Super-Time-Stepping

a simple way of speeding up the explicit scheme for parabolic problems:

Think of the PDE as $\frac{d}{dt}U(t) + AU(t) = 0\,, \quad U(0) = u_0$

Think of the discrete scheme as $U^{n+1} = U^n - \Delta t_n A U^n = (I - \Delta t A)U^n$

$U^0 = u_0$ $\quad$ A is $M \times M$ matrix

symmetric, positive-definite (for parabolic PDE)

Stability condition: $\rho(I - \Delta t\, A) < 1\,, \quad \rho(\cdot) = \text{spectral radius} = \max|\lambda_i|$

$\Rightarrow \Delta t < \Delta t_{expl} := \frac{2}{\lambda_{max}}\,, \quad \lambda_{max} = \text{largest eigenvalue of } A$

(CFL condition) $\quad$ $\left(= \frac{4D}{\Delta x^2} \text{ for } u_t = D u_{xx}\right)$

Idea: Do NOT require stability at each timestep Δt,
only require it at the end of every N steps, for some chosen N
$(1, 5, 10, 20, 100, \ldots)$

So, consider a superstep ΔT consisting of N timesteps $\tau_1, \tau_2, \ldots, \tau_N$
and want to choose $\{\tau_i\}$ to ensure stability over $\Delta T = \sum \tau_i$
and maximize the duration ΔT of the superstep.

So, want $\rho\left(\prod_{i=1}^{N}(I - \tau_i A)\right) < 1$ and $\sum_{i=1}^{N} \tau_i$ as large as possible.

Miracle: This optimization problem has explicit solution, thanks to the remarkable optimality properties of Chebychev polynomials !!!

$$\tau_i = \Delta t_{expl} \cdot \frac{1}{(-1+\nu)\cdot\cos\left(\frac{2i-1}{N}\frac{\pi}{2}\right) + 1 + \nu}\,, \quad i = 1, 2, \ldots, N,$$

where $0 < \nu < \lambda_{min}/\lambda_{max}$

Then $\Delta T \xrightarrow[\nu \to 0]{} N^2 \cdot \Delta t_{expl}$ $\quad$ $N \cdot (N \cdot \Delta t_{expl})$ $\quad$ so N substeps of a superstep cover a time interval N times longer than N explicit steps (each of Δt_{expl})!!!

That's where speedup comes from!

Implementation: Figure out Δt_{expl} from CFL condition (Positive Coeff. Rule).
Pick N, ν (say $N=5$, $\nu=0.001$).
Instead of executing steps of length Δt_{expl},
execute N steps of lengths $\tau_1, \tau_2, \dots, \tau_N$
(no outputing until a superstep is done)

superstep loop:

```
for i = 1:N
    τ_i = ...   (better: precompute τ_i 's and duration ΔT = Σ_1^N τ_i )
    Δt = τ_i
    call EXPLICIT(Δt)
end for
    call OUTPUT if you want (if time ≥ tout)
```

Remarks: 1. $N=1$, $\nu=0$ is the explicit scheme itself, very convenient for debugging.

2. For fixed N, the smaller ν is the faster it runs (bigger ΔT, so fewest supersteps to get to t_{end})
but also less accurate.
So ν is a convenient gauge: close to $\nu=0$: more efficiency, less accuracy
further from $\nu=0$: more accuracy, higher cost

3. Experience shows that $N=5$ to 20 yields best results

4. Works on linear and nonlinear problems equally well

5. Works on 1, 2, 3 dimensions, higher efficiency in higher dimensions.

6. Beats the standard implicit schemes in efficiency and accuracy
with no extra programming!