Generalization of the model problem:

$$u'(t) = -a(t)u(t) + b(t)$$

CN-style discretization:

Step 1: Discretizing domain: $0 = t_0, t_1, t_2, \ldots t_{N_t} = T$
uniform $t_n = n\Delta t$

Step 2: Sample the ODE at discrete points:

$$u'(t_{n+1/2}) = -a(t_{n+1/2})\,u(t_{n+1/2}) + b(t_{n+1/2})$$

Step 3: Approximate u' by finite difference

$$u'(t_{n+1/2}) \approx \frac{u^{n+1} - u^n}{\Delta t}$$

Step 4: Derive the algorithm.
Problem: $u(t_{n+1/2})$? $u^{n+1/2}$ is not a quantity we compute in the mesh, only u^n and u^{n+1}

$$u^{n+1/2} \approx \frac{1}{2}(u^n + u^{n+1})$$

$a(t_{n+1/2})$? Can just evaluate a given $a(t)$
Or: $a(t_{n+1/2}) \approx \frac{1}{2}(a(t_n) + a(t_{n+1}))$

Same for $b(t_{n+1/2})$.

$$\Rightarrow \quad \frac{u^{n+1} - u^n}{\Delta t} = -a(t_{n+1/2})\frac{1}{2}(u^n + u^{n+1}) + b(t_{n+1/2})$$

Solve wrt. u^{n+1}

$$u^{n+1} = \ldots$$

Compulsory exercise: $u' = -a|u|u + b$

$$[|u|u]^{n+\frac{1}{2}} \approx |u|^n \cdot u^{n+1} \Rightarrow \text{linear eq. in } u^{n+1}$$

—

Verification and debugging:

1. Always check a program with a constant solution (not 0 or 1).

$u_e = 3$.
We have to construct the right ODE problem (assume we can specify any $a(t)$ and $b(t)$ in the program)
Any $a(t)$, fit $b(t)$:

$$\underbrace{\frac{d}{dt}3}_{=0} = -a(t)\cdot 3 + b(t) \Rightarrow b(t) = 3a(t)$$

$I = 3 \quad (u(0))$

Problem: any $a(t)$, $u' = -au + \underbrace{3a(t)}_{b}$, $u(0) = 3$

$a(t) = \sin(t)$, t, 2, whatever

Important: Need a sufficiently general implementation with $a(t)$ and $b(t)$ although one might be interested in solving a very specific problem for a particular $a(t)$ and $b(t)$!

The method of manufactured solutions (MMS):
(or: how to always find an analytical solution of the differential equation problem (!))

Given diff.eq. $\mathcal{L}(u) = f$. Choose $u = v$ (some formula)
Set $f = \mathcal{L}(v)$. Solve $\mathcal{L}(u) = \mathcal{L}(v)$. Make initial and boundary conditions on u compatible with v.
Then the program finds a u that approximates v.
Perform convergence rate estimation to verify the implementation.

Example: $u' = -a(t)u + b(t)$.
Choose exact solution: $u_e = \sin(t)$.
Choose any $a(t)$.
Fit b: $\cos(t) = -a(t)\sin(t) + b(t)$
$\Rightarrow b(t) = \cos(t) + a(t)\sin(t)$
$I = \sin(0) = 0 \quad (u(0))$

If we now solve

$$u' = -a(t)u + \cos(t) + a(t)\sin(t)$$
$$u(0) = 0$$

we should get approximations to $\sin(t)$.
Compute errors as $\Delta t \to 0$ and estimate convergence rate.

Nice feature of MMS: Can always generate an analytical solution and hence always compute convergence rates for verification.

Another nice feature: Very often $u \sim t$ or $u \sim t^2$ will solve both the diff. eq. problem (with an appropriate source term) <u>and</u> the discrete equations!

Example: $u' = -au + b$ with Forward Euler

$$[D_t^+ u = -au + b]^n$$

Choose $u = t$ as prescribed solution.
Then $b(t) = 1 + a(t)t$ for any choice of a.

$$[D_t^+ u = -au + 1 + at]^n$$

Is $u^n = t_n = n\Delta t$ also a solution of the FE equations?

$$[D_t^+ t]^n = 1$$

$$\Rightarrow [1 = -at + 1 + at]^n = [0 = 0]^n \text{ Yes!!}$$

Why is this so nice? We know that the output of the program should be $u^n = n\Delta t$ for any $a(t)$ and any time mesh (as long as Δt is small enough to make FE stable).

MMS for discrete equations: Prescribe u^n, insert in scheme, fit the discrete source term.
This is the purpose of the first compulsory exercise.