Boundary Conditions

for advection-diffusion eqs.

A. Donev, Spring 2021

Consider now solving adv-diff

$$u_t + (a\,u)_x = (d\,u_x)_x$$

on $[0, L]$ with BCs.

One of the first challenges we need to deal with is that the allowed BCs are different if $d > 0$ vs. $d = 0$

This means that in general we expect boundary layers for advection-dominated problems

①

To make this explicit, consider

$$u_t + a\,u_x = d\,u_{xx} \qquad, \quad a > 0$$

$$0 < x < L = 1$$

$$u(0, t) = 1 \quad (\underline{\text{inflow}} \text{ boundary})$$

and <u>either</u>
$$\begin{cases} u(1, t) = 0 & (\text{Dirichlet}) \\[2mm] u_x(1, t) = 0 & \left(\begin{smallmatrix}\text{Flux or}\\ \text{Neumann}\end{smallmatrix}\right) \end{cases}$$
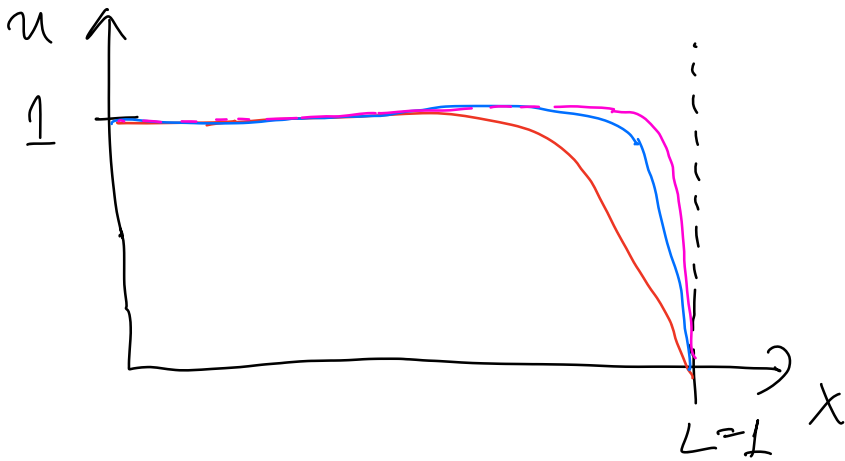
on the <u>outflow</u> boundary

For Dirichlet BC; the steady state solution

$$u_t = 0 = -a\,u_x + d\,u_{xx}$$

is
$$u(x, t) = \frac{e^{a/d} - e^{ax/d}}{e^{a/d} - 1}$$

②

Boundary layer of width

$$l_b \sim d/a \rightarrow 0 \text{ as } d \rightarrow 0$$

If the problem is advection-dominated, i.e.,

Peclet number $P_e = \dfrac{aL}{d} \gg 1$,

then $\dfrac{l_b}{L} \approx \dfrac{d}{aL} \ll 1$

so the boundary layer is very thin.

③

This means that even a fine grid is unlikely to resolve the solution near the boundary.

Does this matter?

Depends!

For flow over the wing of an airplane, we can compute the _lift_ without worrying about the boundary layer. But the (viscous) _drag_ is dominated by the boundary layer.

At an outflow boundary, we mostly care that "stuff" leaves the domain without reflection, so we don't want a boundary layer.

④

In fact, often a boundary
is an open Boundary, meaning
(artificial)
there is <u>not</u> an actual
physical Boundary there and so
the BC is <u>fake</u> anyway.

At such "Boundaries" we
better choose a <u>Neumann</u>
BC $u_x = 0$, since this
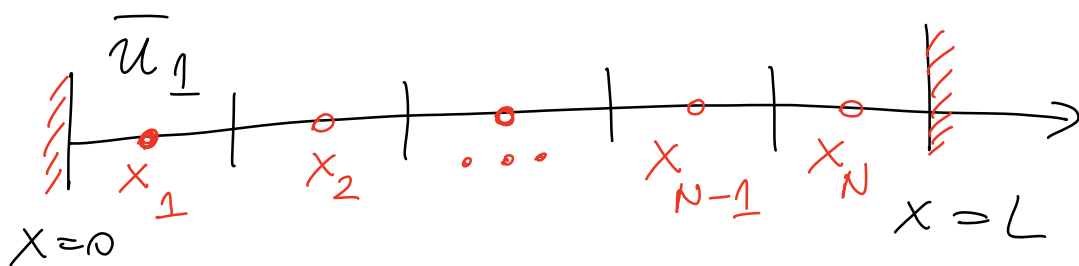makes the steady solution
be

$$u(x, t \to \infty) = 1$$

with <u>no Boundary layer</u>.

It is important to know what
kind of Boundary one has
<u>physically</u> before solving PDE! ⑤

Let's first consider
advection (– dominated) problems

$$\begin{cases} u_t + (a u)_x = 0 \quad , \quad x \in (0, L) \\ u(0, t) = \gamma_0(t) \quad (\text{Inflow}) \\ \underline{NO} \quad BC \quad \text{at} \quad \text{outflow !} \end{cases}$$

$$X_j = (j - 1/2) h \quad , \quad h = \frac{L}{N}$$

Remember that for FV
schemes the __faces__ of the grid
overlap the boundaries:

$$X_{1/2} = 0 \qquad \qquad X_{N+1/2} = L \qquad ⑥$$

The _inflow boundary flux_ is trivial. We need the flux

$$F_{1/2}(t) = a_{1/2} \, u(x_{1/2}, t)$$

$$= a_{1/2} \, g_0(t)$$

For example, to second order

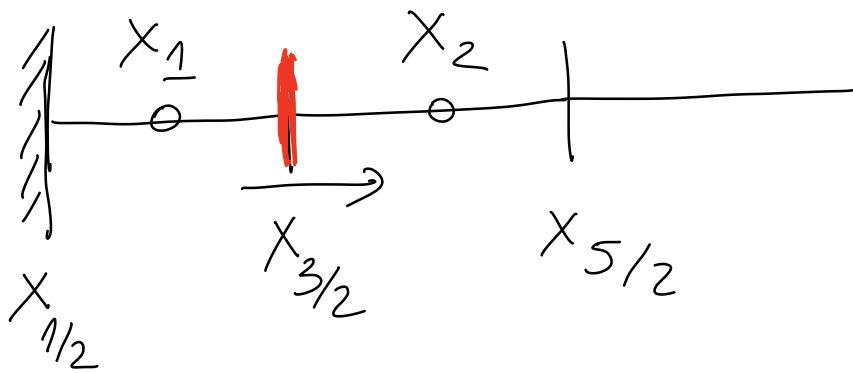$$F_{1/2}^{n+1/2} \approx a_{1/2} \, g_0\left(t^n + \frac{\overline{2}}{2}\right)$$

↑
Don't forget the half step!

But what about $F_{3/2}^{n+1/2}$?

For first order upwinding, there would be no issue

⑦

$$F_{3/2}^n = a_{3/2} u_1$$

We may worry that this will only lead to first-order schemes, but in fact, as we show later, it is OK to use first-order schemes right on the boundary & still get 2nd order, sometimes! Harder issue is stability & hard to analyze & prove

⑧

Another option is to use a centered flux on this face,

$$F^n_{3/2} = a_{3/2} \left(\frac{u_1 + u_2}{2} \right)$$

but we may worry about stability.

Yet a third option is "ghost cells", as we discuss shortly.

In practice: We try things out & see stability / accuracy / robustness

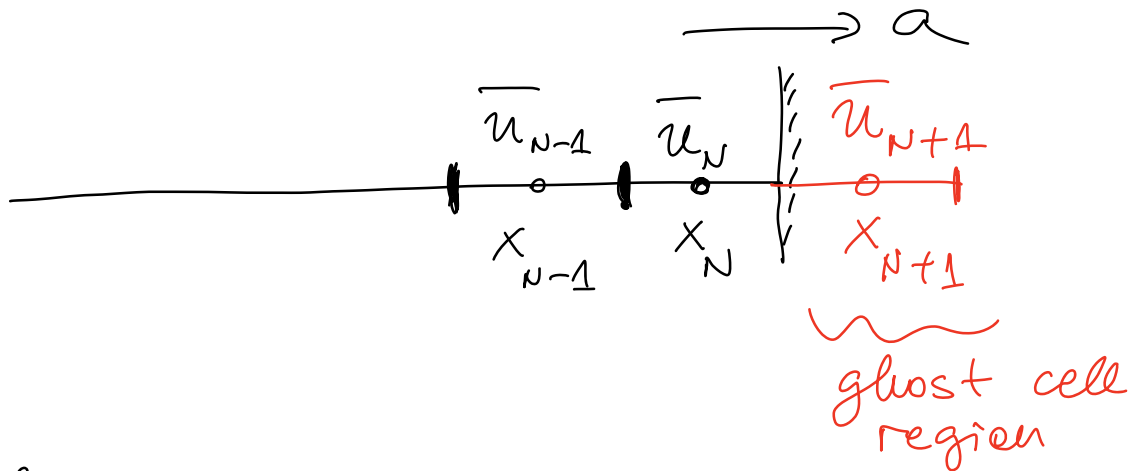The harder part is the outflow boundary, where there is no BC

⑨

<u>Options:</u>

a) In the last one or two cells, switch to a fully one-sided method such as upwind or beam-warming

b) Use "ghost cells" and continue to use the same formulas as in the interior. This is good also for computational performance (no branching in code), is simple, and often works well.
Let's illustrate this approach for the outflow boundary for $a = $ const.

⑩

The idea is to set the values in the ghost cell using extrapolation from the interior using the BC $\underline{\underline{if}}$ there is one. For example:

a) Piecewise $\underline{\underline{constant}}$ extrapolation:

$$U_{N+1} = U_N$$

b) $\underline{\underline{Linear}}$ extrapolation:

$$U_{N+1} = 2U_N - U_{N-1}$$

⑪

Let's see what these do
if we use Lax-Wendroff
NOT in flux form:

$$\overline{u}_N^{n+1} = \overline{u}_N^n - \nu h \left(\frac{\overline{u}_{N-1} - \overline{u}_{N+1}}{2h} \right)$$

$$\underbrace{\phantom{\overline{u}_N^n - \nu h \left(\frac{\overline{u}_{N-1} - \overline{u}_{N+1}}{2h} \right)}}_{\text{centered advection}}$$

$$+ \frac{(\nu h)^2}{2} \left(\frac{\overline{u}_{N-1} - 2\overline{u}_N + \overline{u}_{N+1}}{h^2} \right)$$

Piecewise constant

$$\overline{u}_N^{n+1} = \overline{u}_N^n - \nu h \left(\frac{\overline{u}_{N-1} - \overline{u}_N}{2h} \right)$$

$$+ \frac{(\nu h)^2}{2} \left(\frac{\overline{u}_{N-1} - 2\overline{u}_N + \overline{u}_N}{h^2} \right)$$

$$\bar{u}_N^{n+1} = \bar{u}_N^n - \frac{\nu(1-\nu)}{2}h\left(\frac{\bar{u}_{N-1} - \bar{u}_N}{h}\right)$$

which is ==not consistent== with
the PDE and so is a
bad idea.

If we used ==linear extrapolation==

$$\bar{u}_{N+1} = 2\bar{u}_N - \bar{u}_{N-1} \implies$$

$$\bar{u}_N^{n+1} = \bar{u}_N^n - \nu h\left(\frac{\bar{u}_{N-1} - \bar{u}_N}{h}\right)$$

which means we are using
upwind in the outflow cell,
so this makes sense.
==Is it second order?==
(return to this later)

⑬

Now, imagine we wrote LW or Fromm or Beam Warming in flux form. The problem would be on the outflow face

$$F_{N+1/2} = a \left(\bar{u}_N + \frac{h}{2} S_N (1-\nu) \right)$$

↖ slope estimate

But what is S_N?
For LW, we used downwind slopes,

$$S_N = \frac{\bar{u}_{N+1} - \bar{u}_N}{h} \quad , \quad a > 0$$

So if we use <u>constant</u> extrapolation for the ghost cell $(\bar{u}_{N+1} = \bar{u}_N)$ then $S_N = 0$

$\Rightarrow$ upwind flux $F_{N+1/2} = a \bar{u}_N$

⑭

which is consistent with the
ODE though looks only first
order (to be checked!). So
now constant extrapolation
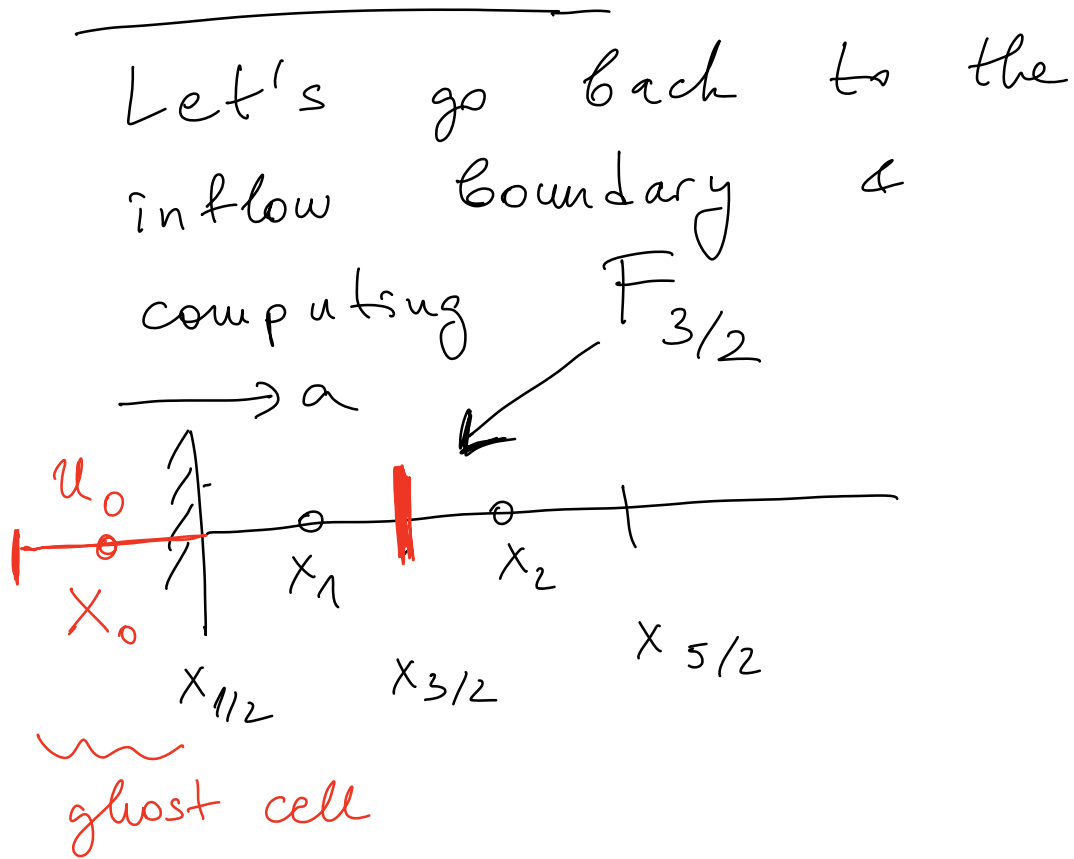is OK.
If we used <u>linear extrapolation</u>
then

$$S_N = \frac{U_N - U_{N-1}}{h}$$

which is the same as the
Beam Warming scheme for
the last <u>face</u> (<u>NOT</u> ~~the~~ same
as switching to BW in last <u>cell</u>)

We see that depending on
what the code does ghost cells
do different things !

(15)

Ghost cells are NOT real cells !

They are just a (computational) tool to implement **one-sided stencils** (finite differences) at the boundaries.

Let's go back to the inflow boundary & computing $F_{3/2}$

$\longrightarrow a$

u_0

x_0

$x_{1/2}$

x_1

$x_{3/2}$

x_2

$x_{5/2}$

~~~
ghost cell

(16)

$$F_{3/2} = a\left(\overline{u}_1 + \frac{h}{2} S_1 (1-\nu)\right)$$

If we use downwind slopes
as in LW, there is no issue,

$$S_1 = \frac{\overline{u}_2 - \overline{u}_1}{h}$$

But what if we are doing
Fromm's method

$$S_1 = \frac{\overline{u}_2 - \overline{u}_0}{2h}$$

Linear extrapolation

$$\overline{u}_0^n = \underbrace{2\gamma_0(t) - \overline{u}_1}_{\text{uses BC!}}$$

$$\Rightarrow S_1 = \frac{(\overline{u}_2 + \overline{u}_1) - 2\gamma_0(t)}{2h}$$

(17)

$$S_1 = \frac{\frac{\overline{u_2} + \overline{u_1}}{2} - \gamma_0(t)}{h}$$

which is a consistent slope estimate, so looks like a good choice. But this also makes sense:

$$S_1 = \frac{u_1 - \gamma_0(t)}{h/2}$$

To get this using ghost cells we would have needed

$$u_0 = u_2 - 4u_1 + 4\gamma_0$$

which is weird & is not quadratic extrapolation, which would be

$$u_0 = \frac{8\gamma_0}{3} - 2u_1 + u_2/3$$

Which choice is "good"? (18)

Getting clear second-order
convergence up to the boundary
is <u>difficult</u> when both advection
& diffusion compete.

More important is to avoid
spurious reflections from
artificial boundaries, since
these polute the solution
in the domain we care about.

In 2D/3D, corners are
even harder & sometimes
singular & not even well-posed
in the PDE itself.

(19)

BCs for diffusion

It is useful here to consider first elliptic PDE

$$\mathcal{L}u = f$$

← elliptic operator WITH BCs

with BCs. General rule:

Discretize elliptic operator

With BCs to (say) 2nd order to get linear system:

$$LU + F^{(BC)} = F \quad \text{(elliptic)}$$

↖ From inhomogeneous BCs

$$\implies U_t = LU + F^{(BC)}(t) \quad \text{(parabolic)} \quad \text{MOL}$$
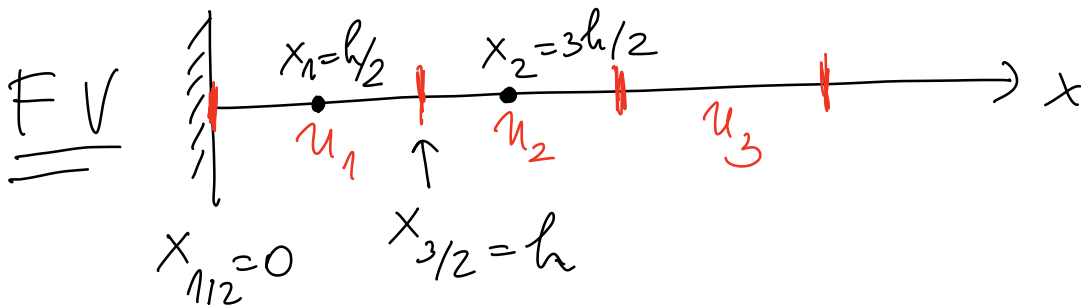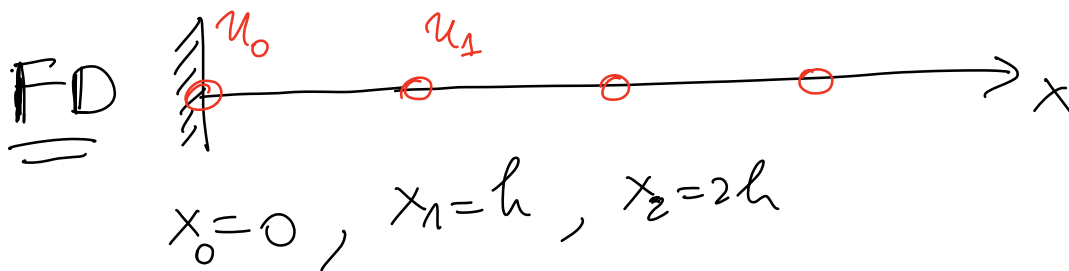
and you will get 2nd order for the parabolic eq. as well

(20)

Let us consider both FD &
FV schemes, i.e., grids where
the boundary is either a grid
point or a face

[Both appear in FV in 2D
sometimes, say incompressible
Navier-Stokes on staggered grid]

$$u_{xx} = f \quad , \quad x \in (0, L)$$

FD

$x_0 = 0, \quad x_1 = h, \quad x_2 = 2h$

FV

$x_{1/2} = 0 \qquad x_{3/2} = h$

(21)

For $\underline{\underline{FD}}$, Dirichlet BC is easy, just set "ghost" cell:

$$u_0 = u(0)$$

$$\Rightarrow \left(u_{xx}\right)_{x=h} \simeq \frac{u_2 - 2u_1 + \overset{= u(0)}{u_0}}{h^2}$$

$$\Rightarrow \begin{cases} \left(Lu\right)_1 = \dfrac{u_2 - 2u_1}{h^2} \\[4mm] F^{(BC)}_1 = \dfrac{u(0)}{h^2} \end{cases} \quad \textcolor{red}{(FD - Dirichlet)}$$

For $\underline{FV}$, homogeneous Neumann is easy since it really means <u>no flux</u> through boundary.

For inhomogeneous BCs, set

$\textcircled{22}$

$$\frac{u_1 - u_0}{h} \simeq u_x(x=0)$$

$$\Rightarrow \quad u_0 = u_1 - h\, u_x(x=0)$$

$$(u_{xx})_{x=h/2} \simeq \frac{u_2 - 2u_1 + u_0}{h^2}$$

$$= \frac{u_2 - u_1}{h^2} - \frac{u_x(x=0)}{h} \Rightarrow$$

$$\begin{cases} (Lu)_1 = \dfrac{u_2 - u_1}{h^2} & \text{(FV-Neumman)} \\[2mm] F_1^{BC} = -\dfrac{u_x(x=0)}{h} \end{cases}$$

(23)

Is this second order?
The easiest way to check
is to check if exact for
quadratic functions. That is,
take
$$u(x) = ax^2 + bx + c$$
$$u_1 = u(h/2)$$
$$u_2 = u(3h/2)$$
& confirm that
$$\frac{u_2 - u_1}{h^2} - \frac{u_x(x=0)}{h} = u_{xx}\left(\frac{h}{2}\right) = 2a$$
which shows 2nd order
(24)

For FD, however, Neumann BC is less obvious.

One option (Section 2.12 in FD book of Le Veque):

$$\frac{U_1 - U_0}{h} = U_x (x=0)$$

and use this equation instead of the PDE & the first point (U_0). This makes it harder to do parabolic for which we need PDE at $x=0$.

Instead, let's put a ghost point U_{-1} & use BC to fill it in:

(25)

E.g.

$$\frac{u_1 - u_{-1}}{2h} = u_x(x=0) \Rightarrow$$

$$u_{-1} = u_1 - 2h\, u_x(0)$$

$$(u_{xx})_{x=0} \simeq \frac{1}{h^2}\left(u_{-1} - 2u_0 + u_1\right)$$

$$= \frac{2}{h^2}\left(u_1 - u_0 - h\, u_x(0)\right) \Rightarrow$$

$$\begin{cases} (L\,u)_0 = \frac{2}{h^2}\left(u_1 - u_0\right) \\[2em] F_0^{BC} = -\frac{2}{h}\, u_x(0) \end{cases}$$

FD
Neumann

26

This makes sense for the elliptic PDE since

$$\left(LU\right)_0 + F_0^{BC} \rightarrow \left(\mathcal{L}\, u\right)(x=0)$$

$$\underset{\parallel}{}\qquad\qquad\qquad \Downarrow PDE$$

$$\frac{2}{h^2}\left(u_1 - u_0 - h\, u_x(0)\right) = f(0)$$

$$\Rightarrow \quad \frac{u_1 - u_0}{h} = u_x(0) + \frac{h}{2} f(0)$$

$$\underset{\rotatebox{-90}{$\approx$}}{} \qquad\qquad\qquad \parallel \qquad \downarrow PDE$$

$$\frac{u(h) - u(0)}{h} \approx u_x\!\left(\tfrac{h}{2}\right) = u_x(0) + \frac{h}{2} u_{xx}(0) + O(h^2)$$

which is a one-sided Taylor series, one order more accurate than the previous attempt $\dfrac{u_1 - u_0}{h} = f(0)$

(27)

Yet another approach is to use a 2nd order one-sided difference instead of ghost cells:

$u_0 \qquad u_1$

$$u (x \text{ near } 0) = a x^2 + b x + c$$

$$u_x(0) = b$$
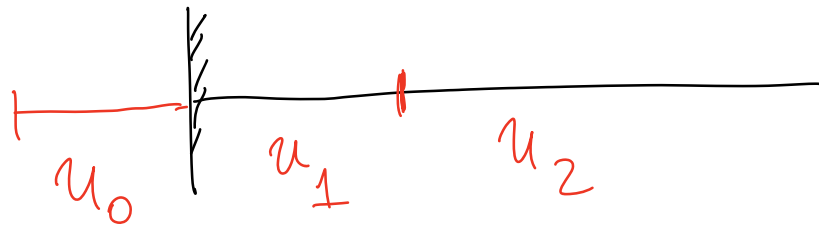
$$u(0) = u_0 = c$$

$$u(h) = a h^2 + b h + c = u_1$$

$$u_{xx}\left(\frac{h}{2}\right) = 2a = \frac{2}{h^2}(u_1 - u_0) - \frac{2}{h} u_x(0)$$

same as what we got before

Since this is exact for quadratic functions, expect 2nd order

(28)

For ==FV, Dirichlet== is less obvious. Let's try using a ghost cell

$$\frac{u_0 + u_1}{2} \approx u(x=0) \quad \swarrow^{BC}$$

$$\Rightarrow u_0 = 2u(0) - u_1$$

(linear extrapolation)

$$(u_{xx})(x = h/2) = \frac{1}{h^2}\left(u_0 - 2u_1 + u_2\right)$$

$$= \frac{1}{h^2}\left(-3u_1 + u_2 + 2u(0)\right)$$

(29)

$$\Rightarrow \begin{cases} (Lu)_1 = \dfrac{1}{h^2}(u_2 - 3u_1) \\[2mm] F_1^{BC} = \dfrac{2u(0)}{h^2} \end{cases}$$

FV Dirichlet #1

The matrix L looks like

$$L = \frac{1}{h^2}\begin{bmatrix} -3 & 1 & & \\ 1 & -2 & \frac{1}{2} & \\ & 1 & -2 & 1 \\ & & 1 & \ddots \end{bmatrix}$$

which is symmetric negative definite, just like the Laplacian operator with Dirichlet BCs. So this is a good discretization in terms of robustness & physical fidelity? Is it 2nd order accurate?

(30)

If we plug in a quadratic function like before, we would get

$$\frac{1}{h^2}\left(-3u_1 + u_2 + 2u(0)\right) =$$

$$\frac{3a}{2} \neq 2a = u_{xx}(h/2)$$

so this is not consistent with PDE at boundary.

Yet numerically this is 2nd order for diffusion equation.

But, if we also have advection, the error introduced into the domain is advected into the domain & we loose 2nd order (demo code)

(31)

Can we get 2nd order
accuracy at boundary?

We want

$$U_{xx}(x=h/2) = \alpha\, u(h/2) + \beta\,(3h/2)$$
$$+ \gamma\, u(0)$$

to be exact for __all__
quadratic functions

$$u(x) = ax^2 + bx + c \quad \Rightarrow$$

(simple 3x3 linear solve)

$$\alpha = -\frac{4}{h^2} \qquad \beta = \frac{4}{3h^2} \qquad \delta = \frac{8}{3h^2}$$

$$\begin{cases} (Lu)_1 = \frac{4}{3h^2}(3u_1 - u_0) \quad \text{\color{red}{FV}} \\[2mm] F_1^{BC} = \frac{8\,u(0)}{3h^2} \end{cases}$$

FV
Dirichlet
#2

(32)

Note that the matrix L is
no longer symmetric! Eigenvalues
may not all be real ... so
we could be sacrificing stability
for accuracy.

An easier & more flexible
way to derive the same BC
& put it in codes (including
non-constant diffusion!) is to
use quadratic extrapolation
to ghost cell

$$U_0 \simeq u(-h/2) = -2U_1 + \frac{U_2}{3} + \frac{8u(0)}{3}$$

$$\Rightarrow \frac{U_0 - 2U_1 + U_2}{h^2} = \frac{4}{3h^2}\left(3U_1 - U_0\right) + \frac{8u(0)}{3h^2}$$

as before $\qquad$ (33)

With non-constant diffusion:

$$F_{1/2} = -d_{1/2}\left(\frac{u_1 - u_0}{h}\right)$$

$$F_{3/2} = -d_{3/2}\left(\frac{u_2 - u_1}{h}\right)$$

$$\left((du_x)_x\right)(x = h/2) = -\frac{F_{3/2} - F_{1/2}}{h}$$

$$= -\frac{\left(3d_{1/2} + d_{3/2}\right)}{h^2}u_1$$

$$+ \left(\frac{\frac{1}{3}d_{1/2} + d_{3/2}}{h^2}\right)u_2 \quad \nwarrow \text{Matrix } L$$

$$+ \frac{8d_{1/2}\, u(0)}{3h^2}$$

(34)

We will not have time to
discuss theory in any detail
here, and the general theory
is called GKSO theory
(Gustafsson, Kreiss, Oliger, Osher)
and is complicated.

An important point to realize
is that the error follows
the same (discretized) PDE
as the solution, (This follows
from linearity) with the local
truncation error as a source
term on the r.h.s.
E.g. error at an inflow
boundary will be advected into
the domain (and is important)

(35)

For elliptic PDEs:

Let $\hat{u}$ be numerical solution
& u be discretized true
                              solution

Error $E = \hat{u} - u$

$$L \hat{u} = f \qquad (\text{Discrete system})$$

Local Truncation error (LTE):

$$\bar{\tau} = L u - f$$

$$L(u - \hat{u}) = LE = \bar{\tau}$$

If we make a larger error
at a boundary, how it affects
the interior depends on solution
of $LE = \bar{\tau}$ which approximates
$\simeq \mathcal{L} e = \bar{\tau}$ (elliptic PDE +
                    source on
                    boundary) (36)

For $\mathscr{L} = \partial_{xx}$ + periodic

$$\bar{\tau} = \frac{h^2}{12} u_{xxxx} + O(h^4)$$

(Taylor series)

$$\mathscr{L}e = \bar{\tau} \implies$$

$$e_{xx} \approx \frac{h^2}{12} u_{xxxx} \implies$$

can be added to $\hat{u}$ to get 4th order accuracy ("compact" stencil)

$$e \approx \frac{h^2}{12} u_{xx} = \frac{h^2}{12} f$$

which is an excellent error estimate in the absence of boundaries.

Now imagine we have BCs, say Neumann on right & Dirichlet on left, and we modify L accordingly.

(37)

E.g. imagine

$$\bar{\tau} = \begin{bmatrix} c_L \; O(h^p) \\ O(h^2) \\ c_R \; O(h^q) \end{bmatrix} \begin{matrix} \leftarrow \text{Left e.g.} \\ p = 0 \; (\text{inconsistent}) \\ \\ \leftarrow \text{Right e.g.} \\ q = \underline{1} \; \binom{\text{first}}{\text{order}} \end{matrix}$$

Dirichlet

$$L = \frac{1}{h^2} \begin{bmatrix} -3 & 1 & 1 & & \\ 1 & -2 & \ddots & & \\ 1 & 1 & \ddots & \ddots & \\ & & & \ddots & 1 \\ & & & 1 & -1 \end{bmatrix} \quad \begin{pmatrix} \text{Discrete} \\ \text{Laplacian} \\ + BCs \end{pmatrix}$$

Neumann

Solve $LE = \bar{\tau}$ & you
will see that

$$E = O(h^2) \; \underline{\text{everywhere}}$$

This is because $L^{-1} = h^2 O(1)$
has an h^2 in front which
helps us! Specifically

(38)

$$L^{-1} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = -\frac{h^2}{2} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ \vdots \end{bmatrix}$$

$$L^{-1} \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} = h^2 \begin{bmatrix} 1 \\ 2 \\ 3 \\ \vdots \\ N \end{bmatrix} - \frac{h^2}{2} \qquad \leftarrow \underline{\text{error grows}}$$

So
$$E = h^2 \left[C_L \, O(h^p) + N \cdot C_R \cdot O(h^q) \right]$$

$$E = C_L \, O(h^{p+2}) +$$
$$L \, C_R \, O(h^{q+1})$$

This means we can get a 2nd
order scheme <u>even</u> if $p = 0$
& $q = \underline{1}$!

To see this from the PDE
perspective $\mathcal{L}e = \bar{\tau}$,
note that e satisfies:

$$
\begin{cases}
e_{xx} = \dfrac{h^2}{12} u_{xxxx} & \text{(interior)} \\[2mm]
e(x=0) = \alpha & \text{(Dirichlet LTE)} \\[2mm]
e_x(x=L) = \beta & \text{(Neumann LTE)}
\end{cases}
$$

$$
\Rightarrow \quad e(x) = \frac{h^2}{12} f(x) + \underset{\underset{\text{constant}}{\uparrow}}{\alpha} + \underset{\underset{\text{grows}}{\sim}}{\beta x}
$$

which tells us that error
introduced at Dirichlet BC will
propagate <u>unchanged</u> through interior,
but error at Neumann boundary
will grow in interior

(40)

For time-dependent PDEs, consider MOL for simplicity

$$\hat{u}'(t) = L\hat{u}(t) + g(t)$$

assume constant

LTE

LTE $u'(t) = Lu + g + \overline{\tau}$

$$e'(t) = u' - \hat{u}' = Le + \overline{\tau}$$

$$e' = Le + \overline{\tau}$$

$$\varepsilon_t = \mathscr{L}\varepsilon + \overline{\tau}$$ } error approx. satisfies same PDE + source

$$e' = L(e + L^{-1}\overline{\tau})$$

$$= L(e + \xi(t))$$

If scheme is stable, then perturbation/error ξ won't grow too fast

(41)

So if $\xi = O(h^P)$ then
$e = O(h^P)$ for finite time.
This means that we need
to control/bound/estimate ξ:

$$L\xi = \overline{\tau}$$

which is the steady-state
equation $\mathcal{L}\xi = \overline{\tau}$, i.e.,
elliptic PDE for parabolic eqs.
This shows that if we
discretize elliptic PDE + BCs
to second order, we will get
second order for the parabolic
one if time integrator is
stable & sufficiently accurate.
For advection-diffusion, error
at inflow boundaries gets
advected into the domain ㊷