Tests on Semi-Lagrangian Transport and Interpolation

M. van Loon
CWI
P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

Abstract. In air pollution models, semi-Lagrangian methods are often used to solve the advective part of the corresponding model equation. Interpolation is an essential part of these methods. In this note, five different interpolation methods will be discussed and results of numerical experiments will be presented. To keep the concentration field nonnegative, filtering techniques are used. Also a monotone interpolation method is examined.

1991 Mathematics Subject Classification: Primary: 65D05. Secondary: 65M25.

1991 CR Categories: G.1.1, G.1.8

Keywords & Phrases: numerical advection, interpolation, semi-Lagrangian transport, air pollution modelling

1. Introduction

For a divergence-free wind field, the following equation is used to model air pollution in 2D (see e.g. [2, 8, 13]):

$$\frac{\partial c}{\partial t} + u\frac{\partial c}{\partial x} + v\frac{\partial c}{\partial y} = f(c) + S(x,y,t), \tag{1.1}$$

where c is a concentration vector, $f(c)$ a chemical reaction term, $S(x,y,t)$ a source term and u and v are components of the wind field in the x- and y-direction, respectively. This equation is to be solved numerically on an $(N+1)\times(N+1)$ equidistant grid covering $[0,1]\times[0,1]$ with points (x_i, y_j), $0 \le i,j \le N$, at time levels $t^n = n\Delta t$.

2. The Semi-Lagrangian Method

The semi-Lagrangian method is in fact a combination of elements of the Eulerian and the Lagrangian approach (see also [12, 13, 14]). It works with an Eulerian grid, with fixed points (x_i, y_j), but it uses characteristics to obtain new concentration values. A major advantage of this approach is that the time step size is not restricted by some Courant condition.

To explain the semi-Lagrangian method in more detail, consider first the homogeneous model equation (i.e. $f(c) = 0$ and $S(x,y,t) = 0$)

$$\frac{\partial c}{\partial t} + u\frac{\partial c}{\partial x} + v\frac{\partial c}{\partial y} = 0. \tag{2.1}$$

Equation (2.1) can be written as

$$\frac{d}{dt} c(\bar{x}(t), \bar{y}(t), t) \;=\; 0. \tag{2.2}$$

Report NM-R9217
ISSN 0169-0388
CWI
P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

where

$$\begin{aligned} \dot{\bar{x}}(t) &= u(\bar{x}(t), \bar{y}(t), t), \\ \dot{\bar{y}}(t) &= v(\bar{x}(t), \bar{y}(t), t). \end{aligned} \tag{2.3}$$

Equation (2.2) expresses that the concentration is constant along the characteristics $(\bar{x}(t), \bar{y}(t))$. To get the value of c in grid point (x_i, y_j) at time $t = t^{n+1}$, we follow the characteristic passing through (x_i, y_j) backwards in time. This characteristic can be found by solving (2.3) with the initial condition

$$\begin{aligned} \bar{x}(t^{n+1}) &= x_i, \\ \bar{y}(t^{n+1}) &= y_j. \end{aligned} \tag{2.4}$$

At time $t = t^n$ the characteristic passes through $(x^n, y^n) = (\bar{x}(t^n), \bar{y}(t^n))$. In general, this will not be a grid point, so interpolation between grid point values has to be carried out to get the value of c in (x^n, y^n). If we have pure advection, i.e. no sources and no chemical reactions, the value $c(x_i, y_j, t^{n+1})$ is equal to $c(x^n, y^n, t^n)$. If there is a source term and chemical reaction, those terms have to be integrated along the characteristic with initial value $c(x^n, y^n, t^n)$.

Since in this note we only want to investigate the behaviour of several interpolation methods, it is sufficient to consider the homogeneous equation (i.e. $f(c) = 0$ and $S(x, y, t) = 0$). From the above, it becomes clear that for each time-step interpolation has to be done for the estimation of c at $t = t^{n+1}$ for every grid point. Thus interpolation plays a very important role and the accuracy of the interpolation method used has direct consequences for the accuracy of the semi-Lagrangian method.

3. Aspects of Interpolation

3.1. Local and Global Interpolation

Interpolation methods can be divided into two categories: local and global methods. Local methods only use a fixed number of neighbouring grid point values, independent of N, to perform the interpolation. Global methods use a number of grid point values dependent on N. Therefore, global interpolation is in general more expensive than local interpolation. Local interpolation methods require for each interpolation a fixed number of floating point operations (flops), independent of N. Therefore, if for every grid point of an $N \times N$ grid interpolation has to be carried out, local interpolation requires $O(N^2)$ flops. Global interpolation methods require $O(N^p)$ flops with $p \geq 2$, because the number of flops required for one interpolation may be dependent on N.

3.2. Accuracy

In the previous section it was stated that the accuracy of the interpolation has direct concequences for the accuracy of the semi-Lagrangian method. Whereas in early studies quadratic interpolation was used, cubic interpolation and occasionally quintic has been used more recently. It is clear that increasing the order of interpolation increases the accuracy of the method, but at additional cost. A good compromise between accuracy and computational cost has to be found. In this note, we will discuss cubic Hermite interpolation, cubic spline interpolation, fifth order Lagrange interpolation and Fourier interpolation.

3.3. Positivity

When chemistry is involved, the computed concentrations of the pollutants have to remain nonnegative. Small negative values may cause large errors, because chemical equilibria are disturbed. Most interpolation methods however do not preserve positivity, in which case after each interpolation step some filtering procedure has to be carried out to make the function c nonnegative. Another way to prevent negative values is using so-called monotone interpolation. This implies that a value, obtained by interpolating in a grid cell, lies between the maximum and the minimum value in the vertices of this grid cell. In this way no negative values can arise, since the initial concentration field is nonnegative. In this note, Hermite interpolation with derivative limiting is an example of a monotone interpolation method.

4. Interpolation Methods

4.1. Hermite Interpolation

Hermite interpolation is used as local interpolation. In 1D, on each sub-interval $[x_i, x_{i+1}]$, $i = 0, \ldots, N\text{-}1$, a cubic Hermite interpolant $p_i(x)$ is determined, satisfying

$$\begin{array}{llll} p_i(x_i) & = c_i \quad , & \dot{p}_i(x_i) & = \dot{c}_i \ , \\ p_i(x_{i+1}) & = c_{i+1} \quad , & \dot{p}_i(x_{i+1}) & = \dot{c}_{i+1} \ , \end{array} \tag{4.1}$$

where the c_i are given data and $\dot{c}_i$ some estimate of the first order derivative at $x = x_i$. This derivative estimate must be at least $O(\Delta x^3)$ accurate, if the Hermite interpolant is to be $O(\Delta x^4)$ accurate. There exist various possible choices for the derivative estimates. Rasch and Williamson [11, 14] tested some possible choices. When using *cubic derivative estimates* the Hermite interpolant becomes identical to the third order Lagrange interpolant. Their results made clear that this gives less good results than the *Akima* and the *Hyman* derivative estimates. The last estimates result in a third order interpolant and will be used in our experiments. For interior points ($2 \leq i \leq N-2$) we thus use the fourth order accurate estimate

$$\dot{c}_i \quad = \quad \frac{c_{i-2} - 8c_{i-1} + 8c_{i+1} - c_{i+2}}{12\Delta x} \quad . \tag{4.2}$$

At the boundaries we use uncentered difference approximations, also proposed by Hyman [9]

$$\begin{aligned} \dot{c}_0 \quad &= \quad \frac{-22c_0 + 36c_1 - 18c_2 + 4c_3}{12\Delta x} \quad , \\ \dot{c}_1 \quad &= \quad \frac{-2c_0 - 3c_1 + 6c_2 - c_3}{6\Delta x} \quad , \\ \dot{c}_N \quad &= \quad \frac{22c_N - 36c_{N-1} + 18c_{N-2} - 4c_{N-3}}{12\Delta x} \quad , \\ \dot{c}_{N-1} \quad &= \quad \frac{2c_N + 3c_{N-1} - 6c_{N-2} + c_{N-3}}{6\Delta x} \quad . \end{aligned}$$

These approximations are $O(\Delta x^3)$ accurate, so the Hermite interpolant is $O(\Delta x^4)$ accurate in the entire domain $[x_0, x_N]$. Note that the interpolant is C^1 continuous. The interpolation formula is given by

$$p_i(\theta) = \alpha_0 \ + \ \alpha_1\theta \ + \ \alpha_2\theta^2 \ + \ \alpha_3\theta^3 \ , \tag{4.3}$$

where

$$\begin{aligned}
\theta &= x - x_i \quad , \\
\alpha_0 &= c_i \quad , \\
\alpha_1 &= \dot{c}_i \quad , \\
\alpha_2 &= \frac{3\Delta_i - \dot{c}_{i+1} - 2\dot{c}_i}{\Delta x} \quad , \\
\alpha_3 &= -\frac{2\Delta_i - \dot{c}_{i+1} - \dot{c}_i}{\Delta x^2} \quad ,
\end{aligned}$$

and Δ_i is the discrete slope on the interval $[x_i, x_{i+1}]$ defined by

$$\Delta_i = \frac{c_{i+1} - c_i}{\Delta x} \quad . \tag{4.4}$$

Because in general the derivative estimates $\dot{c}_i$ and $\dot{c}_{i+1}$ need two grid point values at the left side of the interval $[x_i, x_{i+1}]$ and two at the right side, six grid point values are used to determine $p_i(x)$.

To perform 2D interpolation, a series of six 1D interpolations in x-direction is performed, followed by one interpolation in y-direction. For example, to obtain an interpolated value at the point (x, y), $x_i \leq x \leq x_{i+1}$, $y_j \leq y \leq y_{j+1}$, we first perform six interpolations at rows y_{j+k}, $k = -2, 3$ to get interpolated values at (x, y_{j+k}). These values are used to calculate the derivative estimates at (x, y_j) and (x, y_{j+1}). Finally, we interpolate in y-direction to obtain the desired value at (x, y). From this example it can be seen that, in general, 36 data values are used to perform 2D interpolation in one point.

Standard Hermite interpolation is applied in combination with the modified Ulrich-Ludes filter with r=1, discussed in Section 5.

4.2. Hermite Interpolation with Derivative Limiting

One way to prevent negative concentration values is piecewise monotone interpolation. On each sub-interval $[x_i, x_{i+1}]$ the interpolating function $p_i(x)$ is strictly increasing or decreasing. This implies that the maximum value of the interpolated data will not become larger and the minimum value will not become smaller. Since the last is nonnegative, no negative values can arise. Naturally, the monotonicity property prevents numerical oscillations.

To derive the limited Hermite interpolant $p_i(x)$ on the subinterval $[x_i, x_{i+1}]$ (see also [7]), the same procedure is followed as for standard Hermite interpolation. This time, the estimation of the derivatives is followed by a limiting procedure on the derivative estimates in order to make the interpolant monotone. It can be shown that the interpolant is monotone on the interval $[x_i, x_{i+1}]$ if

$$0 \leq \alpha \leq 3 \quad , \quad 0 \leq \beta \leq 3 \quad , \tag{4.5}$$

where

$$\alpha = \frac{\dot{c}_i}{\Delta_i} \quad , \quad \beta = \frac{\dot{c}_{i+1}}{\Delta_i} \quad , \quad \Delta_i \neq 0. \tag{4.6}$$

It is obvious that $\dot{c}_i$, $\dot{c}_{i+1}$ must be set equal to zero in case $\Delta_i = 0$ in order to get a monotone interpolant. The condition (4.5) is a sufficient, but not a necessary condition. The allowed

values (α, β) form a square in the (α, β)-plane within the monotonicity domain $\Gamma(\alpha, \beta)$, independently found by Fritsch and Carlson [7] and De Boor and Swartz [5]

$$\Gamma \;=\; \Gamma_1 \;\cup\; \Gamma_2, \tag{4.7}$$

where

$$\Gamma_1 \;=\; \{(\alpha,\beta)\,|\,0 \le \alpha \le 3,\; 0 \le \beta \le 3\}, \tag{4.8}$$

$$\Gamma_2 \;=\; \{(\alpha,\beta)\,|\,\phi(\alpha,\beta) \;\le\; 0\}, \tag{4.9}$$

and

$$\phi(\alpha,\beta) \;=\; (3-2\alpha-\beta)^2 \;-\; 3\alpha(\alpha+\beta-2). \tag{4.10}$$

Condition (4.5) is easier to apply than the conditions (4.7)-(4.10) in which α and β may depend on each other. For practical reasons, we will apply condition (4.5). Rasch and Williamson [11] conclude from numerical experiments that using the entire domain Γ results in "minute but discernible improvements" at the cost of extra computation time.

As can be seen from (4.6), every $\dot{c}_i$ is used for $p_{i-1}(x)$ and $p_i(x)$. If $\dot{c}_i$ has to be limited because the monotonicity condition is violated in the interval $[x_i, x_{i+1}]$, this may have effect on the interpolant $p_{i-1}(x)$. If C^1 continuity is required, the limited value of $\dot{f}_i$ must also be used for $p_{i-1}(x)$. If only C^0 continuity is required, the unlimited value of $\dot{c}_i$ may be used for constructing $p_{i-1}(x)$. We will choose for C^0 continuity. This is less restrictive than C^1 continuity. Rasch and Williamson [11, 14] found significant phase errors for the C^1 form. We also found a substantial mass gain in case of the C^1 form.

Fritsch and Carlson [6] extended the limiting procedure to bicubic interpolation. For several reasons, Williamson and Rasch [14] do not follow the approach of Fritsch and Carlson. One of their reasons is that their goal is to construct a 3D advection scheme. Although it seems possible to construct a 3D monotone interpolant, the constraints on the derivative estimates in 3D will be more complicated than they already are in 2D. We will take tensor products in the same way as we did for standard Hermite interpolation.

4.3. Lagrange Interpolation

Lagrange interpolation is also used as local interpolation method. For 1D Hermite interpolation six data points were used for interpolation in a grid interval $[x_i, x_{i+1}]$. With six data points a fifth order Lagrange interpolant can be constructed. In 1D the Lagrange interpolant $F(x)$, fitting data values $(x_0, c_0), (x_1, c_1) ... (x_N, c_N)$ can be defined by

$$F(x) \;=\; F_0(x) \;+\; F_1(x) \;+\; ..... \;+\; F_N(x) \tag{4.11}$$

where

$$F_i(x) \;=\; \left\{ \prod_{\substack{j=0 \\ j\neq i}}^{N} \frac{x - x_j}{x_i - x_j} \right\} c_i. \tag{4.12}$$

Because the smallest values of the factor $|(x-x_0)(x-x_1)(x-x_2)(x-x_3)(x-x_4)(x-x_5)|$ in the truncation error term occur in the interval $[x_2, x_3]$ it seems preferable to have $x \in [x_2, x_3]$.

Hence, for interpolating $x \in [x_i, x_{i+1}]$, we use the data values (x_{i-2}, c_{i-2}), (x_{i-1}, c_{i-1}), (x_i, c_i), (x_{i+1}, c_{i+1}), (x_{i+2}, c_{i+2}) and (x_{i+3}, c_{i+3}) whenever possible. In intervals at the boundaries, $x \in [x_0, x_2]$ or $x \in [x_{N-2}, x_N]$, we use third order Langrange interpolation.

Similar as for Hermite interpolation, we take tensor products of 1D interpolants to perform 2D interpolation. So for 2D Lagrangian interpolation also 36 data values are used.

Lagrange interpolation is applied in combination with the modified Ulrich-Ludes filter with r=1, discussed in section 5.

4.4. Cubic Spline Interpolation

Cubic spline interpolation is a global interpolation method. The concept of a cubic spline function $P(x)$, $x_0 \le x \le x_N$, is the same as for the Hermite function. On each subinterval $[x_i, x_{i+1}]$, the function $P(x)$ is defined by a polynomial $p_i(x)$ such that $P(x)$ is continuous in its first two derivatives. The interpolant $p_i(x)$ is defined by

$$s_i(x) = \theta c_{i+1} + (1-\theta)c_i + h\theta(1-\theta)[(k_i - \Delta_i)(1-\theta) - (k_{i+1} - \Delta_i)\theta] \tag{4.13}$$

where

$$\Delta_i = \frac{c_{i+1} - c_i}{\Delta x}, \quad \theta = \frac{x - x_i}{\Delta x}, \quad h = \Delta x. \tag{4.14}$$

From (4.13) it follows that

$$\dot{p}_i(x_i) = k_i \quad , \quad \dot{p}_i(x_{i+1}) = k_{i+1} \quad , \tag{4.15}$$

so $P(x)$ is continuous in its first derivative. Requiring continuity in the second order derivative of $P(x)$ leads to a linear tridiagonal system of equations for $k_0, k_1, \ldots, k_N$:

$$k_{i-1} + 4k_i + k_{i+1} = 3(\Delta_i + \Delta_{i-1}), \quad i = 1, \ldots, N-1 \quad . \tag{4.16}$$

To solve (4.16) two additional conditions must be given, because there are N+1 unknowns, but only N−1 equations. These conditions can be given in different ways. As in [3] we require the second order derivative of S to be zero at $x = 0$ and $x = 1$. This leads to

$$\begin{array}{rcrcl} 2k_0 & + & k_1 & = & 3\Delta_0 \quad , \\ k_{N-1} & + & 2k_N & = & 3\Delta_{N-1} \quad . \end{array} \tag{4.17}$$

The resulting system of equations can easily be solved, because the corresponding matrix is symmetric and the elements along each diagonal are constant (except for the first and the last). For more detailed information, see Björck and Dahlquist [3] and De Boor [4]. Often Cubic Spline interpolation is used in combination with a time splitting method [2, 12]. Here, 2D interpolation is performed by taking tensor products of 1D interpolations.

Cubic Spline interpolation is used with the Modified Ulrich-Ludes filter, with r=1, as discussed in Section 5.

4.5. Fourier Interpolation

In numerical practice, discrete Fourier transforms (DFT) are often used for the representation of functions. A two-dimensional DFT has the following form (see also [3])

$$c_{ij} = \sum_{k=-M}^{M} \sum_{l=-M}^{M} a_{kl} w_{N+1}^{ik+jl}, \quad i, j \in [0, N] \tag{4.18}$$

where

$$w_{N+1} = exp\left(\frac{2\pi\iota}{N+1}\right) \quad , \quad \iota = \sqrt{-1} \tag{4.19}$$

and

$$M = N/2, \qquad N \text{ even.} \tag{4.20}$$

The coefficients a_{kl} can be calculated by the inverse transformation

$$a_{kl} = \sum_{i=0}^{N}\sum_{j=0}^{N} c_{ij} w_{N+1}^{-(ik+jl)} \quad . \tag{4.21}$$

To perform an interpolation at a grid point (x, y), $x_i \leq x \leq x_{i+1}$, $y_j \leq y \leq y_{j+1}$, we calculate

$$c_{i+\alpha,j+\beta} = \sum_{k=-M}^{M}\sum_{l=-M}^{M} a_{kl} w_{N+1}^{(i+\alpha)k+(j+\beta)l} \tag{4.22}$$

where

$$\alpha = \frac{x - x_i}{\Delta x}, \quad \beta = \frac{y - y_j}{\Delta y} \quad . \tag{4.23}$$

Although direct computation of the a_{kl} and the interpolation for every grid point requires $O(N^4)$ flops, the computational cost may be reduced to $O(N^2 log N^2)$ by using FFT algorithms.

In general, the grid function c_{ij} will not be a periodic function. Therefore, oscillations in the interpolant may occur at the boundaries. We will not be bothered by unwanted oscillations of the interpolant, because the grid functions we will use in our experiments (see below) are zero or have relatively small values at the boundaries. Some experiments made clear that if this is not the case, the solution will be completely disturbed. We will use two different filters, because Fourier interpolation seems to be sensitive for the filter. The Modified Ulrich-Ludes filter with $r = 1$ and the Bartnicki filter will be applied. They are discussed in the next section.

5. Filtering Procedures

After one interpolation step, some grid point values of c may be negative (except for the case that monotone interpolation is used). Negative values often occur in the neighbourhood of (near)discontinuities due to oscillations of the interpolant. In air pollution models, emissions of pollutants can cause (near)discontinuities in the concentration field. It is well-known that a truncated Fourier series as well as Lagrange, Hermite and Spline interpolation produces over- and undershoots near discontinuities. To remove unwanted negative values, a filtering procedure can be carried out. Two different procedures will be used. If we define the total mass M on the grid as

$$M = \sum_{i,j} c_{ij} \quad , \tag{5.1}$$

then both filtering procedures are mass conserving.

5.1. The Modified Ulrich-Ludes Filter

The wavelengths of (unphysical) oscillations introduced by (near) discontinuities are typically $2\Delta x$ or $2\Delta y$. This filtering procedure is designed to remove such oscillations around a zero background. It has been introduced by Ulrich and Ludes [10] and has slightly been modified by Seibert and Morariu [12]. In fact, it is a one-dimensional filter: it removes waves with increasing amplitude along grid lines. After having finished a grid line in one direction, the process must be repeated in reverse direction. In algorithm 1 the filtering procedure proceeding along a grid line from the left to the right is specified (the index j has been dropped).
This procedure is repeated for every grid line (i.e. for all grid lines in the x- or y-direction).

Initialize variables
IF $c_0 < 0$ **THEN**
    $c_1 = c_1 + c_0$
    $c_0 = 0$
ENDIF
for $i = 1, N$ do
    IF $c_i < 0$ **THEN**
        IF $c_{i+1} > c_{i-1}$ **THEN**
            $\Delta M = c_{i-1} + c_i$
            $c_{i-1} = c_i = 0$
            IF $\Delta M < 0$ **THEN**
                $c_{i+1} = c_{i+1} + \Delta M$
            ELSE
                $\Delta M^+ = \Delta M^+ + \Delta M$
            ENDIF
        ENDIF
    ENDIF
next i

ALGORITHM 1 *The Modified Ulrich-Ludes filter algorithm*

When the entire grid has been treated, the residual mass ΔM^+ is distributed over the whole domain

$$\tilde{c}_{ij} = c_{ij} + \frac{(c_{ij})^r}{\sum_{i,j}(c_{ij})^r}\Delta M^+ \quad , \tag{5.2}$$

where $0 \leq r \leq 1$. In our experiments, we will take r=1. This gives a certain anti-diffusive effect, because it enhances relatively large values more than relatively small values. We did also experiments with $r = 0.01$, but only minute differences were noticed. Generally, $r = 1$ gives slightly better results.

5.2. The Bartnicki Filter

This is a straightforward filtering procedure, introduced by Bartnicki [1]. First the negative mass M^- on the grid is calculated

$$M^- = \sum_{\substack{i,j \\ c_{ij}<0}} |c_{ij}| \quad . \tag{5.3}$$

Next, negative values are set equal to zero and the negative mass M^- is equally distributed over the positive grid values

$$c_{ij} = c_{ij} - \frac{M^-}{N_p} \quad , \quad c_{ij} > 0 \quad , \tag{5.4}$$

where N_p is the number of positive concentration values on the grid. This procedure is repeated until all concentration values c_{ij} are nonnegative.

It can easily be proven that this is a convergent process, provided that the total mass on the grid is positive. Typically one or two iterations are necessary to achieve a nonnegative distribution.

6. Numerical Experiments

6.1. Solid Body Rotation

In order to test the several interpolants described above, in combination with a semi-Lagrangian method, the so-called Molenkamp test is performed. This test consists of one or more rotations of a solid body in the plane, which can be realized by substituting

$$\begin{aligned} u &= +2\pi(y - \tfrac{1}{2}) \\ v &= -2\pi(x - \tfrac{1}{2}) \end{aligned} \tag{6.1}$$

into the model equation (2.1), here considered on the domain $\Omega = [0,1] \times [0,1]$. From (6.1) it follows that the characteristics are circles in the xy-plane with centre in $(\frac{1}{2}, \frac{1}{2})$. Three body representations will be used, a Gaussian function defined by

$$c(x,y,0) = \exp(-80[(x - \tfrac{1}{2})^2 + (y - \tfrac{3}{4})^2]), \tag{6.2}$$

a cylinder defined by

$$c(x,y,0) = \begin{cases} 1 & \text{if } (x - \frac{1}{2})^2 + (y - \frac{3}{4})^2 \leq 0.15^2 \\ 0 & \text{elsewhere} \end{cases} \quad . \tag{6.3}$$

and a cone defined by

$$c(x,y,0) = \begin{cases} 1 - \dfrac{z}{R} & \text{if } z \leq R \\ 0 & \text{elsewhere} \end{cases} \quad . \tag{6.4}$$

where

$$z = \left[(x - \tfrac{1}{2})^2 + (y - \tfrac{3}{4})^2\right]^{\frac{1}{2}} \tag{6.5}$$

and

$$R \;=\; \frac{1}{8} \quad . \tag{6.6}$$

The exact solution for the Gaussian can be written as

$$c(x,y,t) = \exp(-80[(x-r(t))^2 + (y-s(t))^2]), \tag{6.7}$$

for the cylinder as

$$c(x,y,t) = \begin{cases} 1 & \text{if } (x-r(t))^2 + (y-s(t))^2 \leq 0.15^2 \\ 0 & \text{elsewhere} \end{cases} \tag{6.8}$$

and for the cone as

$$c(x,y,t) = \begin{cases} 1 \;-\; \dfrac{\tilde{z}}{R} & \text{if } \tilde{z} \leq R \\ 0 & \text{elsewhere} \end{cases} \tag{6.9}$$

where

$$r(t) = (2+\sin(2\pi t))/4, \quad s(t) = (2+\cos(2\pi t))/4 \tag{6.10}$$

and

$$\tilde{z} \;=\; \left[(x-r(t))^2 + (y-s(t))^2\right]^{\frac{1}{2}} \quad . \tag{6.11}$$

To get a well defined problem, boundary conditions must be given at the inflow boundaries. We use Dirichlet conditions.

6.2. Numerical Treatment

The domain Ω is covered with an $(N+1)\times(N+1)$ equidistant grid, where $N = 10$, $N = 20$, $N = 40$ and $N = 80$. To isolate the effect of the interpolation, the departure points will be given exactly. This can easily be done, because the characteristics are circles in the xy-plane and the departure points are only dependent on Δt and not on t. Since we only want to concentrate on interpolation, the time step Δt is chosen in such a way that the characteristics passing through interior points, $i,j \in [1, N-1]$, have departure points inside the domain Ω. This choice avoids a complicated treatment of the boundary conditions. It can be shown that $\Delta t \leq \frac{1}{250}$ satisfies this condition for $N \leq 80$. Results after one rotation will be presented. The total number of time steps then equals 250.

6.3. Results

In the presentation of the results several norms are used to give information about the behaviour of the interpolation methods. The following norms are defined

$$\begin{aligned} \|c\|_\infty &= \max_{ij}|c_{ij}| \quad , \\ \|c\|_2 &= \frac{1}{(N+1)}\sqrt{\sum_{ij} c_{ij}^2} \quad , \\ \text{Mass}(c) &= \frac{1}{(N+1)^2}\sum_{ij} c_{ij} \quad . \end{aligned}$$

We define the error function err_{ij} by the difference of the numerical and the exact solution after k rotations

$$err_{ij} = c_{ij} - c(x_i, y_j, t_k) \tag{6.12}$$

where t_k is the time needed for k rotations. In the tables below, both exact and numerical solution norms are given. Hermite interpolation with derivative limiting is denoted by Hermite1, while standard Hermite is denoted by Hermite2. Hermite2, Lagrange and Cubic Spline are used in combination with the Modified Ulrich-Ludes filter with r=1. DFT1 means Fourier interpolation in combination with the Modified Ulrich-Ludes filter with r=1. DFT2 stands for Fourier interpolation in combination with the Bartnicki filter. For N=80, there are no results available for DFT1 and DFT2 because these experiments required too much computation time. The results of the experiments are given in the Tables 1-12. Also some experiments without filter were carried out. The results of these experiments are presented in Table 13-15 and Figure 4-6.

6.4. Discussion of the results

6.4.1. Diffusion

The maximum norm of the solution and of the error may serve as a measure for the diffusion of the scheme. In this sense, Hermite1 is obviously more diffusive than the other methods. This is especially seen for N=10, see Tables 1,5 and 9. The maximum value decreases by approximately a factor 2, 5 and 8 respectively, whereas other methods perform much better. This is not very surprising, because monotone interpolation does not introduce new maxima. The peak of the Gaussian and the Cone however, lies almost always within a grid cell rather than in a gridpoint. Therefore, Hermite interpolation with derivative limiting will always give a too low value when interpolating in this grid cell. Other methods perform in general better with respect to the maximum norm, but sometimes they show large overshoots in case of the Cylinder, see Tables 1-4 and Figure 1. The overshoot is largest for Lagrange (N=20) and for DFT1 (N=10,40). The maximum of the Gaussian is best maintained by DFT1 and, closely behind, DFT2. Of the local methods, Lagrange and Hermite2 give good results. Lagrange maintains the maximum slightly better than Hermite2. The results of cubic spline seem to be a little better, but comparable to those of Lagrange and Hermite2. When examining the Figures 1-3, it is also seen that DFT1 and DFT2 are less diffusive than the other methods. This is especially seen from Figure 3. Both DFT1 and DFT2 show relatively sharp and high peaks compared to the other methods and their maximum norms are much larger. From Figure 1 we see also that Fourier interpolation is less diffusive. The ground surfaces of Figure 1f and 1g are smaller and only one grid cell in each direction larger than the exact solution. The upper surface however has become very irregular compared to the exact solution but, also compared to the other methods. This irregular surface could be a disadvantage when we have to integrate chemical reaction equations with interpolated values as initial condition.

6.4.2. Mass conservation

An important property for air pollution models is mass conservation. Except for Hermite1, all methods practically conserve mass for N=40,80. For N=20, the resulting mass is also not very different from the initial mass in case of the Gaussian and the Cone. Only for N=10, we see relatively large errors for the mass. Note however that for small numbers of N the total

mass on the grid as defined by (5.1) is not a relevant measure. Hermite1 almost always looses mass. This is especially seen for the Gaussian and the Cone. For example, when looking at the Tables 8 and 12, we see that Hermite1 lost 3-5% of the initial mass whereas the other methods are mass conserving in these cases. Note however that for small numbers of N the total mass on the grid as defined by (5.1) is not a relevant measure.

6.4.3. Accuracy

The maximum norm and the L_2 norm of the error and the solution give information about the accuracy of the methods. From the results for the Gaussian (Tables 5-8) we see that Fourier methods give the most accurate results. Cubic Spline is a little better than Lagrange for N=10,20 but the order is reversed for N=40,80. For non-analytic initial conditions like the Cylinder and the Cone, we also see that the Fourier methods give smaller errors than the other methods. Again, the results for Lagrange and Cubic Spline are comparable. Hermite2 is closely behind. Hermite1 gives obviously the largest errors. In case of the Gaussian, the numerical solutions have approximately the same L_2 norm as the exact solution except for Hermite1. Hermite1 gives always a lower L_2 norm than the other methods. This is also an indication that Hermite1 is more diffusive than other methods. With respect to the L_2 error, DFT1 and DFT2 give the most accurate results. In case of the Gaussian, the L_2 error is a factor 10 or more smaller for N=10,20 and about a factor 5 for N=40. Hermite1 is always more diffusive than the other methods with respect to the L_2 error.

6.4.4. Filtering

Because negative concentration values have to be prevented, filtering procedures were used. Two different procedures were introduced. Both filters seem to give good results, but there are some differences. Because the performance of the Fourier interpolation is sensitive for the filtering procedure, both filters are used in combination with Fourier interpolation. The Modified Ulrich-Ludes filter produces large overshoots for the cylinder, while this phenomenon can hardly be observed for the Bartnicki filter. From Figure 1g however, it can be seen that Fourier interpolation in combination with the Bartnicki filter produces an upper surface which is as irregular as the surfaces produced by Fourier interpolation in combination with the Modified Ulrich Ludes filter (see Figure 1f). Moreover, the ground surface of Figure 1g is also more irregular than the ground surface of Figure 1f. Other interpolation methods are not as sensitive for the filter as Fourier interpolation. Except for DFT interpolation in case of the Cylinder, the filters seem to perform well when looking at the Figures 1-3. The numerical solutions seem to be smooth and no wiggles are observed.

In tabel 13-15 and Figure 4-6, some results for Hermite, Lagrange and Fourier interpolation without filter are presented. It may be concluded that filtering techniques have great influence on the smoothness of the solution and the presence of wiggles. In case of the Cylinder (see fig. 4), interpolation without filter results in wiggles and unsmooth solutions, especially in case of Fourier interpolation. Lagrange interpolation seems to be relatively good without filter. The undershoots are smaller and are only seen at the feet of the body. The results in Table 13-15 are comparable to the results in Tables 1-12. The unfiltered results for the Gaussian seem good. In particular, Fourier interpolation gives a smooth solution (see Figure 5). Hermite shows more wiggles than Lagrange, especially for the Gaussian and the Cone (Figure 5 and

6). Although Figure 6c shows a sharp peak, Fourier interpolation gives wiggles over the entire domain, whereas Lagrange interpolation has wiggles only at the feet of the body. The maximum values are relatively small. Lagrange gives less undershoot than Hermite and also less than DFT in case of $N = 40$.

7. Conclusions

The above results and discussion lead to the following conclusions:

- Hermite interpolation with derivative limiting is in some cases too diffusive and results in an extra loss of mass. Derivative limiting seems to be a too restrictive technique to keep the concentration field nonnegative and free from wiggles.

- Hermite interpolation in combination with a filter gives good results, but filtered Lagrange interpolation of higher order performs generally better at approximately the same computational cost. For smooth solutions fifth order filtered Lagrange interpolation performs much better than filtered Hermite interpolation.

- Cubic Spline interpolation gives good results. However it, it requires more computation time than Hermite and Lagrange. It also needs additional boundary conditions. It performs only a little better than Hermite2 and gives comparable results as Lagrange. Therefore, we do not recommend Cubic Spline interpolation.

- Fourier interpolation methods give the best accuracy of all methods. They are however more expensive than local interpolation methods and can be used only if a periodicity condition is fulfilled. Therefore, Fourier interpolation is not recommended in general. For some specific applications however, Fourier interpolation may be very useful.

- The filters tested here do not prevent wiggles in general. They only keep the solution nonnegative. The Modified Ulrich-Ludes filter gives slightly better results when applied in combination with Fourier interpolation than the Bartnicki filter.

- High order interpolation pays off also for unsmooth solutions. Compare the results for third order Lagrange interpolation (see [14]) and the results for fifth order Lagrange interpolation.

- Lagrange interpolation seems to be a good choice for the semi-Lagrangian method. It uses as much grid point values as Hermite interpolation, but has an higher order of accuracy and it seems to handle (near) discontinuities rather well. It seems worthwhile to examine possibilities to combine the fifth order interpolant with a good filtering procedure which removes (almost) all wiggles.

References

[1] J. Bartnicki. An Efficient Positive Definite Method for the Numerical Solution of the Advection equation. Working Paper WP-86-35, International Institute for Applied Systems Analysis, A-2361 Laxenburg Austria, 1986.

[2] J. Bartnicki, K. Olendrzynski, K. Abert, P. Seibert, and B. Morariu. Numerical Approximation of the transport Equation: Comparision of Five Positive Definite Algorithms. Technical Report WP-90-10, International Institute for Applied Systems Analysis, A-2361 Laxenburg Austria, 1990.

[3] A. Björck and G. Dahlquist. *Numerical Methods.* Prentice Hall, Englewood Cliffs, New Jersey, 1974.

[4] C. de Boor. *A Practical Guide to Splines.* Springer-Verlag, New York, 1978.

[5] C. de Boor and B. Swartz. Piecewise monotone interpolation. *J. Approx. Theory.*, 21: 411–416, 1977.

[6] R.E. Carlson and F.N. Fritsch. Monotone piecewise bicubic interpolation. *SIAM J. Numer. Anal.*, 22: 386–400, 1985.

[7] F.N.Fritsch and R.E. Carlson. Monotone piecewise cubic interpolation. *SIAM J. Numer. Anal.*, 17: 238–246, 1980.

[8] Ø. How, Z. Zlatev, R. Berkowicz, A. Eliassen, and L. Prahm. Comparision of numerical techniques for use in air pollution models with non-linear chemical reactions. *Atmospheric Environment*, 23, No.5: 967–983, 1989.

[9] J.M. Hyman. Accurate monotonicity preserving cubic interpolation. *SIAM J. Sci. Stat. Comput.*, 4: 645–654, 1983.

[10] G. Ludes and W. Ulrich. Numerical simulation of plume-advection over a hill with coupled models. part I: Numerical solution of the transport equation. *Meteor. Atmos. Phys.*, 39: 97–107, 1988.

[11] P.J. Rasch and D.L. Williamson. On shape-preserving interpolation and semi-Lagrangian transport. *SIAM J. Sci. Stat. Comput.*, 4: 656–687, 1991.

[12] P. Seibert and B. Morariu. Improvements of Upstream, Semi-Lagrangian Numerical Advection Schemes. *J. Appl. Meteor.*, January: 117–125, 1991.

[13] A. Staniforth and J. Côté. Semi-Lagrangian Integration Schemes for Atmospheric Models –A Review. *Monthly Weather Review*, 119, No.9, September 1991: 2206–2223, 1991.

[14] D.L. Williamson and P.J. Rasch. Two-Dimensional Semi-Lagrangian Transport with Shape-Preserving Interpolation. *Monthly Weather Review*, 117: 102–129, 1989.

Method	$\|c\|_\infty$	$\|c\|_2$	$\|err\|_2$	$\|err\|_\infty$	mass(c)
Exact	0.100E+01	0.241E+00	0.000E+00	0.000E+00	0.579E−01
Hermite1	0.427E+00	0.117E+00	0.193E+00	0.830E+00	0.550E−01
Hermite2	0.743E+00	0.141E+00	0.189E+00	0.818E+00	0.471E−01
Lagrange	0.964E+00	0.194E+00	0.170E+00	0.964E+00	0.605E−01
Cub. Spline	0.791E+00	0.156E+00	0.122E+00	0.645E+00	0.523E−01
DFT1	0.121E+01	0.180E+00	0.110E+00	0.626E+00	0.480E−01
DFT2	0.931E+00	0.157E+00	0.117E+00	0.574E+00	0.491E−01

TABLE 1. Results for the Cylinder after one rotation, N=10

Method	$\|c\|_\infty$	$\|c\|_2$	$\|err\|_2$	$\|err\|_\infty$	mass(c)
Exact	0.100E+01	0.243E+00	0.000E+00	0.000E+00	0.590E−01
Hermite1	0.719E+00	0.155E+00	0.138E+00	0.745E+00	0.546E−01
Hermite2	0.129E+01	0.210E+00	0.101E+00	0.665E+00	0.605E−01
Lagrange	0.138E+01	0.223E+00	0.979E−01	0.649E+00	0.632E−01
Cub. Spline	0.121E+01	0.205E+00	0.984E−01	0.714E+00	0.597E−01
DFT1	0.127E+01	0.220E+00	0.744E−01	0.757E+00	0.568E−01
DFT2	0.102E+01	0.193E+00	0.802E−01	0.543E+00	0.550E−01

TABLE 2. Results for the Cylinder after one rotation, N=20

Method	$\|c\|_\infty$	$\|c\|_2$	$\|err\|_2$	$\|err\|_\infty$	mass(c)
Exact	0.100E+01	0.256E+00	0.000E+00	0.000E+00	0.654E−01
Hermite1	0.983E+00	0.217E+00	0.909E−01	0.632E+00	0.647E−01
Hermite2	0.117E+01	0.230E+00	0.799E−01	0.703E+00	0.656E−01
Lagrange	0.109E+01	0.231E+00	0.759E−01	0.663E+00	0.654E−01
Cub. Spline	0.113E+01	0.234E+00	0.723E−01	0.738E+00	0.655E−01
DFT1	0.120E+01	0.246E+00	0.484E−01	0.831E+00	0.649E−01
DFT2	0.103E+01	0.228E+00	0.477E−01	0.471E+00	0.635E−01

TABLE 3. Results for the Cylinder after one rotation, N=40

Method	$\|c\|_\infty$	$\|c\|_2$	$\|err\|_2$	$\|err\|_\infty$	mass(c)
Exact	0.100E+01	0.258E+00	0.000E+00	0.000E+00	0.668E−01
Hermite1	0.100E+01	0.240E+00	0.638E−01	0.638E+00	0.669E−01
Hermite2	0.112E+01	0.244E+00	0.582E−01	0.690E+00	0.668E−01
Lagrange	0.109E+01	0.245E+00	0.560E−01	0.663E+00	0.668E−01
Cub. Spline	0.111E+01	0.246E+00	0.537E−01	0.706E+00	0.668E−01

TABLE 4. Results for the Cylinder after one rotation, N=80

Method	$\|c\|_\infty$	$\|c\|_2$	$\|err\|_2$	$\|err\|_\infty$	mass(c)
Exact	0.819E+00	0.127E−00	0.000E+00	0.000E+00	0.325E−01
Hermite1	0.219E+00	0.596E−01	0.994E−01	0.665E−00	0.284E−01
Hermite2	0.443E+00	0.814E−01	0.996E−01	0.662E−00	0.270E−01
Lagrange	0.576E+00	0.117E−00	0.969E−01	0.607E−00	0.363E−01
Cub. Spline	0.433E+00	0.896E−01	0.564E−01	0.401E−00	0.300E−01
DFT1	0.803E+00	0.127E−00	0.732E−02	0.352E−01	0.306E−01
DFT2	0.775E+00	0.120E−00	0.101E−01	0.556E−01	0.297E−01

TABLE 5. Results for the Gaussian after one rotation, N=10

Method	$\|c\|_\infty$	$\|c\|_2$	$\|err\|_2$	$\|err\|_\infty$	mass(c)
Exact	0.100E+01	0.133E+00	0.000E+00	0.000E+00	0.356E−01
Hermite1	0.409E+00	0.821E−01	0.633E−01	0.591E+00	0.277E−01
Hermite2	0.808E+00	0.127E+00	0.235E−01	0.192E+00	0.357E−01
Lagrange	0.856E+00	0.131E+00	0.146E−01	0.144E+00	0.363E−01
Cub. Spline	0.908E+00	0.131E+00	0.112E−01	0.920E−01	0.356E−01
DFT1	0.100E+01	0.134E+00	0.783E−03	0.515E−02	0.355E−01
DFT2	0.998E+00	0.133E+00	0.942E−03	0.431E−02	0.353E−01

TABLE 6. Results for the Gaussian after one rotation, N=20

Method	$\|c\|_\infty$	$\|c\|_2$	$\|err\|_2$	$\|err\|_\infty$	mass(c)
Exact	0.100E+01	0.137E+00	0.000E+00	0.000E+00	0.374E−01
Hermite1	0.723E+00	0.117E+00	0.228E−01	0.277E+00	0.332E−01
Hermite2	0.979E+00	0.136E+00	0.264E−02	0.267E−01	0.373E−01
Lagrange	0.986E+00	0.136E+00	0.125E−02	0.137E−01	0.374E−01
Cub. Spline	0.987E+00	0.136E+00	0.134E−02	0.136E−01	0.374E−01
DFT1	0.100E+01	0.137E+00	0.589E−03	0.487E−02	0.374E−01
DFT2	0.100E+01	0.137E+00	0.614E−03	0.429E−02	0.373E−01

TABLE 7. Results for the Gaussian after one rotation, N=40

Method	$\|c\|_\infty$	$\|c\|_2$	$\|err\|_2$	$\|err\|_\infty$	mass(c)
Exact	0.100E+01	0.138E+00	0.000E+00	0.000E+00	0.383E−01
Hermite1	0.905E+00	0.134E+00	0.546E−02	0.949E−01	0.374E−01
Hermite2	0.998E+00	0.138E+00	0.372E−03	0.477E−02	0.383E−01
Lagrange	0.100E+01	0.138E+00	0.296E−03	0.379E−02	0.383E−01
Cub. Spline	0.998E+00	0.138E+00	0.317E−03	0.336E−02	0.383E−01

TABLE 8. Results for the Gaussian after one rotation, N=80

Method	$\|c\|_\infty$	$\|c\|_2$	$\|err\|_2$	$\|err\|_\infty$	mass(sol)
Exact	0.600E+00	0.795E−01	0.000E+00	0.000E+00	0.134E−01
Hermite1	0.122E+00	0.335E−01	0.719E−01	0.523E+00	0.160E−01
Hermite2	0.172E+00	0.335E−01	0.699E−01	0.519E+00	0.113E−01
Lagrange	0.247E+00	0.501E−01	0.680E−01	0.509E+00	0.154E−01
Cub. Spline	0.187E+00	0.375E−01	0.586E−01	0.433E+00	0.123E−01
DFT1	0.348E+00	0.507E−01	0.339E−01	0.260E+00	0.117E−01
DFT2	0.342E+00	0.529E−01	0.366E−01	0.261E+00	0.128E−01

TABLE 9. Results for the Cone after 1 rotation, N=10

Method	$\|c\|_\infty$	$\|c\|_2$	$\|err\|_2$	$\|err\|_\infty$	mass(sol)
Exact	0.100E+01	0.884E−01	0.000E+00	0.000E+00	0.154E−01
Hermite1	0.208E+00	0.379E−01	0.637E−01	0.796E+00	0.117E−01
Hermite2	0.486E+00	0.641E−01	0.423E−01	0.549E+00	0.154E−01
Lagrange	0.523E+00	0.669E−01	0.332E−01	0.477E+00	0.155E−01
Cub. Spline	0.633E+00	0.734E−01	0.250E−01	0.367E+00	0.154E−01
DFT1	0.955E+00	0.882E−01	0.610E−02	0.446E−01	0.153E−01
DFT2	0.934E+00	0.849E−01	0.706E−02	0.658E−01	0.150E−01

TABLE 10. Results for the Cone after 1 rotation, N=20

Method	$\|c\|_\infty$	$\|c\|_2$	$\|err\|_2$	$\|err\|_\infty$	mass(sol)
Exact	0.100E+01	0.886E−01	0.000E+00	0.000E+00	0.155E−01
Hermite1	0.488E+00	0.615E−01	0.311E−01	0.512E+00	0.126E−01
Hermite2	0.815E+00	0.836E−01	0.972E−02	0.185E+00	0.155E−01
Lagrange	0.830E+00	0.845E−01	0.818E−02	0.170E+00	0.155E−01
Cub. Spline	0.833E+00	0.851E−01	0.773E−02	0.167E+00	0.155E−01
DFT1	0.952E+00	0.882E−01	0.429E−02	0.479E−01	0.154E−01
DFT2	0.945E+00	0.863E−01	0.444E−02	0.550E−01	0.153E−01

TABLE 11. Results for the Cone after 1 rotation, N=40

Method	$\|c\|_\infty$	$\|c\|_2$	$\|err\|_2$	$\|err\|_\infty$	mass(sol)
Exact	0.100E+01	0.894E−01	0.000E+00	0.000E+00	0.159E−01
Hermite1	0.718E+00	0.821E−01	0.972E−02	0.282E+00	0.152E−01
Hermite2	0.885E+00	0.883E−01	0.365E−02	0.115E+00	0.159E−01
Lagrange	0.898E+00	0.886E−01	0.318E−02	0.102E+00	0.159E−01
Cub. Spline	0.901E+00	0.886E−01	0.291E−02	0.994E−01	0.159E−01

TABLE 12. Results for the Cone after 1 rotation, N=80

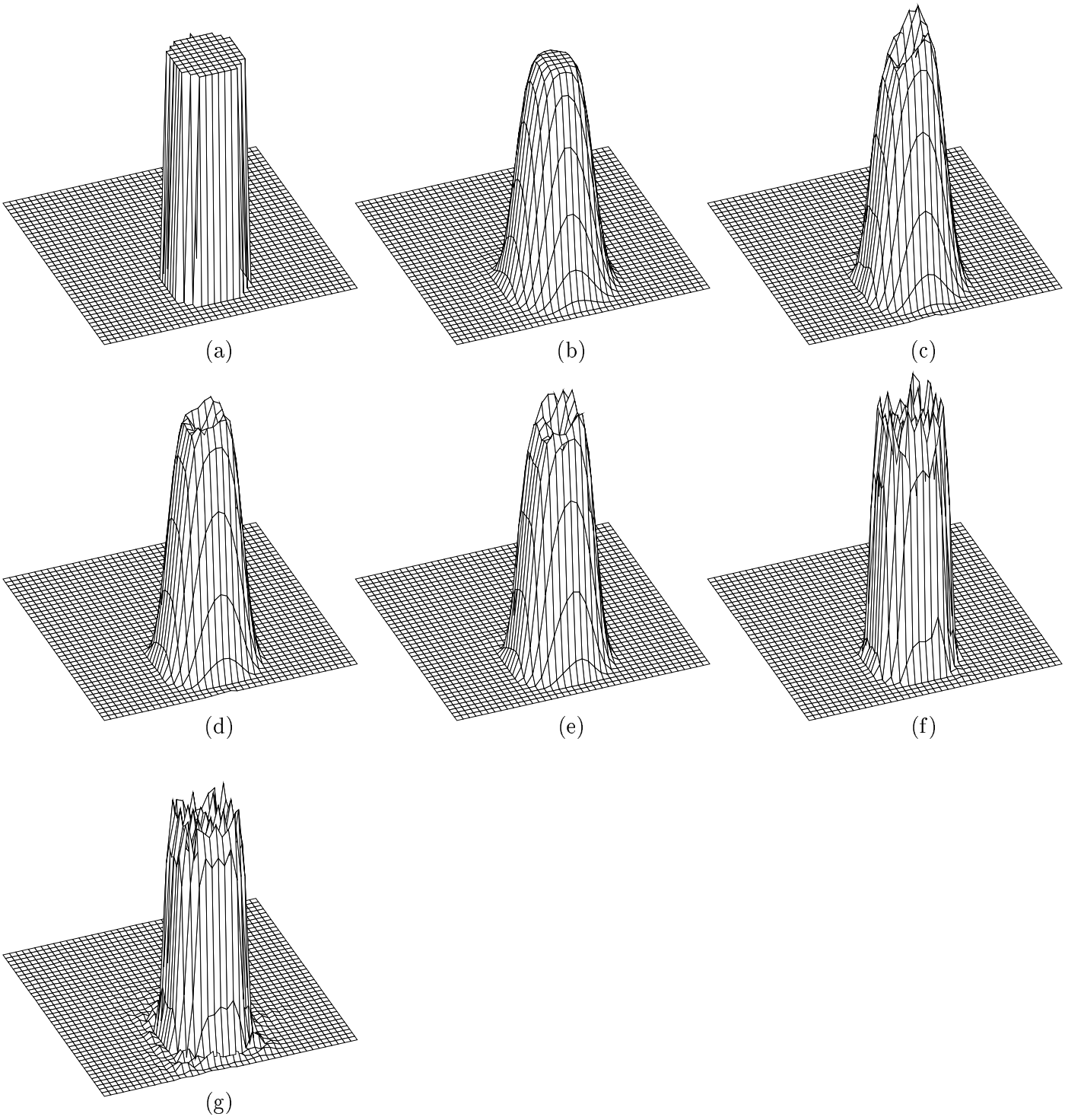


FIGURE 1: Results for the cylinder for N=40 after one rotation, (a) Exact, (b) Hermite1, (c) Hermite2, (d) Lagrange, (e) Cubic Splines, (f) DFT1 and (g) DFT2

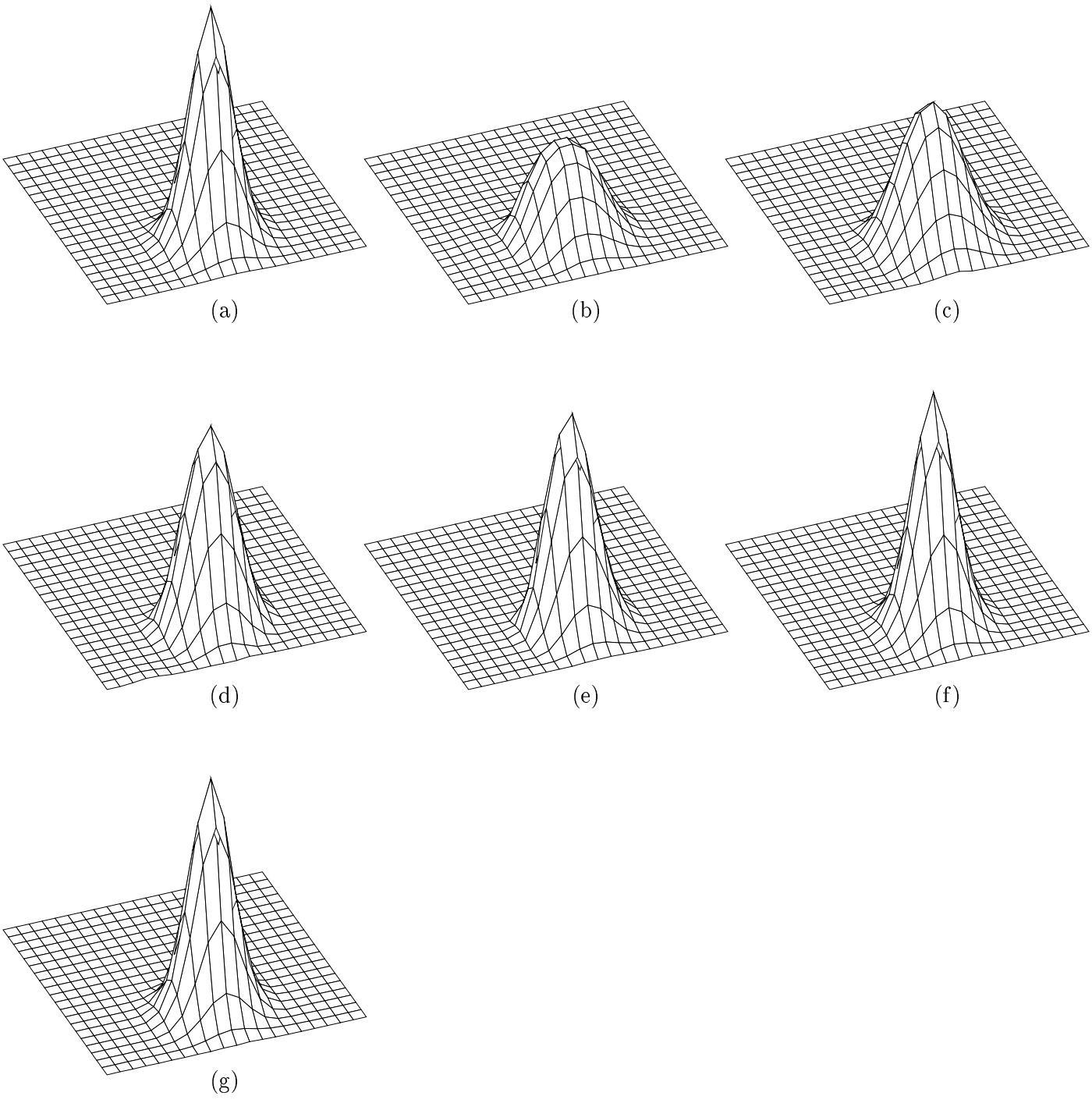


FIGURE 2: Results for the Gaussian, for N=20 after one rotation, (a) Exact, (b) Hermite1, (c) Hermite2, (d) Lagrange, (e) Cubic Splines, (f) DFT1 and (g) DFT2

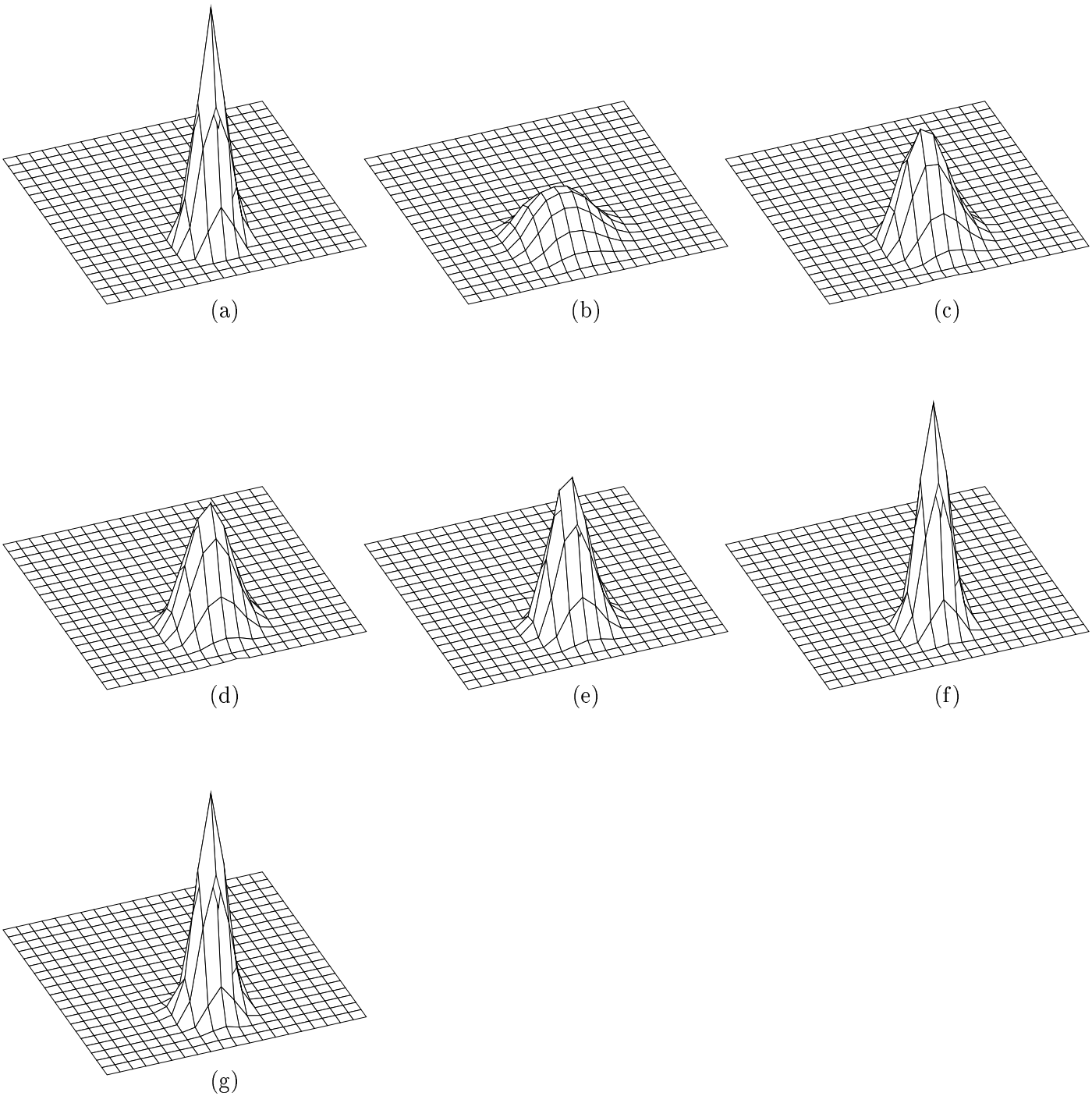


FIGURE 3: Results for the Cone, for N=20 after one rotation, (a) Exact, (b) Hermite1, (c) Hermite2, (d) Lagrange, (e) Cubic Splines, (f) DFT1 and (g) DFT2

Method	N	$\|c\|_\infty$	$\|err\|_2$	$\|err\|_\infty$	mass(sol)	minimum
Hermite	10	0.908E+00	0.172E+00	0.615E+00	0.530E−01	−3.34E−01
Hermite	20	0.128E+01	0.104E+00	0.603E+00	0.600E−01	−1.65E−01
Hermite	40	0.121E+01	0.785E−01	0.616E+00	0.653E−01	−1.85E−01
Lagrange	10	0.870E+00	0.128E+00	0.491E+00	0.553E−01	−1.04E−01
Lagrange	20	0.126E+01	0.986E−01	0.539E+00	0.618E−01	−7.88E−02
Lagrange	40	0.113E+01	0.736E−01	0.583E+00	0.625E−01	−9.07E−02
DFT	10	0.129E+01	0.837E−01	0.332E+00	0.524E−01	−1.96E−01
DFT	20	0.110E+01	0.396E−01	0.201E+00	0.572E−01	−1.43E−01
DFT	40	0.114E+01	0.285E−01	0.204E+00	0.657E−01	−1.37E−01

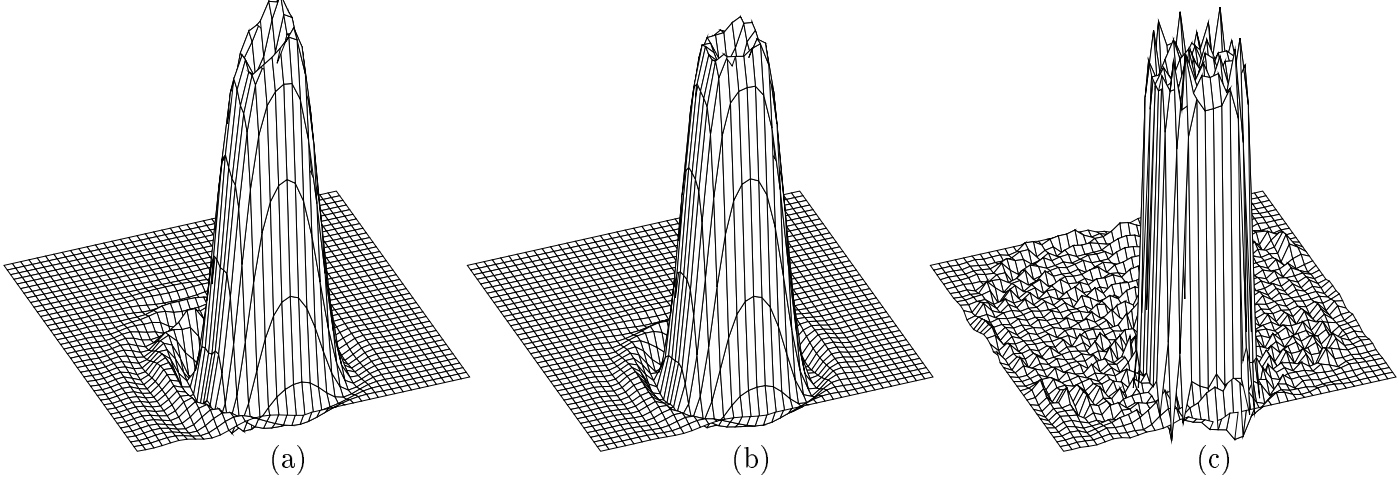
TABLE 13. Unfiltered results for the Cylinder after 1 rotation

Method	N	$\|c\|_\infty$	$\|err\|_2$	$\|err\|_\infty$	mass(sol)	minimum
Hermite	10	0.483E+00	0.895E−01	0.438E+00	0.302E−01	−1.91E−01
Hermite	20	0.814E+00	0.276E−01	0.188E+00	0.356E−01	−6.18E−02
Hermite	40	0.979E+00	0.266E−02	0.269E−01	0.374E−01	−5.96E−03
Lagrange	10	0.453E+00	0.621E−01	0.404E+00	0.323E−01	−4.86E−02
Lagrange	20	0.835E+00	0.170E−01	0.165E+00	0.359E−01	−1.78E−02
Lagrange	40	0.986E+00	0.124E−02	0.139E−01	0.374E−01	−2.14E−04
DFT	10	0.809E+00	0.731E−02	0.251E−01	0.325E−01	−1.15E−02
DFT	20	0.999E+00	0.624E−03	0.384E−02	0.356E−01	−1.41E−03
DFT	40	0.100E+01	0.547E−03	0.487E−02	0.374E−01	−1.33E−03

TABLE 14. Unfiltered results for the Gaussian after 1 rotation

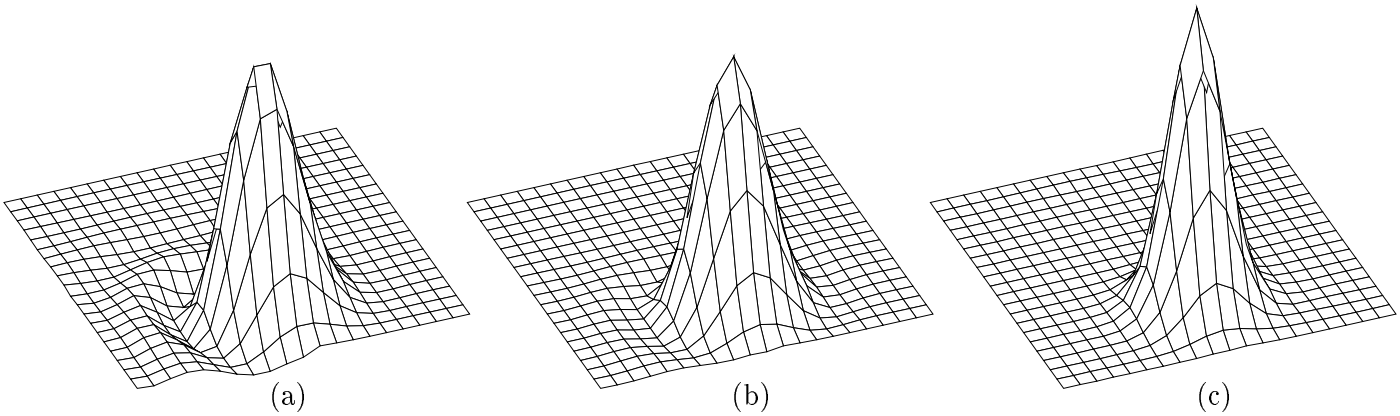
Method	N	$\|c\|_\infty$	$\|err\|_2$	$\|err\|_\infty$	mass(sol)	minimum
Hermite	10	0.248E+00	0.699E−01	0.422E+00	0.128E−01	−1.08E−01
Hermite	20	0.582E+00	0.431E−01	0.461E+00	0.155E−01	−1.09E−01
Hermite	40	0.837E+00	0.962E−02	0.163E+00	0.155E−01	−3.17E−02
Lagrange	10	0.226E+00	0.572E−01	0.398E+00	0.136E−01	−3.16E−02
Lagrange	20	0.584E+00	0.308E−01	0.416E+00	0.153E−01	−4.36E−02
Lagrange	40	0.847E+00	0.781E−02	0.153E+00	0.155E−01	−1.83E−02
DFT	10	0.564E+00	0.211E−01	0.762E−01	0.135E−01	−4.61E−02
DFT	20	0.953E+00	0.523E−02	0.472E−01	0.153E−01	−1.02E−02
DFT	40	0.969E+00	0.240E−02	0.306E−01	0.155E−01	−1.29E−02

TABLE 15. Unfiltered results for the Cone after 1 rotation



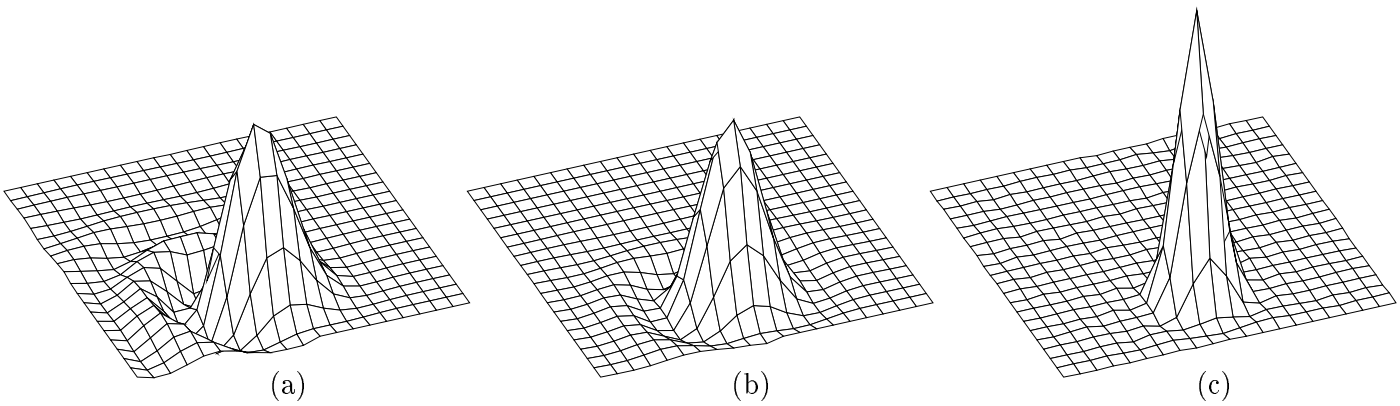
(a) (b) (c)

FIGURE 4: Unfiltered results for the Cylinder for N=40 after one rotation, (a) Hermite, (b) Lagrange and (c) DFT



(a) (b) (c)

FIGURE 5: Unfiltered results for the Gaussian for N=20 after one rotation, (a) Hermite, (b) Lagrange and (c) DFT



(a) (b) (c)

FIGURE 6: Unfiltered results for the Cone for N=20 after one rotation, (a) Hermite, (b) Lagrange and (c) DFT