Wave Inversion Technology, Report No. 3, pages 189-196

A Chebyshev method on 2D generalized coordinates

M. Pohl, F. Wenzel and M. Karrenbach[1]

keywords: *pseudospectral, transfinite interpolation, generalized coordinates*

ABSTRACT

We present a method which allows to generate 2D numerical grids in generalized coordinates from any number of predefined grid lines and solve the wave equation in these coordinates by a Chebyshev pseudospectral method. The grid generation algorithm is based on a transfinite interpolation and defines the mapping function by means of spline interpolations. The stability of the method depends only on the minimum grid spacing.

INTRODUCTION

The numerical modelling of curved interfaces on rectangular grids imposes conditions for the grid spacing to suppress undesired diffraction effects. Due to stability reasons the time stepping directly relates to the grid spacing. A grid refinement thus leads to both a larger data volume and a larger number of time steps. Especially for pseudospectral methods, which require a lower number of grid points per wave length than standard FD methods a correct representation of curved boundaries is of great value.

A major problem in using generalized coordinates is to initialise the grid. Our intention is to introduce arbitrarily shaped layer boundaries in two dimensions without changing the grid boundaries. Grid size and number of grid points remain. In order to define the mapping functions the generalized coordinates must be known. The grid generation requires two steps: (1) computation of the generalized coordinates considering the specified layer boundaries and (2) definition of the mapping functions. The first can be done by means of transfinite interpolations (Hoschek and Lasser, 1992). The layer boundaries divide the model space into several patches. Transfinite interpolations interpolate the boundary curves of each patch using appropriate blending functions but leave the boundary curves unchanged. In the second step we calculate the node dependent shift values and define the mapping functions using spline functions.

[1]**email:** Melanie.Pohl@gpi.uni-karlsruhe.de

CHEBYSHEV METHOD

We use a Chebyshev collocation method to solve the wave equation on the generalized grid. The solution of the wave equation is expanded in terms of Chebyshev polynomials (Gottlieb et al., 1984). In the computational domain the grid is defined by the Gauss–Lobatto collocation points (Carcione and Wang, 1993). Since this grid becomes very fine near the grid boundaries, a one–dimensional stretching function is applied for each coordinate in order to allow increased time steps for the integration (Kosloff and Tal-Ezer, 1993). For the forward integration in time we use a fourth order Runge–Kutta scheme. Absorbing boundary conditions are imposed by a characteristic treatment (Thompsen, 1990).

GRID GENERATION

The model space of the size $M \times N$ is discretized by $n_x \times n_z$ grid points. The coordinates of the unmapped grid are given by the stretched Chebyshev coordinates u_i and v_j

$$\begin{aligned} u_i &= \frac{M}{2}(1 - h_u(\xi_i)) \qquad i = 0, .., n_x \\ v_j &= \frac{N}{2}(1 - h_v(\nu_j)) \qquad j = 0, .., n_z \end{aligned} \tag{1}$$

where ξ_i and ν_j are the Gauss–Lobatto points defined by the extreme values of the Chebyshev polynomials of order n_x and n_z. h_u and h_v are one–dimensional stretching functions in order to overcome the superfine grid spacing near the boundaries. For details see (Carcione and Wang, 1993; Kosloff and Tal-Ezer, 1993). We introduce the mapping by defining r and s arbitrarily shaped boundary functions (e.g. layer boundaries) in each dimension and divide the model space into $r+1 \times s+1$ patches (figure 1). Each patch is represented by the coordinates at the four intersection points P_i and the boundary functions (e.g. spline functions) f_r, f_{r+1}, g_s and g_{s+1} (figure 2). By means of a transfinite interpolation (Hoschek and Lasser, 1992) each patch is discretized by $m_s \times n_r$ nodes and the generalized coordinates x and z are then given by

$$\begin{aligned} x(k,l) &= \phi_0(i_k) g_s(z(k_1,l)) + \phi_1(i_k) g_{s+1}(z(k_2,l)) \\ &\quad - \big(\phi_0(i_k), \phi_1(i_k)\big) \begin{pmatrix} x(k_1,l_1) & x(k_2,l_1) \\ x(k_1,l_2) & x(k_2,l_2) \end{pmatrix} \begin{pmatrix} \phi_0(k_l) \\ \phi_1(k_l) \end{pmatrix} \\ z(k,l) &= \phi_0(j_l) f_r(x(k,l_1)) + \phi_1(j_l) f_{r+1}(x(k,l_2)) \\ &\quad - \big(\phi_0(i_k), \phi_1(i_k)\big) \begin{pmatrix} z(k_1,l_1) & z(k_2,l_1) \\ z(k_1,l_2) & z(k_2,l_2) \end{pmatrix} \begin{pmatrix} \phi_0(k_l) \\ \phi_1(k_l) \end{pmatrix} \end{aligned} \tag{2}$$

where

$$x(k,l_i) = (1 - i_k)\, x(k_1,l_i) + i_k\, x(k_2,l_i); \qquad l_i \in [l_1, l_2]$$

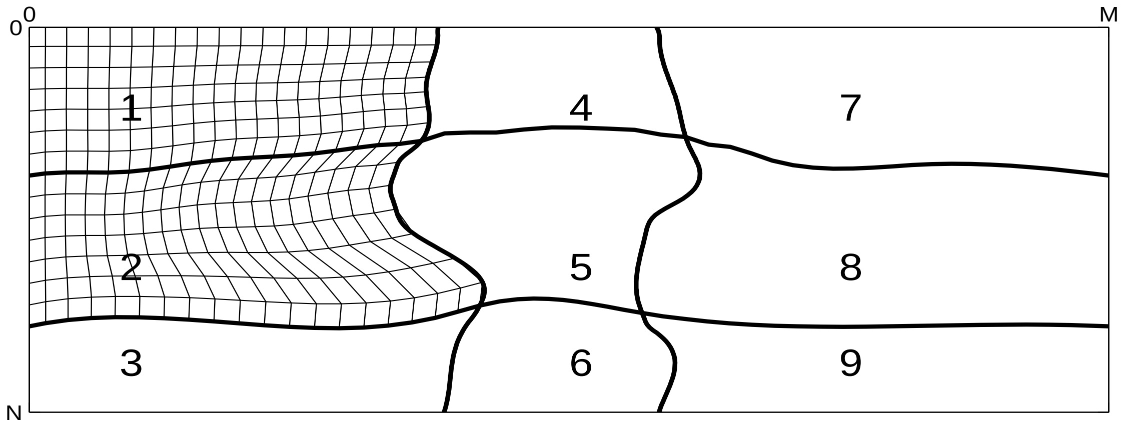

Figure 1: Example of the decomposition of the model space into nine patches. The bold lines mark the boundary functions. The generalized coordinates of the patches 1 and 2 are displayed.

$$\begin{aligned} z(k_i, l) &= (1 - j_l)\, z(k_i, l_1) + j_l\, z(k_i, l_2); \quad k_i \in [k_1, k_2] \\ k_2 &= k_1 + m_s; \quad l_2 = l_2 + n_r; \quad i_k, j_l \in [0, 1]; \quad k = k_1, .., k_2; \quad l = l_1, .., l_2 \end{aligned}$$

ϕ_0 and ϕ_1 are blending functions which interpolate the boundary curves. Since the patches have to be continuous at the boundaries, we use Hermite polynomials

$$\phi_0(i) = 1 - 3i^2 + 2i^3; \quad \phi_1(i) = 3i^2 - 2i^3 \quad i \in [0, 1] \tag{3}$$

which fulfill the condition $\phi_i'(k) = 0$ $(i, k = 0, 1)$.

COORDINATE TRANSFORMATION

The node dependent shift values Δu and Δv of the coordinates u and v can be accomplished analytically by cubic spline interpolants g_x and g_z in x- and z–direction respectively. Thus, the mapping is represented by

$$\begin{aligned} x(i, j) &= u_i + \Delta u(i, j) = u_i + g_x(u_i, v_j) \\ z(i, j) &= v_j + \Delta v(i, j) = v_j + g_z(u_i, v_j) \end{aligned} \tag{4}$$

Calculation of the spatial derivatives of a function f defined on the generalized

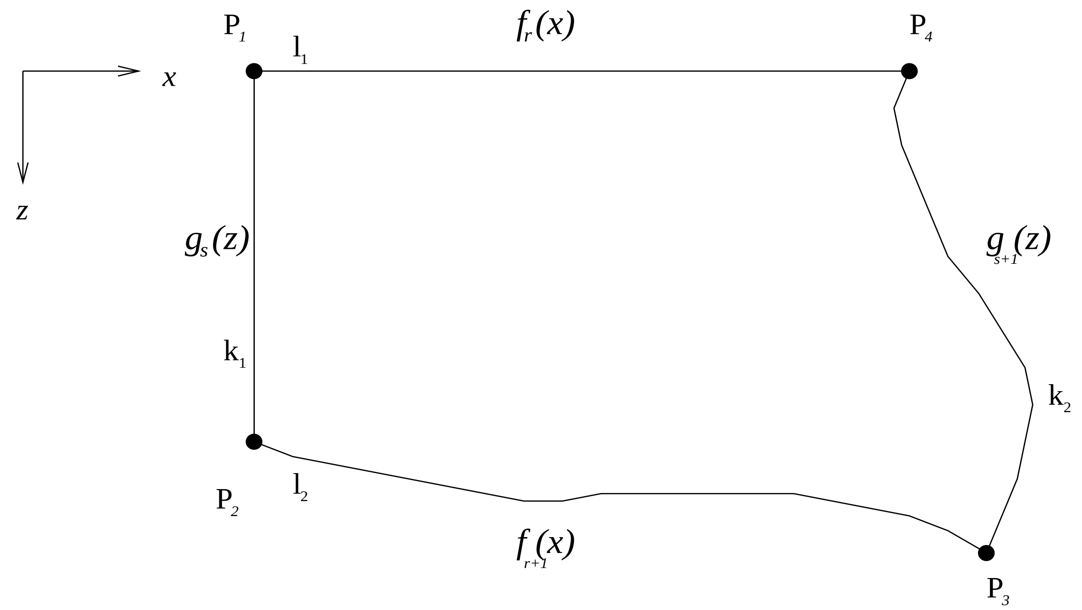

Figure 2: Patch defined by the the points P_1, P_2, P_3 and P_4 and the four boundary functions f_r, f_{r+1}, g_s and g_{s+1}. In the discretized grid the boundary functions represent the l_1^{th}, l_2^{th}, k_1^{th} and k_2^{th} grid line, respectively.

grid, requires spatial derivatives of the mapping functions

$$\frac{\partial f}{\partial x} = \frac{\frac{\partial f}{\partial u} - \frac{\partial f}{\partial v}\overbrace{\frac{\partial g_z}{\partial u}\left(1+\frac{\partial g_z}{\partial v}\right)^{-1}}^{c_x}}{\underbrace{\left(1+\frac{\partial g_x}{\partial u}\right) - \frac{\partial g_x}{\partial v}\frac{\partial g_z}{\partial u}\left(1+\frac{\partial g_z}{\partial v}\right)^{-1}}_{d_x}} = \frac{1}{d_x}\left(\frac{\partial f}{\partial u} - \frac{\partial f}{\partial v}\, c_x\right) \tag{5}$$

$$\frac{\partial f}{\partial z} = \frac{\frac{\partial f}{\partial v} - \frac{\partial f}{\partial u}\overbrace{\left(1+\frac{\partial g_x}{\partial u}\right)^{-1}\frac{\partial g_x}{\partial v}}^{c_z}}{\underbrace{\left(1+\frac{\partial g_z}{\partial v}\right) - \frac{\partial g_z}{\partial u}\left(1+\frac{\partial g_x}{\partial u}\right)^{-1}\frac{\partial g_x}{\partial v}}_{d_z}} = \frac{1}{d_z}\left(\frac{\partial f}{\partial v} - \frac{\partial f}{\partial u}\, c_z\right)$$

c_x, c_z, d_x and d_z are products of the spatial derivatives of the mapping functions. Again, the spatial derivatives of g_x and g_z are derived analytically by means of cubic spline interpolations. The Jacobian of the transformation is

$$J = \left(1+\frac{\partial g_x}{\partial u}\right)\left(1+\frac{\partial g_z}{\partial v}\right) - \frac{\partial g_x}{\partial v}\frac{\partial g_z}{\partial v}. \tag{6}$$

It is a quantitative measure of the change of the grid cell size: $dudv = Jdxdz$.

Velocity Stress Formulation on the transformed grid

In order to calculate the spatial derivatives of the velocity stress formulation of the elastic equations on the transformed grid, new terms have to be added to the existing equations (Fornberg, 1988)

$$
\begin{aligned}
\rho\dot{v}_x &= \frac{1}{d_x}\left(\frac{\partial\sigma_{\mathbf{xx}}}{\partial\mathbf{u}} + \underline{\frac{\partial\sigma_{xx}}{\partial v}}\, c_x\right) + \frac{1}{d_z}\left(\frac{\partial\sigma_{xz}}{\partial u}\, c_z + \frac{\partial\sigma_{\mathbf{xz}}}{\partial\mathbf{v}}\right) \qquad (7)\\
\rho\dot{v}_z &= \frac{1}{d_x}\left(\frac{\partial\sigma_{\mathbf{xz}}}{\partial\mathbf{u}} + \frac{\partial\sigma_{xz}}{\partial v}\, c_x\right) + \frac{1}{d_z}\left(\underline{\frac{\partial\sigma_{zz}}{\partial u}}\, c_z + \frac{\partial\sigma_{\mathbf{zz}}}{\partial\mathbf{v}}\right)\\
\dot{\sigma}_{xx} &= \lambda\left[\frac{1}{d_x}\left(\frac{\partial\mathbf{v_x}}{\partial\mathbf{u}} + \frac{\partial v_x}{\partial v}\, c_x\right) + \frac{1}{d_z}\left(\frac{\partial v_z}{\partial u}\, c_z + \frac{\partial\mathbf{v_z}}{\partial\mathbf{v}}\right)\right] + \mathbf{2}\mu\frac{1}{d_x}\left(\frac{\partial\mathbf{v_x}}{\partial\mathbf{u}} + \frac{\partial v_x}{\partial v}\, c_x\right)\\
\dot{\sigma}_{zz} &= \lambda\left[\frac{1}{d_x}\left(\frac{\partial\mathbf{v_x}}{\partial\mathbf{u}} + \frac{\partial v_x}{\partial v}\, c_x\right) + \frac{1}{d_z}\left(\frac{\partial v_z}{\partial u}\, c_z + \frac{\partial\mathbf{v_z}}{\partial\mathbf{v}}\right)\right] + \mathbf{2}\mu\frac{1}{d_z}\left(\frac{\partial v_z}{\partial u}\, c_z + \frac{\partial\mathbf{v_z}}{\partial\mathbf{v}}\right)\\
\dot{\sigma}_{xz} &= \mu\left[\frac{1}{d_z}\left(\frac{\partial v_x}{\partial u}\, c_z + \frac{\partial\mathbf{v_x}}{\partial\mathbf{v}}\right) + \frac{1}{d_x}\left(\frac{\partial\mathbf{v_z}}{\partial\mathbf{u}} + \frac{\partial v_z}{\partial v}\, c_x\right)\right]
\end{aligned}
$$

Effectively only two additional derivatives (underlined) have to be calculated. In contrast to the spatial derivatives the multiplicative factors c_x, c_z, d_x and d_z (see eq. (5)) have to be calculated only once. If the grid is not transformed $c_x = c_z$ becomes zero, $d_x = d_z$ becomes one and the right hand side reduces to the partial derivatives in bold face ($u = x$ and $v = z$).

DISCUSSION

Computationally the costs for introducing 2D mapping into the Chebyshev code are low. Only two additional spatial derivatives are required. Since all spatial derivatives in this method are calculated analytically by either the Chebyshev method or spline interpolation no numerical dispersion error occurs. The use of generalized coordinates, however, is limited by the minimum grid spacing which is directly related to the time discretisation by the time marching scheme. The explicit Runge–Kutta scheme encounters the condition $\Delta t = \mathcal{O}(N^{-1})$. It is clear that strong lateral and vertical variations in the grid lead to a significant grid refinement, which make the technique nearly impractical. A solution to this problem is to approximate complex geometries (e.g. salt domes) not by a single curve, but to use several curves which are composed to match the geometry.

EXAMPLE

In order to demonstrate the feasibility of this approach we composed a salt dome model and mapped it onto the grid (figure 3). It is not practical to represent the salt dome by a single boundary, since at its base the grid spacing would be very small. The bold lines mark the spline curves which are used to approximate the outline of the salt body.

The Jacobian of the mapping ranges between 0.5 and 2.1 (figure 4). This corresponds qualitatively to the changes in grid spacing. The grid spacing is enlarged where the $J > 1$ and reduced where $J < 1$. The relative changes in the minimum and maximum grid spacing are -0.4 and 0.55.

A snapshot of the vertical component of a simulation run through this model is displayed in figure 5. The velocity within the salt is higher than in the overburden. The wave fronts are not altered by the mapped grid. One might expect diffraction effects at the upper left and right edges of the dome, where the approximating curves intersect, since the outline of the dome is not continuous there. Obviously this is here not the case. This phenomenon is not yet understood and has to be studied in more detail.

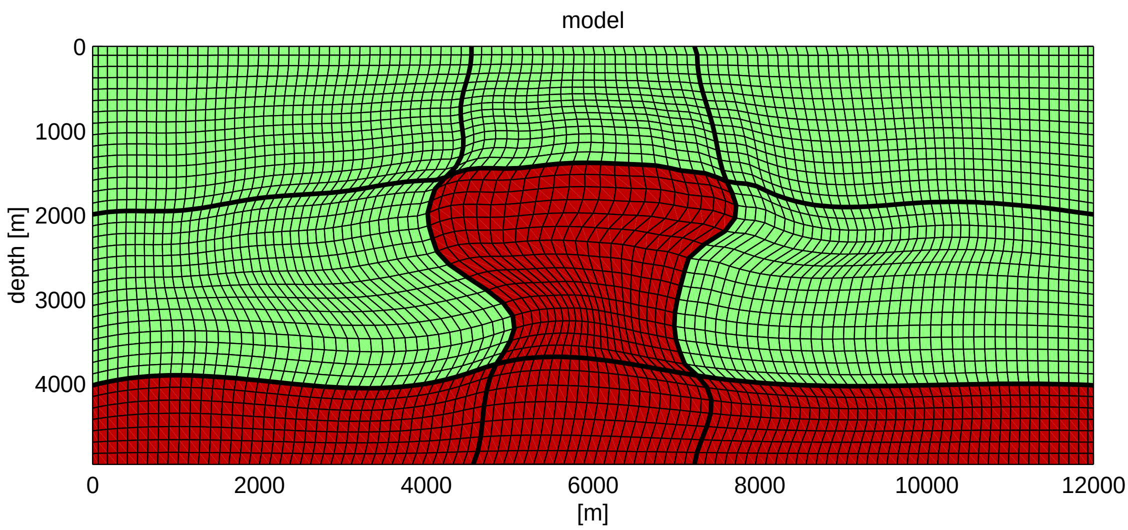

Figure 3: Salt dome model composed of nine patches. The bold lines mark the boundaries of the patches. Every fourth grid line is displayed.

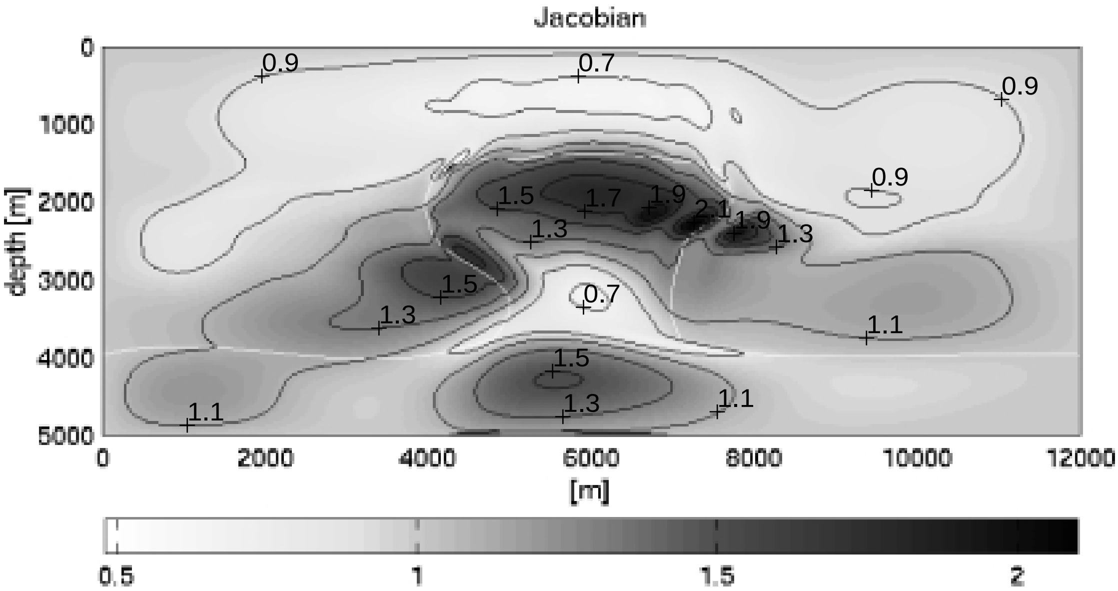

Figure 4: Jacobian of the salt dome model. The white line outlines the salt dome. Contour lines range from 0.5 to 2.1. The grid spacing is enlarged where $J > 1$ and reduced where $J < 1$.

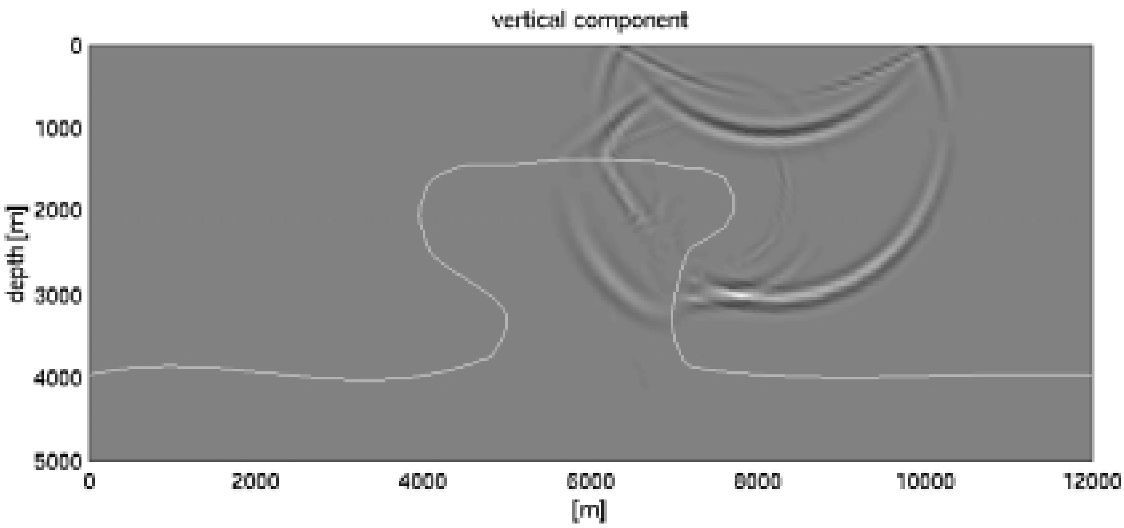

Figure 5: Snapshot through salt model. The white line outlines the salt body. Note that the wave propagation is not disturbed by the curved grid.

CONCLUSION

We developed a method to initialise grids from arbitrarily shaped boundary curves and solve the wave equation on these grids by a pseudospectral method. First simulation runs demonstrate, that it is possible to approximate complex geometries by several curves. Further investigations will comprise a qualitative and quantitative analysis of the factors limiting the grid mapping, a testing of the robustness of the method and a comparison with other methods, e.g. Finite Differences.

REFERENCES

Carcione, J., and Wang, J.-P., 1993, A Chebyshev collocation method for the elastodynamic equation in generalized coordinates , SIAM, Proceedings of the Second International Conference on Numerical Aspects of wave propagation, 269–290.

Fornberg, B., 1988, The pseudospectral method: Accurate representation of interfaces in elastic wave calculations: Geophysics, **53**, no. 5, 625–636.

Gottlieb, D., Hussaini, M., and Orszag, S., 1984, Theory and Applications of Spectral Methods: Spectral Methods for Partial Differential Equations, 1–54.

Hoschek, J., and Lasser, D., 1992, Grundlagen der geometrischen Datenverarbeitung: B.G. Teubner, Stuttgart.

Kosloff, D., and Tal-Ezer, H., 1993, A modified Chebyshev pseudospectral method with an $O(N^{-1})$ time step restriction: J. Comp. Phys., **104**, 457–469.

Thompsen, K., 1990, Time–Dependent Boundary Conditions for Hyperbolic Systems ii: J. Comp. Phys., **89**, 439–461.