# 1

**IBVP 1:** The PDE is $u_t + u_x = 0$, which is the advection equation with positive wave speed. This means that waves travel *to the right*. We need a boundary condition on the inflow, which is the left boundary. We know that we need exactly one BC because the PDE is scalar and first order in space. However, IBVP one includes the condition $u(1,t) = 1$, which is a BC on the right boundary. This IBVP is **ill-posed**.

(If you are unsure which boundary the BC should be imposed on, you can always use the energy method to determine which boundary could contribute to energy growth and therefore requires a BC.)

**IBVP 2:** This is the advection equation with forcing function $F(x,t) = \sin(x)\cos(t)$. We know that the forcing function does not affect well-posedness, so we may pretend like $F(x,t) = 0$. We should have one BC at the left boundary (see discussion for IBVP 1) and one IC, which is precisely what we have. This problem is **well-posed**.

**IBVP 3:** This is the **backwards** heat equation (wrong sign in front of $u_{xx}$), which we know is **ill-posed**. We do not even need to think about the BCs or the IC.

In case you do not remember that this is the backward heat equation, you can also prove this using the energy method or Fourier analysis. The energy method leads to (assuming that $u$ is real)

$$\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\|u\|^2 = -uu_x|_0^1 + \|u_x\|^2.$$

There is no way to bound the growth rate. The term $\|u_x\|^2$ is the dissipation term that we are used to from the heat equation, but with the wrong sign.

Fourier analysis (assuming periodic BCs) yields

$$\frac{\mathrm{d}\hat{u}_k}{\mathrm{d}t} = k^2\hat{u}_k \quad \Longleftrightarrow \quad \hat{u}_k(t) = \hat{u}_k(0)e^{k^2t}.$$

For well-posedness we can tolerate growth of the form

$$\hat{u}_k(t) = Ke^{\alpha t}\hat{u}_k(0)$$

for some *constants* $\alpha$ and $K$, but since the wavenumber $k$ is unbounded, our growth rate can not be bounded in this way.

# 2

This is the wave equation with constant wave speed, a Dirichlet condition on the left boundary, and the condition $c^2u_x = -\alpha u_t$ on the right boundary. The energy method

yields:

$$\underbrace{(u_t, u_{tt})}_{=\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\|u_t\|^2} = c^2(u_t, u_{xx}) = [IBP] = c^2 u_t u_x|_0^L - c^2 \underbrace{(u_{xt}, u_x)}_{=\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\|u_x\|^2}.$$

We get the energy rate

$$\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\left(\|u_t\|^2 + c^2\|u_x\|^2\right) = c^2 u_t(L,t) \underbrace{u_x(L,t)}_{=-\alpha c^{-2} u_t(L,t)} - c^2 \underbrace{u_t(0,t)}_{=0} u_x(0,t).$$

It follows from the Dirichlet condition $u(0,t) = 0$ that $u_t(0,t) = 0$ (since $u(0,t)$ is constant in time). We arrive at:

$$\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\left(\|u_t\|^2 + c^2\|u_x\|^2\right) = -\alpha u_t(L,t)^2.$$

The question is which of the following relations that $u$ satisfies for any smooth initial data.

1. $\|u\|^2 = 0$.
   **False.** Any nonzero initial data contradict this.

2. $\|u\|^2 = -\alpha u_t(L,t)^2$.
   **False.** One can easily select initial data that contradict this.

3. $\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\left(\|u\|^2\right) = 0$.
   **False.** The easiest way to contradict this is probably to set initial data $u = 0, u_t = 1$.

4. $\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\left(\|u\|^2\right) = -\alpha u_t(L,t)^2$.
   **False.** The easiest way to contradict this is probably to set initial data $u = 0, u_t = 1$.

5. $\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\left(\|u_t\|^2 + c^2\|u_x\|^2\right) = 0$.
   **False.** Correct energy, but we know that there is dissipation from the right boundary.

6. $\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\left(\|u_t\|^2 + c^2\|u_x\|^2\right) = -\alpha u_t(L,t)^2$.
   **True!** This is the energy rate that we derived.

7. $\|u_t\|^2 = 0$.
   **False.** Can choose initial data for $u_t$ that does not satisfy this.

8. $\|u_t\|^2 = -\alpha u_t(L,t)^2$.
   **False.** Can choose initial data for $u_t$ that does not satisfy this.

# 3

The total matrix $B = cD + \epsilon A$ is *not* symmetric. CG requires the matrix to be SPD. CG is therefore the worst choice. LU factorization will be more efficient than Gaussian elimination since we are solving many systems with *the same matrix*.

# 4

A correct statement of the weak form is:

Find $u \in V_g$ such that

$$(v, u_t) = -a(v_x, u_x)$$

for all $v \in V_0$.

That is, the correct answers are:

- X1 = Alt. 1
- X2 = Alt. 6
- X3 = Alt. 4

# 5

Three important things to consider:

- A small relative error tolerance ($<< 10^{-3}$) favors a high-order method.
- Complex geometry favors FEM over FD.
- Wave propagation problems (slightly) favor FD over FEM, since there is no mass matrix to invert.

**PDE 1:** Wave propagation (advection), trivial geometry, small error tolerance. Fourth-order FD should be better than second-order FEM.

**PDE 2:** No wave propagation (stationary heat equation), complex geometry, large error tolerance. Perfect for second-order FEM.

**PDE 3:** Wave propagation (Schrödinger), simple geometry (a square), small error tolerance. Perfect for high-order FD.

# 6

**BC1:** $u_x(L, t) = 0$. This is a Neumann condition. The usual SAT ansatz is:

$$SAT_1 = \tau_r H^{-1} \mathbf{e}_r (\mathbf{d}_r^T \mathbf{u} - 0).$$

There is only one alternative that is of this form, with $\tau_r = -1$. Answer:

$$SAT_1 = -H^{-1}\mathbf{e}_r(\mathbf{d}_r^T\mathbf{u} - 0).$$

**BC2:** $u(L,t) = 0$. We have not practiced SAT for Dirichlet BC very much, but we know that $SAT_2$ **must** be consistent with the BC, i.e.,

$$SAT_2 \sim (\mathbf{e}_r^T\mathbf{u} - 0).$$

There is only one alternative of this form, so that must be the correct one. Answer:

$$SAT_2 = -H^{-1}\mathbf{d}_r(\mathbf{e}_r^T\mathbf{u} - 0)$$

This is also similar to the stated SAT for the left boundary, which is a good sign.

**BC3:** $u_x(L,t) + \alpha u(L,t) = 0$. The SBP discretization of this BC is $\mathbf{d}_r^T\mathbf{u} + \alpha\mathbf{e}_r^T\mathbf{u} = 0$. There is only one alternative that matches this. Answer:

$$SAT_3 = -H^{-1}\mathbf{e}_r^T(\mathbf{d}_r^T\mathbf{u} + \alpha\mathbf{e}_r^T\mathbf{u} - 0).$$

# 7

An example solution is included below. It uses the file **operators.py**.

```
import operators as ops
import numpy as np

# Intial data
def f(x):
    return np.exp(-(6*x)**2)


def l2_norm(v, h):
    return h**0.5 * np.linalg.norm(v)


def rk4step(f, v, t, dt):
    k1 = f(v)
    k2 = f(v+0.5*dt*k1)
    k3 = f(v+0.5*dt*k2)
    k4 = f(v+dt*k3)
    v = v + dt/6*(k1 + 2*k2 + 2*k3 + k4)
    t = t + dt
    return v, t
```

```
def main(run_test_parameters=False):

    # Parameters
    m = 101
    xl = -1
    xr = 1

    if run_test_parameters:
        # Parameters for testing
        a = 0
        b = 0.5
        dt_try = 8*1e-5
        T = 0.4

    else:
        # Parameters in final computation
        a = -1
        b = 0.1
        dt_try = 8*1e-5
        T = 0.7

    # Align time step to final time
    mt = int(np.ceil(T/dt_try)) + 1
    dt = T/(mt - 1)

    # Grid
    L = xr - xl
    h = L/(m - 1)
    x = np.linspace(xl, xr, m)


    # Create SBP ops
    H,HI,D1,D2,e_l,e_r,d1_l,d1_r = ops.sbp_cent_4th(m,h)

    # Build matrix
    D = a*D1 + b*D2 + b*HI@d1_l@e_l.T + a/2*HI@e_l@e_l.T - 1/2*HI@e_r@(a
                                        *e_r.T + 2*b*d1_r.T)

    # Right-hand side function, v_t = rhs(v).
    def rhs(v):
        return D@v

    # Initialize solution vector
    v = f(x)
    t = 0

    # Time-stepping loop
    for i in range(mt-1):
        v, t = rk4step(rhs, v, t, dt)

    print(f'Norm of v at t = {t:.2f}: {l2_norm(v, h):.3f}  (using a={a},
                                        b={b}).')
```

```
if __name__ == '__main__':
    main(run_test_parameters=False)
```

# 8

First note that there are typos in the problem formulation. Several $e_\ell^T$ should be $\mathbf{e}_\ell$. Same with $\mathbf{e}_r$. The correct expression is provided below.

To analyze all possibilities at once, let us consider a general SBP-SAT approximation:

$$\mathbf{v}_{tt} = D_2\mathbf{v} + \underbrace{H^{-1}\mathbf{e}_\ell\left(\alpha_\ell \mathbf{e}_\ell^T\mathbf{v}_t + \beta_\ell \mathbf{e}_\ell^T\mathbf{v} + \tau_\ell \mathbf{d}_\ell^T\mathbf{v}\right)}_{SAT_\ell} + \underbrace{H^{-1}\mathbf{e}_r\left(\alpha_r \mathbf{e}_r^T\mathbf{v}_t + \beta_r \mathbf{e}_r^T\mathbf{v} + \tau_r \mathbf{d}_r^T\mathbf{v}\right)}_{SAT_r}$$

The discrete energy method starts by multiplying by $\mathbf{v}_t^T H$ from the left, which yields:

$$\underbrace{(\mathbf{v}_t, \mathbf{v}_{tt})_H}_{=\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\|\mathbf{v}_t\|_H^2} = (\mathbf{v}_t, D_2\mathbf{v})_H + \mathbf{v}_t^T H SAT_\ell + \mathbf{v}_t^T H SAT_r.$$

Let us examine the terms on the right-hand side one by one. By the SBP properties of $D_2$:

$$(\mathbf{v}_t, D_2\mathbf{v})_H = (\mathbf{e}_r^T\mathbf{v}_t)(\mathbf{d}_r^T\mathbf{v}) - (\mathbf{e}_\ell^T\mathbf{v}_t)(\mathbf{d}_\ell^T\mathbf{v}) - \underbrace{\mathbf{v}_t^T M\mathbf{v}}_{=\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{v}^T M\mathbf{v}}.$$

For the terms stemming from $SAT_{\ell,r}$, we have

$$\mathbf{v}_t^T H SAT_\ell = \alpha_\ell(\mathbf{e}_\ell^T\mathbf{v}_t)^2 + \underbrace{\beta_\ell(\mathbf{e}_\ell^T\mathbf{v}_t)(\mathbf{e}_\ell^T\mathbf{v})}_{=\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\beta_\ell(\mathbf{e}_\ell^T\mathbf{v})^2} + \tau_\ell(\mathbf{e}_\ell^T\mathbf{v}_t)(\mathbf{d}_\ell^T\mathbf{v}),$$

$$\mathbf{v}_t^T H SAT_r = \alpha_r(\mathbf{e}_r^T\mathbf{v}_t)^2 + \underbrace{\beta_r(\mathbf{e}_r^T\mathbf{v}_t)(\mathbf{e}_r^T\mathbf{v})}_{=\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\beta_r(\mathbf{e}_r^T\mathbf{v})^2} + \tau_r(\mathbf{e}_r^T\mathbf{v}_t)(\mathbf{d}_r^T\mathbf{v}).$$

Putting everything together and moving all terms with $\frac{\mathrm{d}}{\mathrm{d}t}$ to the left-hand side yields the energy rate

$$\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\underbrace{\left(\|\mathbf{v}_t\|_H^2 + \mathbf{v}^T M\mathbf{v} - \beta_\ell(\mathbf{e}_\ell^T\mathbf{v})^2 - \beta_r(\mathbf{e}_r^T\mathbf{v})^2\right)}_{E} = (-1+\tau_\ell)(\mathbf{e}_\ell^T\mathbf{v}_t)(\mathbf{d}_\ell^T\mathbf{v}) + \alpha_\ell(\mathbf{e}_\ell^T\mathbf{v}_t)^2$$
$$(1+\tau_r)(\mathbf{e}_r^T\mathbf{v}_t)(\mathbf{d}_r^T\mathbf{v}) + \alpha_r(\mathbf{e}_r^T\mathbf{v}_t)^2.$$

For $E$ to be a valid energy, all terms in $E$ need to be nonnegative quantities. This leads to the requirements

$$\beta_\ell \leq 0, \quad \beta_r \leq 0.$$

The right-hand side must be non-positive. The only way to guarantee this is to require

$$\tau_\ell = 1, \quad \tau_r = -1, \quad \alpha_\ell \leq 0, \quad \alpha_r \leq 0.$$

The first alternative that meets all the stability requirements is:

$$\mathbf{v}_{tt} = D_2\mathbf{v} + H^{-1}\mathbf{e}_\ell \left(\mathbf{d}_\ell^T\mathbf{v}\right) - H^{-1}\mathbf{e}_r \left(\mathbf{d}_r^T\mathbf{v}\right),$$

which corresponds to

$$\beta_\ell = 0, \quad \beta_r = 0, \quad \tau_\ell = 1, \quad \tau_r = -1, \quad \alpha_\ell = 0, \quad \alpha_r = 0.$$

The only other correct alternative is

$$\mathbf{v}_{tt} = D_2\mathbf{v} - H^{-1}\mathbf{e}_\ell \left(\mathbf{e}_\ell^T\mathbf{v}_t + \mathbf{e}_\ell^T\mathbf{v} - \mathbf{d}_\ell^T\mathbf{v}\right) - H^{-1}\mathbf{e}_r \left(\mathbf{e}_r^T\mathbf{v}_t + \mathbf{e}_r^T\mathbf{v} + \mathbf{d}_r^T\mathbf{v}\right),$$

which corresponds to

$$\beta_\ell = -1, \quad \beta_r = -1, \quad \tau_\ell = 1, \quad \tau_r = -1, \quad \alpha_\ell = -1, \quad \alpha_r = -1.$$

# 9

An example solution, with some print statements to ensure that things are correct along the way, is included below.

```
import numpy as np
import numpy.linalg as nplg

d1 = -1*np.ones(6)
d0 = 2*np.ones(7)

A = np.diag(d0, 0) + np.diag(d1, 1) + np.diag(d1, -1)
A[0,0] = 6

print('A:')
print(A)
print('---------------------------- \n')

U = np.triu(A, 1)
print('U:')
print(U)
print('---------------------------- \n')

L_plus_D = A - U
print('L + D:')
print(L_plus_D)
print('---------------------------- \n')
```

```
R = - nplg.inv(L_plus_D) @ U
eigs = nplg.eigvals(R)
sr = np.max(np.abs(eigs))
print('Spectral radius: ', sr)
```

# 10

The PDE is

$$u_t = u_{xx} - u_x, \quad x \in (0,1), \quad t > 0,$$

which is the advection-diffusion equation with advection to the right. The energy method yields

$$\underbrace{(u, u_t)}_{=\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\|u\|^2} = \underbrace{(u, u_{xx})}_{IBP} - \underbrace{(u, u_x)}_{=\frac{1}{2}u^2|_0^1} = [IBP] = \left[uu_x - \frac{1}{2}u^2\right]_0^1 - \|u_x\|^2.$$

We have the energy estimate

$$\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\|u\|^2 \leq \left[uu_x - \frac{1}{2}u^2\right]_0^1 = \underbrace{\left[\frac{1}{2}u\left(2u_x - u\right)\right]_0^1}_{:=BT}.$$

For well-posedness, we need $BT \leq 0$. Let us try the sets of BCs one by one.

**BC set 1:**

$$u = 0, \quad x = 0 \quad \text{and} \quad u = 0, \quad x = 1.$$

This set of BCs yields

$$BT = 0.$$

**Well-posed!**

**BC set 2:**

$$u = 0, \quad x = 0 \quad \text{and} \quad u_x - u = 0, \quad x = 1.$$

This set of BCs yields

$$BT = \frac{1}{2}u\left(2u - u\right)|_1 - 0 = \frac{1}{2}u(1,t)^2 \geq 0.$$

**Not well-posed!**

**BC set 3:**

$$2u_x - u = 0, \quad x = 0 \quad \text{and} \quad u_x = 0, \quad x = 1.$$

This set of BCs yields

$$BT = \frac{1}{2}u\left(2 \cdot 0 - u\right)|_1 - 0 = -\frac{1}{2}u(1,t)^2 \leq 0.$$

**Well-posed!**

**BC set 4:**

$$u_x = 0, \quad x = 0 \quad \text{and} \quad u_x = 0, \quad x = 1.$$

This set of BCs yields

$$BT = \frac{1}{2}u\left(0 - u\right)|_1 - \frac{1}{2}u\left(0 - u\right)|_0 = -\frac{1}{2}u(1,t)^2 + \frac{1}{2}u(0,t)^2.$$

**Not well-posed!** (Possible growth at the left boundary)

**BC set 5:**

$$u_x - u = 0, \quad x = 0 \quad \text{and} \quad u_x + u = 0, \quad x = 1.$$

This set of BCs yields

$$BT = \frac{1}{2}u\left(-2u - u\right)|_1 - \frac{1}{2}u\left(2u - u\right)|_0 = -\frac{3}{2}u(1,t)^2 - \frac{1}{2}u(0,t)^2 \leq 0.$$

**Well-posed!**