M3 to M4: from Huygens' to conservation variables

Francis Muir

ABSTRACT

A linear wave propagation scheme, M3, derived from the lattice gas may be re-formulated in terms of conservation variables. This leads to storage savings without increase in computation time and guarantees a perfect shuffle between advection steps.

INTRODUCTION

In a previous paper (Muir, 1987) I have discussed the evolution of a chain of models from the lattice gas (Wolfram, 1986a & 1986b). These models differ in their data types, which range from Boolean through probabilistic to real, but share, amongst other qualities, a common data structure, a Huygens or discrete velocity form.

M4 continues the retreat from the lattice gas by replacing the Huygens' data structure of M3, the linear model system, by conventional conservation field variables.

M4

Motivation

Recall that in M3 data are represented at each point by an n-vector of real numbers, where n is the number of active neighbours to each point— six in the case of the type 2-D hexagonal system.

Recognizing that the data vectors are uniquely determined by the mass and momentum constraints, M4 replaces the neighbour-directed data elements by the

conserved quantities (mass and momentum) themselves, with possibilities of substantial savings in storage requirements. For example, in the case of the 2-D hexagonal system the 6-vector of M3 is replaced by the 3-vector (mass and the momentum 2-vector) of M4. More substantial savings are offered in four dimensions, where a 24-element next-nearest neighbour M3 scheme on a rectangular lattice that grows out of the HLF lattice gas model (d'Humieres et al. 1986) is replaced by a 5-element (mass, momentum 4-vector) M4 scheme, for a storage savings of 79%. This 4-D scheme can then be projected on to a 3-space to give an 18-element (6 nearest and 12 next-nearest neighbours) M3 scheme which can be replaced by a 4-element (mass, momentum 3-vector) M4 scheme, for a storage savings of 78%.

Efficiency

At first glance it might appear that this data compaction (and uncompaction) would be bought at the expense of two additional matrix transformations on the data. A move from the simple M3

Advect Collide

to a more complex M4

Uncompact Advect Collide Compact

but fortunately this turns out to be not so.

Data transforms

Remembering that the data elements in an M3 scheme are masses with velocity labels, the compaction transformation to conservation variables is a particularly simple one. The mass element for M4 is the absolute sum of the M3 masses, and the momentum elements are the several orthogonal vector sums of the mass-velocity products. In the case of our type 2-D hexagonal scheme, the transformation matrix, $\mathbf{T}$, is:

$$\begin{pmatrix} 1.000 & 1.000 & 1.000 & 1.000 & 1.000 & 1.000 \\ 1.000 & 0.500 & -0.500 & -1.000 & -0.500 & 0.500 \\ 0.000 & 0.866 & 0.866 & 0.000 & -0.866 & -0.866 \end{pmatrix}$$

Since the M3 data representation is uniquely determined from the constraints (mass, momentum) and a minimum $\mathbf{L_2}$ principle, and having just described an M3 to M4 mapping operation, the appropriate transformation matrix for returning from M4 to M3 is $\mathbf{T}^\dagger$, the Moore-Penrose inverse of $\mathbf{T}$, which is defined by the four properties:

$$\mathbf{T}\mathbf{T}^\dagger\mathbf{T} = \mathbf{T}$$

$$\mathbf{T}^\dagger\mathbf{T}\mathbf{T}^\dagger = \mathbf{T}^\dagger$$
$$(\mathbf{T}\mathbf{T}^\dagger)^\mathbf{T} = \mathbf{T}\mathbf{T}^\dagger$$
$$(\mathbf{T}^\dagger\mathbf{T})^\mathbf{T} = \mathbf{T}^\dagger\mathbf{T}$$

In the case of our type system $\mathbf{T}^\dagger$, the Uncompact operator, is rather simple, since the three vectors that make up $\mathbf{T}$ are orthogonal—they just need transposing and rescaling:

$$\begin{pmatrix} 0.167 & 0.333 & 0.000 \\ 0.167 & 0.167 & 0.289 \\ 0.167 & -0.167 & 0.289 \\ 0.167 & -0.333 & 0.000 \\ 0.167 & -0.167 & -0.289 \\ 0.167 & 0.167 & -0.289 \end{pmatrix}$$

An M4 timestep now looks like

$$\mathbf{T}^\dagger \; \textit{Advect Collide} \; \mathbf{T}$$

But the *Collide* operation is no more than

$$\mathbf{T}\mathbf{T}^\dagger$$

and the M4 timestep is

$$\mathbf{T}^\dagger \; \textit{Advect} \; \mathbf{T}\mathbf{T}^\dagger\mathbf{T}$$

which, by our (Moore-Penrose) rules of construction is

$$\mathbf{T}^\dagger \; \textit{Advect} \; \mathbf{T}$$

that is, the *Collide* does nothing (idempotent), and can be omitted. Now compare this with the M3 timestep

$$\textit{Advect Collide}$$

which can be written

$$\textit{Advect} \; \mathbf{T}\mathbf{T}^\dagger$$

and it is clear that the only differences between an M3 and an M4 computational chain is in the places where the chain is broken for storage. As previously noted under M3, the *Collide* matrix for our type (FHP) scheme is:

$$\begin{pmatrix} 0.500 & 0.333 & 0.000 & -0.167 & 0.000 & 0.333 \\ 0.333 & 0.500 & 0.333 & 0.000 & -0.167 & 0.000 \\ 0.000 & 0.333 & 0.500 & 0.333 & 0.000 & -0.167 \\ -0.167 & 0.000 & 0.333 & 0.500 & 0.333 & 0.000 \\ 0.000 & -0.167 & 0.000 & 0.333 & 0.500 & 0.333 \\ 0.333 & 0.000 & -0.167 & 0.000 & 0.333 & 0.500 \end{pmatrix}$$

and this checks that *Collide* is $\mathbf{T}\mathbf{T}^{\dagger}$

Extension

This discussion has concerned only the linear M3 scheme. It would apply equally well to the (Navier-Stokes) M2 and M1 schemes (Muir, 1987) with equally substantial storage savings, and the further advantage that memory of the past that can be retained in redundant Huygens schemes is erased—in other words, conservation variables guarantee a perfect shuffle.

REFERENCES

d'Humieres, D., Lallemand, P. and Frisch, U., 1986, Lattice gas models for 3D hydrodynamics: Europhys. Lett. 2 (4) 291-297

Muir, F., 1987, Three experimental modeling systems: SEP–51 pp 119-128

Wolfram, S., 1986a, Theory and Applications of cellular automata: World Scientific

Wolfram, S., 1986b, Cellular automaton fluids 1: basic theory: J. Stat. Phys. 45 3/4 471ff