

## Solutions

### to Worksheet: Stable time-steps for 2nd-order ODE schemes

(a) I computed stability functions for the first two methods:

$$R_{\text{TR}}(z) = \frac{1 + z/2}{1 - z/2},$$

$$R_{\text{EM}}(z) = 1 + z + \frac{z^2}{2}$$

The stability regions  $S = \{z \in \mathbb{C} : |R(z)| \leq 1\}$  have been plotted before, and I put them together together with AB2 below.

(b) AB2 is a multistep method. If we write it as

$$U^{n+2} - U^{n+1} = z \left( \frac{3}{2}f(U^{n+1}) - \frac{1}{2}f(U^n) \right)$$

then  $\rho(\zeta) = \zeta^2 - \zeta$  and  $\sigma(\zeta) = \frac{3}{2}\zeta - \frac{1}{2}$ . By formula (7.5),

$$\pi(\zeta; z) = \rho(\zeta) - z\sigma(\zeta) = \zeta^2 - \zeta - z \left( \frac{3}{2}\zeta - \frac{1}{2} \right) = \zeta^2 - \left(1 + \frac{3}{2}z\right)\zeta + \frac{1}{2}z$$

Now, the stability region  $S$  is the set of  $z \in \mathbb{C}$  so that  $\pi(\zeta; z)$  satisfies root condition (6.34), that is, so that either all roots  $\zeta$  are inside the unit circle, or if there are roots on the boundary then these roots are simple. Since  $\pi(\zeta; z)$  is quadratic, we can compute its roots:

$$\zeta = \frac{1 + \frac{3}{2}z \pm \sqrt{(1 + \frac{3}{2}z)^2 - 2z}}{2}$$

For a figure it suffices to check numerically that both roots have magnitudes less than 1. Alternatively, the boundary locus method of section 7.6.1 can be used; I did both. In fact, the following code generates the figure showing all three stability regions.

*regions.m*

```
% REGIONS Plot stability regions for three second-order ODE IVP schemes
% This version uses imagesc instead of contour.

% set up
N = 300;
x = linspace(-5, 5, N+1); y = x;
[xx,yy] = meshgrid(x, y);
zz = xx + i * yy;

% trapezoid
uu = ((abs(1 + zz/2) ./ abs(1 - zz/2)) <= 1);
subplot(1,3,1), imagesc(x, y, 1 - uu, [-1, 1]), colormap('gray')
grid on, axis square, title('TR')

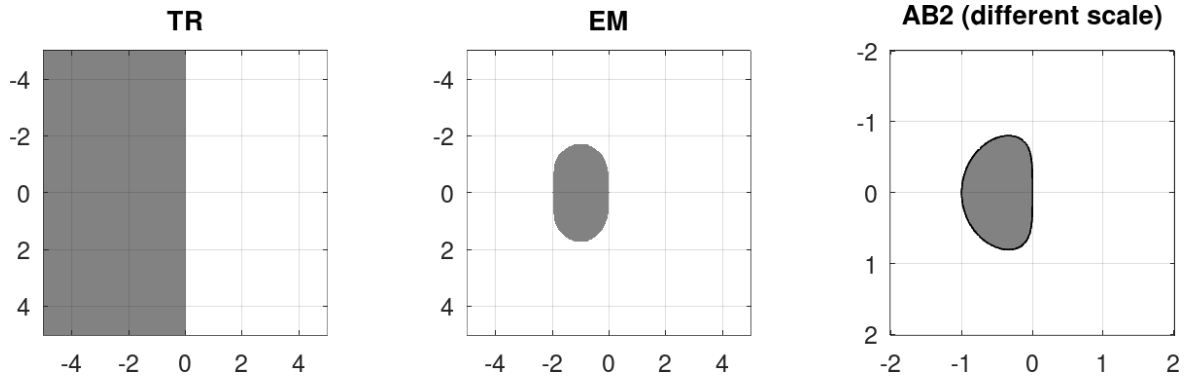
% EM
uu = (abs(1 + zz + zz.^2/2) <= 1);
subplot(1,3,2), imagesc(x, y, 1 - uu, [-1, 1]), colormap('gray')
grid on, axis square, title('EM')

% AB2, both by pointwise decision and boundary locus
% note different scale
N = 200;
```

```

x = linspace(-2, 2, N+1); y = x;
[xx,yy] = meshgrid(x, y);
zz = xx + i * yy;
uu = zeros(size(zz));
for j = 1:N+1
    for k = 1:N+1
        z = zz(j, k);
        zetas = roots([1, -(1+1.5*z), 0.5*z]);
        if abs(zetas(1)) < 1 && abs(zetas(2)) < 1
            uu(j, k) = 1.0;
        end
    end
end
subplot(1,3,3), imagesc(x, y, 1 - uu, [-1, 1]), colormap('gray')
th = 0:pi/100:2*pi;
zeta = exp(i * th);
zbl = (zeta.^2 - zeta) ./ (1.5 * zeta - 0.5);
hold on, plot(real(zbl), imag(zbl), 'k', 'linewidth', 1.0), hold off
grid on, axis square, title('AB2 (different scale)')

```



(c) Note that the  $S_1$  and  $S_3$  matrices are symmetric, so the eigenvalues are real in those cases. The following code finds the eigenvalues. Note that  $S_1$  has two negative and one positive eigenvalue, both eigenvalues in  $S_2$  are purely imaginary, and  $S_3$  has only negative eigenvalues.

*alleigens.m*

```

% ALLEIGENS Compute eigenvalues of the three problems S1,S2,S3

A = [2  2  2;
     2  1  1;
     2  1 -5];
fprintf('S1:\n'), disp(eig(A)')

A = [ 0  2;
     -2  0];
fprintf('S2:\n'), disp(eig(A)')

m = 25; h = 1.0 / (m+1);
A = (1/h^2) * spdiags([ones(m,1), -2*ones(m,1), ones(m,1)], [-1, 0, 1], m,m);
lam = eig(A);
fprintf('S3:\n ')
for j = 1:25, fprintf('%.5f ', lam(j)), end
fprintf('\n')

```

Here is the result:

---

```
>> alleigens
S1:
  -5.56551  -0.52687  4.09237
S2:
  0 - 2i    0 + 2i
S3:
-2694.14240 -2664.71334 -2616.14196 -2549.13655 -2464.67419 -2363.98653
-2248.54183 -2120.02354 -1980.30573 -1831.42581 -1675.55478 -1514.96559
-1352.00000 -1189.03441 -1028.44522 -872.57419 -723.69427 -583.97646
-455.45817 -340.01347 -239.32581 -154.86345 -87.85804 -39.28666
-9.85760
```

---

(d) The main idea is that

*For those eigenvalues  $\lambda_p$  that have negative real parts, corresponding to decaying components of an exact solution, absolute stability*

$$|U^{n+1}| \leq |U^n| \iff z \in S = \{z \in \mathbb{C} : |R(z)| \leq 1\}$$

*is a useful requirement for a reasonable time-step. We find the largest  $k_{stab}$  so that*

$$z = k_{stab} \lambda_p$$

*is in the region of absolute stability of the scheme for all  $\lambda_p$  such that  $\text{Re}(\lambda_p) < 0$ .*

*We allow only time steps  $k > 0$  such that  $k \leq k_{stab}$ .*

For the negative eigenvalues in problem S1, absolute stability is relevant. Since S2 has only purely-imaginary eigenvalues, absolute stability has nothing to say about it, regardless of the method. For S3, all eigenvalues are negative so the stability restrictions for EM and AB2 are severe, with the latter worst. For mere absolute stability,<sup>1</sup> trapezoid has no stability restriction on any of these methods.

Quantitatively, the  $[-2, 0] \subset S_{EM}$  and  $[-1, 0] \subset S_{AB2}$  give the largest portions of the real axis which is inside the stability region of the corresponding scheme, and this allows us to compute  $k_{stab}$  for that scheme on systems S1 and S3. I put my results in a table:

	TR	EM	AB2
S1	*	$k_{stab} = 0.360$	$k_{stab} = 0.180$
S2	†	†	†
S3	*	$k_{stab} = 0.000742$	$k_{stab} = 0.000371$

Certain cases need explanation:

- \*: The implicit TR method has *no stability restriction* because decaying eigenvalues give decaying approximate solutions, and growing give growing.
- †: Since all eigenvalues of S2 have zero real parts, there are *no stability restrictions*. Only  $\lambda$  with  $\text{Re}(\lambda) < 0$  are relevant to absolute stability.

(e) My version of the expert advice:

---

<sup>1</sup>L-stability could be considered. Especially for S3 it suggests trying a stiff decay scheme like BDF2 instead.

- S1: I would recommend using EM subject to the given stability restriction. One would presumably choose time steps much smaller than 0.360 anyway, for accuracy. The reason to choose this method is ease of implementation and low computational cost, as it is an explicit method. It is easier to implement than AB2, because EM is one-step and no start-up fiddles are required, and it is a bit more stable anyway.
- S2: Any of the methods are probably o.k. for any step size which is small enough to give decent accuracy. But note that the eigenvalues  $\lambda = \pm 2i$  correspond to exact solution components which are purely-oscillating, with neither decay nor growth. Because these eigenvalues correspond to points  $z$  which are on the exact boundary of its stability region, method TR has the interesting property that it generates approximate solutions which also neither decay nor grow. I would choose TR given that it is not too hard to implement on a linear problem like this. *(An observation like this explains why TR is standard for the Schödinger equation: it preserves unitarity.)*
- S3: I would choose TR because the stability restrictions for EM and AB2 are severe, and might stop us from doing long time steps. *(That is, when a long time step would be acceptable accuracy-wise.)* Actually a scheme not tested here, namely BDF2, is a good one to recommend. We will see that for smooth initial data both TR and BDF2 work well on discretized heat equations for essentially any time step. But if the initial condition is rough then for large time steps TR will have worse-looking results because it does not kill off the high frequencies as it should; it is common to choose BDF2 for problems like the heat equation with rough data.