JOURNAL OF COMPUTATIONAL PHYSICS 47, 199–210 (1982)

An Explicit Finite-Difference Scheme with Exact Conservation Properties

J. M. SANZ–SERNA

Departamento de Matemáticas,
Facultad de Ciencias, Universidad del País Vasco, Lejona (Vizcaya), Spain

Received October 5, 1981

A finite difference scheme for the numerical study of the Korteweg–de Vries equation is constructed. It is explicit and yet conserves exactly the energy of the computed solutions. The underlying idea can also be applied to more general equations or systems. Numerical experiments are included.

1. INTRODUCTION

When a partial differential equation modelling wave-like phenomena is to be approximated numerically, it is highly desirable that the discrete scheme should conserve the discrete analogues of the quantities that are conserved by the equation. In particular, the conservation of a positive definite quadratic quantity in some cases rules out the occurrence of nonlinear instabilities [14]. It is therefore not surprising that many papers in the past have been devoted to the construction of schemes with *exact conservation properties*. Arakawa's work [3] is now classic; other papers on conservation properties are quoted by Morton [14] and Navon [15].

In most instances, the construction of schemes conserving quadratic quantities is confined to the semidiscrete case, i.e., only the space variables are discretized, while the time is kept continuous so as to approximate the original partial differential equation by a system of ordinary differential equations. In practice, however, the solution of this semidiscrete system must be obtained by means of a numerical method for ordinary differential equations, and as a result, the conservation properties of the semidiscrete approximation may be lost in the integration in time.

In fact, the use of an *explicit* method for the time integration will almost invariably result in failure of the attempt to conserve quadratic quantities, and then nonlinear instability can be a threat. (See [7, p. 480] and the discussion in Section 5.)

In this paper, we introduce a scheme for the integration in time of partial differential equations which is explicit and yet capable of conserving exactly the quadratic functionals conserved by the semidiscrete approximation.

0021-9991/82/080199–12$02.00/0

The new scheme, which is reminiscent of the usual leapfrog technique, will be presented first in the particular case of the Korteweg–de Vries (KdV) equation

$$u_t + uu_x + \varepsilon u_{xxx} = 0, \qquad \varepsilon > 0, \tag{1.1}$$

and extended later to more general situations.

The KdV equation provides a valuable test example. On the one hand, it possesses solutions in closed form which can be employed in the assessment of the accuracy of the numerical approximations. On the other hand, it describes phenomena requiring large-scale time calculations, and this is the sort of situation in which conservation properties are of paramount importance.

References to the numerical solution of (1.1) include Zabusky and Kruskal [22], Vliegenthart [20], Greig and Morris [9], Gazdag [8], Canosa and Gazdag [4], Abe and Inoue [1], Alexander and Morris [2], Whalbin [21], Sanz–Serna and Christie [17], Schoombie [18, 19], Kuo Pen–Yu and Sanz–Serna [10], and Christie *et al.* [5].

In Section 2, we review some of the properties of the Zabusky and Kruskal leapfrog scheme. Sections 3 and 4 describe the new method as applied to the KdV equation. In fact, two versions of the method are proposed: the fixed-step, conservative scheme (FSC) and the self-adaptive step, conservative scheme (SASC). Although we do not advocate the practical use of the FSC method, we have included it here because it is an intermediate step in the construction of the SASC scheme, and because it exhibits certain properties that may be of theoretical interest. In Section 5, the material of the two previous sections is generalized to cover more general equations.

2. The Zabusky–Kruskal Scheme

We are concerned with the initial-value problem given by Eq. (1.1), together with the initial condition

$$u(x, 0) = f(x), \qquad -\infty < x < \infty. \tag{2.1}$$

In 1965, Zabusky and Kruskal suggested the following scheme for the approximation of the solutions of (1.1):

$$\begin{aligned}(1/2k)(U_j^{n+1} - U_j^{n-1}) + (1/6h)(U_{j+1}^n + U_j^n + U_{j-1}^n)(U_{j+1}^n - U_{j-1}^n)\\ + (\varepsilon/2h^3)(U_{j+2}^n - 2U_{j+1}^n + 2U_{j-1}^n - U_{j-2}^n) = 0.\end{aligned} \tag{2.2}$$

Here k, h denote the mesh sizes in the x and t variables, respectively, and U_j^n is an approximation to $u(jh, nk)$. We assume that we are interested in solutions of (1.1) that, for the range of time under consideration, are negligible outside an interval $0 \leqslant x \leqslant L$. (Note that the KdV equation has solutions $u(x, t)$ that decrease exponentially as $|x| \to \infty$.) Then

$$U_0^n = U_1^n = U_{J-1}^n = U_J^n = 0, \qquad n = 1, 2, 3,..., \tag{2.3}$$

where $J = L/h$ are suitable boundary conditions for (2.1) [9, 17–19]. The situation is not greatly altered if periodic boundary conditions rather than (2.3) are considered. The initial condition

$$U_j^0 = f(jh), \qquad j = 0, 1, \ldots, J, \tag{2.4}$$

must be supplemented with a starting procedure to compute U_j^1, and an obvious choice is [20]:

$$\begin{aligned}(1/k)(U_j^1 - U_j^0) + (1/6h)(U_{j+1}^0 + U_j^0 + U_{j-1}^0)(U_{j+1}^0 - U_{j-1}^0)\\ + (\varepsilon/2h^3)(U_{j+2}^0 - 2U_{j+1}^0 + 2U_{j-1}^0 - U_{j-2}^0) = 0.\end{aligned} \tag{2.5}$$

The analysis of the scheme (2.2)–(2.5) is best performed if we introduce the auxiliary system of ordinary differential equations

$$\frac{d}{dt}\mathbf{U}(t) = \mathbf{F}(\mathbf{U}(t)), \tag{2.6}$$

where $\mathbf{U}(t)$ is the vector $[U_2(t), U_3(t), \ldots, U_{J-2}(t)]^{\mathrm{T}}$, and $\mathbf{F}(\mathbf{U})$ is a nonlinear vector function with components

$$\begin{aligned}F_j(\mathbf{U}) = &-(1/6h)(U_{j+1}(t) + U_j(t) + U_{j-1}(t))(U_{j+1}(t) - U_{j-1}(t))\\ &- (\varepsilon/2h^3)(U_{j+2}(t) - 2U_{j+1}(t) + 2U_{j-1}(t) - U_{j-2}(t)),\\ &\qquad j = 2, \ldots, J-2.\end{aligned} \tag{2.7}$$

(We set $U_0(t) \equiv U_1(t) \equiv U_{J-1}(t) \equiv U_J(t) \equiv 0$.)

Clearly, (2.2)–(2.5) can be viewed as the result of the discretization of (2.6) by the midpoint (leapfrog) rule

$$(1/2k)(\mathbf{U}^{n+1} - \mathbf{U}^{n-1}) = \mathbf{F}(\mathbf{U}^n), \qquad n = 1, 2, 3, \ldots \tag{2.8}$$

Euler's method

$$(1/k)(\mathbf{U}^1 - \mathbf{U}^0) = \mathbf{F}(\mathbf{U}^0), \tag{2.9}$$

provides the missing starting value. In formulae (2.8), (2.9), $\mathbf{U}^n$ denotes the vector $[U_2^n, U_3^n, \ldots, U_{J-2}^n]^{\mathrm{T}}$.

We now study the properties of the semidiscrete system (2.6). It is easily found that its order of local accuracy is $O(h^2)$. A bound for the global error is derived in [10]. If we denote by $\mathbf{e}$ the $(J-3)$-dimensional vector that has all its components equal to unity, the following identities hold for any vector $\mathbf{V}$:

$$\mathbf{e}^{\mathrm{T}}\mathbf{F}(\mathbf{V}) = 0, \tag{2.10}$$

$$\mathbf{V}^{\mathrm{T}}\mathbf{F}(\mathbf{V}) = 0. \tag{2.11}$$

It follows that

$$\frac{d}{dt}\mathbf{e}^{\mathrm{T}}\mathbf{U}(t) = \mathbf{e}^{\mathrm{T}}\frac{d}{dt}\mathbf{U}(t) = \mathbf{e}^{\mathrm{T}}\mathbf{F}(\mathbf{U}(t)) = 0, \tag{2.12}$$

and analogously,

$$\frac{d}{dt}(\mathbf{U}^{\mathrm{T}}(t)\,\mathbf{U}(t)) = 0. \tag{2.13}$$

In other words, the semidiscrete scheme (2.6) has the momentum

$$\mathbf{e}^{\mathrm{T}}\mathbf{U}(t) = \sum U_j(t) \tag{2.14}$$

and the energy

$$\mathbf{U}^{\mathrm{T}}(t)\,\mathbf{U}(t) = \sum\, [U_j(t)]^2 \tag{2.15}$$

as conserved quantities, thus reproducing a property of the KdV equation. We emphasize that the conservation of energy is achieved by means of the *Galerkin-like* treatment of the nonlinear term uu_x [14].

Several advantages follow from the existence of conserved quantities. In particular, we note that (2.13) implies boundedness of the solutions of (2.6), and therefore obviates the occurrence of blowup phenomena.

Unfortunately, the discretization in time that must be performed to transform (2.6) into the Zabusky–Kruskal scheme (2.2) destroys the energy-conserving character of the semidiscrete scheme. In more precise terms, one has

$$\sum_j (U_j^{n+1})^2 - \sum_j (U_j^{n-1})^2 = O(k^3), \tag{2.16}$$

and of course, (2.16) is not sufficient to guarantee the boundedness of the solutions as n increases.

In fact, the fully discrete scheme (2.2)–(2.5) can suffer from nonlinear instability, as can be seen in an example. The Zabusky–Kruskal method (2.2)–(2.5) was applied with $h = 0.02$, $k = 0.005$, and $L = 2$ to the smooth initial condition

$$f(x) = 3c\,\mathrm{sech}^2(bx + d), \tag{2.17}$$

where $c = 0.3$, $\varepsilon = 0.000484$, $b = (c/4\varepsilon)^{1/2}$, and $d = -b$. The integration proceeded in a stable way for more than 3000 time steps and then suddenly exploded. There is no doubt that noisy or rough initial data would have led to an earlier blowup.

It is easily seen that the Zabusky–Kruskal scheme conserves the momentum, and the same will be true for the scheme presented in Sections 3 and 4.

We conclude this section with a study of the *linearized* stability condition for the Zabusky–Kruskal scheme, as this will play an important role in subsequent discussions. We consider the linearized equation

$$u_t + \eta u_x + \varepsilon u_{xxx} = 0, \tag{2.18}$$

where n is a real constant. Vliegenthart [20] showed that when the linearized version of (2.2) is employed to discretize (2.18), the von Neumann stability condition (for the periodic problem) takes the form

$$\max_{\xi} |(k/h)\sin\xi(\eta - 2(\varepsilon/h^2)(1-\cos\xi))| \leqslant 1. \tag{2.19}$$

When the scheme is applied to the (nonlinear) KdV equation, we require (2.19) to hold for any η such that $u_{\text{min}} \leqslant \eta \leqslant u_{\text{max}}$, where u_{min} and u_{max} are, respectively, the smallest and largest values of $u(x, t)$. We assume that $u_{\text{min}} = 0$, as this is the case in the examples considered in the next two sections. Then Vliegenthart suggests the stability condition

$$(k/h)(u_{\text{max}} + 4(\varepsilon/h^2)) \leqslant 1, \tag{2.20}$$

which follows from (2.19) and the bounds $|\sin\xi| \leqslant 1$, $|1-\cos\xi| \leqslant 2$. Clearly, the functions $|\sin\xi|$, $|1-\cos\xi|$ do not attain their maxima for the same value of ξ, and therefore the bound (2.20) is not necessary in order to satisfy the von Neumann condition.

In practice, we have found the requirement (2.20) rather pessimistic (and note that linearized stability conditions are usually optimistic). We now analyze (2.19) in detail, so as to derive a more realistic stability condition.

From considerations of periodicity and symmetry, we conclude that it suffices to look at values of ξ between 0 and π. Then $(k/h)\sin\xi\eta$ is *positive* and $-2(k\varepsilon/h^3)\sin\xi(1-\cos\xi)$ is *negative*. Recalling that η ranges between 0 and u_{max}, we see that the condition

$$\max_{0\leqslant\xi\leqslant\pi} 2(k\varepsilon/h^3)\sin\xi(1-\cos\xi) \leqslant 1 \tag{2.21}$$

is *necessary* if (2.19) is to hold for all relevant values of η. Since the maximum of $\sin\xi(1-\cos\xi)$ is $\frac{3}{4}\sqrt{3}$, (2.21) can be rewritten as

$$3\sqrt{3}k\varepsilon \leqslant 2h^3. \tag{2.22}$$

Furthermore, (2.22) is also *sufficient* to guarantee that (2.19) holds (with $0 \leqslant \eta \leqslant u_{\text{max}}$), provided that $(ku_{\text{max}})/h \leqslant 1$. In Table I, we have displayed the maximum time step k allowed by formulae (2.20), (2.22), when $u_{\text{max}} = 0.9$, $\varepsilon = 0.000484$, and $h = 0.02$ or $h = 0.01$. (These are the values used in subsequent experiments.)

TABLE I

Maximum Time Step

	(2.20)	(2.22)
$h = 0.02$	3.5×10^{-3}	6.4×10^{-3}
$h = 0.01$	4.9×10^{-4}	8.0×10^{-4}

It was found that (2.22) provided a better estimate of the allowable time step. For example, runs with $h=0.01$, $k=0.0005$ and $h=0.02$, $k=0.005$ proceeded in a stable manner for very long intervals of time, thus indicating that any potential instabilities should be attributed to nonlinear effects. (Note that linear instabilities would have been apparent from the early stages of the computation.)

3. The Fixed-Step Conservative Scheme

For the time integration of (2.6), we present in this section an explicit method that retains the energy-conserving character of the semidiscrete scheme. The new method is obtained when formula (2.8) is replaced by

$$\mathbf{U}^{n+1}-\mathbf{U}^{n-1}=2\tau_n\mathbf{F}(\mathbf{U}^n), \qquad n=1,2,3,..., \tag{3.1}$$

where

$$\begin{aligned}\tau_n &= \mathbf{F}^{\mathrm{T}}(\mathbf{U}^n)(\mathbf{U}^n-\mathbf{U}^{n-1})/\mathbf{F}^{\mathrm{T}}(\mathbf{U}^n)\,\mathbf{F}(\mathbf{U}^n), && \text{if} \quad \mathbf{F}(\mathbf{U}^n)\neq\mathbf{0},\\ &=0, && \text{otherwise.}\end{aligned} \tag{3.2}$$

The vector $\mathbf{U}^n$ is meant to approximate $\mathbf{U}(nk)$, and for this rason, it is reasonable to use the term *fixed-step* (cf. Section 4). Note that (2.9) still applies. A Taylor expansion reveals that (3.1) is first-order accurate, whilst (2.8) is second-order.

From (3.1), we see that

$$(\mathbf{U}^{n+1})^{\mathrm{T}}\,\mathbf{U}^{n+1}=(\mathbf{U}^{n-1}+2\tau_n\mathbf{F}(\mathbf{U}^n))^{\mathrm{T}}\,(\mathbf{U}^{n-1}+2\tau_n\mathbf{F}(\mathbf{U}^n)), \tag{3.3}$$

and use of (3.2) and (2.11) yields

$$(\mathbf{U}^{n+1})^{\mathrm{T}}\,\mathbf{U}^{n+1}=(\mathbf{U}^{n-1})^{\mathrm{T}}\,\mathbf{U}^{n-1}; \tag{3.4}$$

i.e., the new method conserves the energy.

We note that even if $\mathbf{F}$ had been a linear function, the scheme (3.1) would have led to a nonlinear relation between the vectors $\mathbf{U}^{n+1}$, $\mathbf{U}^n$, and $\mathbf{U}^{n-1}$. This consideration precludes the application of the usual linearized analysis.

In order to compare the performance of the FSC scheme with that of the Zabusky–Kruskal method, both were applied to the initial condition (2.17), with $L=2$, $c=0.3$, $\varepsilon=0.000484$, $b=(c/4\varepsilon)^{1/2}$, and $d=-b$. This problem was used in [2, 9, 17], and has the theoretical solution

$$u(x,t)=3c\,\mathrm{sech}^2(bx-bct+d), \tag{3.5}$$

representing a single soliton with amplitude 0.9 and speec 0.3. The accuracy, efficiency, and stability of the methods will be assessed separately.

(i) *Accuracy.* From the results displayed in Table II, we conclude that the performances of the methods were comparable for $0\leqslant t\leqslant 1$. (See, however, (iv) below.)

TABLE II

L^{∞} Error $\times 10^3$

		ZK	FSC
$h=0.02,\ k=0.005$	$t=0.25$	12.5	12.3
	$t=0.50$	21.5	21.2
	$t=0.75$	29.8	28.7
	$t=1.00$	36.6	34.3
$h=0.01,\ k=0.0005$	$t=0.25$	3.24	2.86
	$t=0.50$	5.45	3.83
	$t=0.75$	7.40	3.25
	$t=1.00$	9.75	2.13

(ii) *Efficiency.* The CPU times (in an IBM 370/148) corresponding to the runs in Table II were 7 and 101 seconds for the FSC scheme, and 5 and 92 seconds for the Zabusky–Kruskal method. In the implementation used, the FSC code stored the vector $\mathbf{F}(\mathbf{U}^n)$, and was therefore more demanding in storage requirements.

(iii) *Stability.* We have already observed that the Zabusky–Kruskal scheme can lead to blowup phenomena. The new method does not suffer from this shortcoming, as it conserves exactly a positive definite quadratic functional. Furthermore, this conservative behaviour is independent of the time step k. In particular, the FSC scheme *generates bounded sequences* $\mathbf{U}^n$, $n = 1, 2, 3,...,$ *for any value of* k.

In the study of wave phenomena, however, the time step must be reduced not only on stability grounds, but also with a view to accuracy, and therefore, this property is not of much practical value. In fact, it was observed that when k was chosen so large as to violate condition (2.22), the FSC scheme produced a bounded but inaccurate solution, whereas the Zabusky–Kruskal method gave rise to machine overflows. A closer examination of the results produced by the FSC scheme showed that for k larger than the maximum step allowed by (2.22), the computed solution described a travelling wave with the correct profile (2.17), moving at a speed smaller than the theoretical. Table III shows the observed speed of the solitary wave in the interval

TABLE III

Computed Speed of the Soliton

Theoretical	0.300
$k=0.0005$	0.300
$k=0.001$	0.240
$k=0.002$	0.125
$k=0.005$	0.055

$0 \leqslant t \leqslant 1$, when h was taken as 0.01. We conclude that the unconditional boundedness of the FSC scheme is achived at the price of retarding the speed of the waves. (Note that any Courant–Friedrichs–Lewy condition of the form $k\lambda|h \leqslant 1$ can be interpreted as imposing an upper limit either on the step k, for given λ and h, or on the speed λ, for given k and h.)

(iv) *Long time intervals.* Experiments involving large ranges of time t revealed that the FSC method was likely to produce larger phase errors than those associated with the Zabusky–Kruskal scheme. This might have been guessed from the presence of the stabilizing mechanism discussed above. Such phase errors can be regarded as a serious disadvantage of the FSC scheme, and can be suppressed by the device described in the next section.

Before we end the study of the FSC method, we wish to point out that, in this method, each value $U_{j_0}^{n+1}$ depends on *all* the values U_j^n and $U_j^{n-1}, j = 2, 3,..., J-2$, at the previous two time levels. Thus, the domain of dependence of the numerical solution is the whole interval $0 \leqslant x \leqslant L$. In this respect, the behaviour of the new method resembles that of the usual implicit schemes.

4. The Self-Adaptive Conservative Scheme

It has been pointed out that the FSC algorithm may suffer from phase errors. The origin of these errors can be traced back to formulae (3.1), (3.2). We see that the left-hand side of (3.1) corresponds to a time increment of $2k$, while on the right-hand side, $2\tau_n = 2k + O(k^2)$. The SASC scheme is given by formulae (3.1), (3.2), (2.9), (2.7), but now $\mathbf{U}^n$ is regarded as an approximation to $\mathbf{U}(t_n)$, where

$$t_0 = 0, \qquad t_1 = k, \qquad t_{n+1} = 2\tau_n + t_{n-1}, \qquad n = 1, 2, 3,.... \tag{4.1}$$

Clearly, the conservation of energy still holds, as the vectors $\mathbf{U}^n$ produced by the SASC method are exactly those originated by the FSC scheme, with only the correspondence between the computed vectors and values of t being different.

The SASC scheme can be regarded as the result of using the midpoint rule (2.8) in variable-step implementation, the steps τ_n being automatically selected so as to guarantee conservation of energy.

Table IV displays some numerical results corresponding to the SASC scheme (L, ε, c, etc. are as before). We observe that for those runs for which the stability condition (2.22) is satisfied (namely, $h = 0.02$, $k = 0.005$ and $h = 0.01$, $k = 0.0005$), the performance of the SASC method is very similar to that of the Zabusky–Kruskal scheme. When k is increased, however, an interesting aspect of the behaviour is the following: the increment $\Delta t_n = t_{n+1} - t_n$, which is initially (i.e., when $n = 0$) equal to k, is steadily reduced as n increases, so as to yield an average time step t_n/n for which (2.22) is satisfied.

TABLE IV

SASC Errors × 10^3

	n	t	L^∞ Error × 10^3	t/n
$h = 0.02$, $k = 0.005$	50	0.2500	12.1	0.00500
	100	0.5000	21.2	0.00500
	150	0.7503	29.5	0.00500
	200	1.007	36.1	0.00500
$h = 0.02$, $k = 0.01$	25	0.1889	15.9	0.00756
	50	0.3479	15.2	0.00696
	75	0.5079	30.2	0.00677
	100	0.6674	26.1	0.00667
$h = 0.01$, $k = 0.0005$	500	0.2501	3.22	0.000500
	1000	0.5006	5.41	0.000500
	1500	0.7516	7.34	0.000500
	2000	1.0031	9.76	0.000500
$h = 0.01$, $k = 0.001$	250	0.2029	2.83	0.000812
	500	0.4020	4.84	0.000804
	750	0.6001	6.74	0.000800
	1000	0.7968	7.84	0.000800

Attempts were also made to use larger values of k than those displayed in Table IV. The stability was preserved, of course, but the accuracy suffered to some extent. We conclude that the best policy is to use the SASC scheme with a value of k equal to or slightly larger than the maximum allowed by (2.22). When used in this manner, the SASC scheme exhibits the following advantages:

(i) It does not suffer from nonlinear instability, as opposed to the Zabusky–Kruskal scheme; and

(ii) the average tipe step t_n/n would be close to the maximum value $(2h^3)/(3\sqrt{3}\varepsilon)$ given by (2.22), whilst for the Zabusky–Kruskal scheme, the time step must be chosen to be a fraction of that maximum value. (For instance, the last run in Table IV has an average step of 0.0008 and, for this time step, the Zabusky–Kruskal scheme causes an overflow at the early stages of the computation.) Thus, the new method requires fewer steps to span the same time interval. (But it should be recalled that Zabusky–Kruskal steps are marginally cheaper.)

Note from Tables II and IV, and from the experiments quoted below, that the previously mentioned advantages are not gained at the expense of a loss of accuracy.

The SASC method was found to perform satisfactorily in tests involving long time intervals. One of these experiments is reported in Table V. The initial condition was once more (2.17), with the values of L, c, ε, b used previously. The parameter d, which governs the initial phase of the soliton, was set equal to $-0.55b$ in order to

TABLE V

Error × 10^3

	ZK		SASC	
	n	Error	n	Error
$t = 1.5$	3000	13.8	2006	13.2
$t = 2$	4000	17.9	2736	16.9
$t = 2.5$	5000	21.8	3495	20.1
$t = 3$	6000	26.4	4264	24.2

have a solution which is negligible at the boundaries for the range of time $0 \leqslant t \leqslant 3$ under consideration. The mesh size h was taken to be 0.01, while k was 0.0008. For comparison, we have included the results corresponding to the Zabusky–Kruskal scheme with $k = 0.0005$ and $h = 0.01$.

5. More General Equations

We assume that a time-dependent partial differential equations has been discretized in space in order to approximate it by the system of ordinary differential equations

$$\frac{d}{dt}\mathbf{U}(t) = \mathbf{F}(\mathbf{U}(t)), \tag{5.1}$$

where $\mathbf{U}(t)$ is the vector of nodal values. It is not assumed that the original partial differential equation is one dimensional in space. We make the assumption that the identity

$$\mathbf{V}^{\mathrm{T}}\mathbf{F}(\mathbf{V}) = 0 \tag{5.2}$$

holds for all vectors $\mathbf{V}$. From (5.2), we conclude, as in Section 2, that the energy $\mathbf{U}^{\mathrm{T}}\mathbf{U}$ is a constant of motion for the solutions of (5.1). One would like to discretize (5.1) in time in such a way that, regardless of the step size k, the fully discrete scheme also conserves the energy. It is possible to prove [16] that such a requirement rules out the use of *explicit* linear-multistep or Runge–Kutta methods, and imposes an upper bound on the order (in time) of the scheme. On the other hand, if (5.1) is discretized by means of the *implicit* method

$$\mathbf{U}^{n+1} - \mathbf{U}^{n} = k\mathbf{F}(\tfrac{1}{2}(\mathbf{U}^{n+1} + \mathbf{U})) \tag{5.3}$$

(which is called the *one-leg trapezoidal rule* in ODEs jargon), one then has

$$(\mathbf{U}^{n+1})^{\mathrm{T}}\,\mathbf{U}^{n+1} = (\mathbf{U}^{n})^{\mathrm{T}}\,\mathbf{U}^{n}. \tag{5.4}$$

Unfortunately, (5.4) does not hold if the nonlinear equations in (5.3) are not solved exactly [6].

The method (3.1), (3.2) for the solution of (5.1) has the conservation property

$$(\mathbf{U}^{n+2})^{\mathrm{T}}\,\mathbf{U}^{n+2} = (\mathbf{U}^{n})^{\mathrm{T}}\,\mathbf{U}^{n}, \tag{5.5}$$

and is at the same time explicit. It is therefore very easy to program and economic to use (in terms of cost per step). The method can be used in either FS or SAS fashion.

Formulae (3.1), (3.2) were first suggested, in a different context, by Lambert and McLeod [11] and Laurie [12]. (See also [13, 16].) The method can be altered [16] in order to accommodate constants of motion other than $\mathbf{U}^{\mathrm{T}}\mathbf{U}$.

Finally, we should like to point out the possibility of enforcing conservation properties in an *a posteriori* manner. The interested reader is referred to Navon [15] and the literature cited by him. (See in particular the work of Isaacson, Isaacson, and Turkel and Sasaki).

Acknowledgment

This work was carried out at the National Research Institute for Mathematical Sciences, Council for Scientific and Industrial Research, Pretoria (South Africa), and the author wishes to express his gratitude for the facilities provided. He also wishes to thank Dr. A. F. Meiring for her assistance with the numerical computations.

Note added in proof. It is possible to show that the FSC scheme is not convergent, while the global errors produced by the SASC scheme behave like $O(k^2 + h^2)$. The interested reader is referred to a forthcoming paper by the present author.

References

1. K. Abe and O. Inoue, *J. Comput. Phys.* **34** (1980), 202.
2. M. E. Alexander and J. Ll. Morris, *J. Comput. Phys.* **30** (1979), 428.
3. A. Arakawa, *J. Comput. Phys.* **1** (1966), 119.
4. J. Canosa and J. Gazdag, *J. Comput. Phys.* **23** (1977), 393.
5. I. Christie, D. F. Griffiths, A. R. Mitchell, and J. M. Sanz-Serna, *IMA J. Numer. Anal.* **1** (1981), 253.
6. K. A. Cliffe, *Int. J. Numer. Meth. Fluids* **1** (1981), 17.
7. J. Gary, *in* "Numerical Methods Used in Atmospheric Models," Global Atmospheric Research Programme (GARP), GARP Publication Series No. 17, Vol. II, 1979.
8. J. Gazdag, *J. Comput. Phys.* **13** (1973), 100.
9. I. S. Greig and J. Ll. Morris, *J. Comput. Phys.* **20** (1976), 64.
10. Kuo Pen-Yu and J. M. Sanz-Serna, *IMA J. Numer. Anal.* **1** (1981), 215.
11. J. D. Lambert and R. J. Y. McLeod, *in* "Numerical Analysis Proceedings, Dundee 1979" (G. A. Watson, Ed.), Springer-Verlag, Berlin, 1980.
12. D. P. Laurie, *in* "Proceedings of the Sixth South African Symposium in Numerical Mathematics, Durban 1980," Computer Science Department, Univ. of Natal, Durban, 1980.

13. R. J. Y. McLeod and J. M. Sanz–Serna, "Geometrically Derived Difference Formulae for the Numerical Integration of Trajectory Problems," Report TWISK 231, Pretoria, September 1981.
14. K. W. Morton, *in* "The State of the Art in Numerical Analysis" (D. A. H. Jacobs, Ed.), p. 699, Academic Press, New York/London, 1977.
15. I. M. Navon, *Mon. Weather Rev.*, **109**(1981), 946.
16. J. M. Sanz–Serna, "An Explicit Ordinary Differential Equation Method with a Useful Stability Property," Report TWISK 232, Pretoria, September 1981.
17. J. M. Sanz–Serna and I. Christie, *J. Comput. Phys.* **39** (1981), 94.
18. S. W. Schoombie, "Finite Element Methods for the Korteweg–de Vries Equation. I. Galerkin Method with Hermite Cubics," Univ. of Dundee Numerical Analysis Report NA/43, 1980. Also to appear in *SIAM J. Numer. Anal.*
19. S. W. Schoombie, "Finite Element Methods for the KdV Equation. II. Petrov–Galerkin Methods with Splines," Univ. of Dundee Numerical Analysis Report NA/44, 1980. Also to appear in *IMA J. Numer. Anal.*
20. A. C. Vliegenthart, *J. Engrg. Math.* **5** (1971), 137.
21. L. Wahlbin, *in* "Mathematical Aspects of Finite Elements" (C. de Boor, Ed.), Academic Press, New York, 1974.
22. N. J. Zabusky and M. D. Kruskal, *Phys. Rev. Lett.* **15** (1965), 240.