LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

LLNL-JRNL-402504

A Cartesian Embedded Boundary Method for the Compressible Navier-Stokes Equations

M. Kupiainen, B. Sjogreen

March 26, 2008

Journal of Scientific Computing

Disclaimer

A Cartesian Embedded Boundary Method for the Compressible Navier-Stokes Equations

Marco Kupiainen*
Björn Sjögreen†

March 21, 2008

Abstract

We here generalize the embedded boundary method that was developed for boundary discretizations of the wave equation in second order formulation in [6] and for the Euler equations of compressible fluid flow in [11], to the compressible Navier-Stokes equations. We describe the method and we implement it on a parallel computer. The implementation is tested for accuracy and correctness. The ability of the embedded boundary technique to resolve boundary layers is investigated by computing skin-friction profiles along the surfaces of the embedded objects. The accuracy is assessed by comparing the computed skin-friction profiles with those obtained by a body fitted discretization.

1 Introduction

In fluid dynamics one is often interested in solving the Navier-Stokes equations in some more or less complicated geometry. Generating a computational structured mesh is often the most time consuming task in the solution procedure.

In this paper we present a new Cartesian grid finite difference method for the compressible Navier-Stokes equations. The method uses embedded boundaries to handle geometries. The embedded boundary discretization is an adaption of the method for the wave equation in [6] to the compressible Navier-Stokes equations. The same method was previously generalized to hyperbolic systems in [11], and an early version for viscous

*Universitè Pierre et Marie Curie,4 Place Jussieu 75252 Paris Cedex 05(kupiainen@lmm.jussieu.fr)
†Center for Applied Scientific Computing, Lawrence Livermore National Lab, Livermore, CA 94551 (sjogreen2@llnl.gov). This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344. UCRL-JNRL-XXXXXX

equations was developed in [8]. The method is second order accurate at the embedded boundary. Furthermore, we will present a simple strategy for for local grid refinement to resolve fine flow features such as boundary layers and shocks.

The purpose of this paper is to demonstrate and highlight the embedded boundary method for finite difference approximations of the compressible Navier-Stokes equations.

Most previous work on embedded boundary methods has focused on finite volume schemes e.g. [10]. Examples are the BoxLib and Chombo frameworks, see [1], see also [3] for a related finite volume method. We here use finite difference approximations to solve the PDEs of fluid flows. Advantages of finite difference methods are a greater flexibility in choice of stencils and boundary approximations. By taking advantage of this flexibility, it has been possible to develop the new method such that it does not suffer from any small cell CFL restriction caused by the very small grid cells that are cut out by the embedded boundary.

We have not been able to prove that the new method is conservative at the embedded boundary. However, conservation becomes less important when equations with physical viscosity, such as the Navier-Stokes equations, are solved, and we have not experienced any practical difficulties from this.

The outline of this paper is as follows. We present the equations and the difference approximation on meshes with local refinements, but away from embedded boundaries, in Section 2. In Section 3 we describe the boundary conditions at the embedded boundaries. Numerical experiments in Section 4 show that the expected accuracy is achieved in the implementation. Furthermore we compute supersonic viscous flow past a cylinder and we investigate the accuracy of the computed skin-friction profile along the embedded surface. We compare these results with results obtained with a high order approximation on a body fitted grid.

2 Equations and Numerical Method

We consider the compressible Navier-Stokes equations for a perfect gas in two and three space dimensions, which can be written in dimensionless units as (using Einstein's summation convention):

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_j}\left(\rho u_j\right) = 0, \tag{1}$$

$$\frac{\partial (\rho u_i)}{\partial t} + \frac{\partial}{\partial x_j}\left(\rho u_i u_j + p\delta_{ij} - \frac{\alpha(T)}{Re}\left(2S_{ij} - \frac{2}{3}\delta_{ij}S_{kk}\right)\right) = 0, \quad i = 1,2,3, \tag{2}$$

$$\frac{\partial e}{\partial t} + \frac{\partial}{\partial x_j}\left((e+p)u_j - \frac{\alpha(T)\gamma}{RePr(\gamma-1)}\frac{\partial}{\partial x_j}\left(\frac{p}{\rho}\right) - \frac{\alpha(T)}{Re}\left(2S_{ij} - \frac{2}{3}\delta_{ij}S_{kk}\right)u_i\right) = 0, \tag{3}$$

where ρ is the density, $u_i,\ i = 1,2,3$ are the velocities in x, y and z directions respectively, p is the pressure, Re is the Reynolds number, and Pr is the Prandtl number. The viscous

shear stress tensor is given by

$$S_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right),$$

and the total energy per unit volume is

$$e = \rho\left(\frac{p}{\gamma - 1} + \frac{u_i u_i}{2}\right). \tag{4}$$

The temperature T is given by

$$T = \frac{Mp}{R_w \rho}, \tag{5}$$

where M is the molar mass of the fluid considered and R_w is the universal gas constant. The function $\alpha(T)$ describes how the viscosity depends on the temperature and is either taken constant ($\alpha(T) \equiv 1$) or calculated using Sutherland's law

$$\alpha(T) = \left(\frac{T}{T_\infty}\right)^{1.5} \frac{T_\infty + 110}{T + 110},$$

where T_∞ is a reference temperature. The speed of sound c, is related to the pressure and density by

$$c^2 = \gamma \frac{p}{\rho},$$

where $\gamma = \frac{C_p}{C_v}$ is the ratio between specific heats and $C_p - C_v = \frac{R_w}{M}$. Both γ and $\frac{R_w}{M}$ are constants. Pr is a constant that relates thermal conductivity to viscosity. For a perfect gas $Pr = 0.72$.

If the flow is planar, all derivatives with respect to x_3 are set to zero and $u_3 \equiv 0$.

For boundary conditions we distinguish between three types of boundaries; inflow, outflow, and solid wall. At the inflow boundaries, we impose Dirichlet conditions on all flow variables. At the outflow boundaries we impose homogeneous Neumann conditions on the velocities and on the temperature, and at solid walls we impose the no-slip adiabatic conditions

$$\begin{aligned} &u_i = 0, \quad i = 1, 2, 3, \\ &\frac{\partial T}{\partial n} = 0. \end{aligned} \tag{6}$$

For the numerical discretization, we will need to add one numerical boundary condition at outflow boundaries and at solid-wall boundaries.

2.1 Numerical Approximation

The Navier-Stokes equations are discretized by a second order accurate total variation diminishing (TVD) finite difference approximation for the convective terms. The viscous terms are discretized by centered differences. The Navier-Stokes system is first approximated on semi-discrete form, and then advanced in time by a Runge-Kutta scheme. The computational domain is the box

$$0 \le x \le L_x \quad 0 \le y \le L_y \quad 0 \le z \le L_z.$$

Geometrical objects in the domain are cut out from the box by the embedded boundary method. This will be described in Section 3, and here we present the numerical method in the absence of embedded boundaries.

We introduce a uniform Cartesian grid with $N_1 \times N_2 \times N_3$ points with equal spacing in all three directions,

$$\begin{aligned} x_i &= (i-1)h, \;\; i = 1, 2, \ldots, N_1 \\ y_j &= (j-1)h, \;\; j = 1, 2, \ldots, N_2 \\ z_k &= (k-1)h, \;\; k = 1, 2, \ldots, N_3 \end{aligned}$$

where

$$h = L_x/(N_1 - 1) = L_y/(N_2 - 1) = L_z/(N_3 - 1)$$

If necessary, the size of the box is adjusted slightly to make h of equal size in all three directions.

The semi-discrete approximation of (1)–(3) in the interior, away from boundaries is

$$\frac{d}{dt}\mathbf{u}_{i,j,k}(t) + D_+^x g^{(x)}_{i-1/2,j,k} + D_+^y g^{(y)}_{i,j-1/2,k} + D_+^z g^{(z)}_{i,j,k-1/2} = D_+^x (g_v^{(x)})_{i-1/2,j,k} + D_+^y (g_v^{(y)})_{i,j-1/2,k} + D_+^z (g_v^{(z)})_{i,j,k-1/2}, \quad (7)$$

where $\mathbf{u} = (\rho \;\; \rho u_1 \;\; \rho u_2 \;\; \rho u_3 \;\; e)$ and where $\mathbf{u}_{i,j,k}$ denotes the approximation of $\mathbf{u}(x_i, y_j, z_k)$. $g^{(x)}_{i-1/2,j,k}$ is the numerical flux of the second order extension to the Godunov scheme approximating the convective fluxes. $g^{(x)}_{i-1/2,j,k}$ uses a Riemann solver with van Albada slope limiter [14] applied to the primitive variables. The numerical flux differences are five point schemes. The forward- and backward difference operators are defined as

$$D_+^x u_{i,j,k} = (u_{i+1,j,k} - u_{i,j,k})/h \qquad D_-^x u_{i,j,k} = (u_{i,j,k} - u_{i-1,j,k})/h$$

and similarly for the y- and z-directions. We also define the centered difference as

$$D_0^x u_{i,j,k} = (u_{i+1,j,k} - u_{i-1,j,k})/2h.$$

The viscous fluxes $(g_v^x)_{i-1/2,j,k}, (g_v^y)_{i,j-1/2,k}$, and $(g_v^z)_{i,j,k-1/2}$ contain first derivatives. For example the x-direction viscous fluxes are

$$(g_v^{(x)})_{i-1/2,j,k} = \begin{pmatrix} 0 \\ \frac{4\alpha_{i-1/2,j,k}}{3} D_-^x u_{i,j,k} - \frac{2\alpha_{i-1/2,j,k}}{3} D_0^y \frac{v_{i,j,k}+v_{i-1,j,k}}{2} - \frac{2\alpha_{i-1/2,j,k}}{3} D_0^z \frac{w_{i,j,k}+w_{i-1,j,k}}{2} \\ \alpha_{i-1/2,j,k}(D_0^y \frac{u_{i,j,k}+u_{i-1,j,k}}{2} + D_-^x v_{i,j,k}) \\ \alpha_{i-1/2,j,k}(D_0^z \frac{u_{i,j,k}+u_{i-1,j,k}}{2} + D_-^x w_{i,j,k}) \\ f_5 + k_{i-1/2,j,k} D_-^x T_{i,j,k} \end{pmatrix}. \quad (8)$$

where

$$f_5 = \frac{u_{i,j,k}+u_{i-1,j,k}}{2}(g_{v,2}^{(x)})_{i-1/2,j,k} + \frac{v_{i,j,k}+v_{i-1,j,k}}{2}(g_{v,3}^{(x)})_{i-1/2,j,k} + \frac{w_{i,j,k}+w_{i-1,j,k}}{2}(g_{v,4}^{(x)})_{i-1/2,j,k}. \quad (9)$$

$g_v^{(x)}$ is a vector with five components, $g_{v,m}^{(x)}$ denotes its m:th component. We define

$$\alpha_{i-1/2,j,k} = \frac{1}{Re}\frac{\alpha(T_{i-1,j,k}) + \alpha(T_{i,j,k})}{2},$$

and similarly for $k_{i-1/2,j,k} = \frac{\alpha_{i-1/2,j,k}\gamma}{Pr(\gamma-1)}$. The discretization of the viscous fluxes means that second derivatives are approximated by standard finite difference formulas, e.g.,

$$u_{xx}(x_i, y_j, z_k) = D_+^x D_-^x u_{i,j,k} + \mathcal{O}(h^2) = \frac{u_{i+1,j,k} - 2u_{i,j,k} + u_{i-1,j,k}}{h^2} + \mathcal{O}(h^2) \quad (10)$$

and

$$u_{xy}(x_i, y_j, z_k) = D_0^x D_0^y u_{i,j,k} + \mathcal{O}(h^2) = D_+^x \frac{D_0^y u_{i,j,k} + D_0^y u_{i-1,j,k}}{2} + \mathcal{O}(h^2). \quad (11)$$

2.2 Approximation near the Boundary

Three different conditions at the boundaries of the computational domain are considered, inflow boundary, outflow boundary, and wall boundary. We describe the boundary procedure for the boundary at $x = 0,\ (i = 1)$. The other sides are similar.

Boundary conditions are needed for $u_{1,j,k}, u_{2,j,k}$, since (7) is a five-point scheme. Here $u_{i,j,k}$ denotes any flow variable. By including a boundary slope limiting procedure into the interior scheme (7) whereby the slope $s_{1,j,k}$, needed for the numerical flux, $g_{1+1/2,j,k}^{(x)}$ is extrapolated to first or second order

$$s_{1,j,k} = s_{2,j,k}, \quad (12)$$

or

$$s_{1,j,k} = 2s_{2,j,k} - s_{3,j,k}, \quad j = 1, 2, \ldots, N_2, k = 1, 2, \ldots, N_3, \tag{13}$$

the interior scheme effectively becomes a three point scheme at the boundary. Therefore, we need only consider boundary values at the outermost point, $i = 1$.

At the outermost points ($i = 1$) the values are imposed by setting Dirichlet data at inflow boundaries.

At outflow boundaries, we use homogeneous Neumann condition

$$\mathbf{u}_{1,j,k} = \frac{4}{3}\mathbf{u}_{2,j,k} - \frac{1}{3}\mathbf{u}_{3,j,k}, \quad j = 1, \ldots, N_2, \quad k = 1, \ldots, N_3.$$

The above Neumann condition is second order accurate and can introduce unphysical values near the boundary. If the solution is smooth this will not cause any problems, but if the solution has low regularity (described in Section 2.4) or is near vacuum conditions, we replace the above condition with the more robust first order Neumann condition

$$\mathbf{u}_{1,j,k} = \mathbf{u}_{2,j,k}, \quad j = 1, \ldots, N_2, \quad k = 1, \ldots, N_3.$$

This reduction of accuracy is done routinely in TVD difference schemes, and only takes place at a small nuber of grid points.

At solid walls we impose the adiabatic wall conditions (6). These give the velocities $(u_m)_{1,j,k} = 0$, $m = 1, 2, 3$. Second order accurate approximation of the temperature normal derivative condition gives

$$T_{1,j,k} = \frac{4}{3}T_{2,j,k} - \frac{1}{3}T_{3,j,k}.$$

As extra numerical boundary condition, we extrapolate the pressure,

$$p_{1,j,k} = 2p_{2,j,k} - p_{3,j,k}. \tag{14}$$

If (14) gives a negative pressure at $i = 1$, we replace the condition by the more robust first order extrapolation

$$p_{1,j,k} = p_{2,j,k}.$$

The constitutive relations give the density and the energy at the wall from the temperature and pressure.

2.3 Local grid refinement

We wish to accurately approximate the Navier-Stokes with as high Reynolds number as possible. The computational can be very large, because it scales roughly as Re^3. We try to reduce this cost by introducing finer grids typically only where there are boundary layers and turbulent wakes.

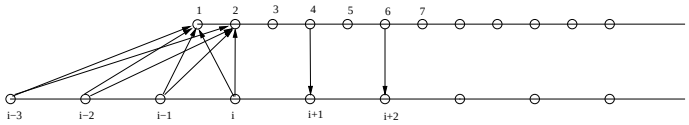


Figure 1: Schematic of grid/grid interpolation in one space dimension.

2.3.1 Grid to grid interpolation

A one dimensional example of how the grid refinements are set up is shown in Fig. 1. The coarse grid with points x_j, $j = 1, 2, \ldots, N$ is given (the lower grid in Fig. 1). Assume that we have decided in some way, that refinement is needed from grid point x_i and upward. Assume that the discretization uses a five point computational stencil. We wish to apply the stencil up to the grid point x_i. We therefore let the points x_{i+1} and x_{i+2} be a part of the discretization. These points will get values by interpolation (injection) from the fine grid. Let the grid points of the fine grid be denoted y_j. The fine grid, in Fig. 1 refined by a factor 2, begins at $y_3 = (x_i + x_{i+1})/2$, the first point where refinement is needed, which is not present in the coarse grid. The points y_1 and y_2 are ghost points, i.e., points that are considered outside the computational domain, but are needed to be able to apply the computational stencil at y_3. The grid to grid interpolation procedure defines the function values at y_1 and y_2 by interpolation from the coarser grid, and the function values at x_{i+1} and x_{i+2} by interpolation from the fine grid. In this one dimensional example, values at all points except y_1 can be defined by injection from the other grid. In two and three space dimensions, there are more interpolated values because of the so called hanging nodes that occur, see Fig. 2.

In three space dimensions, the grid to grid interpolation is done with tensor product Lagrange interpolation. Lagrangian grid to grid interpolation is stable for many discretizations of hyperbolic conservation laws, see, e.g., [13, 2]. We define the rth degree three dimensional interpolant πf of f with the lower left corner of the stencil at $(x_{i_0}, y_{j_0}, z_{k_0})$, as

$$\pi f(x, y, z) = \sum_{m=0}^{r} \sum_{n=0}^{r} \sum_{o=0}^{r} f(x_{i_0+m}, y_{j_0+n}, z_{k_0+o}) L_m(x) L_n(y) L_o(z), \tag{15}$$

where $L_\nu(\xi),\ \nu = m, n, o$ is the standard Lagrange polynomial basis,

$$L_m(x) = \prod_{\substack{p=0 \\ p \neq m}}^{r} \frac{x - x_{p+i_0}}{x_{m+i_0} - x_{p+i_0}},$$

and similarly for $L_n(y)$ and $L_o(z)$. Fig. 1 indicates that for the one dimensional example, the interpolation polynomial has $r = 3$ and has lower left corner at x_{i-3}, i.e., $i_0 = i - 3$. Note that an r degree interpolant gives interpolated values that are of order of accuracy $r + 1$.

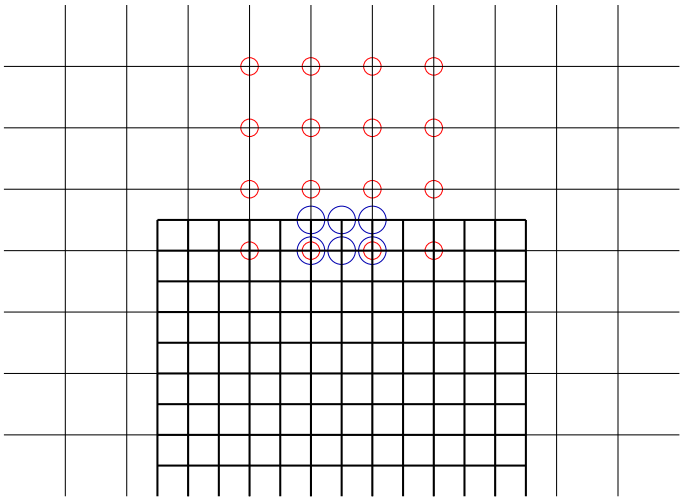


Figure 2: Schematic of grid/grid interpolation in two space dimensions. The larger (blue) circles denote interpolation points. The smaller (red) circles denote the domain of dependence of the fourth order accurate Lagrange interpolation stencil.

The interpolation stencils should be made such that the interpolation is explicit, meaning that the interpolation stencil on the coarse grid (for computing values at y_1 and y_2 in Fig. 1) should not contain points (x_{i+1} and x_{i+2} in Fig. 1) that have been injected from the fine grid.

We choose $r = 3$ because we compute second derivatives when updating the fluxes for the Navier-Stokes equations, and hence we need at least fourth order accurate interpolation to preserve overall second order accuracy, see [2].

2.4 Switching interpolation order

In the presence of sharp gradients, shocks or when the solution is detected to have small pointwise regularity, we use linear interpolation ($r = 1$) instead of higher order interpolation to avoid unwanted oscillations. To detect the non-smoothness we use a so called wavelet filter to approximate the Lipschitz exponent α_i

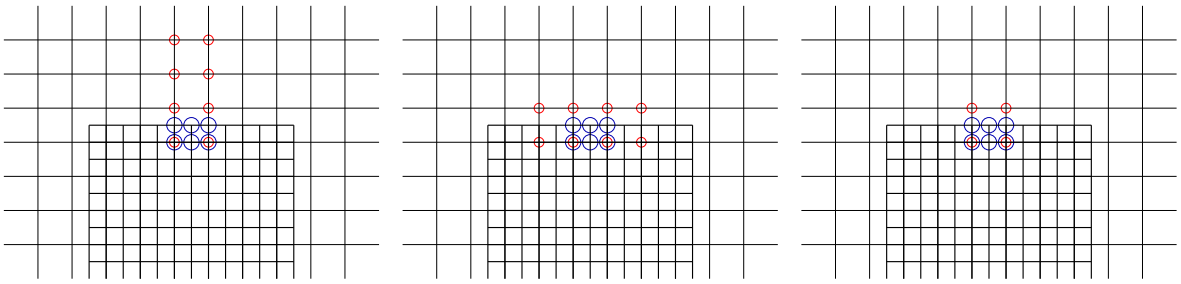
$$|f(\vec{x} + h\vec{e_i}) - f(\vec{x})| \sim Ch^{\alpha_i}, \quad i = 1, \ldots, 3. \tag{16}$$

Consider one space dimension and assume that we are given the function u_i. The difference

$$d_i = u_i - (u_{i+1} + u_{i-1})/2 \tag{17}$$

is a measure of the smoothness of the function. If u_i is smooth, the average at x_i is close to the actual value u_i, making d_i small. In general any averaging difference operator can be used in place of $(u_{i+1} + u_{i-1})/2$. If this procedure is repeated hierarchically, i.e., defining

$$d_i^2 = u_i - (u_{i+2} + u_{i-2})/2, \quad d_i^3 = u_i - (u_{i+4} + u_{i-4})/2, \ldots$$



(a) $\alpha_1 < \frac{1}{2}$ (b) $\alpha_2 < \frac{1}{2}$ (c) $\alpha_1 < \frac{1}{2}$ and $\alpha_2 < \frac{1}{2}$

Figure 3: Schematic of grid/grid interpolation in two space dimensions. The larger (blue) circles denote the points to where the interpolation points are used. The smaller (red) circles denote the domain of dependence of the interpolation stencil.

the functions $d_i, d_i^2, d_i^3, ...$ can be interpreted as wavelet coefficients. Different average operators in (17) correspond to different wavelet bases. It can be shown that the local regularity of u_i, measured as the Liptschitz exponent, α, can be estimated from the decrease of d_i^m with m.
The procedure of approximating α_i is described in detail in [12], where it is used to add artificial dissipation when the solution is non-smooth i.e. has an $\alpha_i < \alpha_0$ ($\alpha_0 \in [0.4, 0.6]$ is reported to work well in [12]). We here use this procedure dimension by dimension to switch to a lower order interpolation stencil in the i-direction when the Lipschitz exponent is less than 0.5. In Fig. 3 we show how the interpolation stencil is modified if any of the points in the domain of dependence has $\alpha_i < \frac{1}{2}$

2.5 Time Discretization

The time integration is done with a second order accurate Runge-Kutta method. Given a Courant number, the time step Δt is taken as the minimum of the viscous and convective part of the operator over all grids and then the solution is advanced forward in time, with the same time step on each grid. The coarsest grid will use a smaller time step than necessary for stability, but we avoid interpolating the inner-grid boundary conditions between different times, see, e.g., [13]. Furthermore, the cost of advancing the solution of the coarser grids is usually only a small fraction of the cost of advancing on the finer grids.

3 Embedded Boundary

Embedded objects cut holes in the Cartesian grid. When the computational stencil is applied at all points outside the embedded objects, a few points inside the objects will be needed. These are the so called ghost points. Boundary conditions at the object boundaries define the values of the solution at the ghost points. We define the interior grid points as the points where the difference approximation is applied. The covered points are the points covered by the embedded object which are not ghost points. The covered points are not used in the computation.

We will here describe two boundary procedures, the method by Kreiss and Petersson from [5, 7], which we from now on will call the KP embedded boundary method, and the embedded boundary method from [11] which we will call the SP (Sjögreen/Petersson) embedded boundary method. The KP embedded boundary method is a very accurate boundary procedure, developed for wave propagation problems. The SP embedded boundary method is a robust, but less accurate, boundary procedure for problems with discontinuous solution. These methods are used to impose the Dirichlet condition

$$u|_B = g_D \tag{18}$$

or the Neumann condition

$$\left.\frac{\partial u}{\partial n}\right|_N = g_N \tag{19}$$

at the embedded boundary. Here g_D and g_N are given data.

The boundary conditions (6) consists of only Neumann and Dirichlet conditions, plus the extrapolation condition (14). In [5, 7, 11], the KP method is defined for Neumann and Dirichlet boundary conditions, and the SP method is defined for Dirichlet and extrapolation boundary conditions. To be able to apply both methods to the Navier-Stokes equations, we therefore need to introduce an extrapolation condition for the KP method and a Neumann condition for the SP method.

3.1 The KP embedded boundary method

In Fig. 4 we outline the embedded boundary procedure from [5]. A normal to the boundary through the ghost point is defined, and function values in the interior of the domain on the normal, ($u_I,\ u_{II}$ in Fig. 4) are obtained by quadratic interpolation in the i-direction. If the normal has slope less than one, the interpolation is instead done in the j-direction. Finally the condition (18) or (19) is approximated by a formula involving the the three values u_I, u_{II}, and the ghost point value $u_{i,j}$. From this formula we obtain from (18)

$$u_{i,j} = c_1 u_I + c_2 u_{II} + c_3 g_D = \sum c^D_{k,l} u_{i+k,j+l} + c_3 g_D \tag{20}$$

where the coefficients depend on the distances Δ and ξ_Γ defined in Fig. 4. The sum extends over the points in the interpolation stencil. For precise formulas for $c_1,\ c_2$, and

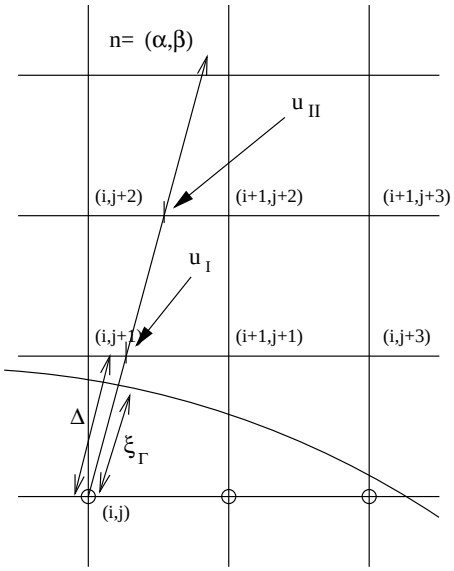


Figure 4: KP embedded boundary method. The indices denote the domain of dependence for the embedded boundary procedure for ghost point (i, j).

c_3,, see [5]. Similarly (19) gives an update formula

$$u_{i,j} = c_1' u_I + c_2' u_{II} + c_3' g_N = \sum c_{k,l}^N u_{i+k,j+l} + c_3' g_N \tag{21}$$

where the coefficients are given in [7].

We impose the numerical condition (14) at the embedded boundary by second order extrapolation along the normal,

$$p_{i,j} = 2p_I - p_{II}, \tag{22}$$

where we decrease the order of extrapolation by one to

$$p_{i,j} = p_I,$$

if (22) gives negative pressure.

3.2 The SP embedded boundary method

When discontinuities are present in the solution, special care has to be taken to make the boundary interpolation robust. We outline the SP method, which is suited for dealing with shock waves, in Fig. 5, where we impose the Dirichlet boundary condition (18). The SP method uses more values along the normal than the KP method, but the horizontal interpolation is linear, i.e.,

$$u_I = \eta_1 u_{i,j+1} + (1 - \eta_1) u_{i+1,j+1}, \tag{23}$$

$$u_{II} = \eta_2 u_{i,j+2} + (1-\eta_2) u_{i+1,j+2}, \tag{24}$$

$$u_{III} = \eta_3 u_{i,j+3} + (1-\eta_3) u_{i+1,j+3}, \tag{25}$$

where $\eta_i \in [0,1]$, $i = 1,2,3$, depend on the location where the normal intersects the horizontal grid lines. When the normal has positive y-component and the angle between the normal and the x-axis is between $\frac{\pi}{4}$ and $\frac{\pi}{2}$, the normal will always intersect the grid line $y = y_{j+1}$ between x_i and x_{i+1}. There are two different cases when the normal intersects the $y = y_{j+2}$ grid line (between x_i and x_{i+1} or between x_{i+1} and x_{i+2}) and similarly three different cases where the normal intersects the $y = y_{j+3}$ grid line. Similarly to the KP method, the interpolation (23)–(25) are done in j-direction if the normal has slope less than one. Denote the distance between the boundary and the ghost point by ξ_Γ and let the distance between the ghost point and grid line $y = y_{j+1}$ along the normal be Δ (see Fig 5). Define new points u_{b_1} and u_{b_2} placed equidistantly along the normal by linear interpolation along the normal at distances $\xi_\Gamma + \Delta$ and $\xi_\Gamma + 2\Delta$ from the ghost point respectively,

$$u_{b_1} = \frac{\xi_\Gamma}{\Delta} u_{II} + \left(1 - \frac{\xi_\Gamma}{\Delta}\right) u_I, \quad u_{b_2} = \frac{\xi_\Gamma}{\Delta} u_{III} + \left(1 - \frac{\xi_\Gamma}{\Delta}\right) u_{II}.$$

A limited boundary slope is defined,

$$s_D := S_{minmod}\left(u_{b_1} - g_D, u_{b_2} - u_{b_1}\right),$$

where

$$S_{minmod}(x,y) = \begin{cases} x, & \text{if } |x| < |y| \text{ and } xy > 0, \\ y, & \text{if } |y| < |x| \text{ and } xy > 0, \\ 0, & \text{otherwise} \end{cases} \tag{26}$$

is the well-known min-mod limiter. The boundary condition (18) is approximated by extrapolation using the limited boundary slope,

$$u_{i,j} = g_D - \frac{\xi_\Gamma}{\Delta} s_D. \tag{27}$$

The above construction is always well-defined, since $h \le \Delta \le \sqrt{2}h$.

In the SP method, we use

$$p_{i,j} = p_I - S_{minmod}\left(p_{III} - p_{II}, p_{II} - p_I\right). \tag{28}$$

to define the pressure at the ghost points. (28) is similar to (14) except for the limiter which improves robustness. If the solution is smooth,

$$S_{minmod}\left(p_{III} - p_{II}, p_{II} - p_I\right) \approx p_{II} - p_I$$

and we recover (14).

We use the second order accurate

$$T_{i,j} = \left(\frac{4}{3} - \frac{\xi}{3\Delta}\right) T_{b_1} - \left(\frac{1}{3} - \frac{\xi}{3\Delta}\right) T_{b_2} - \frac{2\Delta(2\xi+1)}{3} g_N \tag{29}$$

to impose the homogeneous Neumann condition for the temperature ($g_N = 0$).

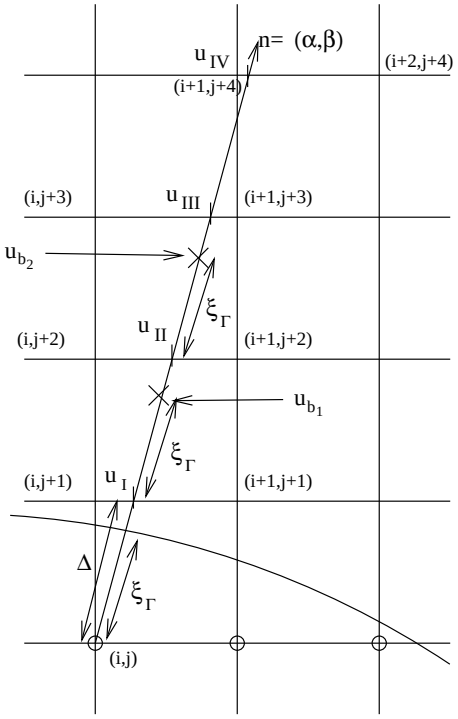


Figure 5: SP embedded boundary method. The indices denote the domain of dependence for the embedded boundary procedure for ghost point (i, j).

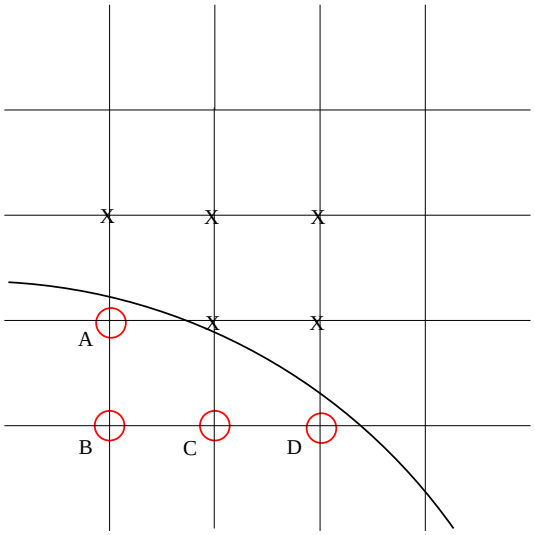


Figure 6: The ghost point A and C are needed when computing $\partial^2/\partial x^2$ and $\partial^2/\partial y^2$ respectively and are updated as described previously. B and D are needed when computing $\partial^2/\partial x \partial y$. The value of D is assigned with the standard algorithm, but since the value of B depends on the value of A or C we make a second update after A and C are found, to update all the ghost points. In practice A and C do not couple back to B. B is a coupled point.

3.3 Navier-Stokes boundary condition

We summarize the KP and SP methods at the embedded boundary as follows. KP method consists of using (20) for the zero velocity boundary condition, using (21) for the zero heat flux boundary condition, and using (14) for the pressure extrapolation. The SP method consists of using (27) for the zero velocity condition, using (29) for the zero heat flux boundary condition, and using (28) for the pressure extrapolation.

3.4 Three space dimensions

The interpolation for updating the ghost points is done analogously in three space dimensions, with the exception that the normal intersects planes instead of lines. The interpolation formulas for the ghost points, (27), (28), and the first equalities of (20) and (21) are still valid in three dimensions, but the formulas for computing the intermediate values, u_I, u_{II}, u_{III}, now involve interpolation in planes.

3.5 Coupled ghost points

One difficulty with the KP and SP methods is that the ghost points sometimes couple, i.e., if the update formula for one ghost point depends on the value at another ghost

point. For the wave equation, it was observed in [7] that when the embedded boundary is smooth and resolved, ghost point values only depend on values inside the domain, and thus do not couple. However, for the Navier-Stokes equations the stencil is larger due to mixed derivatives, e.g., u_{xy}. The mixed derivatives increase the likelihood of coupled ghost points. An example of this is shown in Fig. 6, where the point B is coupled because its update formula according to (20), (21) or (27) depends on point A. However, since A does not depend on any ghost point, it is easy to decouple by first evaluating A and then B.

The embedded boundary procedure and ghost points form a dependency graph with a coupling matrix with a non-zero element if a ghost point depends on another ghost point. We wish to make this matrix lower triangular, if possible. This is achieved with the topological sort algorithm [4], which will terminate if there is a loop in the dependency graph, i.e. it is not possible to make the coupling matrix lower triangular. In this case one would have to solve a linear system of equations for the ghost point values, but in practice, when the embedded boundaries are resolved, the topological sort is successful. Problems are likely to occur when geometries have cusps, sharp inside corners or are unresolved with two boundaries close to each other.

4 Numerical Experiments

In this section we verify the formal order of accuracy of the implementation of the embedded boundary method for the viscous operator and we study the convergence properties of the embedded boundary conditions for a supersonic external flow problem.

4.1 Accuracy

We here verify the second order of accuracy of the approximation of the viscous operator. The nonlinear MUSCL method, which is used for the inviscid fluxes, will switch its order of accuracy between one and two depending on the solution. We therefore solve the PDE with only the viscous terms,

$$\frac{\partial \mathbf{w}}{\partial t} = ViscOp(\mathbf{w}) + f, \tag{30}$$

where $\mathbf{w} = (\rho,\ \rho u_1,\ \rho u_2,\ \rho u_3,\ e)$, and the viscous operator is,

$$ViscOp(\mathbf{w}) = \text{div} \begin{pmatrix} 0 \\ \frac{\alpha(T)}{Re}\left(2S_{ij} - \frac{2}{3}\delta_{ij}S_{kk}\right) \\ \frac{\alpha(T)\gamma}{RePr}(\gamma-1)\frac{\partial}{\partial x_j}\left(\frac{p}{\rho}\right) + \frac{\alpha(T)}{Re}\left(2S_{ij} - \frac{2}{3}\delta_{ij}S_{kk}\right)u_i \end{pmatrix}.$$

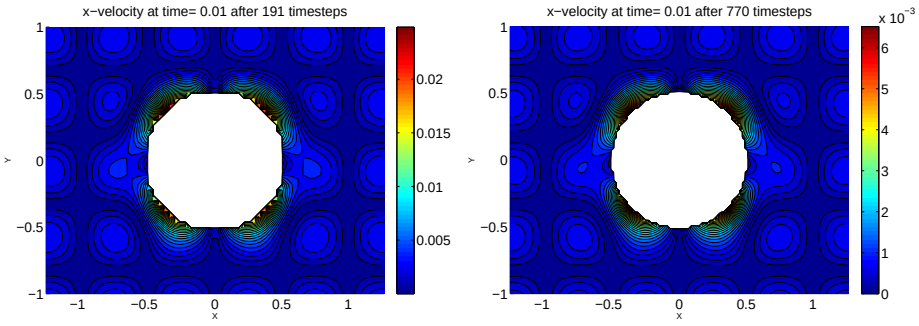
$f = f(x, y, z, t)$ is a known forcing function that has been determined such that the solution of (30) is

$$\begin{aligned}
\rho &= 1 + \frac{\sin(\omega_1 y)\cos(\omega_2 x)\sin(\omega_3 z)\sin(\omega_4 t + \frac{\pi}{2})}{2} \\
u_1 &= \sin(\omega_5 x)\cos(\omega_6 y)\sin(\omega_7 z)\sin(\omega_8 t + \frac{\pi}{2}) \\
u_2 &= \cos(\omega_9 x)\cos(\omega_{10} y)\cos(\omega_{11} t) \\
u_3 &= \sin(\omega_{12} x)\sin(\omega_{13} y)\sin(\omega_{14} z)\cos(\omega_{15} t) \\
T &= x^2 + 1 + \sin(\omega_{16} xyzt + \frac{\pi}{2}),
\end{aligned} \tag{31}$$

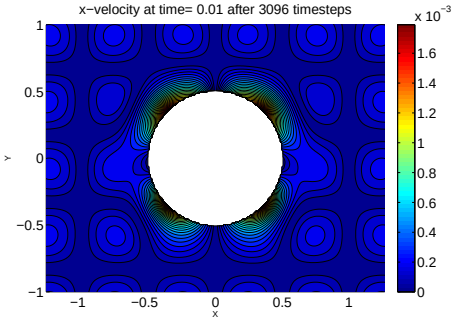
where the total energy is obtained from the temperature by (4) and (5) with constants $Re = 1.0,\ Pr = 0.72,\ \gamma = 1.4$ and $\alpha = 1.0$. The forcing f is calculated as

$$f = \frac{\partial \tilde{\mathbf{w}}}{\partial t} - ViscOp(\tilde{\mathbf{w}})$$

where $\tilde{\mathbf{w}}$ is the function given by (31). In this way we know that the exact solution of (30) is $\tilde{\mathbf{w}}$ and we can therefore measure the absolute error of the numerically computed solution. The test geometry is a cylinder with radius 0.5 centered at the origin. The boundary normal of a cylinder in two space dimensions ranges over all possible angles of interpolation at the embedded boundary. The different ω_i:s can be chosen arbitarily. We have chosen $\omega_4 = \frac{\pi}{2}$, $\omega_8 = \omega_{11} = \omega_{16} = 0$, and $\omega_i = 2\pi$ for $i \neq 4, 8, 11, 16$. The computational domain is $[-2, 2] \times [-2, 2]$ and Dirichlet boundary conditions are used on all four sides of the domain. At the embedded boundary we use the Dirichlet condition (20) for the velocities. We run the simulation until time 0.01, which corresponds to (using Courant number 0.5) 191, 770, and 3096 time steps for the three discretizations respectively. The results are compared component-wise at time 0.01 over all interior grid points. We display the error of the energy component of $\mathbf{w}$ in Tables 1 and 2. Results for the momentum and density components are similar, but not shown here. e_N denotes the approximation of the total energy when the grid has N^2 points, and e is the exact energy. In Tables 1 and 2 we infer that the approximations are second order accurate, because the error decreases with approximately a factor four when the mesh spacing is divided by two. Fig. 7, which displays the pointwise error, clearly shows that the approximation error is largest at the embedded boundary. This is due to the interpolation errors in the embedded boundary conditions. In general, interpolation at a boundary is equivalent to approximating by one-sided operators near the boundary, while the interior scheme is centered and therefore have smaller truncation error. The truncation error oscillates wildly between consecutive grid points along the embedded boundary, but because the problem we solve is parabolic, the non-smooth part of the error decays exponentially away from the embedded boundary.



(a) $N = 100$ (b) $N = 200$



(c) $N = 400$

Figure 7: The pointwise error is largest when the boundary intersects the normal close to an inner point. Note the different scales and the order of convergence. The error on the embedded boundary is of the same order of magnitude as the error in the interior of the domain.

N	$\|e - e_N\|_{L_\infty}$	Quotient
100	1.610611286568719e-02	-
200	4.227500992437072e-03	3.809842480108403e+00
200	4.227500992437072e-03	-
400	1.076486438807756e-03	3.927128888979937e+00

Table 1: The relative L_∞ error in the total energy.

N	$\|e - e_N\|_{L_2}$	Quotient
100	7.994539555804822e-03	-
200	1.983426735739193e-03	4.052418953180670e+00
200	1.983426735739193e-03	-
400	5.008640615789963e-04	4.017933288218640e+00

Table 2: The relative L_2 error in the total energy.

4.2 Mach 3 flow around a cylinder

The challenge with computing approximate solutions to the Navier-Stokes equations using embedded boundaries is the resolution of boundary layers. With body-fitted grids the grid is usually stretched to have very small grid spacing in the direction normal to the wall. This is done because in an attached boundary layer, the need for tangential resolution is much lower than the required resolution in the direction normal to the wall. Typically the ratio of wall normal vs. tangential grid size is one to ten for time accurate simulations, but for steady state calculations it can be of the order of one to hundreds. However, at a point where the boundary layer separates, high resolution in the tangential direction is necessary. We will here investigate these claims by numerical experiments.

The purpose of this study is to evaluate the resolution of boundary layers, and how well the embedded boundary method can predict quantities on the embedded boundary. In particular we will study how the the skin-friction coefficient and temperature on the surface of the cylinder converge as the grid is refined. This tests the Dirichlet boundary condition for the velocity and the Neumann boundary condition for the temperature respectively at the embedded boundary.

We here compute supersonic flow around a cylinder with radius 0.5 with Mach number 3 and Reynolds numbers 500 in the two dimensional domain $(x, y) \in [-10, 10] \times [-5, 5]$. The center of the cylinder is located at $(-1, 0)$. These simulations are time accurate, and resolved in time and space. As initial data, we impose free stream conditions in the entire domain. The discretization on the Cartesian grid is efficient because it has a simpler memory access pattern than an unstructured method and requires less metric information (and thereby less memory accesses and less arithmetic operations) than an approximation on a curvilinear grid. In fact the grid is never used in the computation.

Figure 8 shows velocity magnitude $\sqrt{u_1^2 + u_2^2}$ contours. The computations was run

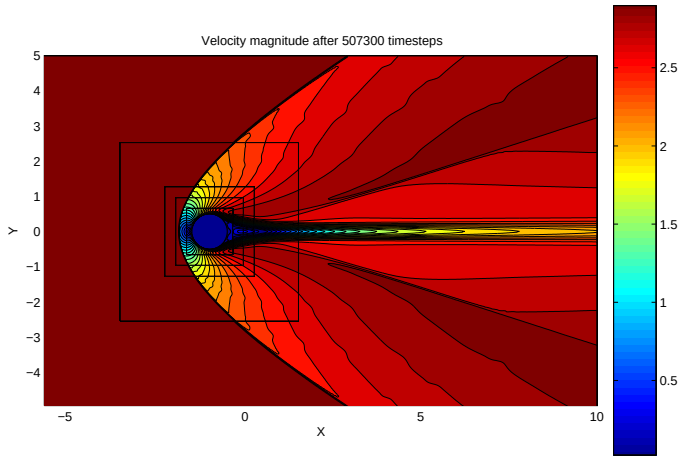


Figure 8: Two dimensional computations of Mach 3 flow past a cylinder. Velocity magnitude contours.

until steady-state. In the computations the timestep had to be restricted by the stability requirement of the viscous operator. We take this as an indication that the flow is resolved.

The solution displayed in Fig. 8 were computed over a base grid with 736×368 points and with four refinement patches, with 373×373, 373×373, 563×563, and 771×771 grid points. The spacing of each refinement patch is a factor 2 finer than the previous. The boundaries of the refinements are outlined in Fig. 8.

4.2.1 Comparison with body fitted computations

We will compare the results obtained by the embedded boundary method with results obtained by solving the same problem using a body fitted grid. For the body fitted grid computations below, we used the solver for the compressible MHD equations, developed in [15], with magnetic fields set to zero.

The domain is discretized by the overset grid configuration displayed in Fig. 9 for the cylindrical flow problem. There are four grids, a base grid that covers the entire domain, a curved grid around the bow shock, a fine polar grid near the cylinder surface, and a fine grid that covers the wake region. We used the overset grid generator Xcog [9] to generate the grids and the interpolation information.

We discretized the Navier-Stokes equations by a sixth order accurate finite difference scheme with summation-by-parts boundary modification of the difference operators on all component grids except the bow shock grid, where we used a TVD type difference scheme. The solution was time marched to steady state, first with a TVD scheme on all grids, and later when the solution is fully developed, with the sixth order method on three of the grids, as described above.

Interesting questions about overall accuracy and error propagation from the bow shock

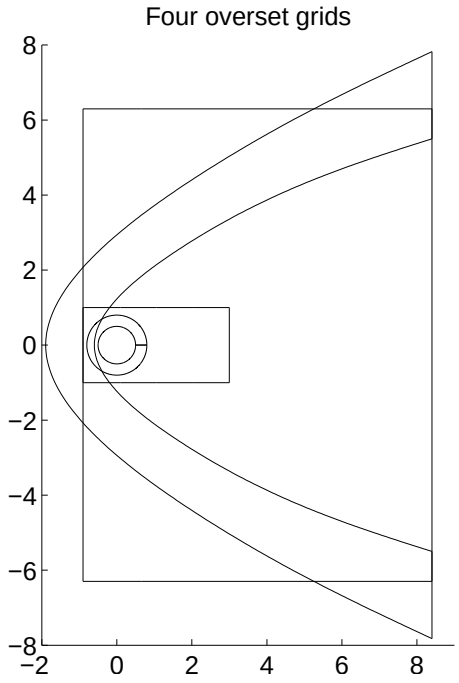


Figure 9: Overset grid domains used for computations with body fitted grids.

are outside the scope of this work. However, it was observed in [15] that the actual grid convergence rate at the body boundary is close to 2nd order.

Fig. 10 shows Mach number contours of a solution with Mach number 3 and Reynolds number 500. The grid boundaries are outlined in color (or grey). The overall flow features are very similar to the embedded boundary solution of the same problem, shown in Fig. 8.

4.2.2 Grid convergence of the skin friction coefficient

Viscous drag, or skin friction, is caused by the fact that we consider here a viscous gas and a no-slip condition is imposed on the velocity. We investigate how accurate and/or converged the normal derivative of the velocity is for different resolutions.

The skin friction coefficient in scaled units is

$$C_f = \frac{\alpha(T)}{Re} \frac{1}{\frac{1}{2}\rho_\infty U_\infty^2} \frac{\partial V}{\partial n}, \tag{32}$$

where V denotes the tangential velocity on the boundary. In two space dimensions we have $V = \vec{u} \cdot \vec{t}$, with $\vec{t}$ the vector tangential to the boundary. $\partial V/\partial n$ at the boundary is computed by (21), where the Neumann data, $\partial V/\partial n$, is the unknown and the ghost point value known. ρ_∞ and U_∞ are the density and velocity magnitude respectively in the free stream state.

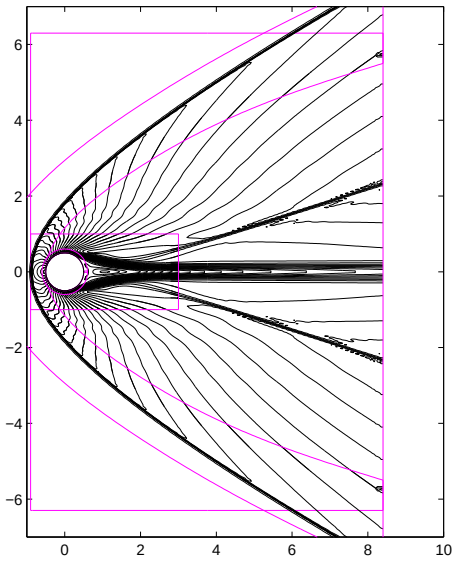


Figure 10: Solution on overset grids with Mach number 3 and Reynolds number 500. Mach number contours.

The results below display C_f as a function of the x-coordinate along the width of the cylinder. The C_f curve rises from the forward stagnation point to a high value at the front side of the cylinder. On the rear part of the cylinder, the friction decreases, and some recirculation (negative C_f) is seen where the boundary layer separates to form the wake.

Two different numerical boundary procedures at the embedded boundary are compared, the KP method as described in Section 3.1 and the SP method as described in Section 3.2. We also compare with results from the body fitted computation described in Subsection 4.2.1.

In Figs. 11–13 we show the grid convergence of C_f for the KP, SP, and body fitted methods. The embedded boundary methods use the second order boundary condition (13) at the embedded boundary. Fig. 11 shows C_f computed by the KP method, on three different sets of grids, each with four refinement grids. The refinement grid closest to the cylinder has grid size $h = 0.0068306$, 0.003406, and 0.0017007 for the three different curves respectively. Fig. 12 show the same computation, but using the SP method. Fig. 13 shows the grid convergence of C_f for the body fitted computation, where the three grid sizes are given in Table 3. The cylinder grid and wake grid have cell aspect ratios approximately equal to one. The spacing in the radial direction on the cylinder, and the uniform spacing in the wake, for the three different overlapping grids are equal to the spacing of the finest refinement grid in the three different set of grids used in the embedded computations. Thus, in Figs. 11–13 the embedded and body fitted computations are made with the same h near the body. For the finest computation this h corresponds to approximately

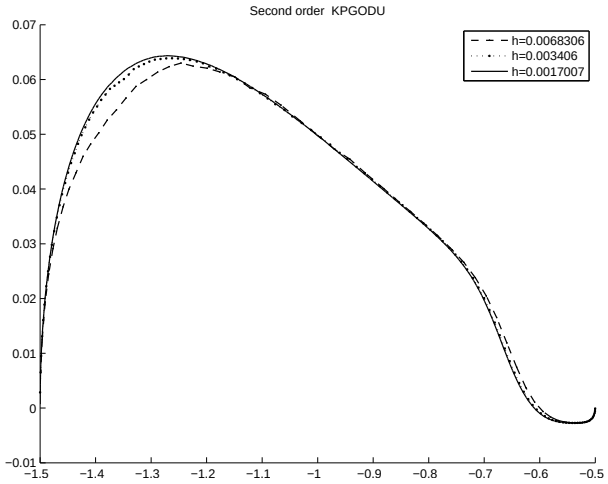


Figure 11: C_f along the upper half of the cylinder computed with the KP embedded boundary method for Mach number 3 and Reynolds number 500.

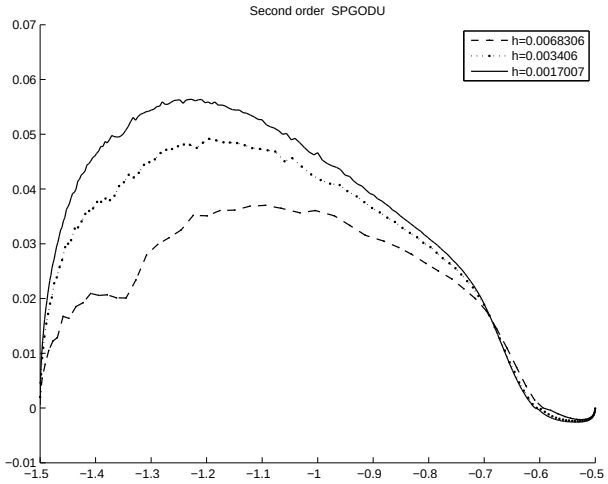


Figure 12: C_f along the upper half of the cylinder computed with the SP embedded boundary method for Mach number 3 and Reynolds number 500.

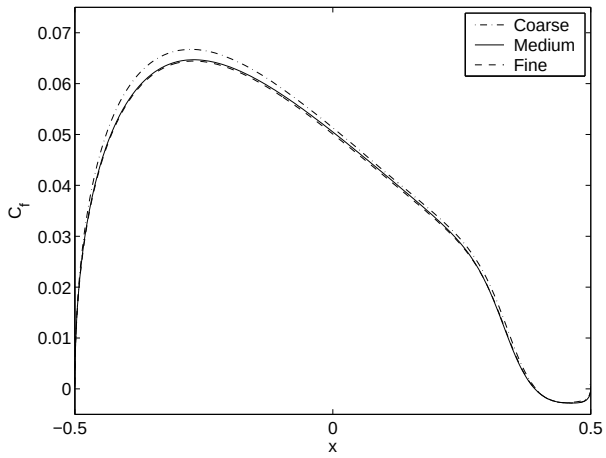


Figure 13: C_f along the upper half of the cylinder computed with the body fitted method for Mach number 3 and Reynolds number 500.

	Coarse	Medium	Fine
base grid	100x136	200x272	200x272
wake grid	400x200	800x400	1200x600
shock grid	160x35	320x70	320x70
body grid	352x15	697x30	1387x60

Table 3: Number of grid points in composite grids for the three computations. The body grid and the wake grid have cell aspect ratios close to one.

26 grid points over the width $1/\sqrt{Re}$.

In Fig. 14 we have collected the C_f curves from the finest grids in Figs. 11–13. The body fitted method and the KP embedded method give results that are indistinguishable in the plot. We conclude that the KP embedded boundary approach gives more accurate results than the SP embedded boundary method, and furthermore that the accuracy of the KP embedded boundary method is comparable to the accuracy of the body fitted method.

It is not unexpected that the KP method is more accurate than the SP method, because the SP method switches between a first and second order accurate boundary condition, whereas the KP method is always of high formal accuracy. The SP method uses limiters to handle shock waves, but the KP method uses centered interpolation stencils. Nevertheless, the KP method gave solutions that were free from unphysical oscillations, since a resolved boundary layer does not contain discontinuities.

Fig. 15 gives another indication that the formal order of accuracy is very important. Fig. 15 shows results from the same computation as in Fig. 14, but with the first order boundary extrapolation (12) used at the embedded boundary instead of the second order

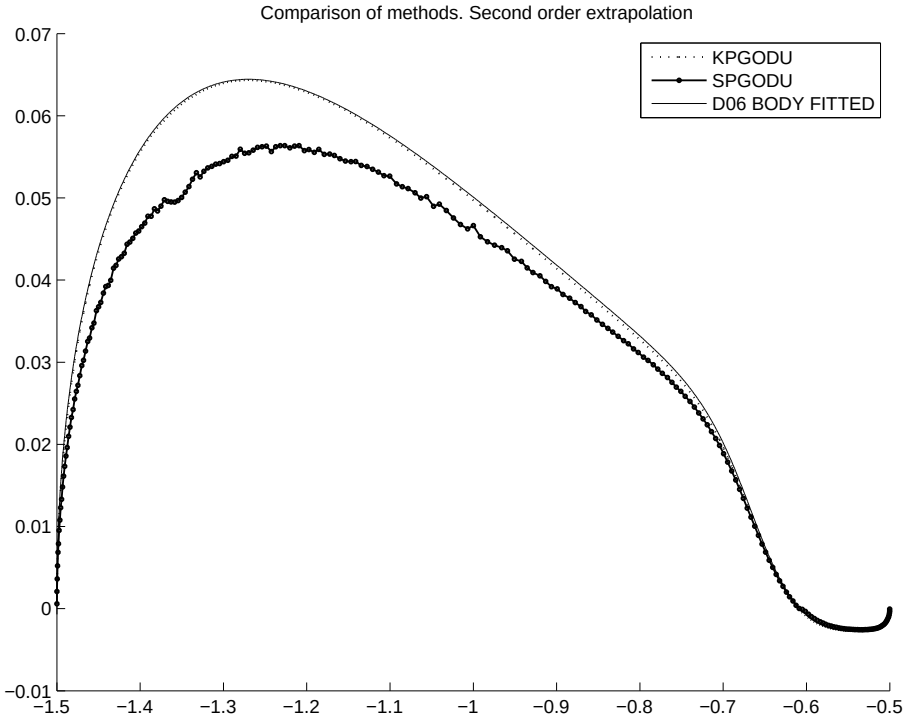


Figure 14: C_f along the cylinder surface with the KP embedded boundary method, the SP embedded boundary method, and the body fitted method. Mach number 3, Reynolds number 500. The finest grid size is $h = 0.0017007$. Second order extrapolation (13).

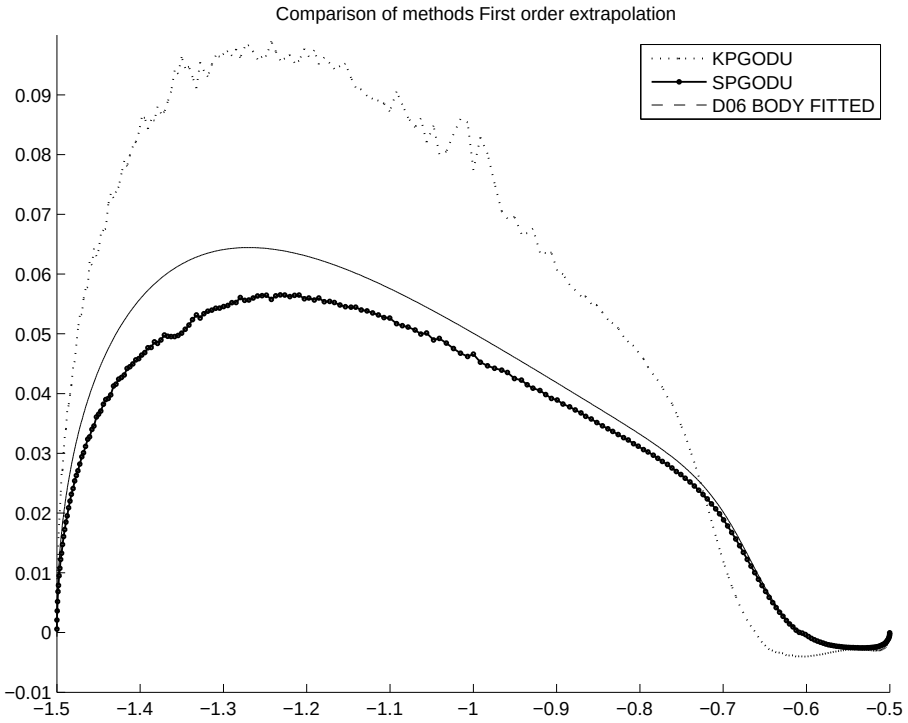


Figure 15: C_f along the cylinder surface with the KP embedded boundary method, the SP embedded boundary method, and the body fitted method. Mach number 3, Reynolds number 500. The finest grid size is $h = 0.0017007$. First order extrapolation (12).

(13). The body fitted C_f curve is the same in both figures.

We conclude that when the physical viscosity is not resolved, which is the case for simulations using the coarse mesh then the skin friction cannot be expected to be accurate raising the need for local grid refinement.

4.2.3 Grid stretching with the body fitted method

The number of points on the cylinder ranges from 352 on the coarse grid to 1387 on the fine grid. Because the boundary layer is attached, the grid size along the cylinder can be coarser than the grid size normal to the body. We coarsen the composite grids Coarse and Medium in the direction tangential to the cylinder, and obtain the grids Coarse-S and Medium-S. These grids have cell aspect ratios approximately 1:10 along the cylinder, the exact number of grid points are given in Table 4. We had to extend the cylinder grid in the Coarse-S composite grid in the radial direction to make the interpolation between the grids well defined, but the grid spacing of the Coarse-S grid is still half of the Medium-S grid. Fig. 16 displays a comparison between C_f for flows computed on the stretched and non-stretched grids. Fig. 16a compares results on the grids Coarse and Coarse-S and

	Coarse-S	Medium-S
base grid	100x136	200x272
wake grid	200x100	400x200
shock grid	160x35	320x70
body grid	35x30	70x30

Table 4: Number of grid points in composite grids forbody grid stretched to aspect ratio 1:10 along the wall. Note: the Coarse-S body grid has thickness 0.2 in the radial direction.

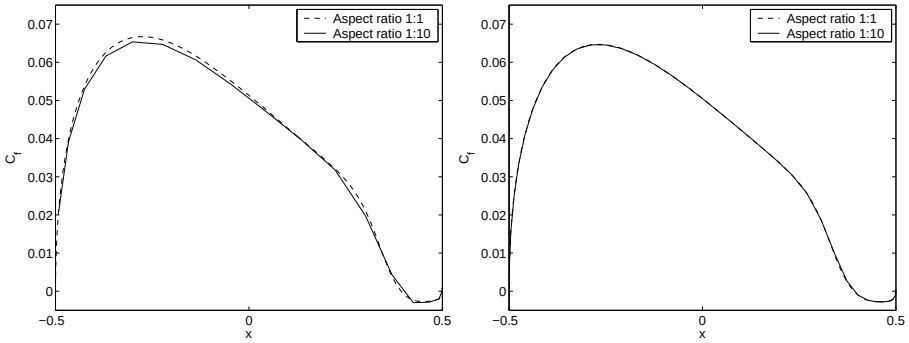


Figure 16: C_f along the upper half of the cylinder for Mach=3 and Re=500. Comparison high aspect ratio vs unit aspect ratio. a) to the left shows Coarse (dashed) and Coarse-S (solid) and b) to the right shows Medium (dashed) vs Medium-S (solid).

Fig. 16b compares Medium vs Medium-S. Some difference at the leading edge and at the separation point is visible on the coarse grids, but on the two medium size grids, the two curves are almost indistinguishable.

The ability to coarsen the grid in one direction is clearly absent in the embedded boundary method. We conclude that for attached laminar boundary layers, this feature makes the body fitted approximation considerably more efficient. However, when resolution is equal in both directions, Fig. 14 shows that the embedded boundary method gives results of similar quality as with the body fitted method. Equal resolution in all direction is needed in direct simulation of turbulent separating flows. Furthermore, with complicated geometries it might not be known a priori at which locations the boundary layer is attached and therefore it would not be possible take advantage of body fitted stretched grids.

4.2.4 The temperature on the boundary

The adiabatic wall condition imposes $\frac{\partial T}{\partial n} = 0$. We evaluate the accuracy of the Neumann boundary condition by plotting the temperature on the surface. Fig. 17 shows the wall temperature obtained with the KP method and 18 displays the wall temperature obtained with the SP method. Similarly to the C_f plots, the KP method appears to be more

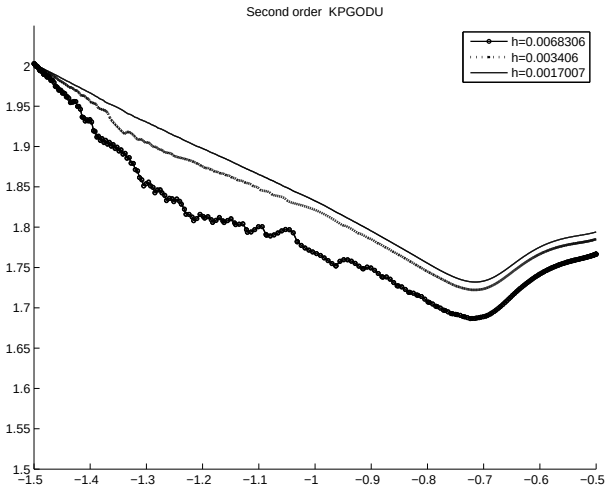


Figure 17: Temperature on the boundary using the KP embedded boundary method. Mach number 3, Reynolds number 500.

accurate than the SP method.

5 Conclusions

We have shown that solutions to the the compressible Navier-Stokes equations in complex geometries can be approximated with the embedded boundary method. This is done maintaining overall second order accuracy using special interpolation procedures to fulfill boundary conditions on geometries inside the computational domain. Furthermore, we have defined a local mesh refinement scheme to enable a finer grid near the embedded objects.

We have compared two different embedded boundary methods and shown that a centered interpolation stencil works well in the boundary layers, even if the flow field contains discontinuities. Furthermore, we have shown that results obtained with a body fitted method are comparable with the results obtained by embedded boundary methods as long as the cell aspect ratios are close to unity. We also found that, for embedded boundary methods to be accurate, it is very important that the formal order of accuracy is maintained at the embedded boundary.

Embedded boundary methods become more advantageous with respect to body fitted methods when the geometry becomes more complicated, espcially in three space dimensions. We have recently extended the embedded boundary method described here to three space dimensions, and used it to do direct numerical simulation of turbulence (DNS). DNS is a suitable application for embedded boundaries, because in such computations it is important to resolve the flow in all directions. Results will be presented in another report.

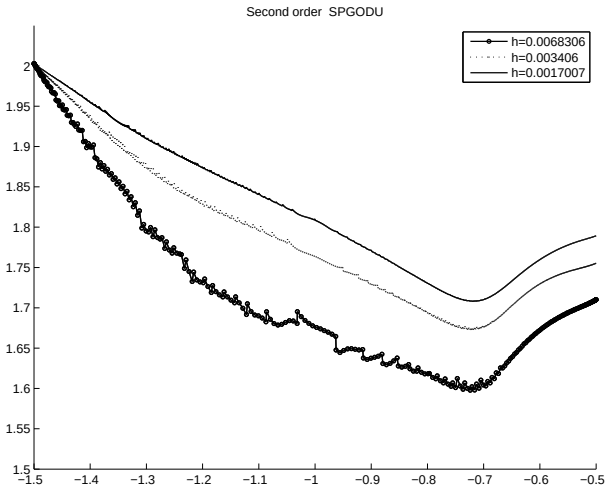


Figure 18: Temperature on the boundary using the SP embedded boundary method. Mach number 3, Reynolds number 500.

The Cartesian grids used for embedded boundary methods are well suited for high order accurate difference methods. Another possible generalization of the method described here would be to define the boundary interpolation to high order accuracy.

References

[1] P. Colella, D. T. Graves, B. J. Keen, and D. Modiano. A Cartesian grid embedded boundary method for hyperbolic conservation laws. *J. Comput. Phys.*, 211:347–366, 2006.

[2] L. Ferm and P. Lötstedt. Accurate and stable grid interfaces for finite volume methods. *Appl. Numer. Math.*, 49:407–224, 2004.

[3] H. Forrer and R. Jeltsch. A higher-order boundary treatment for Cartesian-grid methods. *J. Comput. Phys.*, 140:259–277, 1998.

[4] D. E. Knuth. *The art of Computer Programming*, volume 3, Sorting and Searching. Reading, Massachusetts: Addison-Wesley, second edition edition, 1998.

[5] H.-O. Kreiss and N. A. Peterson. A second order accurate embedded boundary method for the wave equation with Dirichlet data. *SIAM J. Sci. Comput.*, 27:1141–1167, 2006.

[6] H.-O. Kreiss, N. A. Peterson, and J. Yström. Difference approximations for the second order wave equation. *SIAM J. Numer. Anal.*, 40(5):1940–1967, 2002.

[7] H.-O. Kreiss, N. A. Peterson, and J. Yström. Difference approximations of the Neumann problem for the second order wave equation. *SIAM J. Numer. Anal.*, 42:1292–1323, 2004.

[8] O. K. Olsen. Embedded boundary method for Navier-Stokes equations. Master's thesis, Royal Institute of Technology, 2005.

[9] A. Petersson. http://www.andrew.cmu.edu/user/sowen/software/xcog.html. Internet.

[10] J. J. Quirk. An alternative to unstructured grids for computing gas dynamic flows around arbitrarily complex two-dimensional bodies. *Computers Fluids*, 23(1):125–142, 1994.

[11] B. Sjögreen and N. A. Peterson. A Cartesian embedded boundary method for hyperbolic conservation laws. *Commun. Comput. Phys.*, 2:1199–1219, 2007.

[12] B. Sjögreen and H. C. Yee. Multiresolution wavelet based adaptive numerical dissipation control for high order methods. *J. Sci. Comput.*, 20(2):211–255, 2004.

[13] P. Skogqvist. *High Order Adaptive Difference Methods for Combustible Flows*. PhD thesis, Royal Institute of Technology, Stockholm, Sweden, 2001.

[14] G. van Albada, B. van Leer, and J. W.W. Roberts. A comparative study of computational methods in cosmic gas dynamics. *Astron. Astrophys.*, 108:76–84, 1982.

[15] H. Yee and B. Sjögreen. Development of low dissipative high order filter schemes for multiscale Navier-Stokes/MHD systems. *J. Comput. Phys.*, 225(1):910–934, 2007.