

OVERCOMING THE STABILITY RESTRICTION OF EXPLICIT SCHEMES VIA SUPER-TIME-STEPPING

Vasilios Alexiades

Math.Dept., University of Tennessee, Knoxville TN 37996-1300, *alexiaades@utk.edu*
and Math. Sciences Section, Oak Ridge National Lab., Oak Ridge TN 37831-6367

ABSTRACT: Super-time-stepping is a simple method that impressively speeds up explicit time-stepping schemes for parabolic problems. It frees the explicit scheme from the stability restriction on the time-step, rendering it as usable as any implicit scheme, while retaining its simplicity and better accuracy. Unfortunately, it seems hardly anyone knows about it in the computational PDE world, despite the fact that the method is almost 20 years old. We present the basis of the method and document its performance on both linear and nonlinear parabolic problems.

AMS (MOS) subject classification.: 65M20, 65M10

1. INTRODUCTION

Super-time-stepping (STS) is a simple and effective method that speeds up explicit time-stepping schemes for parabolic problems. Even though the method is almost 20 years old, Gentsch [4], hardly anyone in the computational PDE world seems to know about it. This is regrettable, since the method can be very easily incorporated in an existing explicit code and yields impressive savings. I learned about it at the end of 1992 from Droux [3], too late to include it in my book [2] on Stefan-type problems and compare it with the other methods described there. Since then, I have used it on rather demanding, highly nonlinear, 2- and 3-dimensional alloy solidification and welding problems, speeding up the computations by a factor of 4 or 5 (while retaining accuracy), by just inserting a few additional lines in the codes ! That's impressive savings at zero cost.

Conceptually, STS belongs to a class known in numerical ODE circles as Runge-Kutta-Chebyshev methods (van der Houwen [5]). A direct development of (a variant of) the method and its justification for linear equations is presented in Alexiades et al [1], where more details can be found.

The idea of super-time-stepping is to demand stability only at the end of every N steps, constituting one super-step, instead of at every single step. The intermediate steps are chosen non-uniformly from a simple formula (Chebychev numbers). It turns out that executing N super-steps covers a time interval proportional to N^2 times the explicit time-step, thus speeding up the computation by (up to) a factor of N .

2. THE STS SCHEME AND ITS IMPLEMENTATION

The standard explicit scheme (forward Euler time discretization) for a parabolic equation may be written as

$$U^{n+1} = U^n - \Delta t A U^n, \quad n=0, 1, 2, \dots, \quad (1)$$

where Δt is the time step, U is the vector of nodal values, and A is an $M \times M$ symmetric positive definite matrix representing the discretized elliptic operator of the PDE. The Courant-Friedrichs-Lewy condition sufficient for stability is

$$\Delta t \leq \Delta t_{expl} := 2 / \lambda_{\max}, \quad (2)$$

where λ_{\max} is the largest eigenvalue of the matrix A . For example, in the case of the 1-dimensional heat equation $u_t = \alpha u_{xx}$, on a uniform mesh, we have $\lambda_{\max} = 4\alpha/\Delta x^2$, so that $\Delta t_{expl} = \Delta x^2 / (2\alpha)$.

The idea of Super-Time-Stepping is the following. Instead of demanding stability at each step, choose an integer N and demand stability at the end of a **superstep** ΔT consisting of N substeps $\tau_1, \tau_2, \dots, \tau_N$. It is remarkable that these can be chosen **explicitly** so as to ensure stability over the superstep *and maximize* the duration $\Delta T = \sum_{i=1}^N \tau_i$ of the superstep (Genztsch [4], Alexiades et al [1]). Thanks to the optimality properties of some modified Chebychev polynomials, they may be written as, [1],

$$\tau_i = \Delta t_{expl} \left((-1 + \nu) \cos \left(\frac{2i-1}{N} \frac{\pi}{2} \right) + 1 + \nu \right)^{-1}, \quad i = 1, \dots, N, \quad (3)$$

where ν is any number in the interval $(0, \lambda_{\min} / \lambda_{\max}]$, with $\lambda_{\min}, \lambda_{\max}$ the smallest and largest eigenvalues of the matrix A . The resulting duration of the superstep turns out to be, [1],

$$\Delta T := \sum_{i=1}^N \tau_i \rightarrow N^2 \Delta t_{expl}, \quad (\text{as } \nu \rightarrow 0). \quad (4)$$

Comparing this duration to the length of time covered by executing N single explicit steps: $N \Delta t_{expl}$, we see that a superstep consisting of N substeps is N times longer ! Thus, superstepping is N times faster than the standard explicit scheme (when $\nu = 0$).

For each choice of N , the scheme is stable for large enough ν (< 1). The larger the damping factor ν , the shorter ΔT becomes, improving the accuracy at the expense of more computations. The length of the superstep ΔT (which is determined by the spectral properties of A , and the choice of N and ν) is only restricted by accuracy, just like in any unconditionally stable implicit scheme.

The scheme may be easily incorporated in any explicit time-stepping algorithm of the form (1), so in particular to parabolic problems. One determines the explicit time-step Δt_{expl} in the usual way to satisfy the CFL condition, but instead of executing steps of length Δt_{expl} , one executes supersteps of length ΔT as follows: choose N, ν and execute the N substeps $\tau_1, \tau_2, \dots, \tau_N$, of (3), without outputting until the end of each super-step. The only additional expense is the trivial computation in (3), while the execution is accelerated by (up to) a factor of N .

Even though the scheme is constructed and justified in [1] for linear problems only, it turns out to be applicable and work equally well also on nonlinear problems, as evidenced by the results below and in [3].

3. TWO EXACTLY SOLVABLE MODEL PROBLEMS

We consider the following two exactly solvable model problems: a plain heat conduction problem (linear) and the classical 2-phase Stefan problem (nonlinear), in one space dimension.

PROBLEM 1: Consider the heating of a slab in $x > 0$, initially at (normalized) temperature $u(x, 0) = 0$, when $u = 1$ is imposed at $x = 0$. Assuming constant thermophysical properties, in dimensionless variables the temperature $u(x, t)$ satisfies the heat equation $u_t = u_{xx}$. The problem admits an exact (similarity) solution in terms of the error function, given by

$$u(x, t) = 1 - \operatorname{erf}(x / 2\sqrt{t}), \quad x > 0, \quad t > 0. \quad (5)$$

PROBLEM 2: Consider the melting of a slab in $x > 0$, initially solid at a (normalized) temperature $u(x, 0) = -1$, due to $u = +1$ imposed at $x = 0$. A melt front $x = X(t)$ originates from $x = 0$ at $t = 0$, separating liquid in $0 \leq x < X(t)$ from solid in $X(t) < x$, $t > 0$. The phase-change temperature has been normalized to zero. Assuming constant thermophysical properties and, for simplicity here, the same in both phases, in dimensionless variables the unknowns $u(x, t)$ and $X(t)$ satisfy (see [2]) the heat equation $u_t = u_{xx}$ in both phases, and the interface conditions: $u(X(t), t) = 0$, and $X'(t) = St [-u_x(X^-, t) + u_x(X^+, t)]$, where St denotes the *Stefan number* of the process (ratio of sensible to latent heat, here equal to one over latent heat due to our normalizations). This is a moving boundary problem, de facto nonlinear, and one of the few explicitly solvable phase-change problems. It admits a similarity solution (known as the Neumann solution), [2], which in this case takes the form

$$X(t) = 2\Lambda\sqrt{t} \quad (6a)$$

$$u(x, t) = +1 - \frac{\operatorname{erf}\left(\frac{x}{2\sqrt{t}}\right)}{\operatorname{erf} \Lambda}, \quad 0 \leq x \leq X(t), \quad t > 0, \quad (\text{liquid}) \quad (6b)$$

$$u(x, t) = -1 + \frac{\operatorname{erfc}\left(\frac{x}{2\sqrt{t}}\right)}{\operatorname{erfc} \Lambda}, \quad X(t) \leq x, \quad t > 0, \quad (\text{solid}) \quad (6c)$$

with Λ the unique root of the transcendental equation

$$\operatorname{St} \left(\frac{1}{\operatorname{erf} \Lambda} - \frac{1}{\operatorname{erfc} \Lambda} \right) = \sqrt{\pi} \Lambda \exp(\Lambda^2). \quad (6d)$$

Note that PROBLEM 1 is simply the case of zero latent heat (and initial temperature equal to zero).

The most convenient, general, and effective numerical method for treating such problems is the *enthalpy method*, [2], which approximates a weak formulation of the problem (expressing directly the physics), known as the enthalpy formulation. It is a fixed-domain method, in which only the enthalpy (energy) is updated from the conservation law and it determines the phase and temperature. Thus, it is a "front capturing" scheme, as opposed to "front tracking". Its discretization by integrated finite differences (finite volume method) is particularly simple and robust, retaining conservation at the discrete level. It is described and studied in great detail in [2], where the performance of the explicit scheme is compared with two versions of the implicit scheme (S.O.R. and Newton iterations), using the Neumann solution to determine accuracy. In the next section we present a similar comparison with the super-time-stepping scheme.

4. COMPARISON OF NUMERICAL SCHEMES

We test the super-time-stepping scheme and the fully implicit scheme with S.O.R. and with Newton iterations, on PROBLEM 1 and on two choices of the Stefan Number in PROBLEM 2: $St = 0.1$ (slow melting), and $St = 5$ (fast melting), for which the values of Λ in (6) are: 0.189134 and 0.450161, respectively.

We discretize the interval $0 < x < 1$ with $M = 100$ equi-spaced nodes. At the back face $x = 1$ we impose the values of the exact solution (5) or (6) itself, evaluated at $x = 1$ at any desired time. This way, the exact and numerical solutions solve the same problems and comparisons will be valid.

In the super-time-stepping scheme we vary N and ν , whereas in the implicit schemes we increase the time-step Δt in multiples of the explicit time-step $\Delta t_{expl} = \Delta x^2/3$, setting $\Delta t = factor \times \Delta t_{expl}$ with $factor = 5, 10, 20, 40, \dots$. The S.O.R. extrapolation parameter ω is found by trial (linear S.O.R. theory does not apply here), and we report a value of ω that minimized the number of iterations.

To determine the accuracy of each scheme we perform a large number of direct comparisons with the exact solutions. One cannot compare more frequently than N since the sub-steps of a super-step are analogous to iterations of an iterative scheme. We tried to choose the frequency of comparisons so as the total number of comparisons (3rd column of the tables) would be roughly the same on a problem. We calculate three measures of error for each run :

- $maxXerror$ = maximum error in melt-front location over all the comparisons
(only for PROBLEM 2),
- $maxTerror$ = maximum error in Temperature at the fixed locations:
0., .1, .2, ..., .9, 1. (linearly interpolated from nodal values)
over all comparisons, and

Table 1: Performance of the Super-Stepping and Implicit Schemes on the Heat Equation during $0 \leq t \leq 5$, with $M = 100$ nodes							
N	ν	comparisons	maxTerror	max L^1 error	supersteps	time-steps	CPU
STS							
1	0.0	833	0.0006	0.00003	166494	166494	27.8u
7	0.0015	933	0.096	0.015	3729	26103	4.6u
9	0.001	759	0.087	0.022	2280	20495	3.7u
20	0.006	649	0.043	0.086	1301	25968	4.4u
$factor$	ω	comparisons	maxTerror	max L^1 error	time-steps	iterations	CPU
S.O.R.							
20	1.40	834	0.035	0.006	8349	93245	14.1u
40	1.50	834	0.046	0.008	4174	71708	10.6u
80	1.60	695	0.062	0.011	2087	54468	7.9u
160	1.70	1043	0.087	0.018	1043	41263	6.2u
Newton							
40.		834	0.024	0.004	4174	4446	17.3u
80.		695	0.041	0.007	2087	2474	9.8u
160.		521	0.064	0.011	1043	1661	6.5u
200.		834	0.086	0.016	834	1600	6.5u

$\max L^1 \text{ error} = \text{maximum } L^1\text{-error in Temperature at the nodes over}$
all comparisons (computed via trapezoidal rule integration).

In addition, we report the number of supersteps or number of iterations, as well as the total number of time-steps executed in each run. Iterations of the implicit schemes correspond to time-steps of the explicit scheme. Note that since the temperature range is normalized to $-1 \leq u \leq 1$, the temperature errors may be viewed as percent errors. As Δt is increased the errors grow, and only runs with all three errors less than 10% are considered. The computations were performed, in single precision, on a SUN Sparc 10 Model 40 workstation, and we record the CPU units (user time) reported by the UNIX time command. Note that $N = 1, \nu = 0$ is the explicit scheme itself.

Table 2: Performance of the Super-Time-Stepping and Implicit Schemes For $St = 0.1$ (slow melting) during $0 \leq t \leq 5$, with $M = 100$ nodes								
N	ν	comparisons	maxXerror	maxTerror	max L^1 error	supersteps	time-steps	CPU
STS								
1	0.00	833	0.0005	0.038	0.003	166494	166494	35.8u
5	0.006	794	0.01	0.029	0.008	7940	39701	8.6u
10	0.04	834	0.01	0.040	0.014	6673	41721	14.1u
20	0.10	586	0.03	0.029	0.028	5272	66725	21.8u
<i>factor</i>	ω	comparisons	maxXerror	maxTerror	max L^1 error	time-steps	iterations	CPU
S.O.R.								
20	1.50	834	0.006	0.071	0.012	8349	106883	25.9u
40	1.60	834	0.006	0.071	0.012	4174	83530	19.8u
80	1.60	695	0.009	0.067	0.016	2087	60944	14.3u
200	1.70	834	0.009	0.077	0.021	834	41655	9.9u
Newton								
80		695	0.002	0.034	0.006	2087	4859	38.5u
200		834	0.003	0.077	0.019	834	2323	18.6u
320		521	0.004	0.080	0.020	521	1604	12.7u

Table 3: Performance of the Super-Time-Stepping and Implicit Schemes For $St = 5.0$ (fast melting) during $0 \leq t \leq 1$, with $M = 100$ nodes								
N	ν	comparisons	maxXerror	maxTerror	max L^1 error	supersteps	time-steps	CPU
STS								
1	0.0	834	0.004	0.012	0.001	33342	33342	8.0u
5	0.006	794	0.031	0.087	0.056	1589	7946	2.3u
10	0.03	388	0.012	0.023	0.006	1158	11581	2.7u
20	0.12	1155	0.012	0.036	0.047	1156	23121	5.7u
<i>factor</i>	ω	comparisons	maxXerror	maxTerror	max L^1 error	time-steps	iterations	CPU
S.O.R.								
20	1.50	833	0.031	0.066	0.015	1666	29944	7.7u
40	1.60	833	0.076	0.084	0.020	833	21810	5.7u
80	1.70	416	0.117	0.115	0.026	416	16087	4.0u
Newton								
20		841	0.004	0.026	0.008	1683	3536	28.3u
40		841	0.011	0.047	0.013	841	1969	16.0u
80		420	0.014	0.076	0.019	420	1236	9.9u
120		280	0.014	0.105	0.023	280	1018	8.1u

Several observations may be gleaned from the tables.

- The implicit S.O.R. scheme is considerably less accurate than the other schemes. One S.O.R. iteration takes about as long as one explicit step, but it has to perform more iterations than the value of *factor* so it is less efficient than the STS scheme. It becomes competitive only for very large *factors* but at the expense of lower accuracy.
- The implicit Newton scheme is as accurate as the STS scheme and uses far fewer iterations than S.O.R., but each Newton iteration costs so much more that it ends up being slower than S.O.R.
- The Super-Time-Stepping explicit scheme is more accurate and efficient than either one of the implicit schemes. It seems to achieve its best efficiency at around $N = 5$ to $N = 10$ on these problems, same as we found on the much more involved alloy solidification and welding problems. Its accuracy appears to be about an order of magnitude better than that of the implicit schemes and it is achieved at lower cost. In view of the fact that it is also much easier to program and debug (and parallelize) an explicit scheme than an implicit one, we conclude that it is the unquestionable winner in all respects.

ACKNOWLEDGEMENTS

Research partially supported by Science Alliance, a Centers of Excellence Program of the State of Tennessee.

REFERENCES

1. Alexiades, V., Amiez, G., & Gremaud, P.-A., Super-Time-Stepping acceleration of explicit schemes for parabolic problems, *Communications in Numerical Methods in Engineering*, to appear.
2. Alexiades, V., & Solomon, A.D., *Mathematical Modeling of Melting and Freezing Processes*, Hemisphere, Washington DC, 1993.
3. Droux, J.-J., Three-dimensional numerical simulation of solidification by an improved explicit scheme, *Computer Methods in Applied Science & Engineering*, Vol 85 (1991) pp. 57-74.
4. Gentzsch, W., Numerical solution of linear and non-linear parabolic differential equations by a time discretization of third order accuracy, pp. 109-118 in *Proceedings of the 3rd GAMM Conference on Numerical Methods in Fluid Mechanics*, Notes on Numerical Fluid Mechanics, Vol.2, Vieweg, Braunschweig, 1979.
5. van der Houwen, P.J., *Construction of Integration Formulas for Initial Value Problems*, North-Holland, 1977.