ANZIAM J. **63** (2021), 23–38
doi:10.1017/S1446181121000043

SLOW-BURNING INSTABILITIES OF DUFORT–FRANKEL FINITE DIFFERENCING

DAVID GALLOWAY[1] and DAVID IVERS ✉[1]

(Received 26 June, 2020; accepted 1 January, 2021; first published online 30 April, 2021)

Abstract

DuFort–Frankel averaging is a tactic to stabilize Richardson's unstable three-level leapfrog timestepping scheme. By including the next time level in the right-hand-side evaluation, it is implicit, but it can be rearranged to give an explicit updating formula, thus apparently giving the best of both worlds. Textbooks prove unconditional stability for the heat equation, and extensive use on a variety of advection–diffusion equations has produced many useful results. Nonetheless, for some problems the scheme can fail in an interesting and surprising way, leading to instability at very long times. An analysis for a simple problem involving a pair of evolution equations that describe the spread of a rabies epidemic gives insight into how this occurs. An even simpler modified diffusion equation suffers from the same instability. Finally, the rabies problem is revisited and a stable method is found for a restricted range of parameter values, although no prescriptive recipe is known which selects this particular choice.

2020 *Mathematics subject classification*: primary 65M12; secondary 65M06.

Keywords and phrases: finite differences, numerical instability, DuFort–Frankel, leapfrog schemes, diffusion..

1. Introduction

DuFort–Frankel(DFF) finite-differencing [2] is a standard method that is explained in numeroustextbooks [14, 16] and websites. It is typically discussed in the context of the heat equation, for which a von Neumann stability analysis [1] shows that the method is unconditionally stable. DFF has been used extensively for many nonlinear problems involving advection and diffusion, particularly in fluid dynamics (see [1, 17] and references therein). In particular, Roberts and Weiss [18] developed algorithms with two overlaid meshes that are staggered in space and time, saving a factor of 2 in both storage and computing time. These were used successfully for many problems in convection and magnetoconvection (for example, [4, 5, 7, 13, 20]), for solving

[1] School of Mathematics and Statistics, University of Sydney, Sydney, NSW 2006, Australia;
e-mail: david.galloway@sydney.edu.au, david.ivers@sydney.edu.au.

© Australian Mathematical Society 2021

https://doi.org/10.1017/S1446181121000043 Published online by Cambridge University Press

a Fokker–Planck equation [10], and for a neural network problem [6]. With one exception, discussed in Section 6, no difficulties were encountered and the method evolved a reputation for reliability and efficiency—although, being only second-order accurate, it has been somewhat superseded by higher-order or spectral methods. (Roberts and Weiss [18] gave a fourth-order version, but this has been rarely used.)

For completeness, we summarize the derivation here, in preparation for the more complicated analyses that follow later. Richardson's leapfrog scheme for the heat equation is the three-level algorithm for the dependent variable u_j^n, defined at $x = j\Delta x, j = 0, \dots, J$, and $t = n\Delta t, n = 0, 1, 2, \dots,$ taking the form

$$\frac{u_j^{n+1} - u_j^{n-1}}{2\Delta t} = \frac{u_{j+1}^n + u_{j-1}^n - 2u_j^n}{(\Delta x)^2},$$

where Δx and Δt are the fixed mesh intervals in space and time. It is known to be highly unstable, but if the average of time levels $n-1$ and $n+1$ is used for the central element of the diffusion operator, the resulting scheme taking the form

$$\frac{u_j^{n+1} - u_j^{n-1}}{2\Delta t} = \frac{u_{j+1}^n + u_{j-1}^n - u_j^{n+1} - u_j^{n-1}}{(\Delta x)^2}$$

constitutes DFF. This is now implicit, but is easily rearranged to give the explicit updating formula

$$(1+\alpha)u_j^{n+1} = (1-\alpha)u_j^{n-1} + \alpha(u_{j+1}^n + u_{j-1}^n),$$

where $\alpha = 2\Delta t/(\Delta x)^2$. Testing, for instance, the zero solution for stability, assume a disturbance $U_n e^{ikx} = U_n e^{ikj\Delta x}$; there results the linear difference equation

$$(1+\alpha)U_{n+1} - 2\alpha\cos(k\Delta x)U_n - (1-\alpha)U_{n-1} = 0.$$

This has solutions with U_n proportional to r^n, where

$$(1+\alpha)r^2 - 2\alpha\cos(k\Delta x)r - (1-\alpha) = 0. \tag{1.1}$$

Stability or instability depends on whether the roots lie, respectively, inside or outside the unit circle in the complex r-plane. The quadratic can be solved and shown to have roots that lie inside (see, for example, [16]), with the exceptions that when the cosine is -1, the negative root is -1, and when the cosine is 1, the positive root is 1. These isolated points do not affect the stability, as they give no secular growth of any error, but they do hint that a small perturbation might push the scheme over the edge, as in fact we shall find later.

The Schur–Cohn test (that is, the Routh–Hurwitz test with the left half-plane mapped to the interior of the unit circle) gives necessary and sufficient conditions for the roots of a polynomial to lie inside the unit circle [8, 12]. For the quadratic $a_2r^2 + a_1r + a_0 = 0$ the conditions reduce to $|a_2 + a_0| > |a_1|$, which in the current case gives $|2\alpha| > |2\alpha\cos(k\Delta x)|$. This is clearly always true except at the aforementioned two points.

https://doi.org/10.1017/S1446181121000043 Published online by Cambridge University Press

There is also a well-known consistency requirement: the truncation error of the scheme is easily shown to be

$$O(\Delta t)^2 + O(\Delta x)^2 + O(\Delta t/\Delta x)^2.$$

One needs $\Delta t = O(\Delta x)^{1+\epsilon}$ with $\epsilon > 0$ for this to approach zero as $\Delta t, \Delta x \to 0$. In practice, for overall second-order accuracy, one takes $\Delta t = O(\Delta x)^2$, that is, α of order 1. We note in passing that the choice $\alpha = 1$ leads to the amazingly simple scheme

$$u_j^{n+1} = \tfrac{1}{2}(u_{j+1}^n + u_{j-1}^n),$$

still accurate to order $(\Delta t)^2$ and $(\Delta x)^2$. This choice obviates the need for an initial step that uses some other method; in this form it is equivalent to a special case of the forward-time centred-space method found in many textbooks (for example, [16, page 136]).

In Section 2 we present a problem which breaks DFF in an interesting and unexpected way. Specifically, our results exhibit instability at very long times, even after an apparent steady state has been achieved. Section 3 gives a von Neumann stability analysis which explains how this happens; basically an eigenvalue lies outside the unit circle, but only by a tiny amount. Errors therefore take a very long time to accumulate. Section 4 distils the essence of the instability into a very simple model problem, the heat equation with a linear source term. Section 5 discusses variations in the way DFF can be applied to the original problem, and finds one variant which gives a stable scheme for restricted parameter values. Section 6 concludes and cites other work on instability due to nonstandard boundary conditions.

2. An example of instability

The problem we encountered arose when we applied the DFF method to solve the following pair of equations

$$\frac{\partial S}{\partial t} = -IS + bS(1 - S), \tag{2.1}$$

$$\frac{\partial I}{\partial t} = IS - \lambda I + \frac{\partial^2 I}{\partial x^2} \tag{2.2}$$

with boundary conditions $\partial I/\partial x = 0$ at $x = 0, L$ and initial conditions $S = 1$, I a narrow Gaussian pulse centred somewhere in the middle of the domain. This is taken from J. D. Murray's book on mathematical biology [15], and is a simple attempt to model the spread of a rabies epidemic in a one-dimensional isolated country of length L. The context is explained fully in the book, and need not concern us further here, except to point out that the death rate λ and logistic parameter b are both positive. In the absence of the disease, the maximum population has been scaled to unity, $S = 1$; in addition, time has been scaled to give unit diffusivity.

https://doi.org/10.1017/S1446181121000043 Published online by Cambridge University Press

In addition to the two trivial fixed points $(S, I) = (0, 0)$ and $(S, I) = (1, 0)$, it is easily verified that the above equations have a steady-state solution

$$S = \lambda, \quad I = b(1 - \lambda).$$

Its stability is ascertained by linearizing this steady state and evaluating the Jacobian matrix of the right-hand side there. The eigenvalues are solutions to a quadratic, with negative real parts or simple zeros, meaning that any disturbance decays exponentially with time so that the solution is linearly stable.

The above equations apparently have no analytic solutions for the time-dependence. To solve them numerically, our background led us to try a DFF treatment of the diffusive term. This turns out to fail; three other methods—second order Adams–Bashforth (AB2) or mid-point Runge–Kutta (RK2) [12], and a method of lines using MATLAB's ODE45 solver—all yielded successful solutions, but the DFF gave solutions unstable at long times.

Defining Δx and Δt as before, the obvious DFF discretiztion of the above equations is

$$\begin{aligned} \frac{S_j^{n+1} - S_j^{n-1}}{2\Delta t} &= -I_j^n S_j^n + b S_j^n (1 - S_j^n), \\ \frac{I_j^{n+1} - I_j^{n-1}}{2\Delta t} &= I_j^n (S_j^n - \lambda) + \frac{I_{j+1}^n + I_{j-1}^n - I_j^{n+1} - I_j^{n-1}}{(\Delta x)^2}. \end{aligned}$$

The discretized equilibrium is the same, $S_j^n = \lambda$, $I_j^n = b(1 - \lambda)$.

These equations can again be rearranged to give explicit formulae updating S_j^{n-1} and I_j^{n-1}, respectively, to S_j^{n+1} and I_j^{n+1}:

$$\begin{aligned} S_j^{n+1} &= S_j^{n-1} + 2\Delta t S_j^n (b(1 - S_j^n) - I_j^n), \\ I_j^{n+1} &= \frac{(I_j^{n-1}(1 - \alpha) + I_j^n (S_j^n - \lambda) + \alpha (I_{j+1}^n + I_{j-1}^n))}{1 + \alpha}, \end{aligned}$$

where, as before, $\alpha = 2\Delta t/(\Delta x)^2$. The first equation can be used at both interior and boundary points, the second only in the interior. The zero derivative boundary conditions at points 1 and N are set after the interior points with

$$I_1^{n+1} = (4I_2^{n+1} - I_3^{n+1})/3, \quad I_N^{n+1} = (4I_{N-1}^{n+1} - I_{N-2}^{n+1})/3,$$

both of which are accurate to order $(\Delta x)^2$. As with all three-level schemes, the first timestep must be carried out with a different method; we used either explicit Euler or a second-order Runge–Kutta. Surprisingly, the choice makes a difference which will be discussed later. We also successfully ran a few cases where RK2 was used throughout for the timestepping. All our programs were written in MATLAB and ran in seconds or minutes.

Both the AB2 and DFF codes were run for a wide range of values of the parameters b and λ, for several sufficiently small values of Δx, and consistent timesteps were

https://doi.org/10.1017/S1446181121000043 Published online by Cambridge University Press

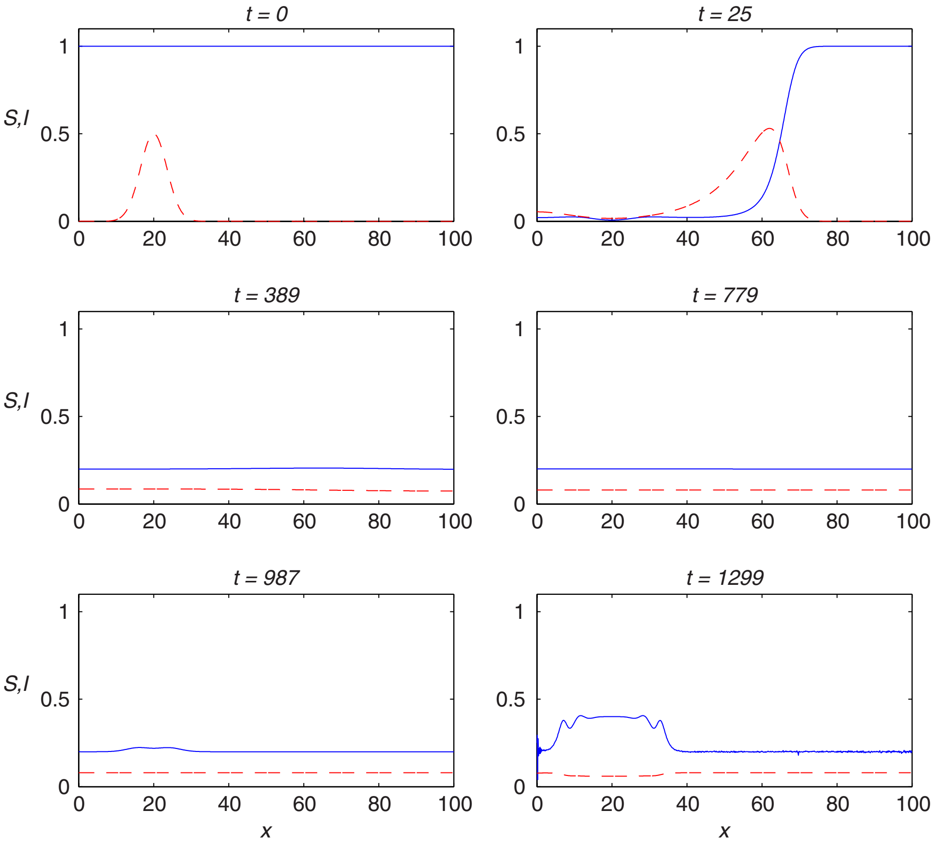

FIGURE 1. Plots of S (the upper curves at $x = 100$) and I (dashed) for $\lambda = 0.2$, $b = 0.1$, $L = 100$, $\Delta x = 0.1$, $\alpha = 0.2$ ($\Delta t = 10^{-3}$). After a stage of propagating waves the solution closely approaches the steady state, $S = 0.2$, $I = 0.08$ about $t = 400$ (see the plots at $t = 389$ and 779). It differs imperceptibly from these values up to about $t = 900$ but by $t = 950$ an instability has started to erupt (see the plot at $t = 987$). The evolution using AB2 is identical except that the instability never appears and the steady state is maintained for all time.

chosen with $\alpha = 0.2$. Typical plots of the solutions at various times are shown in Figure 1, for the values $\lambda = 0.2, b = 0.1, L = 100, \Delta x = 0.1$. The AB2 solution was checked against another code based on MATLAB's ODE45 routine, and we are confident it gives the correct solution, provided a sufficiently small timestep is chosen—a stability analysis for AB2 similar to that described for DFF in the next section shows α must be chosen less than 0.5 for the numerical solution to be stable. The DFF code behaved similarly for a range of other parameter values, and the salient features are listed as follows.

- For the early parts of each run, AB2 and DFF behave indistinguishably, with an initial period of waves sloshing back and forth, followed by an approach to the

https://doi.org/10.1017/S1446181121000043 Published online by Cambridge University Press

correct steady state $S = \lambda$, $I = b(1 - \lambda)$ at later times. The waves are explained by Murray [15].

- At later times, AB2 successfully converges to the steady state. DFF almost gets there, but becomes unstable at very long times.
- Just how long the instability takes to be significant depends on how the initial step Δt is carried out: using an Euler step, it is manifest sooner than with RK2. The solution of the latter case is remarkable in showing an apparent steady state between around $t = 380$ and $t = 980$, after which the instability finally erupts.
- The time until the onset of instability exhibits little or no dependence on the values of Δx and Δt; it does, however, depend critically on the two parameters λ and b from the original problem.
- The instability first arises in the neighbourhood of the location of the initial pulse; even after a very long time the system still remembers the initial condition.

All these facts will find an explanation when the stability of the DFF scheme is investigated in the next section.

3. Why the scheme fails

We can linearize the difference equations in the previous section about their equilibrium solution and study the growth or decay of the Fourier modes by writing

$$S_j^n = \lambda + f_n e^{ijk\Delta x}, \quad I_j^n = b(1 - \lambda) + g_n e^{ijk\Delta x}.$$

Neglecting the quadratic terms yields

$$\begin{aligned} f_{n+1} &= f_{n-1} - 2\lambda\Delta t(g_n + bf_n), \\ g_{n+1} &= \frac{(1-\alpha)g_{n-1} + 2b\Delta t(1-\lambda)f_n + 2\alpha\cos(k\Delta x)g_n}{1+\alpha}. \end{aligned}$$

We can turn this into a system of four first-order homogeneous difference equations by writing $f_{n-1} = A_n$, $g_{n-1} = B_n$,$f_n = C_n$, and $g_n = D_n$:

$$\begin{pmatrix} A_{n+1} \\ B_{n+1} \\ C_{n+1} \\ D_{n+1} \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & -2b\lambda\,\Delta t & -2\lambda\,\Delta t \\ 0 & \dfrac{1-\alpha}{1+\alpha} & \dfrac{2b\,\Delta t(1-\lambda)}{1+\alpha} & \dfrac{2\alpha\cos(k\,\Delta x)}{1+\alpha} \end{pmatrix} \begin{pmatrix} A_n \\ B_n \\ C_n \\ D_n. \end{pmatrix}.$$

This system has solutions proportional to r^n, where r is any one of the eigenvalues of the Jacobian coefficient matrix on the right-hand side. These satisfy the quartic

$$\begin{aligned} &(1+\alpha)r^4 + 2\{\lambda b\Delta t(1+\alpha) - \alpha\cos(k\Delta x)\}r^3 - \{4b(\Delta t)^2\lambda(\lambda-1) + 2 \\ &\quad + 4b\alpha\lambda\Delta t\cos(k\Delta x)\}r^2 + 2(b\Delta t(\alpha-1)\lambda + \alpha\cos(k\Delta x))r + 1 - \alpha = 0. \end{aligned} \tag{3.1}$$

https://doi.org/10.1017/S1446181121000043 Published online by Cambridge University Press

TABLE 1. Eigenvalues r_i (to six decimal places) for selected parameter pairs (α, λ).

(α, λ)	(0.4, 0.1)	(0.4, 0.2)	(0.8, 0.1)	(0.8, 0.2)
r_1	0.999 979	0.999 959	0.999 959	0.999 919
r_2	0.654 653	0.654 653	0.333 333	0.333 333
r_3	−0.654 653	−0.654 653	−0.333 333	−0.333 333
r_4	−1.000 019	−1.000 039	−1.000 039	−1.000 079

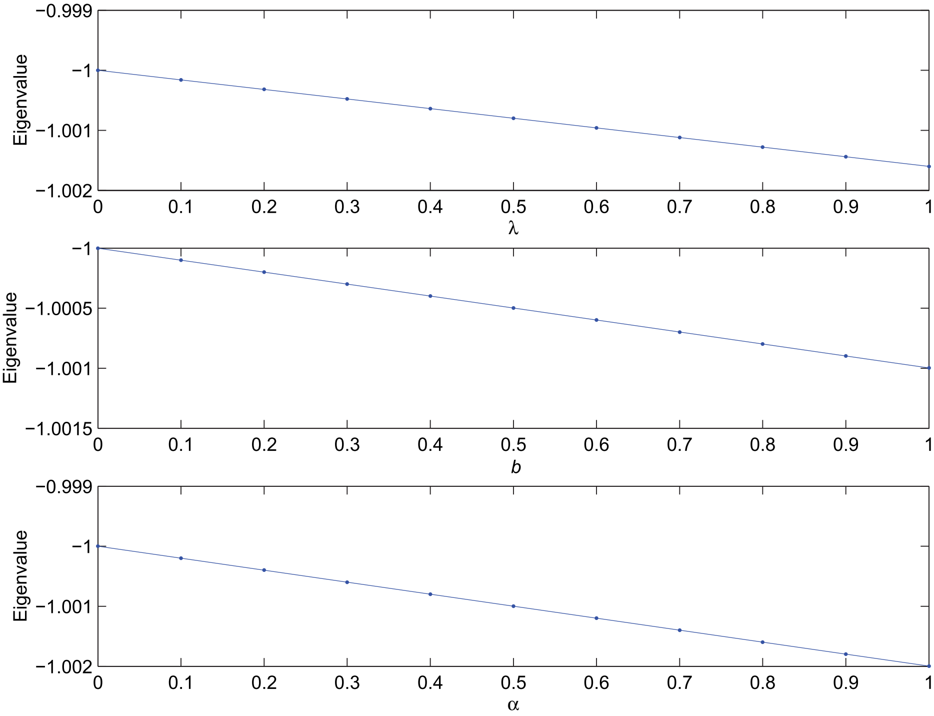

FIGURE 2. Illustration of the linear dependence of the minimum eigenvalue near −1, varying λ (with $b = 0.8$, $\alpha = 0.4$), b (with $\lambda = 0.5$, $\alpha = 0.4$) and $\alpha = 200\Delta t$ (with $\lambda = 0.5$, $b = 0.8$) with $\Delta x = 0.1$.

The formula for the roots of a quartic can be applied to this, but the results are far too complicated to yield any insight. Another possibility is to use the Schur–Cohn test on the coefficients; the formulae are given at the end of [8] (with a typo) but again the results are too involved to be any use. What does work is simply to calculate the eigenvalues numerically with MATLAB in many cases; fortunately, when this is done, a clear pattern emerges. Table 1 shows some typical values.

From numerous similar computations we observe that there are always eigenvalues near ±1, and they are almost insensitive to $\cos(k\Delta x)$, with their differences from ±1 approximately proportional to Δt, b and λ (see Figure 2). The eigenvalue just below −1

https://doi.org/10.1017/S1446181121000043 Published online by Cambridge University Press

provokes the instability and its closeness to -1 is why the latter takes so long to appear.

Armed with these facts, we can algebraically seek Taylor expansions for the roots near ± 1 by writing $r = \pm 1 + \epsilon$, keeping just the constants and the terms linear in ϵ. This can be done by hand and verified with computer algebra. When this is done, the constant term reassuringly vanishes to zero order in Δt and Δx. The eigenvalue near 1 is $1 - b\lambda\Delta t + \cdots$; the one near -1 is $-1 - b\lambda\Delta t + \cdots$. This is enough to explain the facts listed at the end of the previous section.

- A similar analysis for AB2 gives eigenvalues just within the unit circle providing $\alpha < 1/2$, so while DFF fails for any value of α, AB2 is conditionally stable and succeeds.
- The amplitude of the error depends on its initial value. With an Euler first step the error introduced is proportional to $(\Delta t)^2$; with RK2 it is proportional to $(\Delta t)^3$. This explains why the error takes longer to develop with RK2. The Fourier modes of the initial error derive from the nonnegligible parts of the initial condition; these are what are amplified, explaining the fact that the system remembers them.
- For $\lambda > 0$, the eigenvalue near -1 is the cause of the instability. The value -1 indicates that the instability should flip sign between odd and even steps of length Δt; this prediction is borne out by the numerical results. At each timestep any error is amplified by an approximate factor $-1 - b\lambda\Delta t$; after a total integration time $T = n\Delta t$ the amplification factor is $(-1 - b\lambda T/n)^n$. As $n \to \infty$, this tends to $(-1)^n e^{b\lambda T}$, explaining why the growth time is almost independent of the timestep. In fact, with an Euler (or RK2) first step, the time for an initial error of order $(\Delta t)^2$ (or $(\Delta t)^3$) to become of order 1 is estimated by $-2\log(\Delta t)/(\lambda b)$ (or $-3\log(\Delta t)/(\lambda b)$). So an RK2 first step means that the integration appears correct for 1.5 times longer, and there is a weak logarithmic dependence on Δt. This is approximately what is observed computationally.

We will be using the above Taylor expansion procedure repeatedly, for solutions of equation (3.1) and later for three other cases in Section 5, and some remarks are in order. One can formally set Δt to zero in equation (3.1) with α nonzero, which drastically simplifies the Jacobian matrix. Writing $C = \cos(k\Delta x)$, the eigenvalues r then satisfy a quartic which factorizes as

$$(r^2 - 1)\{(1 + \alpha)r^2 - 2\alpha C r - (1 - \alpha)\} = 0.$$

The two roots ± 1 are consistent with what was found above for nonzero Δt. The other two roots $\{\alpha C \pm \sqrt{\alpha^2(C^2 - 1) + 1}\}/(1 + \alpha)$ were treated in equation (1.1): if $|C| < 1$ these roots are stable with modulus less than unity; if $C = 1$ the roots are $1, (\alpha - 1)/(\alpha + 1)$, and if $C = -1$ they are $-1, -(\alpha - 1)/(\alpha + 1)$. This differs from the heat equation case in (1.1) since ± 1 are double roots in (3.1) if $C = \pm 1$, changing the forms of the expansions for nonzero Δt.

https://doi.org/10.1017/S1446181121000043 Published online by Cambridge University Press

We now expand the solutions near ±1 in powers of Δt. We know that since the coefficients of the quartic are analytic functions of the parameters, in particular of Δt, the resulting Taylor series are themselves analytic in a neighbourhood of $\Delta t = 0$ (see, for example, [9, page 82 Ch. II Theorem 2.3]). This means our expansion procedure is valid. Thus we expand $r = \pm 1 + a_1\Delta t + a_2(\Delta t)^2 + \cdots$, substitute this into the full quartic, and calculate the coefficient of Δt on the left-hand side. For the -1 expansion, if this coefficient is negative, the associated eigenvalue gives instability; if it is positive there is stability. (In general, there is one unique value, and the coefficient is real.) The reverse is true for the $+1$ expansion. This gives the results referred to earlier.

For the special cases $C = \pm 1$, the coefficient of Δt in the expansion of the quartic vanishes, because of the double root when $\Delta t = 0$. The expansion must then be carried out to order $(\Delta t)^2$; this shows that a_1 satisfies a quadratic with two possible values, with a_2 appearing with a factor $(C \pm 1)$ and thus making no contribution at this order. This has no consequences for the standard case discussed here, because the $C \neq \pm 1$ modes are all unstable, but it affects things interestingly for Variant 3 in Section 5.

4. A simpler example

The above instability arises, at least to first order in Δt, from the death rate term in the equation for I. This suggests that the following problem may give a distillate of what is happening. Consider the modified heat equation

$$\frac{\partial u}{\partial t} = -\lambda u + \frac{\partial^2 u}{\partial x^2}$$

with a Gaussian initial condition and zero-derivative boundary conditions. Clearly $u \to 0$ as $t \to \infty$ when $\lambda > 0$.

Naive DFF gives the difference scheme

$$(1+\alpha)u_j^{n+1} = (1-\alpha)u_j^{n-1} - 2\Delta t\lambda u_j^n + \alpha(u_{j+1}^n + u_{j-1}^n). \tag{4.1}$$

A stability analysis of the zero solution proceeds as for the ordinary heat equation; the timestep dependence is proportional to r^n, where

$$r^2(1+\alpha) + 2[\lambda\Delta t - \alpha\cos(k\Delta x)]r - (1-\alpha) = 0.$$

Again using the Schur–Cohn test (see Section 1) for the quadratic $a_2r^2 + a_1r + a_0 = 0$, the condition $|a_2 + a_1| > |a_0|$ for stability gives

$$|\alpha| > |\lambda\Delta t - \alpha\cos(k\Delta x)|.$$

This condition is violated when $k\Delta x$ is sufficiently close to an odd multiple of π. For the special case $\alpha = 1$ the roots of the quadratic are $r = 0$ (superstable) and $r = \cos(k\Delta x) - \lambda\Delta t$, so modes $k = \pi/\Delta x, k = 3\pi/\Delta x, \ldots$ are all unstable with growth factor $-1 - \lambda\Delta t + \cdots$. Thus for the modified heat equation DFF is predicted to fail for very large times as in the rabies problem (2.1) and (2.2), with a growing perturbation that

https://doi.org/10.1017/S1446181121000043 Published online by Cambridge University Press

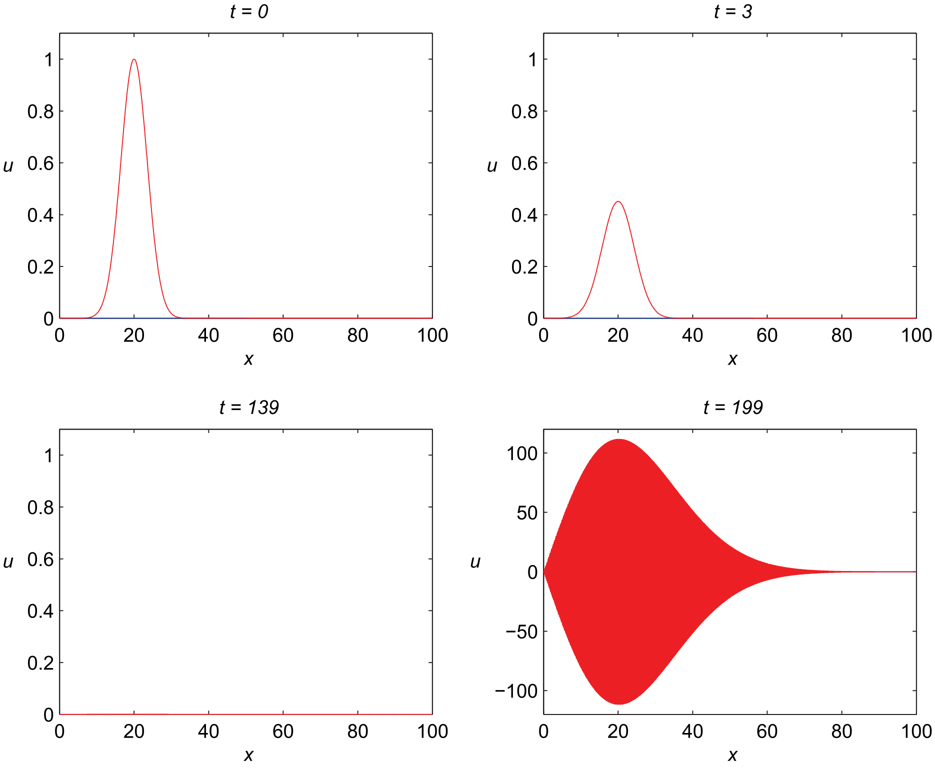

FIGURE 3. Plots of u for $\lambda = 0.2$, $\Delta x = 0.1$, $\alpha = 1$ ($\Delta t = 5 \times 10^{-3}$). The solution closely approaches the steady state, $u = 0$. Between $t = 30$ and $t = 140$, u differs imperceptibly from 0, but by $t = 150$ an instability has started to erupt. The solid appearance of the solution in the final plot is due to the high wave numbers of the most unstable modes, and the negative values arise because the instability is associated with an eigenvalue just below -1.

flips sign between odd and even timesteps. This behaviour is shown by the numerical solution in Figure 3, which uses $\alpha = 1$ in (4.1), obviating the need for a special starting step.

An obvious possible cure is to extend the DFF averaging to the death rate term. The algorithm then becomes

$$(1 + \alpha + \lambda\Delta t)u_j^{n+1} = (1 - \alpha - \lambda\Delta t)u_j^{n-1} + \alpha(u_{j+1}^n + u_{j-1}^n),$$

and the special choice removing the u_j^{n-1} term is $\alpha = 1 - \lambda\Delta t$, that is, $\Delta t = \Delta x^2/(2 + \lambda(\Delta x)^2)$. When implemented, this scheme is stable, and in fact the stability eigenvalues are then $r = 0$ and $r = \cos(k\Delta x)$. This motivates us always to use averaging in both the death rate and diffusive terms in the algorithms to be discussed in the next section.

Of course, if an integrating factor $e^{\lambda t}$ is used, this problem reduces to the ordinary heat equation, for which DFF then gives the correct solution!

https://doi.org/10.1017/S1446181121000043 Published online by Cambridge University Press

5. Attempts to fix the rabies problem

There are several ways to introduce extra DFF averaging into the rabies equations with the goal of finding a stable algorithm. Guided by the results in the previous section, we restrict ourselves to the case where the death rate term is also averaged. There are then three possibilities.

5.1. Variant 1 We do no further averaging in the nonlinear terms. The resulting difference equations are then

$$S_j^{n+1} = S_j^{n-1} + 2\Delta t S_j^n(b(1 - S_j^n) - I_j^n),$$
$$I_j^{n+1} = \frac{I_j^{n-1}(1 - \alpha - \lambda\Delta t) + I_j^n S_j^n + \alpha(I_{j+1}^n + I_{j-1}^n)}{1 + \alpha + \lambda\Delta t}.$$

These can be rearranged as before to give explicit updating formulae, which turn out to be always unstable.

We can analyse the stability exactly as for the standard case; this time the four homogeneous difference equations have a right-hand-side matrix

$$M = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & -2b\lambda\,\Delta t & -2\lambda\,\Delta t \\ 0 & \dfrac{1 - \alpha - \lambda\Delta t}{1 + \alpha + \lambda\Delta t} & \dfrac{2b\,\Delta t(1 - \lambda)}{1 + \alpha + \lambda\Delta t} & \dfrac{2\alpha\cos(k\,\Delta x) + 2\lambda\Delta t}{1 + \alpha + \lambda\Delta t} \end{pmatrix}.$$

Again this has two eigenvalues close to 1 and -1; Taylor series expansions of the roots of the resulting quartic about these points start with $r = 1 - b\lambda\Delta t$ and $r = -1 - b\lambda\Delta t$, just as before.

5.2. Variant 2 We average in the death rate term and in addition we average I but not S in the nonlinear term. This replaces IS in both equations with $(I_j^{n+1} + I_j^{n-1})S_j^n/2$, so that there is still conservative transfer between the two equations. Provided we store I_j^{n-1} and update I_j^{n-1} to I_j^{n+1} first, an explicit rearrangement is still possible, giving

$$\widetilde{I}_j^{n-1} = I_j^{n-1},$$
$$I_j^{n+1} = \frac{I_j^{n-1}(1 - \alpha - \lambda\Delta t + \Delta t S_j^n) + \alpha(I_{j+1}^n + I_{j-1}^n)}{1 + \alpha + \lambda\Delta t - \Delta t S_j^n},$$
$$S_j^{n+1} = S_j^{n-1} - \Delta t(2S_j^n(b(1 - S_j^n) - I_j^{n+1} - \widetilde{I}_j^{n-1}).$$

Application of this algorithm turns out to be unstable again.

https://doi.org/10.1017/S1446181121000043 Published online by Cambridge University Press

Analysing the stability is slightly more complicated than before as an additional elimination stage is necessary. The resulting right-hand-side matrix is

$$M = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & -\dfrac{2\Delta t\lambda}{1+\alpha} & -\dfrac{2b\lambda\,\Delta t(1+\alpha+\Delta t(1-\lambda))}{1+\alpha} & -\dfrac{2\lambda\,\Delta t\,\alpha\cos(k\Delta x)}{1+\alpha} \\ 0 & \dfrac{1-\alpha}{1+\alpha} & \dfrac{2b\,\Delta t(1-\lambda)}{1+\alpha} & -\dfrac{2\alpha\cos(k\,\Delta x)}{1+\alpha} \end{pmatrix}.$$

Yet again, a stability analysis gives two eigenvalues close to 1 and −1; Taylor series expansions around these points give $r = 1 - b\lambda\Delta t$ and $r = -1 - b\lambda\Delta t$, as before.

5.3. Variant 3 We average in the death rate term but this time in addition we average S rather than I in the nonlinear term. This replaces IS in both equations with $(S_j^{n+1} + S_j^{n-1})I_j^n/2$, again giving conservative transfer between the two equations. Provided we store S_j^{n-1} and update S_j^{n-1} to S_j^{n+1} first, an explicit rearrangement is still possible, giving

$$\begin{aligned} \widetilde{S}_j^{n-1} &= S_j^{n-1}, \\ S_j^{n+1} &= S_j^{n-1} - \Delta t((S_j^{n+1} + S_j^{n-1})I_j^n - 2bS_j^n(1 - S_j^n), \\ I_j^{n+1} &= \frac{I_j^{n-1}(1 - \alpha + \Delta t(\widetilde{S}_j^{n-1} - \lambda)) + \alpha(I_{j+1}^n + I_{j-1}^n)}{1 + \alpha + \Delta t(S_j^{n+1} - \lambda)}. \end{aligned}$$

This algorithm was coded up and run for many parameter values. It was found to give the correct stable solution in some but not all cases.

The stability of the scheme can be investigated using the same methods and notation as before; however, the analysis again requires an extra elimination stage, and this time the algebra is yet more complicated. We have

$$M = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \dfrac{1-(1-\lambda)b\Delta t}{1+(1-\lambda)b\Delta t} & 0 & \dfrac{2b(1-2\lambda)\Delta t}{1+(1-\lambda)b\Delta t} & -\dfrac{2\lambda\,\Delta t}{1+(1-\lambda)b\Delta t} \\ \dfrac{2b(1-\lambda)\Delta t}{D} & \dfrac{1-\alpha-\lambda\Delta t}{1+\alpha+\lambda\Delta t} & \dfrac{N_1}{D} & \dfrac{N_2}{D} \end{pmatrix}, \tag{5.1}$$

where

$$\begin{aligned} N_1 &= 2b^2(\Delta t)^2(1-\lambda)(1-2\lambda), \\ N_2 &= 2[\alpha\cos(k\,\Delta x)(1+(1-\lambda)b\Delta t) + \lambda\Delta t], \\ D &= (1+(1-\lambda)b\Delta t)(1+\alpha+\lambda\Delta t). \end{aligned}$$

The eigenvalues determining the stability can again be evaluated in particular cases using MATLAB or Mathematica; for example, for $\Delta t = 0.002$, $\Delta x = 0.1$, $\lambda = 0.2$,

$b = 0.1$, and $\cos(k\Delta x) = 1$, their moduli are 0.999 720, 0.999 980, 0.999 980, 0.428 163 (the middle pair are complex conjugates). So this case is stable.

Two of the eigenvalues are again very close to ±1 so we can expand the quartic about these points in powers of Δt. Setting $r = \pm 1 + a_1\Delta t + a_2(\Delta t)^2 + \cdots$, we find approximate solutions $r = 1 + 4(C-1)\alpha b\lambda\Delta t$ (always less than or equal to 1), and $r = -1 + 4(1+C)\alpha b(2-3\lambda)\Delta t$. This predicts the algorithm should be stable or unstable according as $\lambda < 2/3$ or $\lambda > 2/3$, with the root near -1 producing the instability. However, this is not confirmed by actual calculations. From the MATLAB eigenvalue calculator, we find that in fact the critical value is exactly 0.6. What is going on here?

The answer is subtle: note that for the special modes where $C = \pm 1$, the order-Δt correction to the equation for the eigenvalues near ±1 of the matrix M in (5.1) vanishes. For these cases we actually need to expand the equation to order $(\Delta t)^2$ to get the corrections. Because $r = \pm 1$ are double roots of the $\Delta t = 0$ quartic, the coefficient a_1 satisfies a real-coefficient quadratic with one solution for each of the finite Δt branches bifurcating from this double root. The quadratic is derived using Mathematica, and the product of its roots is found to be $b(3-5\lambda)\lambda$ (this is for the case near $r = -1$, which becomes unstable for lowest λ; the root near $r = +1$ has a factor $1-\lambda$ replacing $3-5\lambda$). When $\lambda > 3/5$ both roots are real and one of them has to be negative, so the corresponding r is less than -1 and there is instability.

We thus have the strange situation that there is an isolated discrete spectrum of k values giving modes more unstable than the case for general k. In the range $3/5 < \lambda < 2/3$, the growth factor per Δt timestep is $-1 - |a_1|\Delta t$. For a total time $T = N\Delta t$ the overall factor becomes $(-1)^N(1+|a_1|\Delta t)^{T/\Delta t}$, which tends to $(-1)^N e^{|a_1|T\Delta t}$ for large N. The growth of catastrophic errors is again independent of the timestep, but as one approaches $\lambda = 3/5$ from above, its appearance should take longer and longer.

When the finite-difference code for Variant 3 is run, we find there is yet another twist to the story. When $\lambda < 3/5$, the steady solution is always stable. When $\lambda > 2/3$ it is always unstable. But for intermediate values of λ there can be solutions whose stability depends on the amplitude of the initial Gaussian pulse given to the infectives I.

Table 2 shows the stability of integrations to $t = 9000$ for a range of λ with $\alpha = 0.2$ and the initial amplitude of the infectives 0.5. Instability always manifested itself at about $t = 3000$ and thereafter sloshed around at large but finite amplitude; the stable cases showed no sign of instability up to $t = 9000$. The first unstable case, $\lambda = 0.632\,395\,019\,531\,25$, became stable when the initial amplitude of the infectives was reduced to 0.25. The last stable case, $\lambda = 0.632\,387\,695\,312\,5$, became unstable when the timestep was doubled by setting $\alpha = 0.4$. In the range $3/5 < \lambda < 2/3$ stability is determined nonlinearly by a variety of factors and a complete understanding is beyond the scope of this paper.

While it is interesting to have found a variant that works under certain circumstances, it is unsatisfactory in that we have no obvious reason why this option should be the one that works, beyond saying that it is the only variant that introduces some averaging for S as well as I. At this stage, it is worth adding that, for the standard

https://doi.org/10.1017/S1446181121000043 Published online by Cambridge University Press

TABLE 2. Stability for varying λ for $\alpha = 0.2$, $b = 0.1$ and the initial amplitude of the infectives set to 0.5.

λ	
0.631 25	stable
0.632 187 5	stable
0.632 343 75	stable
0.632 373 046 875	stable
0.632 387 695 312 5	stable
0.632 395 019 531 25	unstable
0.632 402 343 75	unstable
0.632 421 875	unstable
0.6325	unstable
0.635	unstable
0.637	unstable
0.64	unstable
0.66	unstable
0.68	unstable

case, including a small DFF-averaged diffusion term in the first equation (with extra zero-derivative boundary conditions on S) was tried but was still unstable.

6. Conclusion

We have discovered a straightforward problem which breaks DFF in a disconcerting way; the scheme gives every indication that it is working until it has been run for a very long time. Our analysis has shown that the instability is inherent in the equations themselves, and not the choice of boundary conditions. Taylor [19] discussed how Robin boundary conditions can destabilize the standard DuFort–Frankel treatment of the heat equation, and gave a method to fix the problem. Fornberg [3] gave a treatment of leapfrog instabilities in general, without specific mention of DFF. We have been unable to find other references where the algorithm fails, with the following exception.

Zheligovsky and Galloway [21] attempted to solve a problem in three-dimensional magnetohydrodynamic dynamo theory using an expansion in hexagonally symmetric Fourier modes in the horizontal. Each Fourier mode was evolved with DuFort–Frankel finite-differencing in the vertical direction, giving a large system of heat-like equations with additional advective terms. The context and details are explained fully in the paper; the diffusion coefficient in this case is magnetic diffusivity η. Only when η is tiny is a dynamo possible, so the object was to compute with η as small as feasible. Two versions of the code were produced: one with zero or zero-derivative boundary conditions for all three magnetic field components, and one where the various modes of the z-component satisfied Robin boundary conditions at the top boundary. The first of these worked for all cases considered, but the second (which was the one needed for

https://doi.org/10.1017/S1446181121000043 Published online by Cambridge University Press

the problem being addressed) only gave apparently converged solutions for η values larger than 0.005. Smaller η became unstable at the long times necessary to reach a state of steady magnetic field growth, with the instability flipping sign from one timestep to the next. This is indicative of an eigenvalue falling below -1 by a very small amount just as for the rabies problem, though in this case a detailed stability analysis would be impossible. The mechanism is likely to be that described by Taylor [19], since problems only arose when using the Robin boundary conditions. In the end, this dynamo problem had to be solved by a much less efficient iteration method involving large matrices. In view of the experience, here it might be worth revisiting it with Adams–Bashforth timestepping, which is only slightly more expensive than DFF.

Given that there is a long history of successful use of DFF, it is inappropriate to counsel against using it; rather, anyone applying the method should be aware of what can occasionally happen and proceed with caution. Some of its earlier triumphs might have become unstable if the apparent steady states had been continued for much longer, but there is little doubt that those steady states were largely correct. The pernicious feature of the instability described here is that it takes a remarkably long time to develop. To paraphrase Kramer [11], when DuFort–Frankel is good, it is very very good, but when it is bad, it is horrid!

References

[1] W. F. Ames, "Numerical methods for partial differential equations", *Computer science and scientific computing*, 3rd edn (Academic Press, Boston, MA, 1992); ISBN: 0-12-056761-X.

[2] E. C. Du Fort and S. P. Frankel, "Stability conditions in the numerical treatment of parabolic differential equations", *Math. Tables Other Aids Comput.* **7** (1953) 135–152; doi:10.2307/2002754.

[3] B. Fornberg, "On the instability of leap-frog and Crank–Nicolson approximations of a nonlinear partial differential equation", *Math. Comp.* **27** (1973) 45–57; doi:10.2307/2005246.

[4] D. J. Galloway and D. R. Moore, "Axisymmetric convection in the presence of a magnetic field", *Geophys. Astrophys. Fluid Dyn.* **12** (1979) 73–106; doi:10.1080/03091927908242678.

[5] D. J. Galloway and N. O. Weiss, "Convection and magnetic fields in stars", *Astrophys. J.* **243** (1981) 945–953; doi:10.1086/158659.

[6] R. Henery, L. Farnell, W. G. Gibson and M. R. Bennett, "Potential fields in vascular smooth muscle generated by transmitter release from sympathetic varicosities", *J. Theoret. Biol.* **218** (2002) 531–548; doi:10.1016/S0022-5193(02)93098-5.

[7] C. A. Jones and D. J. Galloway, "Axisymmetric magnetoconvection in a twisted field", *J. Fluid Mech.* **253** (1993) 297–326; doi:10.1017/S0022112093001806.

[8] E. I. Jury and B. H. Bharucha, "Notes on the stability criterion for linear discrete systems", *IRE Trans. Automat. Control* **6** (1961) 88–90; doi:10.1109/TAC.1961.6429319.

[9] T. Kato, *Perturbation theory for linear operators*, 2nd edn (Springer-Verlag, Berlin, 1976); doi:10.1007/978-3-642-66282-9.

[10] J. G. Kirk and D. J. Galloway, "The evolution of a test particle distribution in a strongly magnetised plasma", *Plasma Phys.* **24** (1982) 339–359; doi:10.1088/0032-1028/24/8/512.

[11] S. Kramer, "There was a little girl: its first printings–its authorship–its variants", *Pap. Bibliogr. Soc. Am.* **40** (1946) 287–310; http://www.jstor.org/stable/24298820.

[12] J. D. Lambert, *Numerical methods for ordinary differential systems: The initial value problem* (John Wiley & Sons, Chichester, 1991); ISBN: 978-0-471-92990-1.

https://doi.org/10.1017/S1446181121000043 Published online by Cambridge University Press

[13] D. R. Moore and N. O. Weiss, “2-dimensional Rayleigh–Benard convection”, *J. Fluid Mech.* **58** (1973) 289–312; doi:10.1017/S0022112073002600.

[14] K. W. Morton and D. F. Mayer, *Numerical solution of partial differential equations* (Cambridge University Press, Cambridge, 2005); ISBN: 9780521607933.

[15] J. D. Murray, *Mathematical biology. II: Spatial models and biomedical applications*, 3rd edn, Volume 18 of *Interdiscip. Appl. Math. Ser.* (Springer-Verlag, New York, 2003); ISBN: 978-0-387-95228-4.

[16] J. Noye, “Finite difference techniques for partial differential equations”, in: *Computational techniques for differential equations*, Volume 83 of *North-Holland Math. Stud.* (Elsevier, North-Holland, Amsterdam, 1984) 95–354; doi:10.1016/S0304-0208(08)71201-5.

[17] P. J. Roache, *Computational fluid dynamics* (Hermosa Publishers, Albuquerque, NM, 1976), with an appendix (“On artificial viscosity”) reprinted from *J. Comput. Phys.* **10** (1972) 169–184, Revised printing.

[18] K. V. Roberts and N. O. Weiss, “Convective difference schemes”, *Math. Comp.* **20** (1966) 272–299; doi:10.2307/2003507.

[19] P. J. Taylor, “The stability of the Du Fort–Frankel method for the diffusion equation with boundary conditions involving space derivatives”, *Comput. J.* **13** (1970) 92–97; doi:10.1093/comjnl/13.1.92.

[20] N. O. Weiss, “The expulsion of magnetic flux by eddies”, *Proc. R. Soc. Lond. Ser. A* **293** (1966) 310–328; http://adsabs.harvard.edu/abs/1966RSPSA.293.310W.

[21] V. A. Zheligovsky and D. J. Galloway, “Dynamo action in Christopherson hexagonal flow”, *Geophys. Astrophys. Fluid Dyn.* **88** (1998) 277–293; doi:10.1080/03091929808245477.

https://doi.org/10.1017/S1446181121000043 Published online by Cambridge University Press