A SPLIT STEP APPROACH FOR THE 3-D MAXWELL'S EQUATIONS

JONGWOO LEE AND BENGT FORNBERG

Abstract. Split-step procedures have previously been used successfully in a number of situations, e.g. for Hamiltonian systems, such as certain nonlinear wave equations. In this study, we note that one particular way to write the 3-D Maxwell's equations separates these into two parts, requiring in all only the solution of six uncoupled 1-D wave equations. The approach allows arbitrary orders of accuracy in both time and space, and features in many cases unconditional stability.

1. Introduction

Finite difference time domain (FDTD) solutions of Maxwell's equations are of increasing importance in a wide range of applications. Most FDTD problems feature at least two different spatial scales: a typical wave length, and the size of material objects. In many applications, the former scale is very much larger than the latter. For example, for an integrated circuit operating on a 1 GHz signal, the wave length is around 0.3 m, whereas

Key words and phrases. FDTD, Maxwell's equations, Split-step.

The first author was supported by KOSEF 2000-1-10300-001-5, and by Kwangwoon University during sabbatical year 2001 and the second author supported by NSF grants DMS-9810751 (VIGRE), DMS-0073048, and a Faculty Fellowship from University of Colorado at Boulder.

component/wire sizes are around 10^{-6} m. We naturally want the numerical time step to be determined by accuracy considerations for the time integration, and not be limited by a stability constraint that may be many orders of magnitude smaller (as would be the case, for example for the explicit Yee scheme [14, 13] in the integrated circuit example). As a result, there has been great interest recently in finding numerical schemes which are fast for each time step, but still feature unconditional stability. The ADI-FDTD scheme for the 3-D Maxwell's equations, first proposed in 1999 [17], achieves this (based on an idea introduced independently also in a 2-D version [5]). The purpose - and only novelty - of this present paper is simply to point out that the 3-D Maxwell's equations happens to have a structure that allows a particularly effective implementation of a split-step approach. For each time step, we need only to advance six 1-D wave equations, permitting high computational speed together with unconditional stability. The split-step approach furthermore permits arbitrary orders of accuracy in both space and time.

Split-step procedures have been previously introduced in the context of numerical methods for Hamiltonian systems by Forest and Ruth [1] and Yoshida [15] and have been used successfully in a number of situations, such as a certain nonlinear wave equations [4, 2]. For an introduction and a review on numerical methods for Hamiltonian systems, we refer to [10] and [16].

In the next section, we first briefly recall some well-known features of the split-step approach. We then write down the Maxwell's equations in the particular form which allows their decomposition into six scalar 1-D wave equations. In Section 3, we formulate a simple test example, and we then present a numerical implementation. In order to display temporal errors most clearly, we have here chosen a spectral approach for the resulting 1-D problems (to eliminate spatial discretization errors from the results). In the applications that we envisage, very fine geometrical features will force the use of very fine spatial grids, so truncation errors will indeed come mainly from the time integration even when lower-order methods (with greater boundary flexibility) are used in space. It is of a particular interest to compare time stepping schemes of increasing orders of accuracy. A great number of generalizations are possible, both for the present split-step approach, and also for the ADI-FDTD scheme. Some of these are discussed briefly in the concluding section, and will be presented in much greater detail in a separate study [3]. Since split-step methods are still relatively little known outside the field of symplectic integrators for Hamiltonian systems, a brief appendix is included which may be useful in conveying a heuristic perspective on higher order split-step methods.

2. Numerical procedure

The two main components of the present algorithm are now described in turn.

2.1. **General high-order split-step method.** In its simplest form, featuring only first order accuracy in time, one advances an ODE (or, as in our present application, a system of ODEs)

$$u_t = A(u) + B(u) \tag{2.1}$$

from time t to time $t + \Delta t$ by successively solving

$$\begin{aligned} u_t &= 2A(u), \qquad \text{from } \ t \ \text{ to } t + \tfrac{1}{2}\Delta t, \ \text{ followed by} \\ u_t &= 2B(u), \qquad \text{from } t + \tfrac{1}{2}\Delta t \ \text{ to } \ (t + \tfrac{1}{2}\Delta t) + \tfrac{1}{2}\Delta t = t + \Delta t. \end{aligned}$$

Here, $A(u)$ and $B(u)$ can be very general nonlinear operators (in particular, there is no requirement that A and B commute). The two time increments are each of length $\frac{1}{2}\Delta t$; we denote this by $\left\{\frac{1}{2}, \frac{1}{2}\right\}$. One obtains second-order accuracy in time by instead alternating the two equations in the pattern A, B, A while using the time increments $\left\{\frac{1}{4}, \frac{1}{2}, \frac{1}{4}\right\}$ - known as "Strang splitting" [11]. In 1990 Yoshida [15] devised a systematic way to obtain similar split-step methods of higher order. From an implementation standpoint, one simply chooses certain longer time increment sequences, while again alternating A, B, A, B, ... Table 2.1 shows the coefficients of methods of orders 1, 2, 4, 6, and 8. Although the coefficients are in general not unique, it can be shown that methods of orders above 2 will need to feature at least some negative time increments [12].

In some applications, the main benefit of the split-step approach is to separate an ODE which lacks a closed-form solution into a couple of ODEs which both allow such solutions. The approach has also been used for PDEs

TABLE 2.1. Coefficients of Split-Step methods. The number n in SSn stands for the order of temporal accuracy of a split step method. In general coefficients for SS6 and SS8 are not unique. See the Appendix for details on this.

METHOD	TIME INCREMENT SEQUENCE		
SS1	0.50000 00000 00000 00000	0.50000 00000 00000 00000	
SS2	0.25000 00000 00000 00000	0.50000 00000 00000 00000	0.25000 00000 00000 00000
SS4	0.33780 17979 89914 40851	0.67560 35959 79828 81702	-0.08780 17979 89914 40851
	-0.85120 71919 59657 63405	-0.08780 17979 89914 40851	0.67560 35959 79828 81702
	0.33780 17979 89914 40851		
SS6	0.19612 84026 19389 31595	0.39225 68052 38778 63191	0.25502 17059 59228 84938
	0.11778 66066 79679 06684	-0.23552 66927 04878 21832	-0.58883 99920 89435 50347
	0.03437 65841 26260 05298	0.65759 31603 41955 60944	0.03437 65841 26260 05298
	-0.58883 99920 89435 50347	-0.23552 66927 04878 21832	0.11778 66066 79679 06684
	0.25502 17059 59228 84938	0.39225 68052 38778 63191	0.19612 84026 19389 31595
SS8	0.22871 10615 57447 89169	0.45742 21231 14895 78337	0.29213 43956 99000 73022
	0.12684 66682 83105 67707	-0.29778 97250 73598 45089	-0.72242 61184 30302 57885
	-0.40077 32180 57163 83322	-0.07912 03176 84025 08760	0.44497 46255 63618 95284
	0.96906 95688 11262 99329	-0.00561 77738 38196 20526	-0.98030 51164 87655 40380
	-0.46445 25958 95878 59173	0.05139 99246 95898 22035	0.45281 32300 44769 50634
	0.85422 65353 93640 79233	0.45281 32300 44769 50634	0.05139 99246 95898 22035
	-0.46445 25958 95878 59173	-0.98030 51164 87655 40380	-0.00561 77738 38196 20526
	0.96906 95688 11262 99329	0.44497 46255 63618 95284	-0.07912 03176 84025 08760
	-0.40077 32180 57163 83322	-0.72242 61184 30302 57885	-0.29778 97250 73598 45089
	0.12684 66682 83105 67707	0.29213 43956 99000 73022	0.45742 21231 14895 78337
	0.22871 10615 57447 89169		

and, in particular, for certain nonlinear wave equations (e.g., [2, 4]). For example, the Korteweg-de Vries (KdV) equation $u_t + uu_x + u_{xxx} = 0$ splits into $u_t + 2uu_x = 0$ (which features few numerical difficulties) and $u_t + 2u_{xxx} = 0$ (which is linear and can be solved analytically, thereby bypassing otherwise severe CFL restrictions). We will next see that we can split the 3-D Maxwell's equations in such a way that it decomposes into purely 1-D sub-problems.

2.2. **Split-step formulation of the 3-D Maxwell's equations.** For a linear homogeneous medium with permittivity ε and permeability μ, the curl equations for the 3-D Maxwell's equations with no charges and currents is written by

$$\frac{\partial \mathbf{E}}{\partial t} = \frac{1}{\varepsilon} \nabla \times \mathbf{H}, \tag{2.2a}$$

$$\frac{\partial \mathbf{H}}{\partial t} = -\frac{1}{\mu} \nabla \times \mathbf{E}, \tag{2.2b}$$

where $\mathbf{E} = (E_x, E_y, E_z)$ and $\mathbf{H} = (H_x, H_y, H_z)$. Writing out the equations (2.2) in component form and changing the order of the terms in the right hand side for the last three equations gives

$$\frac{\partial}{\partial t} \begin{bmatrix} E_x \\ E_y \\ E_z \\ H_x \\ H_y \\ H_z \end{bmatrix} = \begin{bmatrix} \frac{1}{\varepsilon}\frac{\partial}{\partial y} H_z \\ \frac{1}{\varepsilon}\frac{\partial}{\partial z} H_x \\ \frac{1}{\varepsilon}\frac{\partial}{\partial x} H_y \\ \frac{1}{\mu}\frac{\partial}{\partial z} E_y \\ \frac{1}{\mu}\frac{\partial}{\partial x} E_z \\ \frac{1}{\mu}\frac{\partial}{\partial y} E_x \end{bmatrix} + \begin{bmatrix} -\frac{1}{\varepsilon}\frac{\partial}{\partial z} H_y \\ -\frac{1}{\varepsilon}\frac{\partial}{\partial x} H_z \\ -\frac{1}{\varepsilon}\frac{\partial}{\partial y} H_x \\ -\frac{1}{\mu}\frac{\partial}{\partial y} E_z \\ -\frac{1}{\mu}\frac{\partial}{\partial z} E_x \\ -\frac{1}{\mu}\frac{\partial}{\partial x} E_y \end{bmatrix} \tag{2.3}$$

or shorter

$$\frac{\partial u}{\partial t} = Au + Bu. \tag{2.4}$$

Here u denotes the vector $(E_x, E_y, E_z, H_x, H_y, H_z)^T$. The split-step approach leads us to repeatedly advance $\frac{\partial u}{\partial t} = 2Au$ and $\frac{\partial u}{\partial t} = 2Bu$ by certain time increments. These two subproblems can be written out explicitly as

shown below. The key point of this paper is to note that each of the sub-problems amounts to three pairs of mutually uncoupled 1-D equations:

$$\left\{ \left\{ \begin{array}{l} \frac{\partial E_x}{\partial t} = \frac{2}{\varepsilon}\frac{\partial H_z}{\partial y} \\ \\ \frac{\partial H_z}{\partial t} = \frac{2}{\mu}\frac{\partial E_x}{\partial y} \end{array} \right\} \left\{ \begin{array}{l} \frac{\partial E_y}{\partial t} = \frac{2}{\varepsilon}\frac{\partial H_x}{\partial z} \\ \\ \frac{\partial H_x}{\partial t} = \frac{2}{\mu}\frac{\partial E_y}{\partial z} \end{array} \right\} \left\{ \begin{array}{l} \frac{\partial E_z}{\partial t} = \frac{2}{\varepsilon}\frac{\partial H_y}{\partial x} \\ \\ \frac{\partial H_y}{\partial t} = \frac{2}{\mu}\frac{\partial E_y}{\partial x} \end{array} \right\} \right\}, \quad \left\{ \left\{ \begin{array}{l} \frac{\partial E_x}{\partial t} = -\frac{2}{\varepsilon}\frac{\partial H_y}{\partial z} \\ \\ \frac{\partial H_y}{\partial t} = -\frac{2}{\mu}\frac{\partial E_x}{\partial z} \end{array} \right\} \left\{ \begin{array}{l} \frac{\partial E_y}{\partial t} = -\frac{2}{\varepsilon}\frac{\partial H_z}{\partial x} \\ \\ \frac{\partial H_z}{\partial t} = -\frac{2}{\mu}\frac{\partial E_y}{\partial x} \end{array} \right\} \left\{ \begin{array}{l} \frac{\partial E_z}{\partial t} = -\frac{2}{\varepsilon}\frac{\partial H_x}{\partial y} \\ \\ \frac{\partial H_x}{\partial t} = -\frac{2}{\mu}\frac{\partial E_z}{\partial y} \end{array} \right\} \right\}. \tag{2.5}$$

Each of the 1-D subsystems in (2.5) can easily be solved numerically. If we choose a method which preserves the L^2-norm for each 1-D sub-problem, the sum of the squares of all the unknowns will be preserved through each sub-step, and therefore also throughout the complete computation. Unconditional numerical stability is then assured.

3. Numerical implementation and test problem

The example below is designed to confirm - in the simplest possible setting - that the split-step approach indeed produces the expected high order in

time solutions to the full 3-D Maxwell's equations. Several generalizations and enhancements are discussed in the conclusion.

For the numerical test, we choose as the computational domain a 3-D periodic cube of size $[0, L]$ in each direction. Each unknown variable is numerically represented by an $N \times N \times N$ cube of discrete values, giving a space step $h = L/N$ in each direction.

In order to compare efficiency of proposed time split-step methods with that of a traditional FDTD scheme, we also implemented the Yee scheme for the 3-D Maxwell's equations (2.2), see [14, 13], however using pseudospectral implementation for the spatial approximation. We will denote this version of time stepping method by YS.

3.1. **Solution of the linear 1-D subproblems.** Various numerical methods are available for solving the 1-D linear subproblems in (2.5). Since we wish to focus on the time stepping in this study, we note that the subproblems can also be solved analytically both over an infinite continuous interval and over a finite discrete periodic interval. We can write a generic case of the equations in (2.5) as

$$\begin{cases} \dfrac{\partial u}{\partial t} = \alpha \dfrac{\partial v}{\partial x} \\ \\ \dfrac{\partial v}{\partial t} = \beta \dfrac{\partial u}{\partial x} \end{cases} \tag{3.1}$$

with initial conditions

$$\begin{cases} u(x,0) = u_0(x) \\ v(x,0) = v_0(x) \end{cases},$$

where α and β are assumed to be constants of the same sign. The D'Alembert solution for an infinite interval becomes

$$\begin{cases} u(x,t) = \frac{1}{2}\left\{u_0(x+ct) + u_0(x-ct) + \mathrm{sgn}(\alpha)\sqrt{\frac{\alpha}{\beta}}\,[v_0(x+ct) - v_0(x-ct)]\right\} \\ v(x,t) = \frac{1}{2}\left\{v_0(x+ct) + v_0(x-ct) + \mathrm{sgn}(\beta)\sqrt{\frac{\beta}{\alpha}}\,[u_0(x+ct) - u_0(x-ct)]\right\} \end{cases},$$

where $c = \sqrt{\alpha\beta}$. In the spatially discrete L-periodic case, the solution is best described in discrete Fourier space:

$$\begin{cases} \widehat{u}(\omega,t) = \widehat{u}_0(\omega)\cos\delta_\omega ct + i\ \mathrm{sgn}(\alpha)\sqrt{\frac{\alpha}{\beta}}\ \widehat{v}_0(\omega)\sin\delta_\omega ct \\ \widehat{v}(\omega,t) = \widehat{v}_0(\omega)\cos\delta_\omega ct + i\ \mathrm{sgn}(\beta)\sqrt{\frac{\beta}{\alpha}}\ \widehat{u}_0(\omega)\sin\delta_\omega ct \end{cases}, \qquad (3.2)$$

where $\delta_\omega = \frac{2\pi\omega}{L}$ for $\omega = 0, 1, \ldots, \frac{N}{2}$, and $\delta_\omega = \frac{2\pi(\omega-N)}{L}$ for $\omega = \frac{N}{2}, \frac{N}{2} + 1, \ldots, N-1$. To advance the discrete solution of (3.1) forward (say, again, from time 0 to time t), we first convert the discrete initial data from physical- to Fourier space, then apply (3.2) and finally return again to physical space. The discrete Fourier transforms (DFTs) needed for this

$$\begin{cases} \widehat{f}(\omega) = \dfrac{1}{N}\displaystyle\sum_{j=0}^{N-1} f(x_j)\ e^{-i\frac{2\pi}{N}j\,\omega}, \quad \omega = 0, 1, \ldots, N-1, \\ f(x_j) \ = \ \displaystyle\sum_{\omega=0}^{N-1} \widehat{f}(\omega)\ e^{i\frac{2\pi}{N}j\,\omega}, \quad j = 0, 1, \ldots, N-1 \end{cases}$$

are best carried out by the FFT algorithm.

3.2. **Some exact solutions of the periodic 3-D Maxwell's equations.**

Let $\mathbf{k} = \frac{2\pi}{L}(k, \ell, m)$ for given integers k, ℓ, m such that $k^2 + \ell^2 + m^2 \neq 0$ and $\mathbf{n}$ a nonzero vector in the space such that $\mathbf{k} \cdot \mathbf{n} = 0$. If we let $\mathbf{r} = (x, y, z)$, then the following functions constitute a set of exact solutions of the 3-D

Maxwell's equations (2.3) of spatial period L in each direction.

$$\mathbf{E} = (E_x, E_y, E_z) = E_0 \cos\left(\mathbf{k}\cdot\mathbf{r} - |\mathbf{k}|ct + \delta\right)\ \hat{\mathbf{n}}, \tag{3.3a}$$

$$\mathbf{H} = (H_x, H_y, H_z) = E_0\sqrt{\frac{\varepsilon}{\mu}} \cos\left(\mathbf{k}\cdot\mathbf{r} - |\mathbf{k}|ct + \delta\right)\ (\hat{\mathbf{k}}\times\hat{\mathbf{n}}), \tag{3.3b}$$

where E_0 and δ are constants, and $c = 1/\sqrt{\varepsilon\mu}$ and $\hat{\mathbf{k}} = \mathbf{k}/|\mathbf{k}|$ and $\hat{\mathbf{n}} = \mathbf{n}/|\mathbf{n}|$. The above equations is an example of electric and magnetic fields in a plane wave with propagation vector $\hat{\mathbf{k}}$ and polarization vector $\hat{\mathbf{n}}$.

We use the following solutions as our test example, which can be obtained from (3.3) with $L = 1$, $\mathbf{k} = 2\pi(1,1,1)$, $\mathbf{n} = (1,-2,1), \delta = 0$ and $E_0 = \sqrt{6}$.

$$E_x = \cos(2\pi(x+y+z) - 2\sqrt{3}\pi ct), \qquad H_x = \sqrt{3}\sqrt{\frac{\varepsilon}{\mu}}E_x, \tag{3.4a}$$

$$E_y = -2E_x, \qquad H_y = 0, \tag{3.4b}$$

$$E_z = E_x, \qquad H_z = -\sqrt{3}\sqrt{\frac{\varepsilon}{\mu}}E_x. \tag{3.4c}$$

3.3. **Numerical results.** We approximate the solutions of the 3-D Maxwell's equations (2.3) by using time split-step methods in Table 2.1 based on (2.5) and (3.2), and by the YS method on the domain $[0,1]^3$. In these numerical experiments, we use the functions given by (3.4) with $\varepsilon = \mu = 1$ and $t = 0$ as the initial condition. For the spatial step size, we use $h = \Delta x = \Delta y = \Delta z = 1/N$, where N is a power of 2. For the temporal step size we use $\Delta t = 1/M$, where M is the number of time steps from $t = 0$ to $t = 1$. We measure the spatial accuracy of a numerical method using maximum errors

between the approximated solutions and the exact solutions (3.4) over the $N \times N \times N$ gridpoints in the computational spatial domain at $t = 1$.

Since there are several 6th and 8th order split-step methods, i.e., there are sequences of coefficients other than those in Table 2.1, we have checked convergence and accuracy of those methods with $N = 32$. See the Appendix and [15] for the issues of multiplicity of such sequences. We describe below tests with different split-step methods of 6th and 8th order. We denote them by SS6a, SS6b and SS6c for 6th order methods, and by SS8a, ..., SS8g for 8th order methods. Figures 3.1 and 3.2 show their convergence and accuracy as a function of the number M of time steps. Different sequences of coefficients give quite different accuracy for the same number of time steps, but the convergence rates are always the same as predicted.

Figure 3.1 is placed near here.

Figure 3.2 is placed near here.

In this test we find that the SS6a and SS8d methods are the best for 6th and 8th order methods, respectively, in terms of cost effectiveness, as shown in Figures 3.1 and 3.2. The sequences SS6 and SS8 of coefficients in Table 2.1 are those for the SS6a and SS8d methods, respectively. In the remaining part of our numerical experiments, we use the sequences in Table 2.1. We should also note that it is possible to use sequences of 6th and 8th order which contain more than the numerical number of steps. At a slightly larger computational cost per time step, the error constant can be somewhat lower still; see [7].

Figure 3.3 shows the spatial accuracy of split-step methods and the YS method at the final time $t = 1$ in terms of maximum errors as a function of the number M of time steps. It confirms the rate of convergence of each method.

In order to make a fair comparison of the effectiveness of each method, we need to measure the *computational cost* of each method to obtain a certain spatial accuracy at the final time $t = 1$ (i.e., to compensate for the fact that higher order split-step methods use a larger number of substeps). We estimate the cost of each method in terms of the total number of FFTs carried out to obtain approximated solutions of a certain accuracy to the 3-D Maxwell's equations at the final time.

Figure 3.3 is placed near here.

Figure 3.4 is placed near here.

Figure 3.4 shows total numbers of FFTs of the proposed time split-step methods and for the YS scheme carried out to get a certain accuracy at the final time. In most cases split-step methods of orders between 4 and 8 are found to be most effective, significantly improving on traditional 2nd order time stepping methods (as for example, the YS method).

In cases where the waves are highly over-resolved in order to capture fine geometrical structures, CFL restrictions will for the Yee scheme force extremely fine time steps. The lower cut-off of the YS-curves in Figures 3.3 and 3.4 will then move far to the right (in proportion to N; i.e., with possibly several orders of magnitude). On the other hand, our proposed split-step

method removes any need to reduce the temporal step size even if the spatial geometry requires a heavily dense gridding.

4. Conclusions

In this study we have presented an easily implemented high order and unconditionally stable time stepping strategy for the 3-D Maxwell's equations and tested it numerically in the special case of periodic geometry. We also compared our proposed time stepping methods with a traditional 2nd order time stepping method YS. In most cases, time split-step methods of orders between 4 and 8 are found to be most effective, significantly improving on traditional 2nd order time stepping methods. The SS6 method appears to be a particularly attractive choice.

With these results established, many further directions of study become available;

1. The substeps in the split-step procedure need not be solved with higher than second order of accuracy in time (so for example, a simple Crank-Nicolson approach suffices for the 1-D wave equations).
2. Entirely different approaches(than split-step) are available to reach higher orders in time combined with unconditional stability, e.g., enhancing ADI-FDTD or SS2 with Richardson extrapolation [9] or with deferred correction [8].
3. The best way to implement boundary conditions in split-step methods can vary from case to case, and should be investigated.

4. The Maxwell's equations (2.2) imply conservation of $\nabla \cdot \mathbf{E}$ and $\nabla \cdot \mathbf{H}$. The numerical conservation of the corresponding quantities can vary between schemes, and need to be studied.

Regarding issue 1 above, the key result is obtained by Yoshida [15], who showed that a $2p$th-order accurate method S_{2p} can be obtained using the following formula:

$$S_{2p}(\tau) = S_2(w_k\tau)\, S_2(w_{k-1}\tau) \cdots S_2(w_1\tau)\, S_2(w_0\tau)\, S_2(w_1\tau) \cdots S_2(w_{k-1}\tau)\, S_2(w_k\tau).$$

Here the constants $w_k, k = 0, 1, \ldots, 2^{p-1} - 1$ satisfy certain nonlinear systems of algebraic equations, and S_2 is a method of second order accurate in time. For the relations between current split-step and the above equation, see [15]. See also [4, 7]. The comparison suggested under item 2 will be presented in our next study [3]. Items 3 and 4 remain to be investigated.

5. Appendix

This appendix is intended to convey heuristically the basic idea behind the high-order split-step concept. We will also make some comments on 6th and 8th order methods and their coefficients as shown in Table 2.1.

5.1. **Concept behind high-order split-step methods.** We can for simplicity think of $A(u)$ and $B(u)$ in equation (2.1) as matrices A and B times a vector u (the split-step procedure allows for more general nonlinear operators, but in our present context, A and B will become discrete linear operators, i.e., matrices, following the space discretization). We can write

the analytic solution of the system (2.1) of ODEs as

$$u(t) = e^{(A+B)\,t}u(0),$$

where we by $e^{(A+B)\,t}$ mean

$$\begin{aligned} e^{(A+B)\,t} &= I + t(A+B) + \tfrac{t^2}{2!}(A+B)^2 + \cdots \\ &= I + t(A+B) + \tfrac{t^2}{2!}(A^2 + AB + BA + B^2) + \cdots \end{aligned} \tag{5.1}$$

(taking note of the fact that A and B in general do not commute). The solution operator using SS1 amounts to replacing the exact operator $e^{(A+B)\,t}$ by

$$\begin{aligned} e^{Bt}e^{At} &= (I + tB + \tfrac{t^2}{2!}B^2 + \cdots)(I + tA + \tfrac{t^2}{2!}A^2 + \cdots) \\ &= I + t(A+B) + \tfrac{t^2}{2!}(A^2 + 2BA + B^2) + \cdots . \end{aligned} \tag{5.2}$$

The expansion in (5.2) differs from the one in (5.1) first in the t^2-term. This tells that this particular splitting is only first order accurate. Carrying out the equivalent expansion for the SS2 scheme gives

$$\begin{aligned} e^{\frac{1}{2}At}e^{Bt}e^{\frac{1}{2}At} &= (I + \tfrac{t}{2}A + \tfrac{t^2}{8}A^2 + \cdots)(I + tB + \tfrac{t^2}{2}B^2 + \cdots)(I + \tfrac{t}{2}A + \tfrac{t^2}{8}A^2 + \cdots) \\ &= I + t(\tfrac{1}{2}A + B + \tfrac{1}{2}A) + \\ &\qquad t^2(\tfrac{1}{8}A^2 + \tfrac{1}{2}AB + \tfrac{1}{4}A^2 + \tfrac{1}{8}A^2 + \tfrac{1}{2}BA + \tfrac{1}{2}B^2) + \cdots \\ &= I + t(A+B) + \tfrac{t^2}{2!}(A^2 + AB + BA + B^2) + \cdots . \end{aligned}$$

This agrees with (5.1) throughout the t^2-term, ensuring that the SS2 approximation indeed is accurate to second order.

TABLE 5.1. Values of w_1, w_2, w_3 for 6th order Split-Step methods.

	SS6a	SS6b	SS6c
w_1	-1.17767 99841 78871 00695	-2.13228 52220 01451 52071	0.00152 88622 84249 27025
w_2	0.23557 32133 59358 13368	0.00426 06818 70792 01620	-2.14403 53163 05389 31060
w_3	0.78451 36104 77557 26382	1.43984 81679 76783 09093	1.44778 25623 99297 93290

A similar verification in the case of SS4 will produce an expansion that reproduces (5.1) also through the next two powers of t (not displayed in (5.1)). The SS4 scheme was originally found (via direct algebraic expansions similar to the ones above) in 1987 [6]. Closed form expressions are in this case available for the fractional step lengths c_i and d_i in the expansion

$$e^{(A+B)t} = e^{c_1 At} e^{d_1 Bt} e^{c_2 At} e^{d_2 Bt} e^{c_3 At} e^{d_3 Bt} e^{c_4 At} + O(t^5),$$

namely

$$c_1 = c_4 = \frac{1}{2(2 - 2^{1/3})}, c_2 = c_3 = (1-2^{1/3})c_1, \quad d_1 = d_3 = 2c_1, \;\; d_2 = -2^{4/3}c_1.$$

The key contributions by Yoshida were to demonstrate that it is possible to find sequences of time increments that give time stepping accuracies of any order, and also to device a practical algorithm for computing these sequences of increments. For details on this, see the original reference [15].

5.2. **Coefficients of split-step methods of order 6 and 8.** The coefficients for 6th and 8th order split-step methods in Table 2.1 are not unique. They are obtained from a (numerical) solution of a set of m simultaneous algebraic equations of m unknowns $w_1, w_2, \ldots, w_m$, where $m = 3$ and $m = 7$ for 6th and 8th order split-step methods, respectively. The coefficient sequence $\{c_1, d_1, \ldots, d_{2m+1}, c_{2m+2}\}$ of 6th and 8th order split-step methods

TABLE 5.2. Values of $w_\ell, \ell = 1, \ldots, 7$ for 8th order Split-Step methods

	SS8a	SS8b	SS8c
w_1	-1.61582 37415 00653 78479	-0.00169 24858 77717 06559	0.31179 08124 18466 77255
w_2	-2.44699 18237 04245 88929	2.89195 74431 58173 91248	-1.55946 80382 14702 58821
w_3	-0.00716 98941 97095 33210	0.00378 03958 83626 68224	-1.67896 92825 96738 47121
w_4	2.44002 73261 66344 06382	-2.89688 25033 04239 86859	1.66335 80996 33505 17976
w_5	0.15773 99281 23708 32134	2.89105 14897 21989 00061	-1.06458 71478 91825 07054
w_6	1.82020 63097 06980 06933	-2.33864 81510 10419 43098	1.36934 94641 68817 70056
w_7	1.04242 62086 99704 26435	1.48819 22920 29213 10080	0.62903 06502 10370 79990
	SS8d	SS8e	SS8f
w_1	0.10279 98493 91796 44070	0.02277 38840 12631 22598	2.70742 45624 18790 06024
w_2	-1.96061 02329 75310 80761	2.52778 92731 80283 39183	-2.45896 67162 13455 99441
w_3	1.93813 91376 22525 98658	-0.07191 80053 65070 50748	2.64024 54464 98768 20351
w_4	-0.15824 06353 68050 17520	0.00536 01892 13752 38083	-2.36286 67205 68674 20427
w_5	-1.44485 22368 60605 15769	-2.04809 79588 34902 05644	1.58972 50062 60202 77710
w_6	0.25369 33365 66211 35415	0.10799 04677 18098 27961	-1.78020 65575 56607 60924
w_7	0.91484 42462 29791 56675	1.30300 16575 75168 38491	1.64725 65223 24468 23179
	SS8g		
w_1	2.91407 22457 15539 86303		
w_2	-2.36086 97991 65913 18673		
w_3	-1.96008 27978 85508 98896		
w_4	-0.80818 66671 15686 57282		
w_5	0.80309 13815 81478 63885		
w_6	1.95798 64664 14544 17459		
w_7	1.50667 98223 69523 23625		

for one unit time increment in Table 2.1 are obtained from the following relations:

$$d_1 = d_{2m+1} = \frac{1}{2}w_m, \quad d_2 = d_{2m} = \tfrac{1}{2}w_{m-1}, \ \ldots, \ d_m = d_{m+2} = \tfrac{1}{2}w_1, \ d_{m+1} = \tfrac{1}{2}w_0,$$

$$c_1 = c_{2m+2} = \frac{1}{4}w_m, \quad c_2 = c_{2m+1} = \tfrac{1}{4}(w_m + w_{m-1}), \ \ldots, \ c_{m+1} = c_{m+2} = \tfrac{1}{4}(w_1 + w_0),$$

where $w_0 = 1 - 2(w_1 + \cdots + w_m)$. See [15] for details.

In Table 5.2, we present two more solutions SS8f and SS8g, not listed in [15], for $\{w_\ell\}$(found by using Newton's method). The sequences for SS6 and SS8 in Table 2.1 correspond to SS6a and SS8d in Table 5.1 and 5.2, respectively.

References

[1] E. Forest and R.D. Ruth, Fourth order symplectic integration, *Phys. D*, **43** (1990), 105–117.

[2] B. Fornberg and T.A. Driscoll, A fast spectral algorithm for nonlinear wave equations with linear dispersion, *J. Comput. Phys.*, **155** (1999), 456–467.

[3] J. Lee and B. Fornberg, Some unconditionally stable time stepping methods for the 3-D Maxwell's equations, In preparation.

[4] R. McLachlan, Symplectic integration of Hamiltonian wave equations, *Numer. Math.*, **66** (1994), 465–492.

[5] T. Namiki, A new FDTD algorithm based on alternating-direction implicit method, *IEEE Trans. Microwave Theory Tech.*, **47** (1999), 2003-2007.

[6] F. Neri, Lie algebras and canonical integration, Dept. of Physics, University of Maryland, Preprint, 1987.

[7] D.I. Okunbor and R.D. Skeel, Canonical Runge-Kutta-Nyström methods of order five and six, *J. Comput. Applied Math.*, **51** (1994), 375–382.

[8] V. Pereyra, Accelerating the convergence of discretization algorothm, *SIAM J. Numer. Anal.*, **4** (1967), 508–532.

[9] L. Richardson, The deferred approach to the limit, *Phil. Trans. A,* **226** (1927), 299–349.

[10] J.M. Sanz-Serna and M.P. Calvo, *Numerical Hamiltonian Problems.* Chapman & Hall, London, UK, 1994.

[11] G. Strang, On construction and comparison of difference schemes, *SIAM J. Numer. Anal.*, **5** (1968), 506–516.

[12] M. Suzuki, General theory of fractal path integrals with applications to many-body theories and statistical physics, *J. Math. Phys.*, **32** (1991), 400–407.

[13] A. Taflove and S. Hagness, *Computational Electrodynamics: The Finite-Difference Time-Domain Method*. Artech House, Boston, MA, 2nd ed., 2000.

[14] K.S. Yee, Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media, *IEEE Trans. Antennas Propagat.*, **14** (1966), 302–307.

[15] H. Yoshida, Construction of higher order symplectic integrators, *Phys. Lett. A*, **150** (1990), 262–268.

[16] ———, Recent Progress in the theory and application of symplectic integrators, *Celes. Mech. Dyn. Astr.*, **56** (1993), 27–43.

[17] F. Zheng, Z. Chen, and J. Zhang, Toward the development of a three-dimensional unconditionally stable finite-difference time-domain method, *IEEE Trans. Microwave Theory Tech.*, **48** (2000), 1550–1558.

Kwangwoon University, Department of Mathematics, 447-1 Wolgye-Dong, Nowon-Gu, Seoul 139-701, Korea

E-mail address: jlee@mail.gwu.ac.kr

University of Colorado, Department of Applied Mathematics, 526 UCB, Boulder, CO 80309, USA

E-mail address: fornberg@colorado.edu

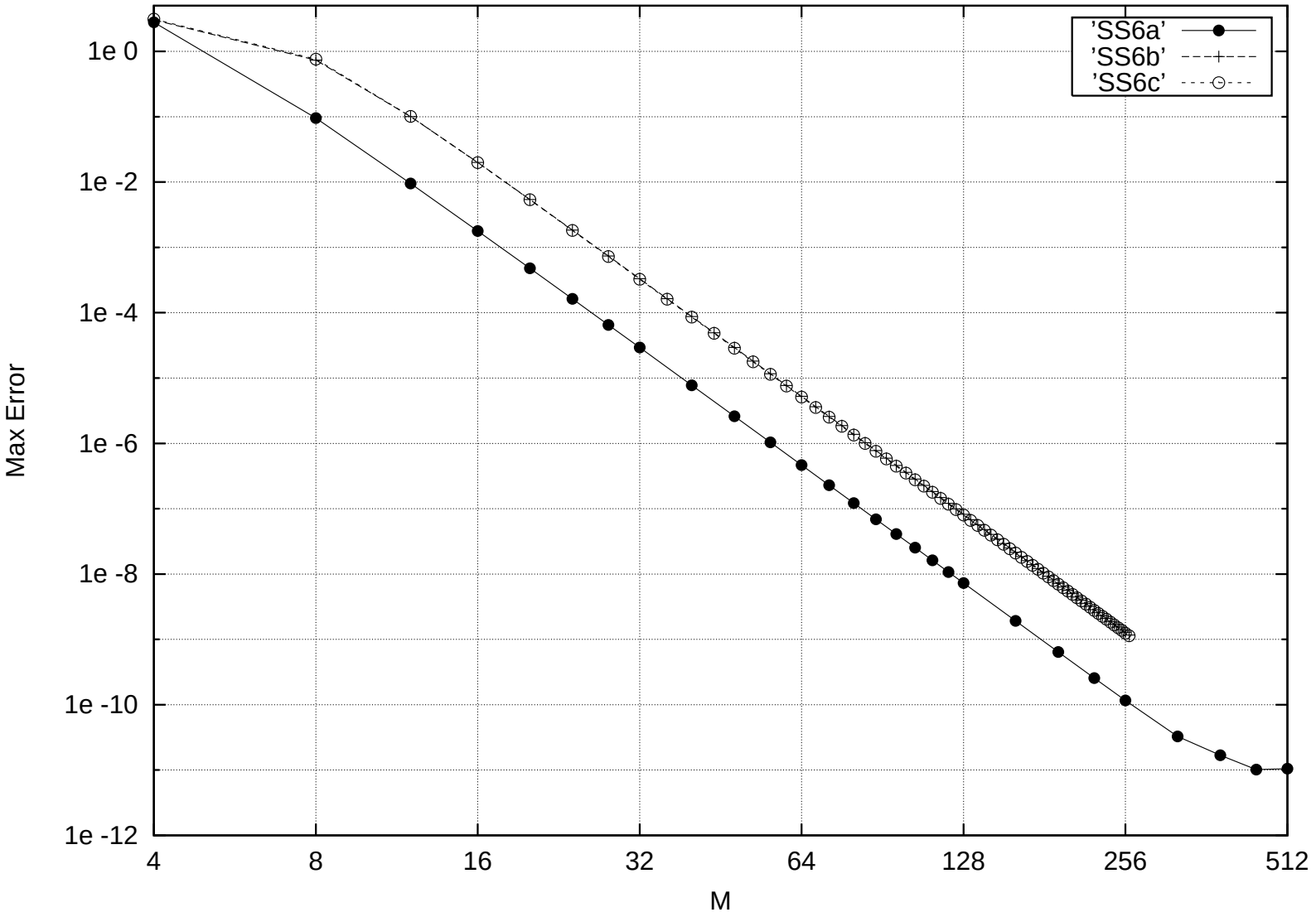

Figure 3.1: The maximum spatial errors of 6th order spilt-step methods SS6a, SS6b, and SS6c for $N = 32$. Actual computed data are marked by various symbols. The vertical axis denotes the spatial accuracy (in $\log_{10}$ scale) at time $t = 1$. The horizontal axis denotes the number M of time steps (in $\log_2$ scale). See the Appendix for some explanations on 6th order split-step methods.

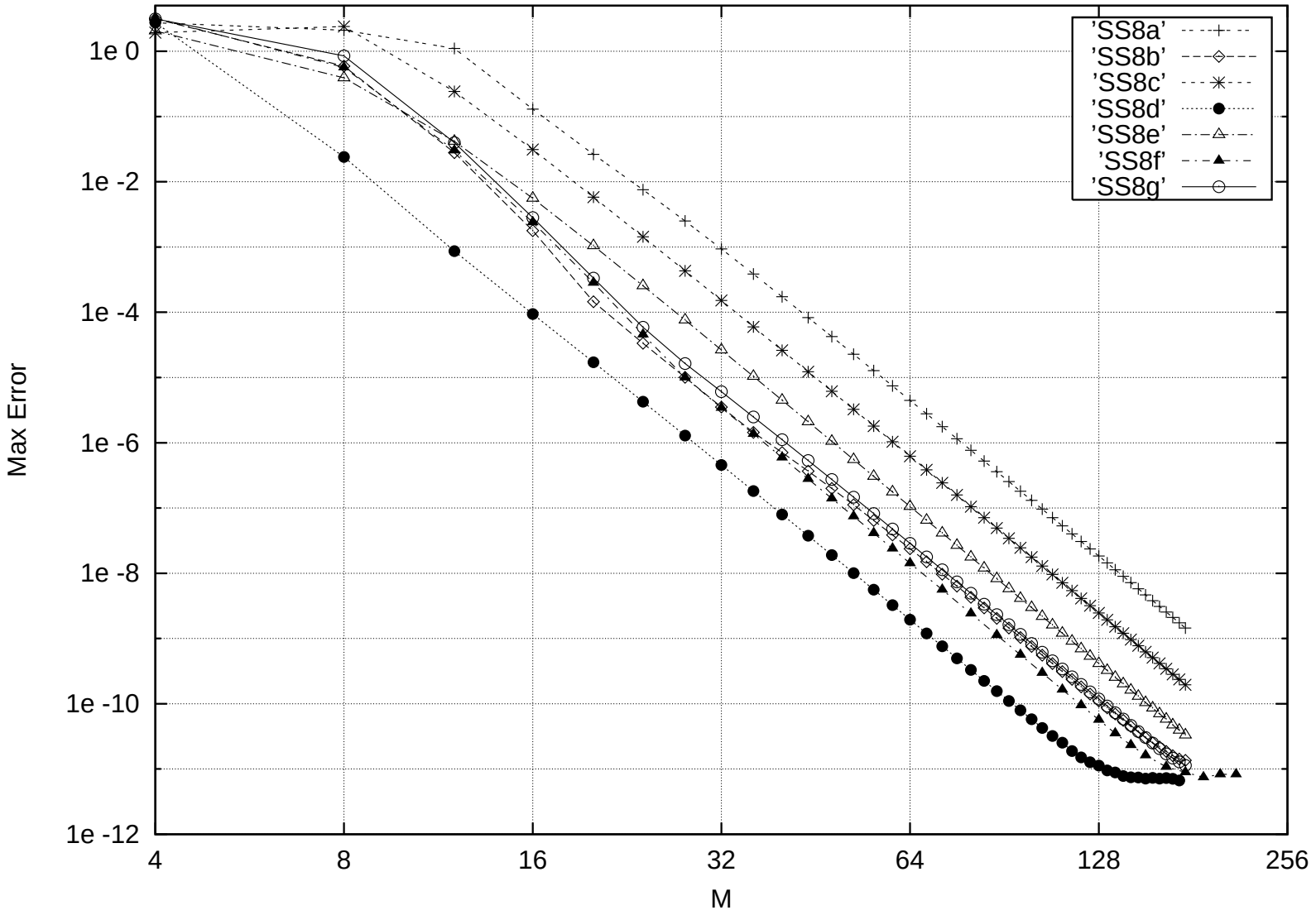

Figure 3.2: The same as Figure 3.1 but for 8th order split-step methods SS8a, SS8b, SS8c, SS8d, SS8e, SS8f, and SS8g for $N = 32$.

See the Appendix for some explanations on 8th order split-step methods.

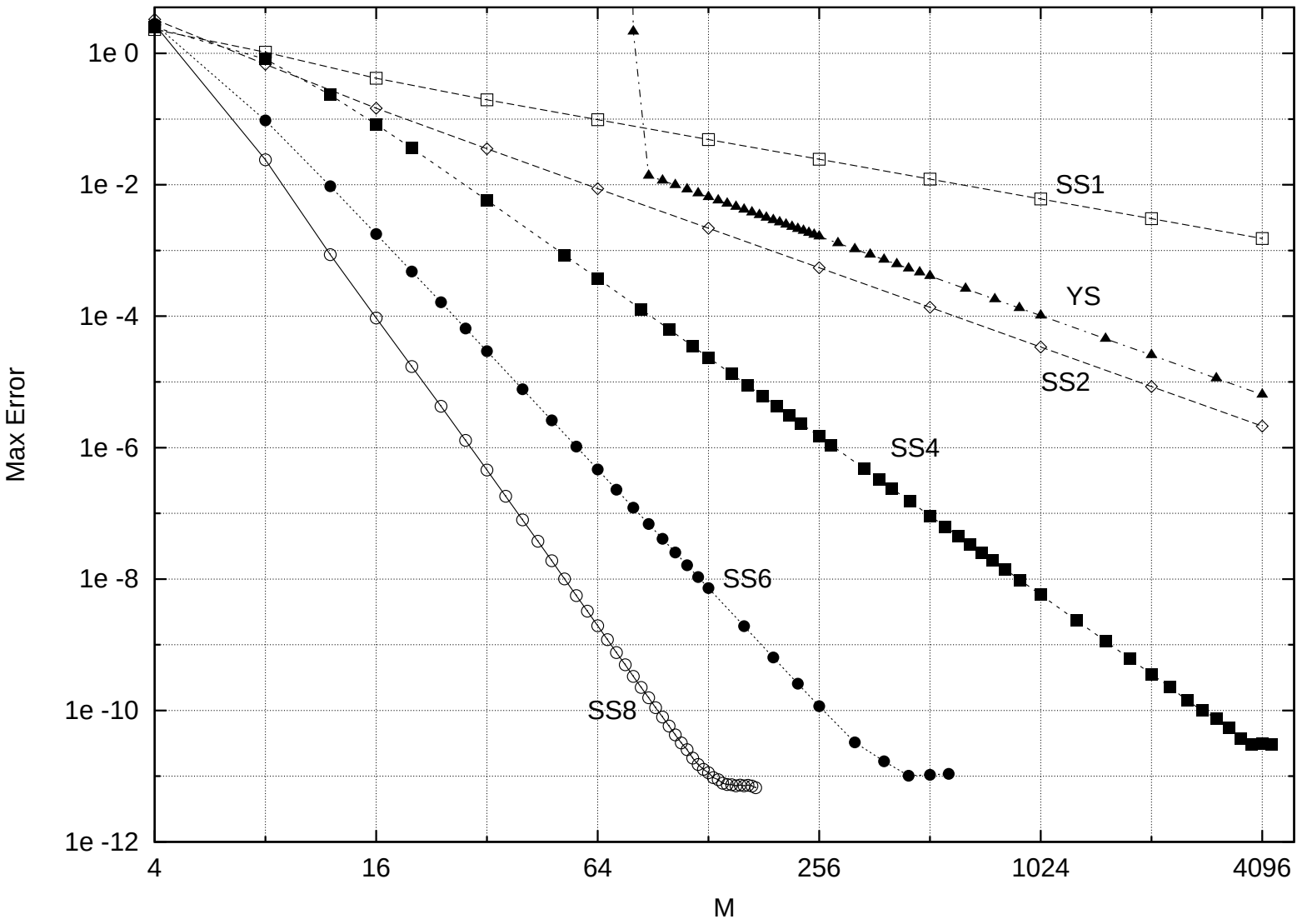

Figure 3.3: The maximum spatial errors of SS1, SS2, SS4, SS6, SS8, and YS with $N = 32$. Actual computed data are marked by various symbols. The vertical axis denotes the spatial accuracy measured in terms of maximum error (in $\log_{10}$ scale) at time $t = 1$. The horizontal axis denotes the number M of time steps (in $\log_2$ scale).

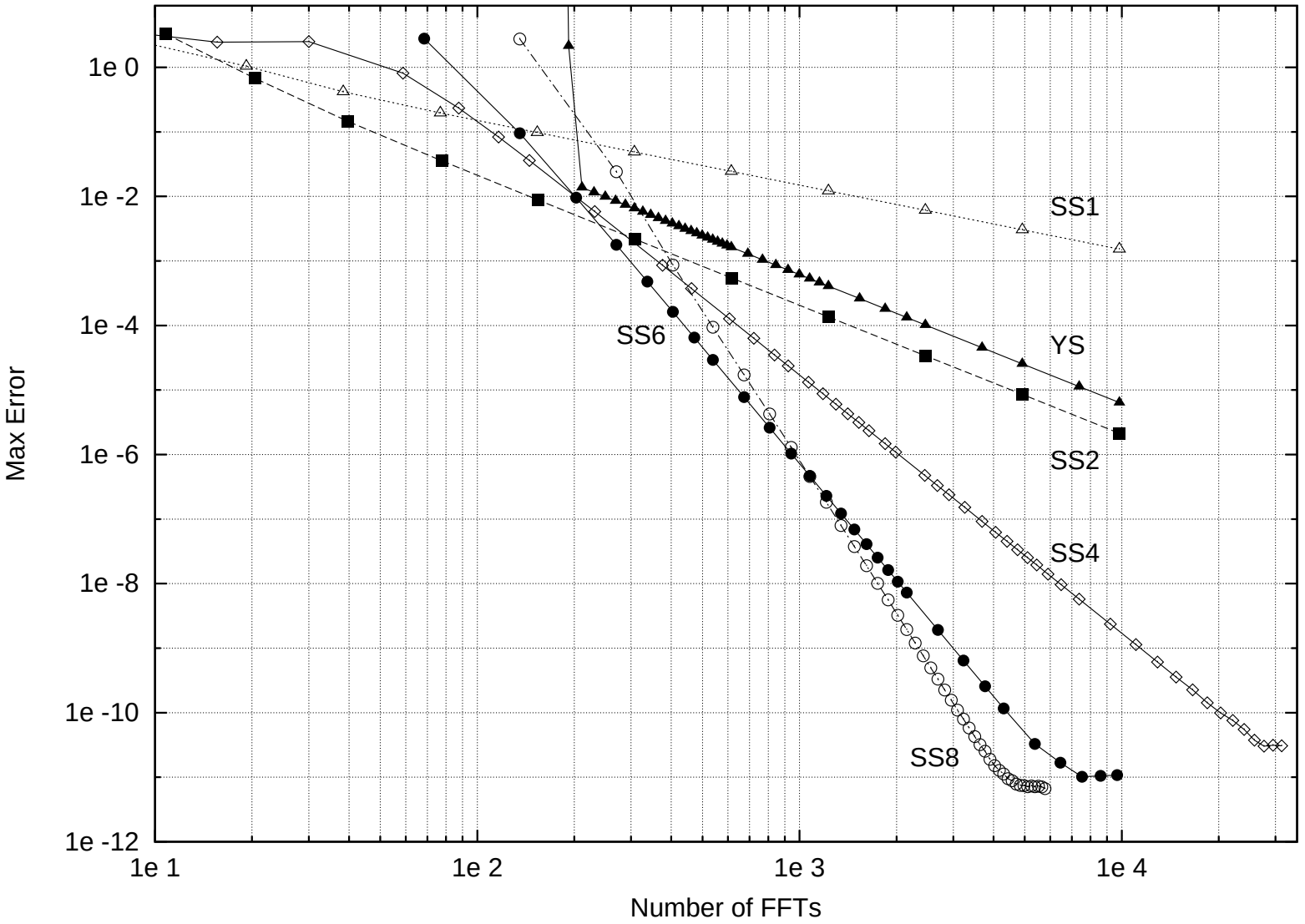

Figure 3.4: Computational costs of SS1, SS2, SS4, SS6,SS8, and YS for $N = 32$. Actual computed data are marked by various symbols. The vertical axis denotes spatial accuracy (in $\log_{10}$ scale) at the final time $t = 1$. The horizontal axis denotes numbers of FFTs (in $\log_{10}$ scale).