Super-Time-Stepping

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

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

· Think of the discrete scheme as

$$\vec{U}^{n+1} = \vec{U}^n - \Delta t_n A \vec{U}^n = (I - \Delta t\, A)\vec{U}^n, \quad A \text{ is } M\times M \text{ matrix}$$

$$\vec{U}^0 = u_0$$

symmetric, positive-definite (for parabolic)

Stability condition:

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

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

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

Idea: Don't require stability at the end of <u>each</u> step Δt, only require it at the end of (every) N steps. (pick some $N = 1, 2, 5, 10, 20, 100, \dots$)

So, consider a superstep ΔT consisting of N time-steps $\tau_1, \tau_2, \dots, \tau_N$ and want to choose τ_i to ensure stability over $\Delta T = \sum \tau_i$ and maximize the duration of the superstep ΔT.

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

<u>Miracle</u>: This optimization problem has <u>explicit</u> solution, thanks to the remarkable optimality properties of Chebychev polynomials:

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

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

Then $\quad \Delta T \xrightarrow[\nu\to 0]{} N^2 \Delta t_{expl} \; !!!$ so N substeps of a superstep over a time interval N times longer than

$N \cdot (N \Delta t_{expl})$ ← N explicit steps (each of Δt_{expl}) !!!

Implementation: Figure out Δt_{expl} from the CFL condition (Positive Coeff. Rule)

Pick N, ν (say $N=5$, $\nu=.001$)

Instead of executing steps of length Δt_{expl},

execute N steps of lengths $\tau_1, \tau_2, \ldots, \tau_N$

(no outputting until a superstep is done):

```
DO i=1,N
   τ_i = .....   {better: precompute τ_i's, and duration = Σ_1^N τ_i }
   Δt = τ_i
   CALL EXPLICIT(Δt)
ENDDO
   CALL OUTPUT if you want
```

Remarks: 1. For fixed N, the smaller ν is the faster it runs but also less accurate

So ν is a convenient gauge: close to $\nu=0$: more efficiency less accuracy and further from $\nu=0$: more accuracy (at higher cost)

2. Experience shows that $N=5$ to 20 yields best (efficient) results

3. Works on linear and nonlinear problems equally well

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

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

6. $N=1$, $\nu=0$ is the explicit scheme itself.