where $\boldsymbol{u} = [u\ v]^T$. Separately, the diffusion operator $\boldsymbol{D}$, and the Jacobian $\boldsymbol{J}$ each have eigenvalues with negative real part. Analysis of Turing instability begins by identifying conditions under which an eigenvalue of $(\boldsymbol{J}+\boldsymbol{D})$ can still have a positive real part.

With the assumption that all eigenvalues of $\boldsymbol{M}$ have negative real part, $e^{t\boldsymbol{M}}$ is stable for large times t. If the matrix $\boldsymbol{M}$ is real symmetric then the matrix exponential $e^{h\boldsymbol{M}}$ is also well behaved for small times. Otherwise, whilst eigenvalues do govern the long-time behavior of $e^{t\boldsymbol{M}}$, the transient behavior can be different if there is a significant *pseudospectrum* [61]. This can happen when the eigenvectors of the matrix are not orthogonal, and the convection-diffusion operator is an important example [49].

Even if the matrix exponentials $e^{t(\boldsymbol{A}+\boldsymbol{B})}$, $e^{t\boldsymbol{A}}$, and $e^{t\boldsymbol{B}}$ are stable separately, we don't yet know about the stability of their multiplicative combination, as in say, first order splitting, $e^{h\boldsymbol{A}}e^{h\boldsymbol{B}}$. A *sufficient condition* for stability is that the symmetric parts

$$\text{symmetric part} \qquad \boldsymbol{A}_{sym} \equiv \frac{\boldsymbol{A}+\boldsymbol{A}^T}{2}$$

of $\boldsymbol{A}$ and of $\boldsymbol{B}$ are negative definite. In that case we have stability of both ordinary splitting and symmetric Strang splitting because separately $||e^{h\boldsymbol{A}}|| \leq 1$ and $||e^{h\boldsymbol{B}}|| \leq 1$. This result was proved by Dahlquist and the idea is closely related to the log norm of a matrix. One way to show this is to observe that the derivative of $||e^{h\boldsymbol{A}}u||^2$ is $((\boldsymbol{A}+\boldsymbol{A}^T)e^{h\boldsymbol{A}}u, e^{h\boldsymbol{A}}u) \leq 0$. Another way is to let $\boldsymbol{A} = \boldsymbol{A}_{sym} + \boldsymbol{A}_{anti}$, and observe that $e^{h\boldsymbol{A}}$ is the limit of $e^{h\boldsymbol{A}_{sym}/n}e^{h\boldsymbol{A}_{anti}/n}e^{h\boldsymbol{A}_{sym}/n}\ldots e^{h\boldsymbol{A}_{anti}/n}$. Each $||e^{h\boldsymbol{A}_{sym}/n}|| \leq 1$ since eigenvalues of $\boldsymbol{A}_{sym}$ are negative and the matrix is symmetric. Each $||e^{h\boldsymbol{A}_{anti}/n}|| \leq 1$ since eigenvalues of $e^{h\boldsymbol{A}_{anti}/n}$ are purely imaginary and the matrix is orthogonal.

Returning to the convection-diffusion example (4), we can now see that the splitting method is stable. In that case, note that $\boldsymbol{P} = \boldsymbol{Q} + h\boldsymbol{D}$, where $\boldsymbol{Q}$ is the antisymmetric part and $h\boldsymbol{D}$ is the symmetric part. Observing that the diffusion matrix $\boldsymbol{D}$ is symmetric negative definite, we see that such a splitting is strongly stable with symmetric Strang splitting.

Having established that splitting is stable and accurate over finite times, we have now investigated many of the concerns of the original paper on Strang splitting:

```
Surprisingly, there seem to be no recognized rules for the
comparison of alternative difference schemes. Clearly there
are  three fundamental criteria -- accuracy, simplicity, and
stability -- and we shall evaluate each of the competing
schemes in these terms.
```

– " On the Construction and Comparison of Difference Schemes"
Gilbert Strang, *SIAM J. Numer. Anal.*, 1968.

Perhaps another criterion could have been added – how well the method transitions from an early transient behavior to late stage steady state behavior. It turns out

that most splitting schemes do not exactly capture the all-important steady state. We review next a new "balanced splitting" scheme that corrects this error.

2.6 Ordinary splitting does NOT preserve the steady state

Suppose u_∞ is a steady state of (1). By definition

$$(\boldsymbol{A}+\boldsymbol{B})u_\infty = 0 \qquad \text{and} \qquad e^{\boldsymbol{A}+\boldsymbol{B}}u_\infty = u_\infty.$$

In special cases, such as when $\boldsymbol{A}u_\infty = \boldsymbol{B}u_\infty = 0$, both first order splitting (2), and second order splitting (3) preserve steady states of the original ODE. However, in general, standard splitting approximations do *not* preserve the steady state u_∞: $e^{h\boldsymbol{A}}e^{h\boldsymbol{B}}u_\infty \neq u_\infty$ and $e^{\frac{1}{2}h\boldsymbol{A}}e^{h\boldsymbol{B}}e^{\frac{1}{2}h\boldsymbol{A}}u_\infty \neq u_\infty$.

3 Balanced splitting: a symmetric Strang splitting that preserves the steady state

We again consider our linear ODE $\mathrm{d}v/\mathrm{d}t = (\boldsymbol{A}+\boldsymbol{B})v$. In **balanced splitting** [53] a constant vector, c, is computed at the beginning of each step. Then c is added to $\boldsymbol{A}v$ and subtracted from $\boldsymbol{B}v$ in the substages of the splitting approximation; the parts still add to $(\boldsymbol{A}+\boldsymbol{B})v$.

A first idea (simple balancing) is to choose c so that $\boldsymbol{A}v + c = \boldsymbol{B}v - c$. Then the first stage solves

$$\mathrm{d}v/\mathrm{d}t = \boldsymbol{A}v + c, \qquad c = \frac{1}{2}(\boldsymbol{B}-\boldsymbol{A})v_0, \qquad v_0 = u_0, \tag{6}$$

for the solution[3] $v^+ = e^{h\boldsymbol{A}}v_0 + (e^{h\boldsymbol{A}} - \boldsymbol{I})\boldsymbol{A}^{-1}c$, at time h . Now the second stage solves

$$\mathrm{d}v/\mathrm{d}t = \boldsymbol{B}v - c, \qquad v_0 = v^+,$$

for the solution $e^{h\boldsymbol{B}}v^+ - (e^{h\boldsymbol{B}} - \boldsymbol{I})\boldsymbol{B}^{-1}c$ at time h. We call this method 'nonsymmetric balanced splitting'. By adding and subtracting a constant vector, we see this as a modification of first order splitting (2), but the modified version has an advantage near steady state. Actually this choice of $c = \frac{1}{2}(\boldsymbol{B}-\boldsymbol{A})v$ frequently leads to instability [53].

Of course there is also a simple modification of the second order splitting (3) approximation, where we add and subtract a constant at each stage. In symmetric balanced splitting, we solve the '$\boldsymbol{A}$ stage' for a time step $\frac{1}{2}h$, then the '$\boldsymbol{B}$ stage'

[3] Here we assume $\boldsymbol{A}$ and $\boldsymbol{B}$ are invertible. The non-invertible case is treated by the variation-of-parameters formula [58].

for a time step h, and finally the '$\boldsymbol{A}$ stage' again for a time step $\frac{1}{2}h$. That is, in the 'symmetric balanced splitting method', the first stage $\{\, \mathrm{d}v/\mathrm{d}t = \boldsymbol{A}v + c,\ v_0 = u_0 \,\}$, is the same as before except that we solve for the solution over a smaller interval $h/2$. Then $v^+ = e^{\frac{1}{2}h\boldsymbol{A}}v_0 + (e^{\frac{1}{2}h\boldsymbol{A}} - \boldsymbol{I})\boldsymbol{A}^{-1}c$, is the initial condition for the second stage. We solve for v^{++} over the time interval h. The third stage is $\{\, \mathrm{d}v/\mathrm{d}t = \boldsymbol{A}v + c,\ v_0 = v^{++} \,\}$ over the remaining half step $h/2$. The output, $v(h) = \boldsymbol{R}v(0)$, is the approximation at the end of the whole step, where

$$\boldsymbol{R} = \frac{1}{2}\Big(\boldsymbol{I} - \boldsymbol{A}^{-1}\boldsymbol{B} + e^{\frac{1}{2}h\boldsymbol{A}}e^{h\boldsymbol{B}}e^{\frac{1}{2}h\boldsymbol{A}}(\boldsymbol{I} + \boldsymbol{A}^{-1}\boldsymbol{B}) + e^{\frac{1}{2}h\boldsymbol{A}}(e^{h\boldsymbol{B}} - \boldsymbol{I})(\boldsymbol{B}^{-1}\boldsymbol{A} - \boldsymbol{A}^{-1}\boldsymbol{B})\Big). \tag{7}$$

In the special case that $\boldsymbol{A} = \boldsymbol{B}$, the formula simplifies to $\boldsymbol{R} = e^{\frac{1}{2}h\boldsymbol{A}}e^{h\boldsymbol{B}}e^{\frac{1}{2}h\boldsymbol{A}}$ so symmetric balanced splitting is identical with symmetric Strang splitting in this case. This is what we expect because in this case $c = 0$. To improve stability we may choose different balancing constants, thereby moving from simple balanced splitting to *rebalanced splitting* [53]. One good choice [53, equation 7.7] is $c_{n+1} = (-v_{n+1} + 2v_n^{++} - 2v_n^{+} + v_n)/2h + c_n$, which involves all values from the previous step.

3.1 Balanced splitting preserves the steady state

Having introduced the method of balanced splitting, we now confirm its most important property. Recall that we are at a steady state if and only if the derivative is zero. Hence we may check that a steady state, u_∞, of the original system (1). is also a steady state of the new balanced splitting approximation by direct substitution and evaluation of the derivative. Suppose that we start at steady state, i.e. $v_0 = u(0) = u_\infty$. The first stage of balanced splitting is

$$\mathrm{d}v/\mathrm{d}t = \boldsymbol{A}u_\infty + c = \boldsymbol{A}u_\infty + \frac{1}{2}(\boldsymbol{B} - \boldsymbol{A})u_\infty = \frac{1}{2}(\boldsymbol{A} + \boldsymbol{B})u_\infty = \boldsymbol{0},$$

where we have used the defining property of the steady state, i.e. $(\boldsymbol{A} + \boldsymbol{B})u_\infty = 0$. Similarly for the second stage $\mathrm{d}v/\mathrm{d}t = 0$, so $v(h) = u(0) = u_\infty$. Thus a steady state of the original ODE is also a steady state of the balanced splitting method. The same observation shows that other variations of the balanced splitting method (such as symmetric balanced splitting) also preserve the steady state. This also gives the intuition behind the particular choice of the constant c – it is chosen in just the right way to 'balance' each substep.

Two special cases for which ordinary splitting may be preferable to balanced splitting are:

- In the special case that $\boldsymbol{A}$ and $\boldsymbol{B}$ commute, ordinary splitting is exact. However, balanced splitting does not share this property.

- In the special case that $\boldsymbol{A}u_\infty = \boldsymbol{B}u_\infty = \boldsymbol{0}$, ordinary splitting preserves the steady state, so balanced splitting does not offer an advantage in this case.

The eigenvalues of $\boldsymbol{R}$ in (7) will tell us about the stability of simple balanced splitting – that remains an area of active interest [53], as does stability of operator splitting more generally [50]. The main message is that balanced splitting has applications to important problems where ordinary splitting approximations fail to capture the steady state [53].

3.2 Splitting fast from slow

Splitting fast processes from slow processes is very common in applied mathematics. After averaging away the fast processes, a simplified model is reached, which is sometimes known as a quasi-steady-state approximation. The principles go further than splitting, but splitting is the first step. Potentially, time-scale separation provides another application for balanced splitting: quasi-steady state approximations are not always guaranteed to preserve the steady state of the original model. We wonder if a balanced splitting can be extended to efficient simulation of stochastic processes with fast and slow time-scales [6, 15, 46].

4 A very special Toeplitz-plus-Hankel splitting

We now describe a very special splitting: a Toeplitz-plus-Hankel splitting [59]. Unlike the previous examples, where exponentials of separate terms were computed separately (e.g. in first order splitting (2)) as a computationally efficient approximation, in the coming example (11) the exact solution is split into two parts, merely to gain a novel perspective through the lens of splitting. We see solutions to the wave equation as the sum of a Toeplitz solution and a Hankel solution. It transpires that reflections at the boundary come from the Hankel part of the operator (Figure 1).

4.1 All matrix functions $f(\boldsymbol{K})$ are Toeplitz-plus-Hankel

We begin with the $N \times N$ tridiagonal, symmetric positive definite Toeplitz matrix [64, 29]:

$$\boldsymbol{K} = \begin{bmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{bmatrix} \qquad h = \frac{1}{N+1}. \tag{8}$$

Perhaps this the most studied matrix in all of computational mathematics [55, 56]. Its eigenvalues and eigenvectors are known:

Eigenvalues of $\boldsymbol{K}$	$\lambda_k = 2-2\cos(k\pi h), \qquad k = 1,\dots,N$
Eigenvectors of $\boldsymbol{K}$	$\boldsymbol{v}_k = \sqrt{\frac{2}{N+1}}\Big(\sin(k\pi h), \sin(2k\pi h), \dots, \sin(Nk\pi h)\Big)^T$
Function of $\boldsymbol{K}$	$f(\boldsymbol{K})_{m,n} = \frac{2}{N+1}\sum_{k=1}^{N} f(\lambda_k)\sin(mk\pi h)\sin(nk\pi h)$

They produce the spectral decomposition

$$\textbf{Spectral theorem} \qquad \boldsymbol{K} \;=\; \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^T \;=\; \sum_{1}^{N} \lambda_k \boldsymbol{v}_k \boldsymbol{v}_k^T \tag{9}$$

where the matrix $\boldsymbol{K}$ is constructed from its eigenvalues in the diagonal matrix $\boldsymbol{\Lambda}$ and its eigenvectors in the columns of $\boldsymbol{V}$. This *diagonalization* separates $\boldsymbol{K}$ into a sum of rank one symmetric matrices $\lambda_k \boldsymbol{v}_k \boldsymbol{v}_k^T$. Now *any matrix function* [28] comes easily via this diagonalization: $f(\boldsymbol{K}) = \boldsymbol{V} f(\boldsymbol{\Lambda}) \boldsymbol{V}^T = \sum_1^N f(\lambda_k) \boldsymbol{v}_k \boldsymbol{v}_k^T$.

Entries of the rank one matrices $\boldsymbol{v}_k \boldsymbol{v}_k^T$ are products of sines. By rewriting those products $\sin(m\theta)\sin(n\theta)$ in terms of $\cos((\boldsymbol{m}-\boldsymbol{n})\theta)$ (which leads to a Toeplitz part) and $\cos((\boldsymbol{m}+\boldsymbol{n})\theta)$ (which leads to a Hankel part), we learn that the rank one matrix

$$\boldsymbol{v}_k \boldsymbol{v}_k^T = \boldsymbol{T}_k + \boldsymbol{H}_k \tag{10}$$

is Toeplitz-plus-Hankel, for all k [59]. Explicitly,

$$\text{Toeplitz} \qquad \big(\boldsymbol{T}_k\big)_{mn} = \frac{1}{N+1}\cos\big((m-n)k\pi h\big)$$
$$\text{Hankel} \qquad \big(\boldsymbol{H}_k\big)_{mn} = -\frac{1}{N+1}\cos\big((m+n)k\pi h\big).$$

This shows that $\boldsymbol{K}$ has the strong Toeplitz-plus-Hankel property: the rank one matrices $\boldsymbol{v}_k \boldsymbol{v}_k^T$ coming from the eigenvectors can be written as a sum of a Toeplitz matrix and a Hankel matrix.

We quickly recall that *Toeplitz matrices* are those with constant diagonals (entries depend on $m-n$). *Hankel matrices* have constant antidiagonals (entries depend on $m+n$). By applying a Toeplitz matrix and a Hankel matrix to the same input, you see shifts in opposite directions. Toeplitz shifts the output forwards, while Hankel shifts the output backwards:

$$\textbf{Toeplitz} \qquad \begin{bmatrix} \mathrm{b}\ \mathrm{a} & & \\ \mathrm{c}\ \mathrm{b}\ \mathrm{a} & & \\ \mathrm{c}\ \mathrm{b}\ \mathrm{a} & & \\ \mathrm{c}\ \mathrm{b} & & \end{bmatrix} \begin{bmatrix} 0\ 0 \\ 1\ 0 \\ 0\ 1 \\ 0\ 0 \end{bmatrix} = \begin{bmatrix} \mathrm{a} \\ \mathrm{b}\ \mathrm{a} \\ \mathrm{c}\ \mathrm{b} \\ \mathrm{c} \end{bmatrix} \qquad \text{forward shift}$$

$$\textbf{Hankel} \qquad \begin{bmatrix} \mathrm{a}\ \mathrm{b} & & \\ \mathrm{a}\ \mathrm{b}\ \mathrm{c} & & \\ \mathrm{a}\ \mathrm{b}\ \mathrm{c} & & \\ \mathrm{b}\ \mathrm{c} & & \end{bmatrix} \begin{bmatrix} 0\ 0 \\ 1\ 0 \\ 0\ 1 \\ 0\ 0 \end{bmatrix} = \begin{bmatrix} \mathrm{a} \\ \mathrm{a}\ \mathrm{b} \\ \mathrm{b}\ \mathrm{c} \\ \mathrm{c} \end{bmatrix} \qquad \text{backward shift}$$

Combining (10) with (9), we now see $f(\boldsymbol{K})$ as the sum of a Toeplitz matrix and a Hankel matrix:

$$\textbf{Matrix function} \qquad \begin{aligned} f(\boldsymbol{K}) &= \boldsymbol{V} f(\boldsymbol{\Lambda}) \boldsymbol{V}^T = \sum_1^N f(\lambda_k)(\boldsymbol{T}_k + \boldsymbol{H}_k) \\ &= \underbrace{\sum_1^N f(\lambda_k)\boldsymbol{T}_k}_{\text{Toeplitz}} + \underbrace{\sum_1^N f(\lambda_k)\boldsymbol{H}_k}_{\text{Hankel}} \end{aligned} \tag{11}$$

If we choose $f(z) = z^{-1}$, then we split the inverse matrix into Toeplitz and Hankel parts: $\boldsymbol{K}^{-1} = \boldsymbol{T} + \boldsymbol{H}$. With $N = 3$, this $\boldsymbol{T}$ and $\boldsymbol{H}$ are

$$\boldsymbol{K}^{-1} = \frac{1}{4}\begin{bmatrix} 3\ 2\ 1 \\ 2\ 4\ 2 \\ 1\ 2\ 3 \end{bmatrix} = \frac{1}{8}\begin{bmatrix} 5 & 2 & -1 \\ 2 & 5 & 2 \\ -1 & 2 & 5 \end{bmatrix} + \frac{1}{8}\begin{bmatrix} 1\ 2\ 3 \\ 2\ 3\ 2 \\ 3\ 2\ 1 \end{bmatrix}.$$

In summary, the matrix $\boldsymbol{K}$, and all functions of that matrix, are Toeplitz-plus-Hankel [59]. In the sequel, we make the particular choice $f(z) = \exp(\pm it\sqrt{z}/\Delta x)$ or its real part, $f(z) = \cos(t\sqrt{z}/\Delta x)$. Then $f(\boldsymbol{K})$ solves a wave equation.

Before we proceed to the wave equation, we make one small observation about the example of the convection-diffusion operator $(\boldsymbol{P} + \boldsymbol{D})$ in (4). It is an important instance of a nonsymmetric matrix where the *pseudospectra* plays a role in the analysis [49]. That nonsymmetric matrix certainly does not have the strong Toeplitz-plus-Hankel property. However, with the help of a simple diagonal matrix $\boldsymbol{Z}$ the similar matrix $\boldsymbol{S} \equiv \boldsymbol{Z}(\boldsymbol{P} + \boldsymbol{D})\boldsymbol{Z}^{-1}$ is symmetric, and $\boldsymbol{S}$ *does* have the strong Toeplitz-plus-Hankel property. The ith diagonal entry $z_i = \boldsymbol{Z}_{i,i}$ is found by setting $z_1 = 1$, and $z_{i+1} = z_i\sqrt{b_i/a_i}$, where $a_i = \boldsymbol{M}_{i+1,i}$, and $b_i = \boldsymbol{M}_{i,i+1}$, and $\boldsymbol{M} = (\boldsymbol{P} + \boldsymbol{D})$. We hope to explore Toeplitz-plus-Hankel properties for convection-diffusion operators in the future.

4.2 The wave equation is Toeplitz-plus-Hankel

Our model problem is on an interval $-1 \leq x \leq 1$ with zero Dirichlet boundary conditions $u(-1,t) = u(1,t) = 0$. The second derivative u_{xx} is replaced by second differences at the mesh points $x = -1, \ldots, -2\Delta x, -\Delta x, 0, \Delta x, 2\Delta x, \ldots, 1$, where $\Delta x = 2/(N-1)$. The familiar wave equation can be approximated with the help of the second difference matrix $\boldsymbol{K}$:

$$\text{Wave equation} \quad \frac{\partial^2}{\partial t^2}\boldsymbol{u} = \frac{\partial^2}{\partial x^2}\boldsymbol{u} \qquad \text{becomes} \qquad \frac{\mathrm{d}^2}{\mathrm{d}t^2}\boldsymbol{u} = -\frac{\boldsymbol{K}}{\Delta x^2}\boldsymbol{u}. \tag{12}$$

Time remains continuous in this finite difference, semi-discrete approximation. One solution to the semi-discrete approximation in (12) involves exponentials or cosines of matrices:

$$\text{Solution} \qquad \boldsymbol{u}(t) = f(\boldsymbol{K})\boldsymbol{u}(0) = \cos\left(t\sqrt{\boldsymbol{K}}/\Delta x\right)\boldsymbol{u}(0).$$

Our purpose here is to apply the Toeplitz-plus-Hankel splitting, so we again set $\boldsymbol{T} = \sum_k f(\lambda_k)\boldsymbol{T}_k$ and $\boldsymbol{H} = \sum_k f(\lambda_k)\boldsymbol{H}_k$, with $\boldsymbol{T}_k$ and $\boldsymbol{H}_k$ as in (10). Now with $f(z) = \cos(t\sqrt{z}/\Delta x)$ in (11) we see this same solution as the sum of two parts:

$$\boldsymbol{u}(t) = f(\boldsymbol{K})\boldsymbol{u}(0) = \underbrace{\boldsymbol{T}\boldsymbol{u}(0)}_{\text{Toeplitz}} + \underbrace{\boldsymbol{H}\boldsymbol{u}(0)}_{\text{Hankel}}.$$

Unlike the approximate splitting into products of exponentials discussed in the previous sections of this chapter, here we see an exact splitting into a sum. Thus we have split the wave equation into a Toeplitz part and a Hankel part. Now we can separately investigate the behavior of the solutions coming from each part.

Figure 1 shows the exact solution to the wave equation via d'Alembert's formula, as if the equation were on the whole real line with no boundaries. We use this as a reference to compare to the solution of the same problem with Dirichlet boundary conditions. We see the consequences of the boundary in the differences between these solutions. Figure 1 also shows, separately, the solutions coming from the Toeplitz part ($\boldsymbol{T}\boldsymbol{u}(0)$) and the Hankel part ($\boldsymbol{H}\boldsymbol{u}(0)$). Their sum solves the Dirichlet problem exactly. The most interesting behavior happens at the boundary. Before reaching the boundary, the solution is essentially Toeplitz. After reaching the boundary, the solution is essentially Hankel. The reflection at the boundary comes from the Hankel part of the operator.

The Toeplitz-plus-Hankel splitting described here is very special, but in this example the splitting does show reflections at the boundary in a new light: the reflections come from the Hankel part of the operator. The design of absorbing boundary conditions, or perfectly matched layers, is a big subject in computational science, that we will leave untouched [17, 27, 35]. We conjecture that Figure 1 can be understood by the *method of images* [59, 23]. That would involve identifying the solution of the Dirichlet boundary condition version of the problem here with the solution of a closely related problem having *periodic* boundary conditions. Periodic behavior is

far from ideal when designing absorbing boundary conditions – we don't want the wave to come back to the domain later. Nevertheless, whilst the approximation of the Sommerfeld boundary condition results in a small reflection here, it is intriguing that the Toeplitz part of the solution is seemingly reflectionless in Figure 1.

Outlook

This chapter introduced the basic ideas behind operator splitting methods. We focused on the application of splitting methods to solve differential equations. Historically, that has been their greatest application, though by now the splitting idea has found wide applications such as in optimization. We reviewed some of the main applications in biology especially, such as splitting reaction terms from diffusion terms in reaction-diffusion PDEs. Operator splitting is an old idea of numerical analysis, so it is pleasing that new ideas and new applications keep appearing even today. Perhaps one of the biggest contemporary applications of splitting involves coupling models across scales, such as appropriate coupling of mesoscopic reaction diffusion master equation models to finer, microscopic models [19, 18, 26, 25]. On that front, no doubt more great work on splitting methods is still to come.

Toeplitz-plus-Hankel parts of the wave equation

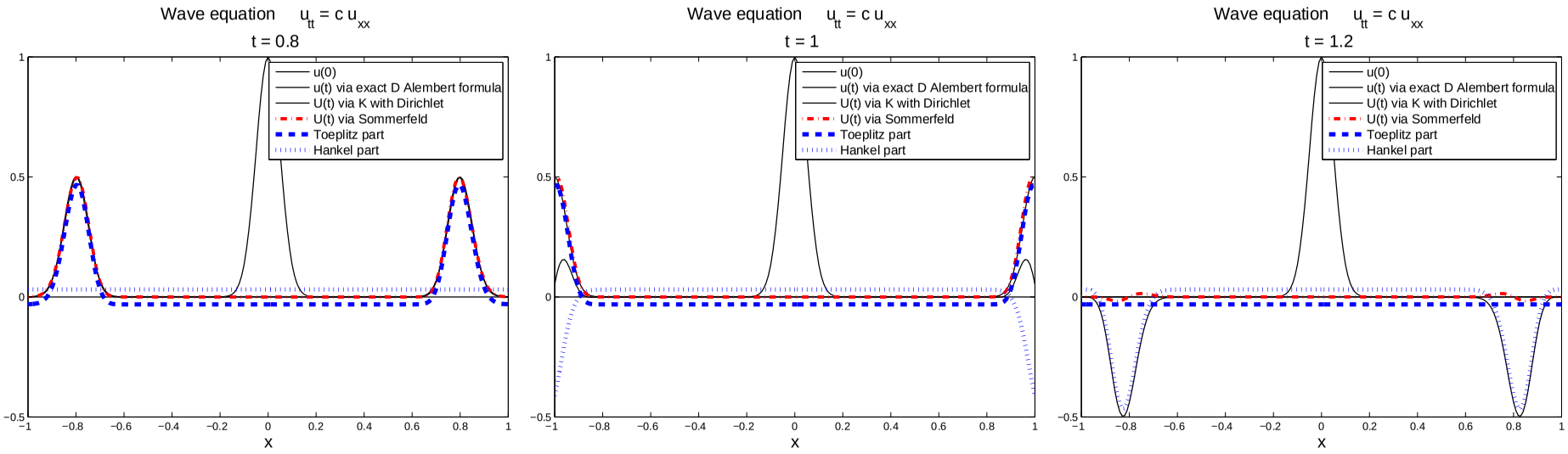

Fig. 1 Solutions of the wave equation $u_{tt} = u_{xx}$ at three snapshots in time: before ($t = 0.8$), during ($t = 1.0$) and after ($t = 1.2$) the wave reaches the boundary of the computational domain. The exact solution (d'Alembert's formula) and various finite difference approximations, with $\Delta x = 2/(N-1)$, $N = 201$, are shown. With the Sommerfeld radiation condition, $u_x = \pm u_t$, at the ends of our finite computational domain, we hope our computational solution is in perfect agreement with the solution to the wave equation on the whole line (with no boundary). However, a finite difference approximation of the Sommerfeld condition is not exact, and the small reflection visible at $t = 1.2$ is a manifestation of the error. With Dirichlet boundary conditions, in $\boldsymbol{K}$, **the solution splits exactly into the sum of a Toeplitz solution and a Hankel solution.** Before reaching the boundary, the solution is purely Toeplitz, while the Hankel part is absent. After reaching the boundary, it is the other way around: the solution is purely Hankel, while the Toeplitz part is absent. Hankel parts come from the boundary. The Toeplitz part is reflectionless.

References

1. Ascher, U., Ruuth, S., Wetton, B.: Implicit-explicit methods for time-dependent partial differential equations. SIAM Journal on Numerical Analysis (1995)
2. Asmussen, S., Glynn, P.W.: Stochastic Simulation: Algorithms and Analysis. Springer (2007)
3. Blanes, S., Casas, F., Chartier, P., Murua, A.: Optimized high-order splitting methods for some classes of parabolic equations. Math. Comput. (2012)
4. Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J.: Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. Foundations and Trends in Machine Learning pp. 1–122 (2011)
5. Butcher, J.: Numerical Methods for Ordinary Differential Equations. Wiley (2003)
6. Cao, Y., Gillespie, D.T., Petzold, L.R.: The slow-scale stochastic simulation algorithm. J. Chem. Phys. **122**, 014,116–1–18 (2005)
7. Chorin, A.J.: Numerical solution of the Navier-Stokes equations. Math. Comp. pp. 745–762 (1968)
8. Dahlquist, G.: Convergence and stability in the numerical integration of ordinary differential equations. Math. Scand. pp. 33–53 (1956)
9. Dahlquist, G.: A special stability problem for linear multistep methods. BIT Numerical Mathematics pp. 27–43 (1963)
10. Dahlquist, G., Björk, A.: Numerical Methods. Prentice-Hall (1974)
11. Daubechies, I.: Ten Lectures on Wavelets. SIAM (1992)
12. Descombes, S.: Convergence of a splitting method of high order for reaction-diffusion systems. Mathematics of Computation **70**(236), 1481–1501 (2001)
13. Descombes, S., Schatzman, M.: Directions alternées d'ordre élevé en réaction-diffusion. Comptes Rendus de l'Académie des Sciences. Série 1, Mathématique **321**(11), 1521–1524 (1995)
14. Douglas, J., Rachford, H.H.: On the numerical solution of heat conduction problems in two and three space variables. Trans. Amer. Math. Soc. pp. 421–439 (1956)
15. E, W., Liu, D., Vanden-Eijnden, E.: Nested stochastic simulation algorithm for chemical kinetic systems with disparate rates. J. Chem. Phys. **123**(19), 194,107 (2005)
16. Engblom, S.: Strong convergence for split-step methods in stochastic jump kinetics (2014). http://arxiv.org/abs/1412.6292
17. Engquist, B., Majda, A.: Absorbing boundary conditions for numerical simulation of waves. Proc Natl Acad Sci U S A **74**, 765–1766 (1977)
18. Ferm, L., Lötstedt, P.: Numerical method for coupling the macro and meso scales in stochastic chemical kinetics. BIT Numerical Mathematics **47**(4), 735–762 (2007)
19. Ferm, L., Lötstedt, P., Hellander, A.: A hierarchy of approximations of the master equation scaled by a size parameter. J. Sci. Comput. **34**, 127–151 (2008)
20. Giles, M.: Multi-level Monte Carlo path simulation. Operations Research **56**, 607–617 (2008)
21. Goldstein, T., Osher, S.: The Split Bregman Method for L1-Regularized Problems. SIAM Journal on Imaging Sciences **2**(2), 323–343 (2009)
22. Golub, G.H., Van Loan, C.F.: Matrix Computations. Johns Hopkins (1996)
23. Haberman, R.: Applied Partial Differential Equations. Prentice Hall (2013)
24. Hairer, E., Lubich, C., Wanner, G.: Geometric numerical integration: Structure-preserving algorithms for ordinary differential equations. Springer (2006)
25. Hellander, A., Hellander, S., Lötstedt, P.: Coupled mesoscopic and microscopic simulation of stochastic reaction-diffusion processes in mixed dimensions. Multiscale Model. Simul. pp. 585–611 (2012)
26. Hellander, A., Lawson, M., Drawert, B., Petzold, L.: Local error estimates for adaptive simulation of the reaction-diffusion Master Equation via operator splitting. J. Comput. Phys (2014)
27. Higdon, R.: Numerical Absorbing Boundary Conditions for the Wave Equation. Mathematics of Computation **49**, 65–90 (1987)
28. Higham, N.J.: Functions of Matrices. SIAM (2008)
29. Horn, R., Johnson, C.: Matrix Analysis. Cambridge University Press (2013)

30. Hundsdorfer, W., Verwer, J.: Numerical Solution of Time-Dependent Advection-Diffusion-Reaction Equations. Springer (2003)
31. Iserles, A.: A First Course in the Numerical Analysis of Differential Equations. Cambridge University Press (1996)
32. Jahnke, T., Altintan, D.: Efficient simulation of discrete stochastic reaction systems with a splitting method. BIT **50**, 797–822 (2010)
33. Lax, P.D., Richtmyer, R.D.: Survey of the stability of linear finite difference equations. Comm. Pure Appl. Math pp. 267–293 (1956)
34. Lee, J., Fornberg, B.: A split step approach for the 3-D Maxwell's equations. J. Comput. Appl. Math. **158**, 485–505 (2003)
35. Loh, P.R., Oskooi, A.F., Ibanescu, M., Skorobogatiy, M., Johnson, S.G.: Fundamental relation between phase and group velocity, and application to the failure of perfectly matched layers in backward-wave structures. Phys. Rev. E **79**, 065,601 (2009). DOI 10.1103/PhysRevE.79.065601. URL http://link.aps.org/doi/10.1103/PhysRevE.79.065601
36. Lubich, C., Oseledets, I.: A projector-splitting integrator for dynamical low-rank approximation. BIT Numerical Mathematics **54**, 171–188 (2014)
37. MacNamara, S., Burrage, K., Sidje, R.: Application of the Strang Splitting to the chemical master equation for simulating biochemical kinetics. The International Journal of Computational Science **2**, 402–421 (2008)
38. MacNamara, S., Burrage, K., Sidje, R.: Multiscale Modeling of Chemical Kinetics via the Master Equation. SIAM Multiscale Model. & Sim. **6**(4), 1146–1168 (2008)
39. Maini, P.K., Baker, R.E., Chuong, C.M.: The Turing model comes of molecular age. Science **314**, 1397–1398 (2006)
40. Marchuk, G.I.: Some application of splitting-up methods to the solution of mathematical physics problems. Aplikace matematiky pp. 103–132 (1968)
41. Marchuk, G.I.: Splitting and alternating direction methods. In: P. Ciarlet, J. Lions (eds.) Handbook of Numerical Analysis, vol. 1, pp. 197–462. North-Holland, Amsterdam (1990)
42. McLachlan, R., Quispel, G.: Splitting methods. Acta Numer. **11**, 341–434 (2002)
43. Moler, C., Van Loan, C.: Nineteen Dubious Ways to Compute the Exponential of a Matrix, 25 Years Later. SIAM Review **45(1)**, 3–49 (2003)
44. Murray, J.: Mathematical Biology: An Introduction. New York : Springer (2002)
45. Ninomiya, S., Victoir, N.: Weak approximation of stochastic differential equations and application to derivative pricing. Applied Mathematical Finance **15** (2008)
46. Pavliotis, G., Stuart, A.: Multiscale Methods: Averaging and Homogenization. Springer (2008)
47. Peaceman, D.W., Rachford Jr., H.H.: The numerical solution of parabolic and elliptic differential equations. Journal of the Society for Industrial and Applied Mathematics (1955)
48. Rauch, J.: The Turing Instability. URL: http://www.math.lsa.umich.edu/ rauch/
49. Reddy, S.C., Trefethen, L.N.: Pseudospectra of the convection-diffusion operator. SIAM J. Appl. Math (1994)
50. Ropp, D., Shadid, J.: Stability of operator splitting methods for systems with indefinite operators: Reaction-diffusion systems. Journal of Computational Physics **203**, 449–466 (2005)
51. Saad, Y.: Iterative Methods for Sparse Linear Systems. SIAM, Society for Industrial and Applied Mathematics, Philadelphia (2003)
52. Schatzman, M.: Toward non commutative numerical analysis: High order integration in time. Journal of Scientific Computing **17**, 99–116 (2002)
53. Speth, R., Green, W., MacNamara, S., Strang, G.: Balanced splitting and rebalanced splitting. SIAM Journal of Numerical Analysis **51**(6), 3084–3105 (2013)
54. Strang, G.: On the construction and comparison of difference schemes. SIAM J. Numer. Anal. **5**(2), 506–517 (1968)
55. Strang, G.: Computational Science and Engineering. Wellesley-Cambridge Press (2007)
56. Strang, G.: Introduction to Linear Algebra. Wellesley-Cambridge Press (2009)
57. Strang, G.: Essays in Linear Algebra. Wellesley-Cambridge Press (2012)
58. Strang, G.: Differential Equations and Linear Algebra. Wellesley-Cambridge Press (2014)

59. Strang, G., MacNamara, S.: Functions of difference matrices are Toeplitz plus Hankel. SIAM Review **56**(3), 525–546 (2014)
60. Trefethen, L.: Numerical analysis. In: T. Gowers, J. Barrow-Green, I. Leader (eds.) Princeton Companion to Mathematics, pp. 604–615. Princeton University Press (2008)
61. Trefethen, L., Embree, M.: Spectra and Pseudospectra: The Behavior of Nonnormal Matrices and Operators. Princeton University Press (2005)
62. Usadi, A., Dawson, C.: 50 years of ADI methods: Celebrating the contributions of Jim Douglas, Don Peaceman, and Henry Rachford. SIAM News **39** (2006)
63. Wanner, G.: Dahlquist's classical papers on stability theory. BIT Numerical Mathematics **46**, 671–683 (2006)
64. Widom, H.: Toeplitz matrices. In: I.I. Hirschman (ed.) Studies in Real and Complex Analysis. Prentice-Hall (1965)
65. Yoshida, H.: Construction of higher order symplectic integrators. Phys. Lett. A. **150**, 262–268 (1990)