Reply to referee's comments

Reviewers' comments are in plain and the author's reply is in italic text.

General response:

We would like to thank the two Reviewers for their in depth perspicacious comments that contributed to improving the presentation of our paper.

In summary, to address the comments of the reviewers, the following work has been carried out:

1. *The first two cases have been redone to show more convincing convergence as the mesh is refined. For the 1st and 2nd cases, the time step is reduced by a factor to ensure a small Courant number with the smaller elements sizes. We have thus re-plotted the results in Figs 2~9.*

2. *The 3rd case has been redone with a negative concentration background of -0.2 in the subdomain [0.24,0.76]×[0.12,0.88] as suggested by reviewer. The maximum number of nodes for adaptive schemes is set to be 15000. Table 1 and Figs 10~15 have been updated to reflect these new results. A new Fig 16 has been added to show the distribution of CFL number over the domain.*

3. *A new case, case 4 based on a real – large scale atmospheric geometry – and flow, has been added to demonstrate the capability of this new adaptive multiscale model. Figs 17~20 show the results obtained from this new case.*

4. *Case 4 is the simulation of the dispersion of power plant plumes. Diffusion and source terms have therefore been introduced into the equations in section 2.*

5. *Section 3 has been revised and more details of the adaptive mesh techniques have been added.*

6. *Section 2.2 has been rewritten and details of numerical schemes have been provided.*

<u>Anonymous Referee #1</u>

Summary: the authors present details of an adaptive grid-resolution approach to solving the classic tracer advection problem whereby locally movable higher resolution grids are employed in areas of tracer distribution where steep gradients and small features are better simulated with finer resolution. Overall the paper is sound and presents promising results. I suggest a couple minor comments related to the tests presented, and all the tests should be repeated using a nonzero (preferably a negative)

background rather than zero background, since there is nothing special about zero, but many algorithms assign inappropriate significance to zero. Positive-definite schemes are not necessarily useful with tracers with large backgrounds are present, and the value of this scheme increases if it can be shown to advect negative tracers. Please redo the 3rd test with a negative background.

RESPONSE:
Thanks for your comments. As suggested, the 3rd test has been redone with a negative concentration background of -0.2 in the subdomain [0.24,0.76]×[0.12,0.88]. The maximum number of nodes for the adaptive schemes is set to be 15000. Figs 10~15 in the paper have been replotted. A new Fig. 16 has been added to show the distribution of CFL number over the domain.

Other minor points would improve the manuscript

1. In Eq (6) the "sup" operator is used. I am not familiar with this nomenclature. The authors should briefly qualitatively explain this operator and maybe provide a reference.

   RESPONSE:
   Here "sup" is the abbreviation of supremum. This has been explained in the paper.

2. At line 220 the authors note that computation efficiency is obtained by reducing the number of grid cells. However, this reduction is very dependent on the nature of the scalar fields being advected. For many air pollution scenarios, tracer fields can be very noisy with multiple point sources and advection feature, and it is possible that this adaptive grid will use considerably greater grids than a fixed grid approach. For example, in Fig. 1 if another tracer blob were added to the tracer field, or of one of the three shapes were removed, the number of grid cells would change dramatically.

   RESPONSE:
   The computation efficiency has been estimated through a comparison between the fixed and adaptive schemes where the minimum mesh size of the adaptive grid is set to be the same with the mesh size of the fixed uniform grid. Detailed discussion has been provided in cases 1-3.

   Here as suggested, we have added a new case (case 4) to show the computational efficiency, where there are 100 emission sources in the domain. Again, it is shown that the results using adaptive meshes are in agreement with those using fixed meshes with a high mesh resolution of 2.5 km while the number of nodes decreases by a factor of 16 with use of adaptive meshes. The corresponding discussion is provided in section 5.4.

3. At line 270 the authors describe the Staniforth swirling test. This is an interesting test where the advected tracer distribution becomes sheared into smaller and smaller swirls that become infinitesimally small as time progresses, and this raises an interesting question about comparing the "exact" solution with numerical approximations. As time goes to infinity, Walcek & Aleksic (their fig 13) show that the tracer distribution turns into an essentially unchanging button/pillow-like appearance. This "pillow" appearance might in fact be an EXACT solution, AT

THE RESOLUTION OF THE NUMERICAL SIMULATION. When comparing their algorithm with fixed-grid modeling domain, the authors should average the "exact" solution over the identical averaging volumes used in the fixed or adaptive grid models. Even here, the adaptive grid should be averaged onto the same fixed grid and then compared. I think it unfair to compare simulations at different resolutions.

RESPONSE:
The "exact" solutions were plotted at times after initialization by calculating back trajectories along streamlines of these swirling flow. As displayed in Figs 11-12, the distribution becomes sheared into smaller and smaller swirls as time evolves. To adequately represent the "exact" solution, sampling interval Δx or Δy near the edges of the vortex should be set to 1/25600 so that these small-scale features can be qualitatively represented. However, for a given spatial resolution, there is a limiting time beyond which it is no longer possible to adequately represent all the space scales of the exact solution in a qualitative manner.

Therefore, Staniforth et al.(1987) defined two flow regimes (short time periods and long time periods) that have different evaluation criteria for the numerical advection schemes. For long time periods, it is necessary to average the "exact" solution over the identical averaging volumes used in the fixed or adaptive grid models. However, in this paper, we focus on the evaluation of the first regime (short time periods) so that the numerical solutions should be compared with the "exact" one in a qualitative manner. As shown in Figs 11-12, the anisotropic adaptive schemes can effectively represent infinitesimally small-scale features using almost the same number of fixed uniform grid nodes. But if the adaptive grid has been averaged onto the same fixed grid, these small features would disappear due to insufficient resolution.

Further clarification has been made in section 5.3

4. At line 300 the authors state that the regular grid contains 40000 grid cells. This simulation domain consists of a grid of 4x4 (16) swirling vortex circulations that are materially isolated from one another. The initial tracer is spread over only six of those vortices, and all tracer mass remains within those six swirling cells. Therefore the regular grid really only needs 6/16 (or 3/8ths) of the 40000 cells to simulate this tracer evolution with time or 15000 cells. This is the true number of cells required for any non-adaptive grid. The authors should reduce the domain size for this test to be restricted to the six cells containing tracer mass. All of the remaining domain is only advecting a constant. Either reproduce this test using the reduced domain, or change the reference from 40000 cells to 15000 cells.

RESPONSE:
The 3rd test has been reproduced using the reduced domain $[0.24,0.76]\times[0.12,0.88]$ that only cover those 6 swirling vorticies. The maximum number of nodes for the adaptive schemes is set to be 15000. The results have been presented in Figs 10-15.

5. Again for the Staniforth test: Fig. 15 shows that the adaptive grid method is using considerably greater number of grid cells than the 15000 cells required (not 40000,

see note above) by the fixed grid beyond a critical time, and this might even be a problem for this method. I assume the authors utilize some method for stopping grids from becoming infinitesimally small? Please explain how to stop this grid-adaptive method from going too small in size.

RESPONSE:
For robustness of the mesh adaptivity procedure, and to limit refinement/coarsening of the mesh it is possible to set the maximum and minimum allowed edge length sizes. These constraints are achieved through manipulations to the metric, which in turn controls an optimization procedure. Section 3 has been re-written and more details of adaptive mesh techniques have been added.

Anonymous Referee #2

1. It is not clear to me what is novel in this paper, or even whether the goal of the paper is aligned with the aims of this journal. The advection algorithms and also the adaptive refinement algorithms are all implemented in Fluidity, but from the paper it is not at all clear whether the authors of the paper were involved in some new implementation in this version of the code, or are simply testing the code on some particular test problems. The title of the paper, explicitly mentioning Fluidity 4.1.9, makes it sound like the code is specifically designed for the problem discussed in the paper and the paper serves to describe the full code. However, in Section 4 it is stated that Fluidity solves 2D and 3D Navier-Stokes equations and multiphase flow problems over topography, while this paper only concerns scalar advection in two dimensions. So the paper does not seem to describe or test very much of Fluidity. Moreover there is no real discussion of a "new air quality model" anywhere in the paper. Standard 2D advection test problems are used. Advection equations may be used in air quality models but there does not seem to be anything specific to this application, and advection equations arise in many other situations, so it seems misleading to include this term in the title.

   RESPONSE:
   While the individual methods – the advection methods, the mesh adaptivity methods – are not novel, this is the first time that the integrated approaches of full 3D adaptive meshes and advanced numerical discretization techniques have been applied to demanding advection-diffusion problems suitable for testing the advection capability of an atmospheric model. This has been clarified in the abstract.

   In this work, we used Fluidity version 4.1.9, but not limited to. Thus we deleted Fluidity 4.1.9 from the title. Section 4 is shorten. Fluidity is briefly introduced. However, the N-S equations still remain in section 4 since they are important in our future work.

   In this paper, we only focus on integrating this advanced mesh adaptivity methods into air quality modelling. It is well known that the dynamic and chemical processes of air pollution involve a wide range of scales. The initial transformation of emissions from urban and industrial centers and dispersion of plumes occur on relatively small scales, which are responsible for regional or

global air quality problems. But it is a gargantuan computational challenge to modeling large regions with uniform resolution at the finest relevant scale. Therefore, mesh adaptation may be a very effective way to encompass different scales (e.g., local, urban, regional, global) in a unified modeling system. An unstructured adaptive mesh model would be the next generation model for air pollution problems. This has been added to the first paragraph in introduction.

The advanced numerical discretization techniques used in the transport air quality model are described in section 2 and adaptive meshes techniques in section 3. Both sections 2 and 3 are updated (see the general response).

In the revised version, to further demonstrate the advantage of adaptive meshes, we added a 3D advection-diffusion case and used realistic wind data and topography, where the mesh was adapted in 3D and time. This is a first step towards applications in realistic cases.

2. Are these specific advection algorithms and/or the adaptive mesh refinement algorithms significantly different in 4.1.9 than they were in 4.1.8? Or are the authors just noting the particular version that they happened to use for these tests of algorithms that have long been a part of Fluidity? If the latter, what is the novel algorithm or software development? A large number of papers have already been written on advection algorithms of the sort used here, which are often tested on similar problems. The anisotropic refinement algorithm is not described in any detail so it is also not clear if there is anything new here. This all needs to be better clarified.

   RESPONSE:
   As stated above, the novelty is the integration of methods. An integrated method of advanced anisotropic hr-adaptive mesh and discretization numerical techniques has been, for first time, applied to multi-scale transport-diffusion problems, which is based on a discontinuous Galerkin/control volume discretization on unstructured meshes. This has been clarified in the abstract.

   Again, we used Fluidity version 4.1.9, but not limited to.

   Section 3 has been re-written. The anisotropic method has been described in detail.

3. The application of the algorithms to the test problems is not well described, e.g. the description on page 4345 of the error metric tensor is inadequate. In (13) it is stated that H is the Hessian matrix, but of what? The full discretization in terms of all degrees of freedom? How are the elements of this tensor used to determine where to refine?

   RESPONSE:
   Please see the updated version of section 3. The formulae of Hessian, interpolation error, minimum and maximum mesh sizes have been provided, and the anisotropic method has been described.

4. It would be very useful if the authors would make the code available to

accompany this paper, so that readers could potentially better understand the details of the tests performed. This would also be very useful to any reader who is interested in implementing something similar in Fluidity.

RESPONSE:
All the test problems in the paper have been operated in Fluidity model. The source code of Fluidity is available under https://github.com/FluidityProject/fluidity. The user manual and examples are also available. We can offer all setup scripts of the test problems so that the readers can run these test problems directly after installing Fluidity.

5. It is not well explained why it is necessary to use an implicit method for the hyperbolic advection equation, for which explicit methods are more easily implemented and generally preferred for efficiency reasons. It is stated that very large CFL numbers (e.g. 80) are used, and presumably this is because of the highly anisotropic cells with very large aspect ratios. I assume these are stretched in the advection direction, as suggested by Figure 14. Presumably these very high CFL numbers result from comparing e.g. the velocity in the x-direction in this figure to the width of the cells in the y-direction. If the CFL number were truly this large in terms of the number of grid cells the flow advects through in one time step (e.g. if the flow were in the y-direction in Figure 14) then I believe the implicit method would be extremely dissipative and fairly useless, even if it did remain stable. However, this is not discussed in enough detail to figure out what is going on.

   RESPONSE:
   In our work, for discontinuous Galerkin discretization, the explicit Euler scheme is used in conjunction with an advection subcycling method based upon a CFL criterion or a fixed number of subcycles. For the CV discretization, the explicit scheme is easier to implement but strictly limited by the CFL number. Here a new timestepping θ scheme is used to eliminate the time-step restrictions and maintain high accuracy as far as possible, where θ ($1/2 \leq \theta \leq 1$) is chosen to 0.5 for most of elements while big enough (close to 1) for a small fraction of individual elements with a large CFL number (see Fig.16). In this way, the use of a large time step is acceptable when applying adaptive mesh techniques into comprehensive air quality models, which can make the computation much more efficient.

   This has been clarified in the revised version of section 2.2 and the corresponding numerical schemes have been described in detail.

   A new figure (Fig. 16) has been added in case 3, to show the distribution of CFL number over the domain and used to explain the new timestepping θ method.

6. On page 4347, line 25, "advection subcycling" is mentioned but is not explained. Does this mean smaller time steps are used in smaller cells? If so, how are these time steps chosen? Since there is a continuous distribution of cell sizes this is not clear, nor is it clear what is done when adjacent cells are using different size time steps and hence updated a different number of times.

RESPONSE:
For discontinuous Galerkin discretization, an advection subcycling method based upon a CFL criterion or a fixed number of subcycles is adopted in modelling advection flows, that is, the timestep Δt is split to N subtimestep to satisfy the specified Courant number. Further explanation has been added in the revised section 2.2.

7. The anisotropic refinement illustrated in Figure 14 may work well for this flow field in which the streamlines are constant in time and hence the flow is always in a fixed direction at each point in the domain, but it is not at all obvious that the approach used here would work for advection in a real fluid flow (such as the sort Fluidity presumably computes when solving the Navier-Stokes equations, or the sort alluded to in the title of the manuscript). In most flows the direction of flow at each point will be changing dynamically. Even if the adaptive grid is constantly deformed in every time step, the flow would generally not be exactly aligned with the highly anisotropic cells and I suspect this would severely impact the accuracy. All three of the test problems presented in this paper have the feature that the flow directions are time-invariant (even problem 2, where the flow speed varies, has constant direction at each point). I believe the algorithm should be tested on more challenging problems.

   RESPONSE:
   To demonstrate the capability of the adaptive model and estimation of accuracy of solutions, we added a new case (case 4) to simulate the dispersion of power plant plumes, where, the meteorological fields are provided by the mesoscale meteorological model WRF(v3.5) and stored at hourly intervals during 5-day period. For 2D case, a comparison of results using the fixed and adaptive meshes results is plotted in Figs. 18-19. The results using adaptive meshes are in agreement with those using fixed meshes with a high mesh resolution of 2.5 km while the number of nodes decreases by a factor of 16 with use of adaptive meshes.

   We also extended 2D to 3D case, the results are shown in Fig. 20, where the mesh is adapted in 3D and time. It can be seen high resolution meshes are located within the boundary layer and around the power plant stacks.(for details, see section 5.4).

8. The test problems also have large regions of the domain where the solution is constant and hence very few grid cells are needed. This is perhaps reasonable since the point of adaptive refinement is to handle problems where the features needing refinement are relatively isolated. But comparisons of accuracy versus number of cells is then somewhat arbitrary for these problems, since making the domain larger relative to the region where the solution is non-constant would greatly increase the number of grid cells needed for a given resolution on a uniform grid but have no impact on the number of cells needed for the adaptive algorithm. Hence one can make this ratio arbitrarily large by making the domain large, and test problem 3 in particular has a domain that is far larger than reasonable for the given problem.

   RESPONSE:
   We agree with the reviewer and there is always issue in comparing different

methods especially when they are substantially different. None the less this is not a reason not to try to make a comparison. It should be mentioned that these four test problems are benchmark numerical experiments used for testing different numerical advection schemes. We did not make the domain or the ratio larger arbitrarily. But for test problem 3, the initial tracer is spread over only six vortices. Therefore, the 3^{rd} test has been reproduced using the reduced domain $[0.24,0.76]\times[0.12,0.88]$ that cover six swirling vortex containing tracer mass. The results have been presented in Figs 10-15.

9. There is no discussion in the paper of what order of accuracy the advection algorithm is expected to have for smooth solutions, nor even a mention of what order polynomials are used in the continuous or discontinuous Galerkin methods. This is strange, since the presumed advantage of using such methods over simpler and perhaps more efficient finite difference or finite volume methods is that they can achieve higher order. A potential user of Fluidity would surely want to know what orders are supported, along with some evidence that it delivers.

   RESPONSE:
   The equation for calculation of the order of accuracy has been added (see Eq. 26) and corresponding discussion has been provided in cases 1-2.

   For the discontinuous Galerkin methods, polynomials of different degrees k can be used as discontinuous test and trial functions to avoid taking derivatives of discontinuous functions. Within an element, the functions are continuous, and everything is well defined. In this paper, piecewise quartic shape functions (polynomial degree $k = 4$) are used to achieve high-order accurate.
   As an alternative finite volume method, the control volume (CV) methods may be thought of as the lowest order discontinuous Galerkin method, using a dual mesh constructed around the nodes of the parent finite element mesh. In two dimensions this is constructed by connecting the element centroids to the edge midpoints. Once the dual control volume mesh has been defined, it is possible to discretize the advection equation using piecewise constant shape functions within each volume. Although higher-order accuracy is difficult to achieve within the framework of CV method, it is relatively easy to understand and implement using much less computational cost compared with the DG methods.

   The CV and DG methods are usually used in conjunction with unstructured meshes, which are very flexible to capture highly complex solutions and are well suited for hr-adaptivity and parallelization. Even though a number of issues remain, in particular those related to the computational cost of models produced using unstructured mesh methods compared with their structured mesh counterparts. Mesh adaptivity represents an important means to improve the competitiveness of unstructured mesh models, where high resolution is only used when and where necessary. This is the major advantage of using such methods.

   In the next question, we will discuss the order of accuracy for smooth solutions.

10. None of the test problems have smooth initial data for which this accuracy could

be tested. I think some test should be performed of the order of accuracy on smooth data in addition to showing the performance on the sort of data used in the test problems shown.

RESPONSE:
In the first two test problems, we consider a slotted cylinder, a sharp cone, and a smooth hump as the initial solid bodies. The hump as smooth initial data has been considered. In order to discuss what order of accuracy the advection algorithm is expected to have for smooth solutions, we redo the 1st test problem only considering the smooth hump as the initial data. Here, in order to guarantee convergence, it is necessary to use small enough time steps to keep $\Delta t/\Delta x$ *fixed as the grid is refined. The effective order of accuracy* $p = log_2\left(E_1(h)/E_1(h/2)\right)$ *on smooth hump data estimated using* $h = 1/200$ *equals {1.98, 1.52, 1.54, 1.13} for {CV_Fix, CV_Adapt, DG_Fix, DG_Adapt} schemes respectively.*

Due to limitation of pages, we did not add the above smooth case in the paper. However, we mentioned it in section 5.1 by saying:
"If we only consider the hump-smooth profile as the initial data, the order of accuracy can increase to be {1.98, 1.52, 1.54, 1.13}."

11. The error plots in Figures 2 and 5 are logarithmic in x and linear in y, which is not a useful way to display the error. A log-log plot would make it easier to determine the order of accuracy.

RESPONSE:
Figs. 2 and 5 have been replotted in log-log form. The order of accuracy has been discussed in section 5.

12. Moreover, Figures 2 and 5 also seem to show that the error asymptotes to non-zero values as the grid is refined for most of the methods displayed, which means the methods are not even converging, let alone exhibiting any reasonable order of accuracy. This seems to be a serious problem.

RESPONSE:
The convergence issue was caused by the large time step size. Cases 1 ans 2 have been re-run with a small time step. The figures in sections 5.1 and 5.2 have been re-plotted with the new results.

13. What are the units of CPU time in Figures 2 and 5? Seconds? If so, then apparently the uniform grid DG method in Figure 5 requires 11 hours of CPU time for one revolution of two-dimensional advection on a 400 by 400 grid! Even the adaptive DG code seems to take around 2 hours with h = 1/800, which seems quite excessive for this problem. Of course it would also be useful to state what computer these timings were done on, and how many cores were used since it is stated in the paper that Fluidity uses MPI for parallelization to thousands of cores, although it is not stated whether this is used in these examples.

RESPONSE:
The units of CPU time in Figures 2 and 5 are seconds. We did not use MPI for parallelization. All computations were performed on a workstation using the

Gfortran Compiler for Linux. The simulation workstation has 8 processors and a 4GB random-access memory (RAM). The processor used in workstation is Intel(R) Core(TM) i7-2600 CPU @ 3.40GHz. A single processor with frequency of 3.40GHz was used since the test cases were simulated in serial.

This has been clarified in section 5.

14. If I am interpreting the timings right, I suspect the slowness of the code is due to the use of an implicit method for the advection problem. This leads me again to question the wisdom of such methods for this problem, since there are good explicit block-structured AMR algorithms implemented in software such as AMROC, Boxlib, Chombo, Clawpack, SAMRAI, etc. that I believe works quite efficiently on the sort of test problems presented here. More justification is needed for the value of the methods implemented in Fluidity than is presented in this paper.

RESPONSE:
For implicit issue, please see the response to question 5. The slowness of codes is due to the use of unstructured meshes – this is a common issue in unstructured mesh models. In unstructured mesh finite element modelling, it involves the integration of equations over the domain. It thus spends most of time on assembling the matrices at each time step, especially for nonlinear problems. We used a number of numerical techniques to reduce the CPU time, for example, the timestepping θ scheme to eliminate the time-step restrictions (please see section 2.2 and response to question 5). The use of adaptive meshes will also reduce the number of nodes, thus increasing the computational efficient (please see discussion in cases 1 and 2 in section 5) although it may take time on adapting the mesh at certain time level.

As stated in introduction (or see response to question 9), the unstructured adaptive mesh technique is important in next generation models since it may be the only way to model multi-scale flow dynamical problems in large regions. Due to advanced computational technologies, the issue of CPU times can be sorted out using MPI. Fluidity is parallelized using MPI and is capable of scaling to many thousands of processors.

15. In (15), epsilon is a relative error tolerance that is presumably some positive value chosen by the user, so why is epsilon_min needed to ensure the denominator is nonzero?

RESPONSE:
For example, if $\epsilon = 0.01$, then the tolerance on the denominator of the metric formulation will be 1% of the value of the field c , and so it will scale the target interpolation error with the magnitude of the field. Since the value of the field c may be zero in some region of the domain (e.g., the background of tracer field has been set to be zero in the 1^{st} and 2^{nd} test), it is necessary to set the minimum tolerance ϵ_{min} to ensure that the denominator never becomes zero.

16. On page 4345 line 15, I am not sure what is meant by imposing different limits on the cell sizes in different directions.

RESPONSE:
For robustness of the mesh adaptivity procedure, and to limit refinement/coarsening of the mesh it is possible to set maximum and minimum allowed edge length sizes. The inputs to these quantities are tensors allowing one to impose different limits in different directions. Assuming that these directions are aligned with the coordinate axes allows one to define diagonal tensors. These constraints are achieved through manipulations to the metric, which in turn controls an optimization procedure. They are therefore not hard constraints and one may observe the constraints being broken (slightly) in places.

17. Page 4347, line 20 and I am not sure what is meant by the "Sweby limiter". Sweby's paper discussed many limiter such as minmod, superbee, etc., but I am not sure what Sweby limiter is referred to here.

RESPONSE:
Although Sweby(1984) discussed many limiter functions, the "Sweby limiter" here is not referred to any one of them, but only use the Sweby's TVD region on the normalized variable diagram(NVD) as a criterion to limit the face value calculated by the finite element interpolation approach. Any combination of normalized face and donor values falling within this region is left unchanged. Values falling outside this region are 'limited' along lines of constant normalized donor value back onto the top or bottom of the stable region. The high order flux is obtained from the finite element interpolation of the CV values (for details, see AMCG, 2014). So, we use the name "CV-TVD limiter" instead of "Sweby limiter" in the revised version.

Manuscript prepared for Geosci. Model Dev. Discuss.
with version 2015/04/24 7.83 Copernicus papers of the LaTeX class copernicus.cls.
Date: 26 September 2015

A new multiscale air quality transport model ~~(Fluidity, 4.1.9)~~ using fully unstructured anisotropic adaptive mesh technology

J. Zheng[1,2,4], **J. Zhu**[2], **Z. Wang**[1], **F. Fang**[3], **C. C. Pain**[3], **and J. Xiang**[3]

[1]State Key Laboratory of Atmospheric Boundary Layer Physics and Atmospheric Chemistry, Institute of Atmospheric Physics, Chinese Academy of Sciences, Beijing 100029, China
[2]International Center for Climate and Environment Sciences, Institute of Atmospheric Physics, Chinese Academy of Sciences, Beijing 100029, China
[3]Applied Modelling and Computation Group, Department of Earth Science and Engineering, Imperial College London, Prince Consort Road, London, SW7 2BP, UK
[4]University of Chinese Academy of Sciences, Beijing 100049, China

Correspondence to: F. Fang (f.fang@imperial.ac.uk)

Abstract

~~A new~~ An integrated method of advanced anisotropic hr-adaptive mesh ~~technique has been~~and discretization numerical techniques has been, for first time, applied to modelling of ~~multiscale transport phenomena~~multi-scale advection-diffusion problems, which is based on a ~~-~~discontinuous Galerkin/control volume discretization on unstructured meshes. Over existing air quality models typically based on static-structured grids using a locally nesting technique, the advantage of the anisotropic hr-adaptive model has the ability to adapt the mesh according to the evolving pollutant distribution and flow features. That is, the mesh resolution can be adjusted dynamically to simulate the pollutant transport process accurately and effectively. To illustrate the capability of the anisotropic adaptive unstructured mesh model, three benchmark numerical experiments have been setup for two-dimensional (~~2-D) transport~~ 2D) advection phenomena. Comparisons have been made between the results obtained using uniform resolution meshes and anisotropic adaptive resolution meshes. Performance achieved in 3D simulation of power plant plumes indicates that this new adaptive multiscale model has the potential to provide accurate air quality modeling solutions effectively.

1 Introduction

It is well known that the interaction of multiscale physical processes in atmospheric phenomena poses a formidable challenge for numerical modelling (Kühnlein, 2011). Large scale processes can trigger small scale features that again have an important influence/feed-back to the large scale (Behrens, 2007). For example, the processes of tropical cyclone involve a range over a continuous spectrum of scales from the large-scale flow environment $\sim\mathcal{O}(10^6\text{–}10^7)\,\mathrm{m}$, tropical cyclone itself $\sim\mathcal{O}(10^5\text{–}10^6)\,\mathrm{m}$, embedded eyewall and rainbands $\sim\mathcal{O}(10^3\text{–}10^4)\,\mathrm{m}$, down to microscales of the boundary layer turbulence $\sim\mathcal{O}(10\text{–}10^2)\,\mathrm{m}$ (Kühnlein, 2011). ~~Due to the highly disparate scalesinvolved, global mesh refinement is not a viable option. Thus~~ For air pollution, the dynamic

and chemical processes also involve a wide range of scales. The initial transformation of emissions from urban and industrial centers or dispersion of plumes from large power plant stacks occur on relatively small scales, but would be engaged to much larger scales after long range transport. It is a gargantuan computational challenge to modeling large regions with uniform resolution at the finest relevant scale. Therefore, mesh adaptation may be the only effective way to ~~resolve these multiscale geophysical flows accurately (Kühnlein, 2011; Weller et al., 2010; Nikiforakis, 2009)~~ encompass different scales (e.g. local, urban, regional, global) in a unified modeling system to better capture the interactions among the processes relevant at each scale. (Garcia-Menendez and Odman, 2011; Kühnlein, 2011; Weller et al., 2010; Nikiforakis, 2009) .

So far, the accurate numerical modelling of advection (or transport) remains a central problem for many applications such as air pollution, atmospheric chemistry, meteorology and other physical sciences. There have been many studies on the numerical advection schemes (e.g. PPM, Bott and Walcek etc.) which have been used in many air quality models (e.g. CMAQ, CMAx, NAQPMS etc.) (Colella and Woodward, 1984; Bott, 1989; Walcek and Aleksic, 1998). These advection algorithms were implemented based on a fixed uniform mesh system. The successive global refinement can be used to capture the details of small scale flow features, but is prohibitively expensive and not feasible for practical applications. Alternatively, the nesting technique, placing finer meshes within coarser meshes, is often used for achieving local higher resolution in many air quality models (Garcia-Menendez and Odman, 2011; Frohn et al., 2002; Wang, 2001). In static mesh nesting, the solutions obtained from the global coarse mesh model provide the boundary conditions for the nested mesh regional model, in turn, the solutions in the global model are updated with the high resolution solutions. However this may lead to spurious oscillations at the interface between the coarse mesh and nested fine mesh, especially when concentration gradients is large cross the interface. Although the numeral techniques such as blending, nudging, and selective damping approaches can be used to remove these oscillations, the small scale features on the fine meshes may be damped (Garcia-Menendez and Odman, 2011; Zhang et al., 1986; Debreu and Blayo, 2008; Alapaty et al., 1998). Moreover, due to highly unsteady at-

mospheric flows, it is almost impossible to construct a static optimal nested mesh suitable for an accuracy simulation over a long time period. The use of dynamically adaptive mesh techniques can therefore be considered so that the mesh resolution can be adjusted locally in response to the evolution of the flow and passive tracer (Piggott et al., 2009; Behrens, 2007).

In contrast to locally nested mesh techniques, adaptive mesh techniques not only can resolve multiscale processes in a consistent way, but also can enable to follow and capture the features of flows as time evolves. Dynamic mesh adaptation can be achieved, either by relocating mesh nodes or by locally increasing (and decreasing) the number of nodes in time and space. The former, known as mesh movement (i.e. r-adaptivity), can be used to improve the accuracy of solutions by optimally re-locating mesh nodes to resolve the small scale features of interest (Garcia-Menendez and Odman, 2011; Srivastava et al., 2000; Lagzi et al., 2009; Kühnlein et al., 2012; Nikiforakis, 2009). However, the accuracy of solutions using r-adaptivity is restricted by a priori for achieving an optimal dynamic mesh (where the total number of nodes is fixed). The latter, known as mesh enrichment (i.e. h-adaptivity), can guarantee a minimum solution accuracy level by providing sufficient resolution where and when it is needed (Baker et al., 2013; Constantinescu et al., 2008; Piggott et al., 2005). Various h-adaptive techniques based on structured meshes as well as the r-adaptive techniques on unstructured/structured meshes have been explored in atmospheric modeling. And some of these techniques have been applied to air quality models (Garcia-Menendez and Odman, 2011). Recently, significant research efforts have been focused on application of this new adaptive mesh techniques in ocean modeling (Pain et al., 2005; Piggott et al., 2009, 2008a, b).

This article applies a new anisotropic hr-adaptive mesh technique into ~~two-dimensional (2-D)~~ air quality transport(advection) modelling. This adaptive unstructured mesh technique provides the dynamic spatial and temporal resolution to capture moving features, e.g. moving fronts or ~~dust storm~~power plant plumes. Using the hr-adaptive technique, existing elements can be split (h-adaptive) or element vertices can be moved (r-adaptive), to periodically modify the mesh geometry. Hence, the purpose of this article is to demonstrate,

through example problems, the capability of anisotropic mesh adaptivity for modelling of multiscale transport phenomena.

The remaining structure of this article is as follows: Sect. 2 describes numerical ~~advection~~ methods, including discontinuous Galerkin (DG) and control volume (CV) methods based on unstructured meshes. Section 3 covers the topics of mesh adaptivity, error measures and interpolation. Section 4 introduces ~~3-D~~ 3D unstructured anisotropic adaptive mesh model (Fluidity). ~~In Sect. 5 , results~~ Section 5 discusses its performance in three benchmark ~~tests are presented and discussed for a two-dimensional scalar advection problem~~ advection problems and a model problem for dispersion of power plant plumes. Conclusions are drawn in Sect. 6.

2 Numerical ~~advection~~ methods for transport equation

As a model problem, we consider the ~~advection~~ generic transport equation for a scalar quantity c, is given in conservative form by:

$$\frac{\partial c}{\partial t} + \nabla \cdot (\boldsymbol{u} c) - \nabla \cdot (\overline{\overline{\kappa}} \nabla c) = s, \tag{1}$$

where $\boldsymbol{u} = (u, v, w)^{\mathsf{T}}$ is the velocity vector~~.~~, $\overline{\overline{\kappa}}$ is the diffusivity (tensor) and s represents any source or reaction terms. If $\overline{\overline{\kappa}} = 0$ and $s = 0$, Eq. (1) reduces to the advection equation:

$$\frac{\partial c}{\partial t} + \nabla \cdot (\boldsymbol{u} c) = 0, \tag{2}$$

2.1 Spatial discretization

Integrating Eq. (2) by part over the computational domain Ω, its weak form can be written:

$$\int_{\Omega} \left(\phi \frac{\partial c}{\partial t} - \nabla \cdot \phi \cdot \left(\boldsymbol{u} c - \overline{\overline{\kappa}} \nabla c \right) - \phi s \right) \mathrm{d}\Omega + \int_{\partial\Omega} \left(\phi \widehat{\boldsymbol{n} \cdot \boldsymbol{u} c} - \phi \widehat{\boldsymbol{n} \cdot \overline{\overline{\kappa}} \nabla c} \right) \mathrm{d}\partial\Omega = 0. \tag{3}$$

2.1.1 Discontinuous Galerkin discretization

As a locally conservative, stable and high-order accurate method, the discontinuous Galerkin methods can easily construct discontinuous approximations on unstructured meshes to capture highly complex solutions and are well suited for hr-adaptivity and parallelization (Cockburn et al., 2000; Cockburn and Shu, 2001; Flaherty et al., 2002; Hesthaven and Warburton, 2007). Moreover DG methods, as a generalization of finite volume methods, can directly make numerical fluxes and slope limiters available in the finite element framework (Burbeau et al., 2001; Hoteit et al., 2004; Krivodonova, 2007; Krivodonova et al., 2004).

Integrating Eq. (2) over a single element and summing over all elements, we obtain:

$$\sum_e \left\{ \int_e \left(\phi \frac{\partial c}{\partial t} - \nabla \cdot \phi \cdot \left(\boldsymbol{u} c - \bar{\bar{\kappa}} \nabla c \right) - \phi s \right) \mathrm{d}e + \int_{\partial e} \left(\phi \widehat{\boldsymbol{n} \cdot \boldsymbol{u} c} - \phi \widehat{\boldsymbol{n} \cdot \bar{\bar{\kappa}} \nabla c} \right) \mathrm{d}\partial e \right\} = 0, \tag{4}$$

where, the hatted term represents fluxes across the element facets. If $\bar{\bar{\kappa}} = 0$ and $s = 0$, equation (4) becomes a pure advection equation:

$$\sum_e \left\{ \int_e \left(\phi \frac{\partial c}{\partial t} - \nabla \phi \cdot \boldsymbol{u} c \right) \mathrm{d}e + \int_{\partial e} \phi \widehat{\boldsymbol{n} \cdot \boldsymbol{u} c} \, \mathrm{d}\partial e \right\} = 0, \tag{5}$$

Due to the discontinuous nature of fields, there is no unique value for the flux term, however the requirement that c is a conserved quantity, does demand that adjacent elements make a consistent choice for the flux between them. In this work, two advective flux schemes, the upwind and local Lax-Friedrichs flux methods, are used to represent $\widehat{\boldsymbol{n} \cdot \boldsymbol{u} \, c}$ for DG methods (AMCG, 2014). In $\widehat{\boldsymbol{n} \cdot \boldsymbol{u} \, c}$, the advecting velocity $\boldsymbol{u}$ can be calculated by either averaging it on each side of the face or applying a Galerkin projection to project the velocity onto a continuous basis.

In the upwind flux formulation, the value of c at each quadrature point on the face is taken to be the upwind value, that is, if fluid is into/out of the element then it is the value on the exterior/interior side of the face. Integrating the advection term by ~~part~~ parts twice, then Eq. (~~4~~5) becomes (AMCG, 2014):

$$\sum_e \left\{ \int_e \left(\phi \frac{\partial c}{\partial t} - \phi\, \hat{\boldsymbol{u}} \cdot \nabla c \right) \mathrm{d}e + \int_{\partial e \cap \partial\Omega} \boldsymbol{n} \cdot \hat{\boldsymbol{u}}\, (c_b - c_{\text{int}}) \mathrm{d}\partial e + \int_{\partial e \setminus \partial\Omega} \boldsymbol{n} \cdot \hat{\boldsymbol{u}}\, (c_{\text{ext}} - c_{\text{int}}) \mathrm{d}\partial e \right\} = 0, \tag{6}$$

where, $\hat{\boldsymbol{u}}$ represents the flux velocity and a weakly imposed boundary condition $c = c_b$ is applied on the inflow part of boundaries; c_{ext} and c_{int} are the values on the exterior and interior side of the face respectively.

In local Lax–Friedrichs flux formulation, the tracer advection is given by:

$$\widehat{\boldsymbol{n} \cdot \boldsymbol{u}\, c} = \frac{1}{2} \boldsymbol{n} \cdot \hat{\boldsymbol{u}} (c_{\text{int}} + c_{\text{ext}}) - \frac{C}{2} (c_{\text{int}} - c_{\text{ext}}), \tag{7}$$

where for each facet $s \subset \partial e$:

$$C = \sup_{x \in s} |\hat{\boldsymbol{u}} \cdot \boldsymbol{n}|. \tag{8}$$

Here, "sup" is the abbreviation of supremum.

To ensure nonlinear stability and effectively suppress spurious oscillations, the slope limiting techniques are used here (Kuzmin, 2010; Cockburn and Shu, 2001; Luo et al., 2007).

2.1.2 Control volume discretization

The control volume discretization uses a dual mesh constructed around the nodes of the parent finite element mesh. Once the dual control volume mesh has been defined, it is possible to discretise the ~~advection~~ transport Eq. (~~2~~1)using piecewise constant shape functions

within each volume, v. Integrating Eq. (21) by parts within a volume, v and summing over all volumes, we obtain:

$$\sum_{v}\left\{\int_{v}\left(\frac{\partial c}{\partial t}-s\right)\mathrm{d}v+\sum_{k}\int_{\partial v_k}\left(\widehat{\boldsymbol{n}\cdot\boldsymbol{u}}\,c-\widehat{\boldsymbol{n}\cdot\bar{\bar{\kappa}}\nabla c}\right)\mathrm{d}\partial v_k\right\}=0,. \tag{9}$$

here

For the flux term $\widehat{\boldsymbol{n}\cdot\boldsymbol{u}}\,c$, the velocity is well-defined since the control volume facets are in the center of the elements of the parent mesh where it is continuous. The face value of c_k is computed at each quadrature point of the facet k using the finite element interpolation approach, i.e. interpolating it using the finite element basis functions on the parent mesh. Usually the first order quadrature is performed on the control volume facets, however if higher order control volume facet quadrature is selected then k refers to each quadrature point on the facet. To avoid the spurious oscillations, the Sweby CV-TVD flux limiter is used to make the solutions total variation diminishing (Sweby, 1984; Leonard, 1991; Waterson and Deconinck, 2007; Wilson, 2009; LeVeque, 2002) (Sweby, 1984; AMCG, 2014) .

For diffusion term $\widehat{\boldsymbol{n}\cdot\bar{\bar{\kappa}}\nabla c}$, ∇c is treated with control volumes-element based gradients, equal order Bassi-Rebay and staggered mesh Bassi-Rebay discretization (for details, see (AMCG, 2014)).

2.2 Time discretization

The semi-discrete matrix form of Eq. (3) can be written as

$$\mathbf{M}\frac{\mathrm{d}\boldsymbol{c}}{\mathrm{d}t}+\mathbf{A}(\boldsymbol{u})\boldsymbol{c}+\mathbf{K}\boldsymbol{c}=\boldsymbol{r}, \tag{10}$$

in which the vector $\boldsymbol{c}=(c_1,\ldots,c_{\mathcal{N}})^T$ contains the solution of variable c at nodes ($\mathcal{N}$ is the number of nodes);~~$\boldsymbol{r}$ is the right-hand side vector containing boundary, source and~~

~~absorption terms;~~, $\mathbf{M}$ is the mass matrix~~; and~~, $\mathbf{A}(\boldsymbol{u})$ is the advection operator, $\mathbf{K}$ is the diffusion operator and $\boldsymbol{r}$ is the right-hand side vector containing boundary, source and absorption terms, where for continuous Galerkin discretization:

$$\mathbf{M}_{ij} = \int_\Omega \phi_i \phi_j, \quad \mathbf{A}_{ij} = -\int_\Omega \nabla\phi_i \cdot \boldsymbol{u}\phi_j, \quad \mathbf{K}_{ij} = -\int_\Omega \nabla\phi_i \cdot \bar{\bar{\kappa}} \nabla\phi_j, \quad i,j \in (1,2,\ldots,\mathcal{N}). \tag{11}$$

The time derivative term at time level $n+1$ is treated using the θ-method to yield

$$\mathbf{M}\frac{\boldsymbol{c}^{n+1}-\boldsymbol{c}^n}{\Delta t} + \mathbf{A}(\boldsymbol{u}^{n+\theta n})\boldsymbol{c}^{n+\theta} + \mathbf{K}\boldsymbol{c}^{n+\theta} = \boldsymbol{r}^{n+\theta}. \tag{12}$$

where $\theta \in [0,1]$ and the terms $\boldsymbol{c}^{n+\theta}$ are given by,

$$\boldsymbol{c}^{n+\theta} = \theta\boldsymbol{c}^{n+1} + (1-\theta)\boldsymbol{c}^n. \tag{13}$$

Equation (12) can be rearranged for unknown vector $\boldsymbol{c}^{n+1}$:

$$\left(\mathbf{M} + \theta\Delta t\left(\mathbf{A}\left(\boldsymbol{u}^{n+\theta n}\right) + \mathbf{K}\right)\right)\boldsymbol{c}^{n+1} = \left(\mathbf{M} - (1-\theta)\Delta t\left(\mathbf{A}\left(\boldsymbol{u}^{n+\theta n}\right) + \mathbf{K}\right)\right)\boldsymbol{c}^n + \boldsymbol{r}^{n+\theta}. \tag{14}$$

Equation (14) can be solved in two stages:

$$\mathbf{M}\frac{\boldsymbol{c}_* - \boldsymbol{c}^n}{\Delta t} + \mathbf{A}\left(\boldsymbol{u}^n\right)\boldsymbol{c}^{n+\theta} = \boldsymbol{r}_D^{n+\theta} \tag{15}$$

$$\mathbf{M}\frac{\boldsymbol{c}^{n+1} - \boldsymbol{c}_*}{\Delta t} + \mathbf{K}\boldsymbol{c}^{n+\theta} = \boldsymbol{r}_N^{n+\theta} + \boldsymbol{r}_s^{n+\theta}, \tag{16}$$

where $\boldsymbol{r}$ in Eq. (14) is split into Dirichlet $\boldsymbol{r}_D$ and Neumann boundary components $\boldsymbol{r}_N$, and a source component $\boldsymbol{r}_s$.

For discontinuous Galerkin discretization, the explicit Euler scheme ($\theta = 0$) is used in Eq. (15). An advection subcycling method based upon a CFL criterion or a fixed number of subcycles is adopted in modelling advection flows, that is, the timestep Δt is split to N subtimestep $\Delta t_{sub} = \frac{\Delta t}{N}$ to satisfy the specified Courant number:

$$\mathbf{M}\boldsymbol{c}_{\text{new}} = \left(\mathbf{M} - \frac{\Delta t}{N}\mathbf{A}\left(\boldsymbol{u}^n\right)\right)\boldsymbol{c}_{\text{old}} + \boldsymbol{r}_D^{n+\theta}. \tag{17}$$

To guarantee a bounded solution, the slope limiter is applied to $\boldsymbol{c}_{\text{new}}$ after each subtimestep. Note that the matrix $\mathbf{M} - (\Delta t/N)\mathbf{A}(\boldsymbol{u}^n)$ is constant within one timestep. Therefore the process of solving eq.(17) only involves the matrix-vector multiplication, thus reducing a large amount of the CPU time required for assembling the matrices, especially when unstructured meshes are used.

For control volume discretization, an explicit scheme is simple but strictly limited by the CFL number which can be restrictive on adaptive meshes as the minimum mesh size can be very small. Here, we adopt a new timestepping θ scheme based on traditional Crank–Nicolson scheme ($\theta_{aim} = 1/2$) because of its robustness, unconditional stability and second-order accurate in time (Pavlidis et al., 2015; Versteeg and Malalasekera, 2007; Donea and Huerta, 2003) . For the given time step, the value of θ_{min} can be estimated at each CV face based on the satisfaction of a total variation diminishing (TVD) criterion. Therefore, for each control volume v, we can choose $\theta_v \in [\theta_{min}, 1]$ to be as close to θ_{aim} as possible. That is, $\theta_v = \max\{\theta_{min}, \theta_{aim}\}$. In this way, it can eliminate the local time step restriction for physically realistic and bounded solution although it may be in cost of losing some local accuracy. (for details, see (Pavlidis et al., 2015)).

3 Mesh adaptivity

The optimization-based adaptivity technique developed by the Applied Modelling and Computation Group (AMCG) at Imperial College London (AMCG, 2014), is introduced

in this section. It utilizes dynamic adaptation of a fully unstructured triangular (or tetrahedral) mesh in two (or three)-dimensions, as presented in (Pain et al., 2001, 2005; Piggott et al., 2009). The unstructured and adaptive meshes allow computational effort to resolve important fluid dynamics at diverse scales. The key objective of using adaptive mesh methods is to reduce the overall computational cost in achieving an error goal; thus ensuring that fine resolution is used only when and where it is needed ~~(Fang et al., 2010)~~ (Fang et al., 2010; Pain et al., 2001) . A error metric tensor to guide an adaptive meshing algorithm can be defined ~~(Fang et al., 2010)~~ (Pain et al., 2001) :

$$\bar{\mathbf{M}}_e = \frac{\gamma}{|\epsilon|}|\mathbf{H}|\frac{\gamma}{\epsilon}|\mathbf{H}|, \tag{18}$$

where, ~~**H is the Hessian matrix,** ϵ~~ is the required level of ~~error~~ errors and γ a scalar constant ~~.~~ (here, $\gamma = 1$), **H** is the Hessian matrix of variable fields (here, the tracer concentration c):

$$\mathbf{H} = \begin{pmatrix} \frac{\partial^2 c}{\partial x^2} & \frac{\partial^2 c}{\partial x \partial y} & \frac{\partial^2 c}{\partial x \partial z} \\ \frac{\partial^2 c}{\partial y \partial x} & \frac{\partial^2 c}{\partial y^2} & \frac{\partial^2 c}{\partial x \partial z} \\ \frac{\partial^2 c}{\partial z \partial x} & \frac{\partial^2 c}{\partial z \partial y} & \frac{\partial^2 c}{\partial z^2} \end{pmatrix}.$$

The absolute value of the symmetric Hessian matrix is defined as ~~(Fang et al., 2010)~~ (Pain et al., 2001) :

$$|\mathbf{H}| = \mathbf{V}\Lambda\mathbf{V}^{\mathsf{T}}, \tag{19}$$

where, the matrices **V** and Λ contain the eigenvectors $\boldsymbol{e}_i$ and eigenvalues ~~Λ_i~~ λ_i of the Hessian matrix **H** respectively. The required edge length in the direction $\boldsymbol{e}_i$ to achieve the required level of errors ϵ can be obtained (Piggott et al., 2009) :

$$h_i = \frac{1}{\sqrt{\epsilon\lambda_i}}. \tag{20}$$

The rotation matrix $\mathbf{V}$ in combination with Λ can be used to adapt the original element to an anisotropic element required for the given level of errors. To bound the aspect ratio of elements in physical space, the eigenvalues of the metric can be modified (Pain et al., 2001) :

$$\hat{\lambda}_j = \max\left\{\lambda'_j, \frac{1}{a^2}\max_{i=1}^{3}\lambda_i\right\}, \quad j = 1,2,3, \tag{21}$$

where

$$\lambda'_j = \min\left\{\frac{1}{h_{min}^2}, \max\left\{|\lambda_j|, \frac{1}{h_{max}^2}\right\}\right\}, \quad j = 1,2,3, \tag{22}$$

where, a is the a given aspect ratio of elements, h_{min} and h_{max} are the minimum and maximum sizes of elements respectively.

To represent small-scale dynamics, a relative error metric formulation is suggested:

$$\mathbf{M}_e = \frac{\gamma|\mathbf{H}|}{\max(|\epsilon|\cdot|f|, |\epsilon_{\min}|)} \frac{\gamma|\mathbf{H}|}{\max(\epsilon\cdot|c|, \epsilon_{\min})}, \tag{23}$$

where, ~~f~~ c is the field under consideration, ϵ is now a relative tolerance, and $\epsilon_{\min}$ is the minimum tolerance used to ensure that the denominator never becomes zero.

To guide refinement/coarsening of the mesh, the maximum and minimum mesh sizes are set to allow one to impose different limits in different directions (for details, see AMCG, 2014). Assuming that these directions are aligned with the coordinate axes allows one to define diagonal tensors. The maximum and minimum number of nodes are also set for mesh adaptivity. This is effected by computing the expected number of nodes from the given metric. If the expected number of nodes is greater than the maximum number of nodes, the metric resolution is homogeneously decreased so that the expected number of nodes is the maximum number of nodes.

Another key issue of mesh adaptivity is to interpolate any necessary data from the previous mesh to the adapted one. The consistent interpolation is often adopted in mesh adaptivity. However, the consistent interpolation can introduce a suboptimal interpolation error, unsuitability for discontinuous fields, and lack of conservation. An alternative conservative interpolation approach, the Galerkin projection is proposed for discontinuous fields. A supermeshing algorithm (Farrell et al., 2009) is used for implementation of the Galerkin projection.

4 Introduction of a ~~3-D~~ 3D unstructured anisotropic adaptive mesh model (Fluidity~~, 4.1.9~~)

The new multiscale air quality transport model has been developed with a ~~3-D~~ 3D unstructured and adaptive mesh model (Fluidity, developed by the Applied Modelling and Computation Group (AMCG) at Imperial College London). Fluidity, an open source LGPL model, ~~is a general purpose multiphase CFD code which is capable of modelling a wide range of fluid phenomena involving single and multiphase flows. It~~ numerically solves the ~~2-D~~2D/~~3-D~~ 3D Navier–Stokes equation (being non-hydrostatic, to model dense water formation and flows over steep topography) and field equations with a range of control volume and finite element ~~discretisation methods. Its modelling framework has been developed through the cross-fertilisation of techniques and experiences in engineering fluids, oceanography, atmospheric dynamics, air pollution, impact cratering, inversion, radiation transport, coupled multi-physics modelling, multi-phase flows and fluid-structure interactions. This has led to~~ discretization methods. It includes a number of novel, advanced methods based upon adapting and moving anisotropic unstructured meshes, advanced finite element and control volume ~~discretisations~~discretization, and a range of numerical stabilization and Large Eddy Simulation (LES) turbulence models. Among existing unstructured mesh models, Fluidity is the only model that can simultaneously resolve both small- and large-scale fluid flows while smoothly varying resolution and conforming to complex topography. The model employs ~~3-D~~ 3D anisotropic mesh adaptivity to resolve and reveal fine scale features as

they develop while reducing resolution elsewhere. A number of interpolation methods (e.g., non conservative point-wise and conservative methods) are available for mesh-to-mesh interpolations between adaptations.

Fluidity is parallelized using MPI and is capable of scaling to many thousands of processors. It has a user-friendly GUI and a python interface which can be used to calculate diagnostic fields, set prescribed fields or set user-defined boundary conditions (for details, see https://www.imperial.ac.uk/engineering/departments/earth-science/research/research-groups/amcg/).

5 Numerical examples

To illustrate the efficiency and accuracy of anisotropic adaptive schemes, ~~three~~ Four benchmark problems have been adopted which are representative and challenging enough to predict how the new adaptive multiscale model would behave in future real-life applications (LeVeque, 1996; Kuzmin, 2009; Staniforth et al., 1987; Walcek and Aleksic, 1998; Bott, 1989, 1993, 2010).

In the following comparative study, we consider FEM_Fix and FEM_Adapt schemes (FEM represents CV or DG) based on the control volume and discontinuous Galerkin discretization~~in the same computational domain $\Omega = [0,1] \times [0,1]$~~. The CV_Fix_L and DG_Fix_L schemes use fixed uniform triangular meshes while the CV_Adapt_L and DG_Adapt_L schemes use adaptive meshes (where L represents the different mesh schemes, as shown in Table 1). For CV discretization, a finite element interpolation is used at the control volume faces ~~in combination~~ with a ~~Sweby slope~~ CV-TVD limiter to bound the solution. The time discretization used here is the ~~classical~~ new timestepping θ scheme based on the Crank–Nicolson scheme~~($\theta = 0.5$)~~. For DG discretization, the upwind flux is chosen in combination with vertex-based slope limiter. The slope limiter used with the discontinuous Galerkin formulation only guarantee a bounded solution in conjunction with an explicit advection scheme. Therefore, advection subcycling based upon a CFL criterion is necessary for DG discretization (AMCG, 2014).

Equation (14) is solved by the generalized minimum residual method (Saad, 1993). The successive over-relaxation preconditioned is invoked to speed up convergence at large time steps.

To assess the difference between the analytical solution c and its numerical approximation c_h, we introduce the error norms:

$$E_1 = \int_\Omega |c - c_h| \mathrm{d}\Omega = \|c - c_h\|_1, \tag{24}$$

$$E_2 = \sqrt{\int_\Omega |c - c_h|^2 \mathrm{d}\Omega} = \|c - c_h\|_2. \tag{25}$$

The order of accuracy in modelling is used to assess the numerical convergence rate:

$$p = log_2(E_1(h)/E_1(h/2)), \tag{26}$$

where h is the mesh size.

All computations were performed on a workstation using the Gfortran Compiler for Linux. The simulation workstation has 8 processors and a 4GB random-access memory (RAM). The processor used in workstation is Intel(R) Core(TM) i7-2600 CPU @ 3.40GHz. A single processor with frequency of 3.40GHz was used since the test cases were simulated in serial.

5.1 Case one: solid body revolution

A standard test problem applied to the ~~convection~~ advection equation (2) in ~~2-D~~ 2D is solid body revolution (LeVeque, 1996; Kuzmin, 2009). The incompressible velocity field in the domain $\Omega = [0,1] \times [0,1]$ is represented by

$$\boldsymbol{u}(x,y) = (0.5 - y, x - 0.5), \tag{27}$$

which corresponds to a counterclockwise rotation around the center $(0.5, 0.5)$ of Ω. Following LeVeque (1996), we consider a slotted cylinder, a sharp cone, and a smooth hump as the initial solid bodies defined within the circle centered at each reference point (x_0, y_0):

$$r(x,y) = \frac{1}{r_0}\sqrt{(x-x_0)^2 + (y-y_0)^2} \leq 1, \tag{28}$$

where $r_0 = 0.15$. After each full revolution ($t = 2\pi k$), the exact solution return to the initial distribution as depicted in Fig. 1. For the slotted cylinder, the reference point is $(x_0, y_0) = (0.5, 0.75)$ and

$$c(x,y,0) = \begin{cases} 1 & \text{if } |x - x_0| \geq 0.03 \text{ or } y \geq 0.85, \\ 0 & \text{otherwise.} \end{cases} \tag{29}$$

The cone is centered at $(x_0, y_0) = (0.5, 0.25)$ and its geometry is given by

$$c(x,y,0) = 1 - r(x,y). \tag{30}$$

The peak of the smooth hump is located at $(x_0, y_0) = (0.25, 0.5)$ and the shape function is

$$c(x,y,0) = \frac{1 + \cos(\pi r(x,y))}{4}. \tag{31}$$

In the rest of the domain Ω, the solution of Eq. (2) is initialized by zero. The challenge of this numerical test case is to preserve the shape of the rotating bodies as time evolves. The mesh size used for the FEM_Fix_L schemes and the FEM_Adapt_L schemes are listed in Table 1. The time step is set to $\Delta t = 0.01$ for all different mesh schemes. For DG discretization, the explicit advection subcycling scheme with a tight CFL criterion (here 0.1) is used to make sure that the simulation is converging as the mesh is refined. For CV discretization, although the timestepping θ scheme based on the Crank–Nicolson scheme can maintain high accuracy, the subcycling number is set to be $\{2, 4\}$ for $h = \{1/400, 1/800\}$

respectively such that the sub-time step is small enough to guarantee convergence and higher accuracy.

Figure 2 shows the errors of results at $t = 2\pi$ (one full revolution) and the CPU time required. It can be seen that compared with the CV method, the DG method is more accurate but requires more computer memory and CPU time. For the CV method, the accuracy of results using the adaptive mesh scheme is very close to that using the fixed mesh (global mesh refinement) scheme while the CPU time required by the adaptive mesh scheme is reduced by ~~a factor of up to 10.~~ 75 % . For the DG method, to achieve a given level of accuracy of results, for example, ~~$E_1 = 0.0025$ and $E_2 = 0.025$~~ $E_1 = 0.0055$ and $E_2 = 0.035$, by using adaptive meshes, the CPU time can be reduced by ~~85 %~~ 45 % of that required using fixed meshes. ~~The CPU time required using the adaptive mesh schemes is much less than that using the fixed mesh schemes due to the significant reduction of the number of nodes~~ With increasing mesh resolution the CPU time for the adaptive schemes increase at a much slower rate than those for the fixed (global mesh refinement) approach (see Fig. 3 and Table 1). Compared with that in the fixed mesh (global mesh refinement) schemes, the problem size is reduced by 68–97.7 % using the adaptive mesh schemes. Hence, the use of adaptive meshes provides an efficient approach to lower the storage requirement, thus leading to the reduction of the overall computing time while remaining the accuracy of numerical results. To estimate the rate of convergence, the order of accuracy is calculated in Eq. (26 (here, $h = 1/200$). The order of accuracy is {0.83, 0.54, 0.95, 0.72} for {CV_Fix, CV_Adapt, DG_Fix, DG_Adapt} schemes respectively. It is argued that no-smooth profiles in the complex problems presented here, lead to a low order of accuracy, that is, a low convergence rate. If we only consider the hump-smooth profile as the initial data, the order of accuracy can increase to be {1.98, 1.52, 1.54, 1.13}.

Figure 4 shows the numerical results at $t = 2\pi$ (after one full revolution) using the adaptive and fixed mesh schemes. For comparison purpose, the FEM_Fix_1 and FEM_Adapt_4 schemes are chosen since the number of nodes in these two mesh schemes are almost same, where $N = 10\,201$ for FEM_Fix_1 scheme while $N \approx 11\,500$ for FEM_Adapt_4 scheme. The solutions of CV_Fix_1 and DG_Fix_1 are computed on a structured uniform

mesh of triangular elements with mesh size $h = 1/100$ and $\Delta t = 0.01$. It can be seen that there is severe erosion of the slotted cylinder when the fixed mesh scheme is adopted. The adaptive mesh scheme provides an improvement in accuracy of results. It is shown that with use of adaptive meshes (especially DG Adapt_Adapt_4), the initial shape of bodies is preserved well.

5.2 Case two: swirling flow

The capability of the adaptive mesh model has been further demonstrated in modelling swirling flow phenomena. The set up of the simulation in this case is similar with case one, however the velocity field is provided by the formula (LeVeque, 1996; Kuzmin, 2009):

$$\boldsymbol{u}(x,y,t) = (\sin^2(\pi x)\sin(2\pi y)g(t), -\sin^2(\pi y)\sin(2\pi x)g(t)) \tag{32}$$

where $g(t) = \cos(\pi t/T)$ on the time interval $0 \le t \le T$ (here $T = 1$).

The initial mass distribution will be deformed by the time-dependent velocity field which gradually slows down to zero and reverses its direction at $t = T/2$. Thus, the initial profile will be reproduced at the final time $t = T$ as depicted in Fig. 1. Due to the flow here is time-variable, the time step is set small enough to be $\Delta t = 0.00125$ for all different mesh schemes.

A comparison of results using fixed and adaptive meshes is illustrated in Figs. 5–8. Again, it can be observed that by using the adaptive mesh scheme in the model, both the CPU time and number of nodes required are significantly reduced for a given level of accuracy of results (see Fig. 6). To improve the stability of solutions when the mesh resolution is increased, the explicit advection subcycling based upon a CFL criterion is used for DG discretization while the Crank–Nicolson scheme for CV discretization. In this case, the order of accuracy is {0.81, 0.68, 0.92, 0.79} for {CV_Fix, CV_Adapt, DG_Fix, DG_Adapt} schemes respectively. Again the convergence rate is low due to non-smooth profiles in solutions.

The numerical solutions in Figs. 7 and 8 (at time levels $t = T/2$ and T) were computed by different fixed and adaptive mesh schemes. Again adaptive mesh modelling is able to

present better deformation of shapes at $t = T/2$ (Fig. 7) and preserve the initial shape after one full revolution ($t = T$) much better than fixed mesh modelling Fig. 8.

Figure 9 displays the change of adaptive meshes as time evolves. It is observed the dynamic mesh adaptation algorithm is capable of following the evolution details of transient flows. As the simulation progresses, the mesh has to be adapted not only to the current solution profile but also to its expected shape in the future. It can be seen that the mesh is adapted to capture the details of local flows, i.e, increasing the resolution around the shape's boundary with anisotropic elements and then capturing the shape of deformed bodies.

5.3 Case three: swirling deformation

A comparison of the anisotropic adaptive mesh schemes with the Walcek (or Bott) scheme (Walcek and Aleksic, 1998; Bott, 2010) adopted by many air quality models has been undertaken in this section. The case used here was described in Staniforth et al. (1987). In this case, we only focus on the subdomain $[0.24, 0.76] \times [0.12, 0.88]$ of Ω. A cone is initially centered at $(x_0, y_0) = (0.5, 0.5)$ with a negative (-0.2) background as shown in Fig. 10 and its geometry is given by

$$c(x, y, 0) = 1.2(1 - r(x, y)) - 0.2. \tag{33}$$

The velocity field defined by the following stream function (Staniforth et al., 1987):

$$\psi(x, y) = A \sin(kx) \cos(ky) \tag{34}$$

with

$$\boldsymbol{u}(x, y) = (-\psi_y, \psi_x) = (\, Ak \sin(kx) \sin(ky),\, Ak \cos(kx) \cos(ky)) \tag{35}$$

where $A = 0.08, k = 4\pi$.

The results using anisotropic adaptive meshes have been compared with those using Walcek scheme presented in Walcek and Aleksic (1998) . The analytical solutions were

depicted in (Staniforth et al., 1987) Staniforth et al. (1987) defined two flow regimes (short time periods and long time periods) that have different evaluation criteria for the numerical advection schemes. Here, we focus on the evaluation of the first regime (short time periods) so that the numerical solutions should be compared with the analytical solutions in a qualitative manner. Figures 11 and 12 show the comparison of three different schemes' results with the analytical solution at time $t = 3T/20$ and $t = T/5$, where $T = 2.6376$. The solutions of the Walcek scheme were computed on a structured uniform mesh with $h = 1/200$ and $\Delta t = 0.003297$. For FEM_Adapt_128 schemes (see Table 1), they were computed on dynamic adaptive mesh with constant $\Delta t = 0.006594$. The minimum mesh size is 7.8125×10^{-6} while the maximum mesh size is 0.2.

It can be observed the initial c field is splited into two rotations within the areas of the two central vertices as time evolves. Since the spatial gradient of solutions increases as time evolves especially at the boundaries of the central vortices, high resolution of meshes around the boundaries is needed to present the sharp shape accurately. Due to lack of high resolution of meshes, the solutions using the Walcek scheme fail to represent the analytical one and maintain the shape distribution. For the Walcek scheme, at time $t = 3T/20$ (see Fig. 11), the gradients of numerical distribution begin to disappear at the upper and middle boundaries of the central vortices and nearly completely disappear at time $t = T/5$ (Fig. 12). By adapting the mesh in time and space, the mesh resolution increases around the boundaries of each vertice, thus improving the accuracy of results. There is close agreement between the adaptive mesh modelling results and the analytical ones although the gradients for CV_Adapt_128 scheme are not as strong as the exact solution.

The sequence of triangulations presented in Fig. 13 demonstrates that the dynamic mesh adaptation algorithm succeeds in locally refining the mesh in the vicinity of steep fronts so as to reduce the amount of numerical diffusion and follow steep fronts as time evolves. To further reduce the number of elements, the anisotropic adaptive algorithm has been used for all the above adaptive mesh scheme, allowing the mesh is adapted along different directions. As shown in Fig. 14 which depicts a closeup view of locally adapted mesh, the adapted mesh size across the boundaries is small enough to capture the sharp fronts

while large along the boundaries since the c field does not change much. Therefore, the mesh sizing desired in anisotropic adaptive algorithm is not only a function of space, it is also a function of direction. At a given point, the desired mesh sizing differs in different directions.

Figure 15 shows the number of nodes required for CV_Adapt_128 scheme is less than the node number (~~40,000~~15,808) for fixed Walcek scheme during most of the simulation period. However, as local mesh resolution increases with time, the max CFL number of CV_Adapt_128 scheme exceeds ~~unit.~~ 10 and even reach 80. To keep the stabilization of solutions, the ~~Crank–Nicolson scheme is adopted here~~timestepping θ scheme is used to eliminate the time-step restrictions and maintain high accuracy as far as possible, where θ_v $(1/2 \leq \theta_v \leq 1)$ is chosen to 0.5 for most of elements while big enough(close to 1) for a small fraction of individual elements with a large CFL number (see Fig. 16). In this way, the use of a large time step is acceptable when applying adaptive mesh techniques into comprehensive air quality models, which can make the computation much more efficient. As shown in Fig. ~~15~~11–12, in combination with the ~~Crank–Nicolson method~~timestepping scheme, the adaptive mesh CV modelling solutions can maintain stable and accurate without reducing the time step size even if the max CFL number of CV_Adapt_128 exceeds 80. All of these can further illustrate the efficiency and the potential of dynamic mesh adaptation for future real applications in air quality model.

5.4 Case four: power plant plumes

In this case, the anisotropic adaptive mesh model is applied to an advection-diffusion problem (Eq. (2)): atmospheric dispersion of emissions from power plants. This is a first step towards applying the adaptive mesh model to realistic cases. The SO_2 emission of power plants was obtained from the Regional Emission inventory in ASia (REAS 2.1) data developed by National Institute of Environmental Sciences of Japan. As shown in Fig. 17 (a), the simulated domain covers the whole Shanxi-Hebei-Shandong-Henan region of China with $1090\,km \times 1060\,km$, and there are about 100 power plants in this area. The meteorological fields are provided by the mesoscale meteorological model WRF (v3.5) with

a horizontal resolution of $5\,km \times 5\,km$ and 20 vertical sigma layers. The simulation started at 00:00 UTC on 10 January 2013 and ran through to the 15 January 2013. In this case, the CV method is used for simulation of power plant plumes.

We started with a numerical investigation of a simplified 2D test. The mixing layer height is $600\,m$ and the turbulent horizontal diffusivity is $100\,m^2/s$. The horizontal wind fields are obtained by averaging the lowest five layers of WRF's meteorological fields and stored at hourly intervals during 5-day period. For fixed mesh schemes, three mesh resolution levels in horizontal are used: $10\,km \times 10\,km$ (level 1), $5\,km \times 5\,km$ (level 2) and $2.5\,km \times 2.5\,km$ (level 3). For coarse meshes (level 1), there are 110×107 nodes and 23 108 elements. The total number of fixed elements increases by a factor of 4 when doubling the horizontal mesh resolution. For adapt mesh schemes, the minimum (maximum) mesh size is set to be $2km$ ($30km$), and the maximum number of nodes is set to be 12000 which is the same as that of the fixed coarse mesh scheme (level 1). To represent the emission sources accurately, the fixed mesh with a high resolution of $2\,km$ is used around the power plant points within a radius of $6\,km$ (see Fig. 17 (b)).

Figure 18 shows SO_2 concentrations at 21:00 UTC on 12 January after spin-up of simulations. An artificial dilution effect can be seen when coarse meshes are used in modelling. This can be improved by increasing the mesh resolution or applying an adaptive mesh scheme. The results using adaptive meshes are in agreement with those using fixed meshes with a high mesh resolution of $2.5\,km$ while the number of nodes decreases by a factor of 16 with use of adaptive meshes. The evolution of adaptive meshes displayed in Fig. 19 illustrates that the adaptive algorithm is able to capture not only the detailed small-scale plume structures near the point sources, but also the regional high concentrations at large downwind distances.

To further demonstrate the adaptive mesh model's ability in 3D modelling, we extended the above 2D case to 3D dispersion of plumes. According to the terrain data of the modeling domain, the initial 2D adaptive mesh (see Fig. 17 (b)) can be extruded to create a layered 3D mesh from the top $20km$ (above sea level) to the terrain surface, with 11 terrain-following layers. There are seven layers within the lowest $1km$ above the terrain surface (see Fig. 20

(a)). The power plant emissions were injected into the third layer about $200m$ above the surface. Similarly, the three-dimensional velocity fields produced by WRF were interpolated from the fixed mesh in WRF onto the adaptive mesh. The vertical eddy diffusivity is parameterized based on a scheme by Byun and Dennis (1995) . Figure 20 shows the evolution of 3D SO_2 concentrations visualization which includes surface concentrations and the corresponding adaptive mesh, as well as the 3D pollutant plumes defined as a constant concentration surface for concentrations greater than $100\mu g/m^3$. It can be seen that full 3D mesh adaptivity has been used to improve the ability of the model to capture the details of flow dynamics and follow the evolution of power plant plumes.

6 Conclusions

In this paper, a new anisotropic adaptive mesh technique has been introduced and applied to modelling of multi-scale transport phenomena, which is a central component in air quality modelling systems. The first two benchmark test cases using the fixed mesh and adapted mesh schemes have been setup to illustrate the efficiency and accuracy of anisotropic adaptive mesh technique, which is an important means to improve the competitiveness of unstructured mesh air quality models. The ~~last~~ third case presents the irreplaceable advantage of this new adaptive mesh method to reveal detailed small scale plume structure (large gradients) that cannot be resolved with static grids, using comparable computational resources. Dispersion of power plant plumes, as a real model problem, has been simulated in the last case to illustrate that the adaptive algorithm is able to capture the detailed small-scale plume structures near each point source as well as the regional high concentrations at large downwind distances.

It is demonstrated that the dynamic anisotropic adaptive mesh technique can be used to automatically adapt the mesh resolution to follow the evolving pollutant and transient flow features in time and space, thus reducing the CPU time and memory requirement significantly. In combination with the timestepping θ scheme based on the Crank–Nicolson method, the adaptive mesh air pollution model is able to maintain the stability and accu-

racy of results without reducing the time step size when the minimum mesh size is getting smaller. This is of great significance for the future applications in multiscale modeling.

The ~~last~~ third test case serves as a proof-of-concept to further illustrate the capability of anisotropic mesh adaptivity techniques. In this case, the swirling deformation flow exhibits very high aspect ratios (1000, for example), which means that the pollutant distribution can possess very strong anisotropies as time evolves. Hence, the anisotropic mesh adaptation provides a very useful and effective way to simulate and represent this special atmospheric phenomena.

In summary, the results obtained in this work show the capability and potential of adaptive mesh methods to simulate multiscale air pollutant transport problems (spanning a range of scales) with higher numerical accuracy. The mesh adaptation can be used to improve the mesh resolution when and where it is needed without performing successive global refinement which is prohibitively expensive, and therefore, not feasible for realistic applications. Future work will consider ~~emissions and meteorological data as inputs~~ chemical reactions to further demonstrate the capability of dynamic adaptive mesh techniques.

Code availability

Fluidity code developed by the Applied Modelling and Computation Group (AMCG) at Imperial College London is available under the GNU General Public License (https://github.com/FluidityProject/fluidity). The user manual and examples are also available.

Acknowledgements. This work was carried out under funding from the Chinese Academy of Sciences (CAS) Strategic Priority Research Program (Grant No. XDB05030200), the UK's Natural Environment Research Council (projects NER/A/S/2003/00595,NE/C52101X/1 and NE/C51829X/1), the Engineering and Physical Sciences Research Council (GR/R60898, EP/I00405X/1 and EP/J002011/1) and the Imperial College High Performance Computing Service. The authors would like to thanks to J. Percival for many helpful discussions and support from others in AMCG. The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under Grant agreement n 603663 for the research project PEARL (Preparing for Extreme And Rare events in coastaL regions). Pain is greatful for the support of the EPSRC MEMPHIS multi-phase flow programme grant. The authors acknowledge the reviewers and

Editor for their in depth perspicacious comments that contributed to improving the presentation of this paper.

References

Alapaty, K., Mathur, R., and Odman, T.: Intercomparison of spatial interpolation schemes for use in nested grid models, Mon. Weather Rev., 126, 243–249, 1998.

AMCG: Fluidity Manual, Applied Modelling and Computation Group, Imperial College London, 2014.

Baker, C., Buchan, A., Pain, C., Farrell, P., Eaton, M., and Warner, P.: Multimesh anisotropic adaptivity for the Boltzmann transport equation, Ann. Nucl. Energy, 53, 411–426, 2013.

Behrens, J.: Adaptive Atmospheric Modeling: Key Techniques in Grid Generation, Data Structures, and Numerical Operations with Applications, Springer Science & Business Media, Vol. 54, 1–22, 2007.

Bott, A.: A positive definite advection scheme obtained by nonlinear renormalization of the advective fluxes, Mon. Weather Rev., 117, 1006–1016, 1989.

Bott, A.: The monotone area-preserving flux-form advection algorithm: reducing the time-splitting error in two-dimensional flow fields, Mon. Weather Rev., 121, 2637–2641, 1993.

Bott, A.: Improving the time-splitting errors of one-dimensional advection schemes in multidimensional applications, Atmos. Res., 97, 619–631, 2010.

Burbeau, A., Sagaut, P., and Bruneau, C.-H.: A problem-independent limiter for high-order Runge–Kutta discontinuous Galerkin methods, J. Comput. Phys., 169, 111–150, 2001.

Byun, D. W. and Dennis, R.: Design Artifacts in Eulerian Air Quality Models – Evaluation of the Effects of Layer Thickness and Vertical Profile Correction on Surface Ozone Concentrations, Atmos. Environ., 29, 105–126, 1995.

Ciarlet, P. G. and Lions, J. L.: Handbook of Numerical Analysis, Volume 7, Techniques of Scientific Computing (Part 3), pages 713-1020. Elsevier, 2000. 55

Cockburn, B. and Shu, C.-W W.: Runge–Kutta discontinuous Galerkin methods for convection-dominated problems, J. Sci. Comput., 16, 173–261, 2001.

Cockburn, B., Karniadakis, G. E., and Shu, C.-W W.: The Development of Discontinuous Galerkin Methods, Springer, 1–20, 2000.

Colella, P. and Woodward, P. R.: The piecewise parabolic method (PPM) for gas-dynamical simulations, J. Comput. Phys., 54, 174–201, 1984.

Constantinescu, E. M., Sandu, A., and Carmichael, G. R.: Modeling atmospheric chemistry and transport with dynamic adaptive resolution, Computat. Geosci., 12, 133–151, 2008.
Debreu, L. and Blayo, E.: Two-way embedding algorithms: a review, Ocean Dynam., 58, 415–428, 2008.
Donea, J., Huerta, A.: Finite element methods for flow problems. John Wiley Sons, 2003.
Fang, F., Pain, C., Navon, I., Gorman, G., Piggott, MD Allison, P., and Goddard, A.: A POD goal-oriented error measure for mesh optimization, Int. J. Numer. Meth. Fl., 63, 185–206, 2010.
Farrell, P., Piggott, M., Pain, C., Gorman, G., and Wilson, C.: Conservative interpolation between unstructured meshes via supermesh construction, Comput. Method. Appl. M., 198, 2632–2642, 2009.
Flaherty, J. E., Krivodonova, L., Remacle, J.-F., and Shephard, M. S.: Aspects of discontinuous Galerkin methods for hyperbolic conservation laws, Finite Elem. Anal. Des., 38, 889–908, 2002.
Frohn, L. M., Christensen, J. H., and Brandt, J.: Development of a high-resolution nested air pollution model: the numerical approach, J. Comput. Phys., 179, 68–94, 2002.
Garcia-Menendez, F. and Odman, M. T.: Adaptive grid use in air quality modeling, Atmosphere, 2, 484–509, 2011.
Hesthaven, J. S. and Warburton, T.: Nodal discontinuous Galerkin methods: algorithms, analysis, and applications, Vol. 54, Springer, 2007.
Hoteit, H., Ackerer, P., Mosé, R., Erhel, J., and Philippe, B.: New two-dimensional slope limiters for discontinuous Galerkin methods on arbitrary meshes, Int. J. Numer. Meth. Eng., 61, 2566–2593, 2004.
Krivodonova, L.: Limiters for high-order discontinuous Galerkin methods, J. Comput. Phys., 226, 879–896, 2007.
Krivodonova, L., Xin, J., Remacle, J.-F. F., Chevaugeon, N., and Flaherty, J. E.: Shock detection and limiting with discontinuous Galerkin methods for hyperbolic conservation laws, Appl. Numer. Math., 48, 323–338, 2004.
Kühnlein, C.: Solution-adaptive moving mesh solver for geophysical flows, PhD thesis, lmu, Ludwig-Maximilians-Universität, München, 1–14, 2011.
Kühnlein, C., Smolarkiewicz, P. K., and Dörnbrack, A.: Modelling atmospheric flows with adaptive moving meshes, J. Comput. Phys., 231, 2741–2763, 2012.
Kuzmin, D.: Explicit and implicit FEM-FCT algorithms with flux linearization, J. Comput. Phys., 228, 2517–2534, 2009.

Kuzmin, D.: A vertex-based hierarchical slope limiter for p-adaptive discontinuous Galerkin methods, J. Comput. Appl. Math., 233, 3077–3085, 2010.
Lagzi, I., Turányi, T., Tomlin, A. S., and Haszpra, L.: Modelling photochemical air pollutant formation in Hungary using an adaptive grid technique, Int. J. Environ. Pollut., 36, 44–58, 2009.
Leonard, B.: The ULTIMATE conservative difference scheme applied to unsteady one-dimensional advection, Comput. Method. Appl. M., 88, 17–74, 1991.
LeVeque, R. J.: High-resolution conservative algorithms for advection in incompressible flow, SIAM J. Numer. Anal., 33, 627–665, 1996.
LeVeque, R. J.: Finite Volume Methods for Hyperbolic Problems, Vol. 31, Cambridge University Press, 2002.
Luo, H., Baum, J. D., and Löhner, R.: A Hermite WENO-based limiter for discontinuous Galerkin method on unstructured grids, J. Comput. Phys., 225, 686–713, 2007.
Nikiforakis, N.: Mesh generation and mesh adaptation for large-scale Earth-system modelling, Philos. T. R. Soc. A, 367, 4473–4481, 2009.
Pain, C., Umpleby, A., De Oliveira, C., and Goddard, A.: Tetrahedral mesh optimisation and adaptivity for steady-state and transient finite element calculations, Comput. Method. Appl. M., 190, 3771–3796, 2001.
Pain, C., Piggott, M., Goddard, A., Fang, F., Gorman, G., Marshall, D., Eaton, M., Power, P., and De Oliveira, C.: Three-dimensional unstructured mesh ocean modelling, Ocean Model., 10, 5–33, 2005.
Pavlidis, D., Gomes, J. L., Xie, Z.: Compressive advection and multi-component methods for interface-capturing. International Journal for Numerical Methods in Fluids, 2015.
Piggott, M., Pain, C., Gorman, G., Power, P., and Goddard, A.: h, r, and hr adaptivity with applications in numerical ocean modelling, Ocean Model., 10, 95–113, 2005.
Piggott, M., Gorman, G., Pain, C., Allison, P., Candy, A., Martin, B., and Wells, M.: A new computational framework for multi-scale ocean modelling based on adapting unstructured meshes, Int. J. Numer. Meth. Fl., 56, 1003–1015, 2008a.
Piggott, M. D., Pain, C. C., Gorman, G. J., Marshall, D. P., and Killworth, P. D.: Unstructured adaptive meshes for ocean modeling, Geoph. Monog. Series, 177, 383–408, 2008b.
Piggott, M., Farrell, P., Wilson, C., Gorman, G., and Pain, C.: Anisotropic mesh adaptivity for multi-scale ocean modelling, Philos. T. R. Soc. A, 367, 4591–4611, 2009.
Saad, Y.: A flexible inner-outer preconditioned GMRES algorithm, SIAM J. Sci. Comput., 14, 461–469, 1993.

Srivastava, R. K., McRae, D., and Odman, M.: An adaptive grid algorithm for air-quality modeling, J. Comput. Phys., 165, 437–472, 2000.

Staniforth, A., Côté, J., and Pudykjewicz, J.: Comments on "Swolarkiewicz's Deformational Flow", Mon. Weather Rev., 115, 894–900, 1987.

Sweby, P. K.: High resolution schemes using flux limiters for hyperbolic conservation laws, SIAM J. Numer. Anal., 21, 995–1011, 1984.

Versteeg, H. K., Malalasekera, W.: An introduction to computational fluid dynamics: the finite volume method. Pearson Education, 243-266, 2007.

Walcek, C. J. and Aleksic, N. M.: A simple but accurate mass conservative, peak-preserving, mixing ratio bounded advection algorithm with FORTRAN code, Atmos. Environ., 32, 3863–3880, 1998.

Wang, Y.: An explicit simulation of Tropical cyclones with a triply nested movable mesh primitive equation model: TCM3. Part I: Model description and control experiment, Mon. Weather Rev., 129, 1370–1394, 2001.

Waterson, N. P. and Deconinck, H.: Design principles for bounded higher-order convection schemes–a unified approach, J. Comput. Phys., 224, 182–207, 2007.

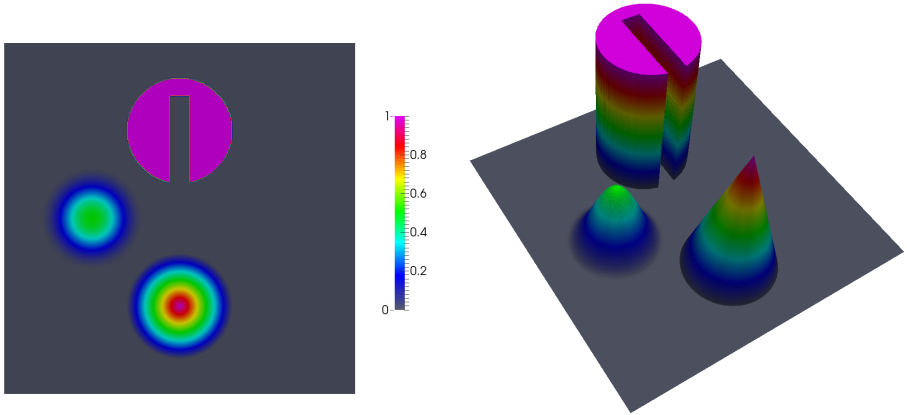
Weller, H., Ringler, T., Piggott, M., and Wood, N.: Challenges facing adaptive mesh modeling of the atmosphere and ocean, B. Am. Meteorol. Soc., 91, 105–108, 2010.

Wilson, C.: Modelling multiple-material flows on adaptive unstructured meshes, PhD thesis, Imperial College London, 2009.

Zhang, D.~~L~~ L., Chang, H.~~R~~ R., Seaman, N. L., Warner, T. T., and Fritsch, J. M.: A two-way interactive nesting procedure with variable terrain resolution, Mon. Weather Rev., 114, 1330–1339, 1986.

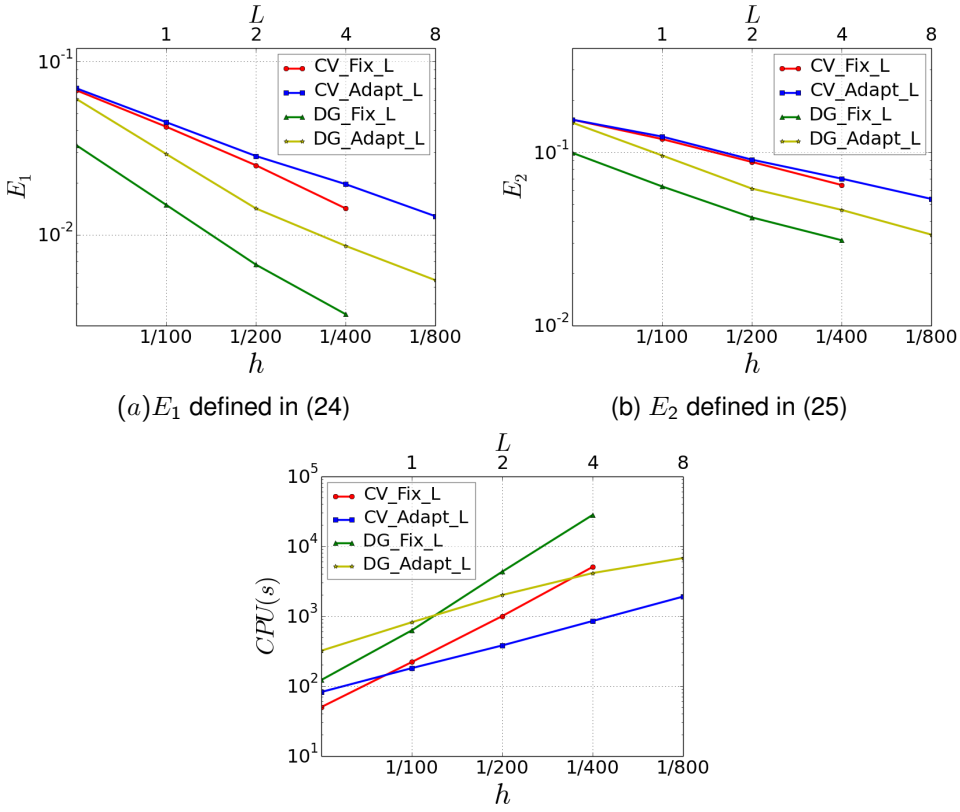
Table 1. Basic configuration for FEM_Adapt_L and FEM_Fix_L schemes (where FEM represents CV or DG; the maximum mesh size is set to be 0.2).

Mesh schemes (L)		1	2	4	8	128
Minimum mesh size (h)		0.01	0.005	0.0025	0.00125	7.8125×10^{-6}
The maximum number of nodes	FEM_Adapt_L	3500	7500	15 000	25 000	4515 000
The number of nodes	FEM_Fix_L	10 201	40 401	160 801	641 601	163 865 601



(a) 2D (b) 3D

Figure 1. Initial distribution/exact solution at $t = 2\pi$ in ~~2-D~~ 2D and ~~3-D~~ 3D view.



$(a)E_1$ defined in (24)

(b) E_2 defined in (25)

(c) CPU time

Figure 2. Case one – solid body revolution: the errors in the c field solutions and the CPU time (as a function of the mesh size h) required for one revolution ($t = 2\pi$), where h is the mesh size for FEM_Fix_L schemes while the minimum mesh size for FEM_Adapt_L schemes~~, using the same $\Delta t = 0.01$~~.

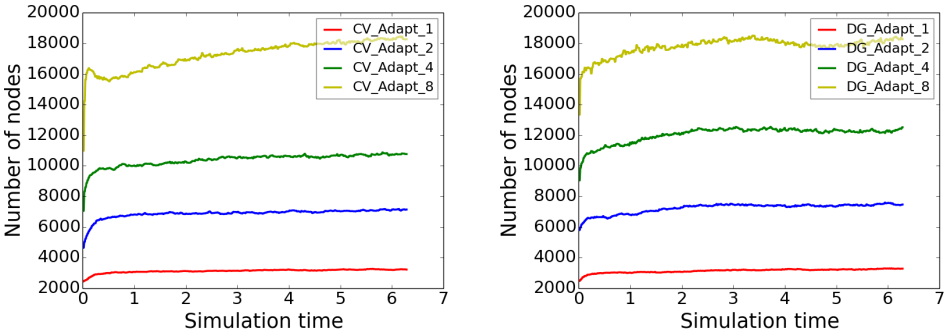
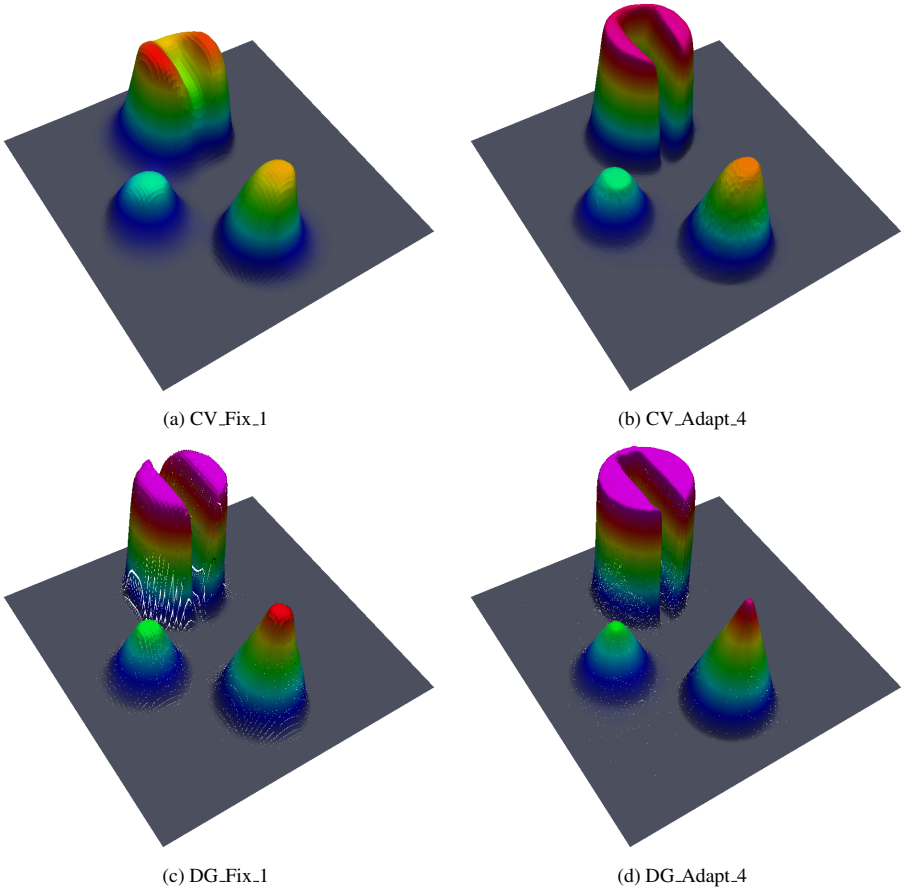


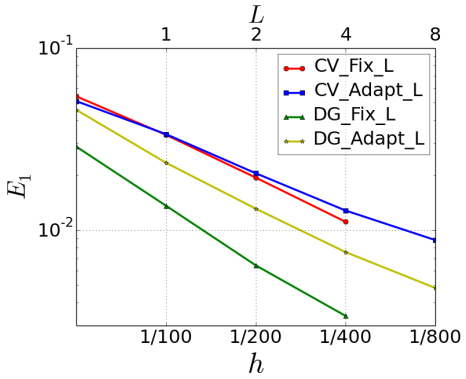
Figure 3. Case one – solid body revolution: the evolution of number of nodes for **(a)** CV_Adapt, **(b)** DG_Adapt.



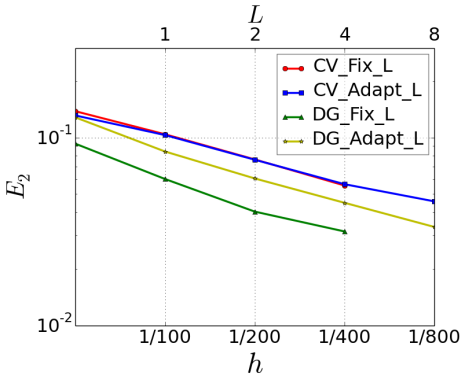
(a) CV_Fix_1 (b) CV_Adapt_4

(c) DG_Fix_1 (d) DG_Adapt_4

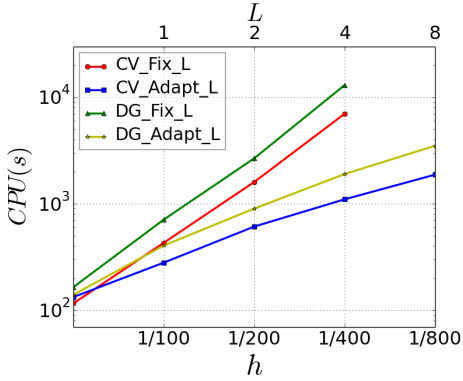
Figure 4. Case one – solid body revolution: the results from the fixed and adaptive mesh schemes using almost the same nodes number N, where $N = 10\,201$ for FEM_Fix_1 scheme while $N \approx 11\,500$ for FEM_Adapt_4 scheme, at $t = 2\pi$.



(a) E_1 defined in (24)

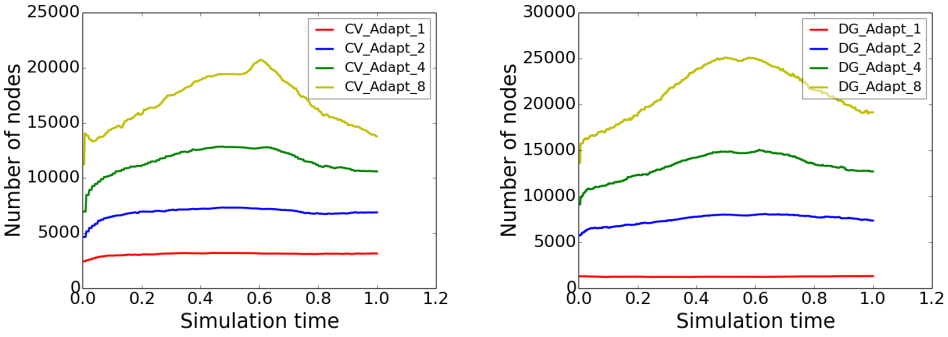


(b) E_2 defined in (25)



(c) CPU time

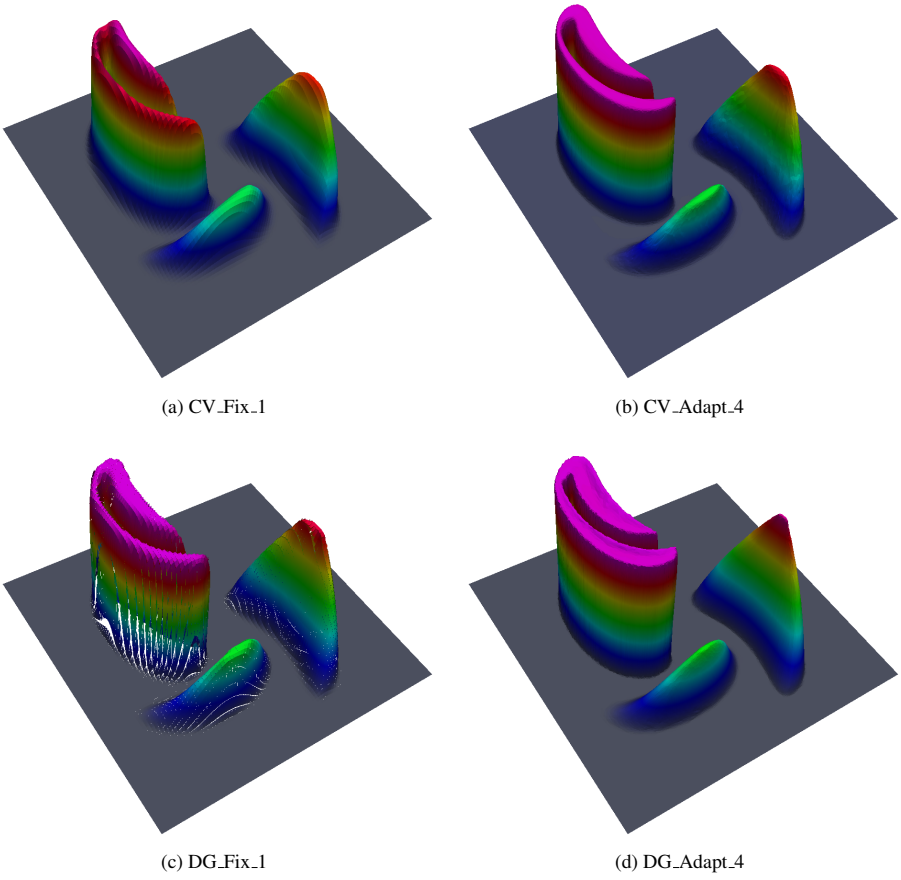
Figure 5. Case two – swirling flow: the errors in the c field solutions and the CPU time (as a function of the mesh size h) required for one revolution, where h is the mesh size for FEM_Fix_L schemes while the minimum mesh size for FEM_Adapt_L schemes, using the same ~~$\Delta t = 0.0025$~~$\Delta t = 0.00125$.



(a) CV_Adapt

$b) DG\_Adapt$

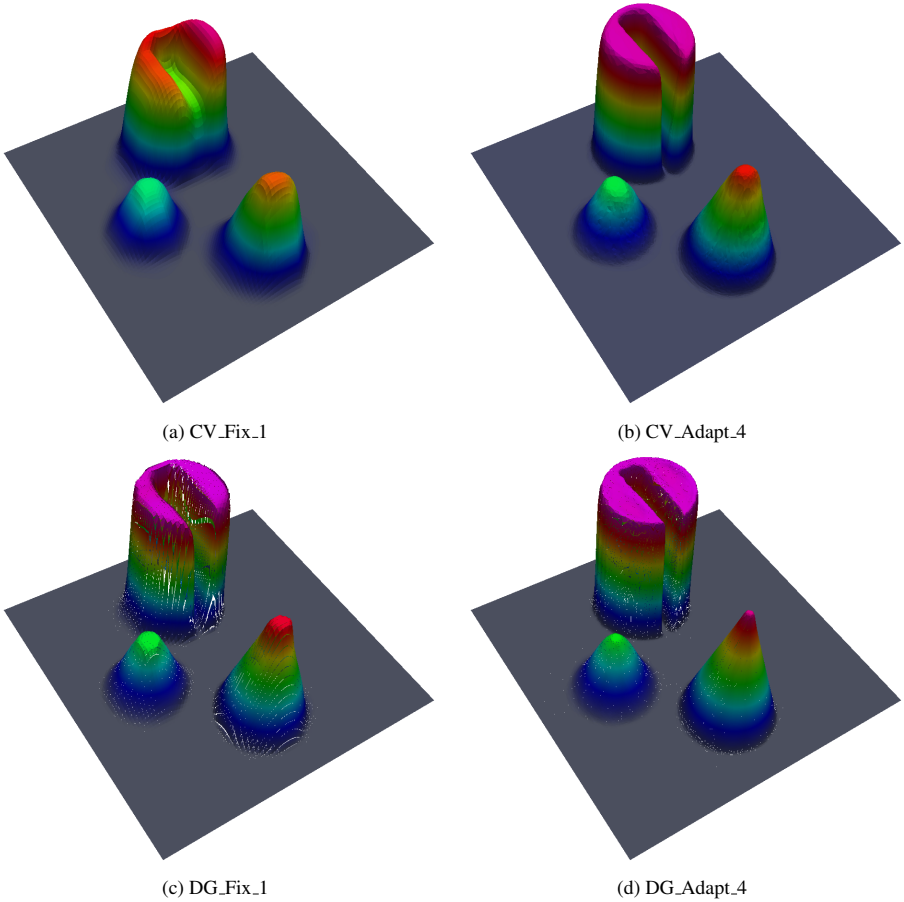
Figure 6. Case two – swirling flow: the evolution of number of nodes for **(a)** CV_Adapt, **(b)** -DG_Adapt.



(a) CV_Fix_1 (b) CV_Adapt_4

(c) DG_Fix_1 (d) DG_Adapt_4

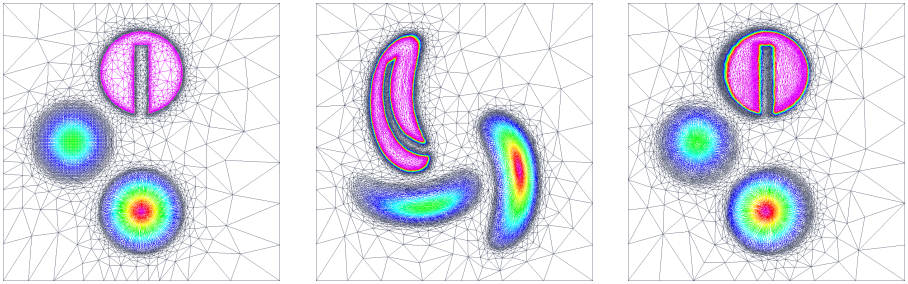
Figure 7. Case two – swirling flow: the results from the fixed and adaptive mesh schemes using almost the same nodes number N, where $N = 10\,201$ for FEM_Fix_1 scheme while $N \approx 12\,000$ for FEM_Adapt_4 scheme, at $t = T/2 (= 0.5)$.



(a) CV_Fix_1 (b) CV_Adapt_4

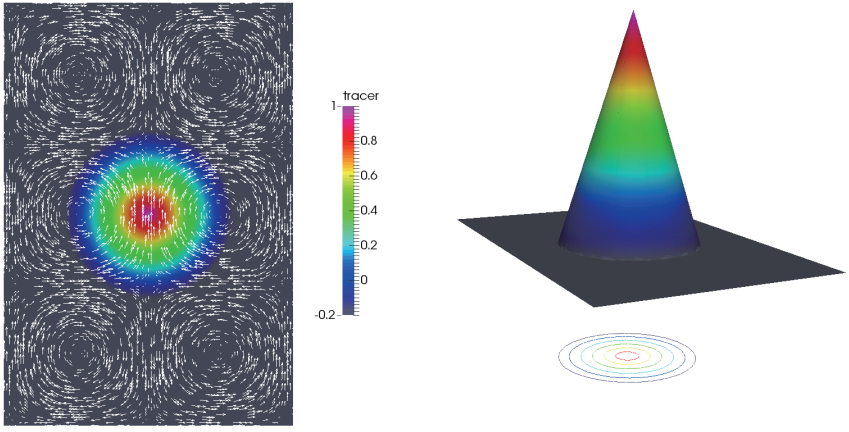
(c) DG_Fix_1 (d) DG_Adapt_4

Figure 8. Case two – swirling flow: the results from the fixed and adaptive mesh schemes using almost the same nodes number N, where $N = 10\,201$ for FEM_Fix_1 scheme while $N \approx 12\,000$ for FEM_Adapt_4 scheme, at $t = T(= 1)$.



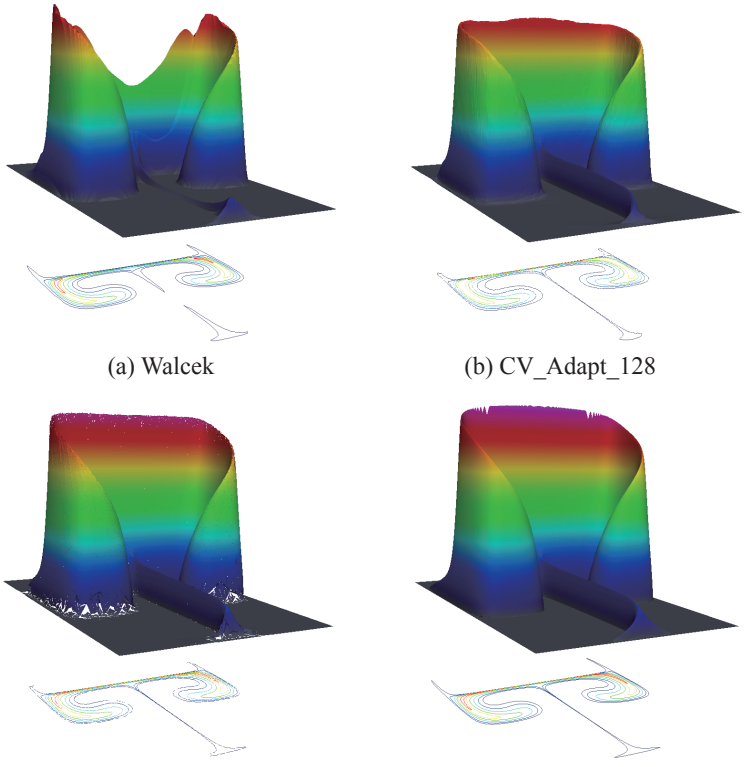
(a) $t = 0$ (b) $t = 0.5$ (c) $t = 1$

Figure 9. Case two – swirling flow: the evolution of the adaptive mesh colored with tracer value c, where DG_Adapt_4 scheme is used.



(a) 2D (b) 3D

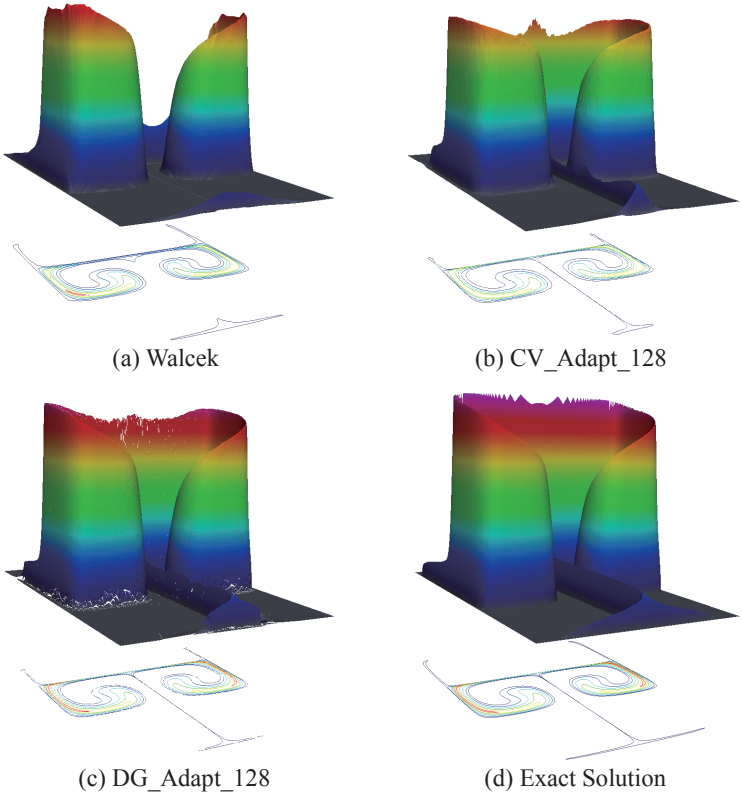
Figure 10. Case three – swirling deformation: initial distribution and velocity field.



(a) Walcek (b) CV_Adapt_128

(c) DG_Adapt_128 (d) Exact Solution

Figure 11. Case three – swirling deformation: comparison of the analytical solution with the results from different schemes using almost the same number of nodes ~~$N \approx 40\,000$~~ $N \approx 15\,000$, at $t = 3T/20$, where $T = 2.6376$.



(a) Walcek (b) CV_Adapt_128

(c) DG_Adapt_128 (d) Exact Solution

Figure 12. Case three – swirling deformation: comparison of the analytical solution with the results from different schemes using almost the same number of nodes ~~$N \approx 40\,000$~~$N \approx 15\,000$, $t = T/5$, where $T = 2.6376$.

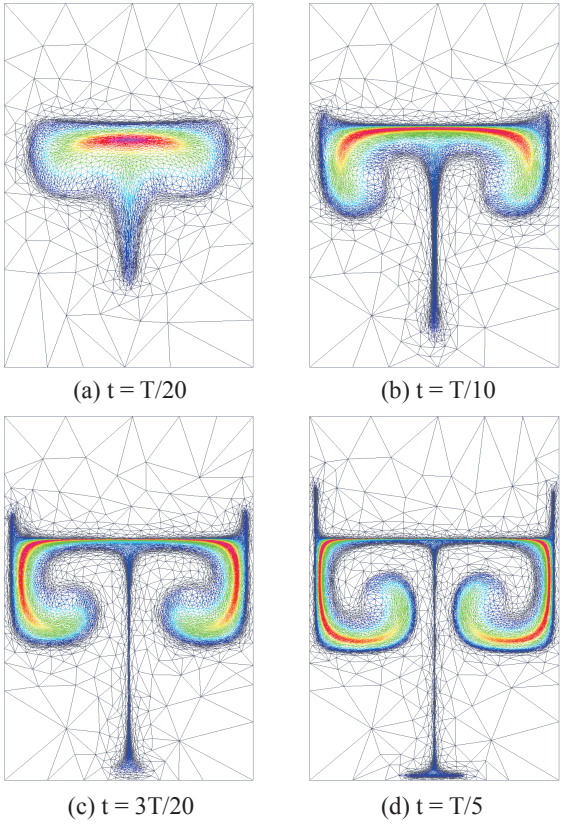
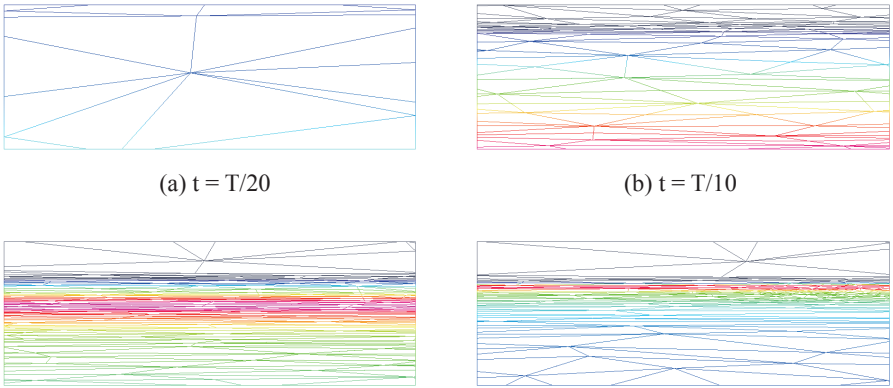


Figure 13. Case three – swirling deformation: the evolution of the adaptive mesh colored with tracer value c, where DG_Adapt_128 scheme is used.



(a) t = T/20 (b) t = T/10

(c) t = 3T/20 (d) t = T/5

Figure 14. Case three – swirling deformation: the evolution of the adaptive mesh colored with tracer value c, in the subdomain $[0.49, 0.51] \times [0.62, 0.627]$, using DG_Adapt_128 scheme.

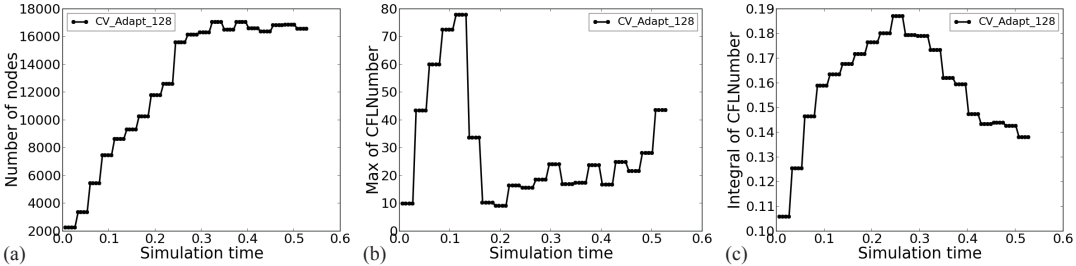
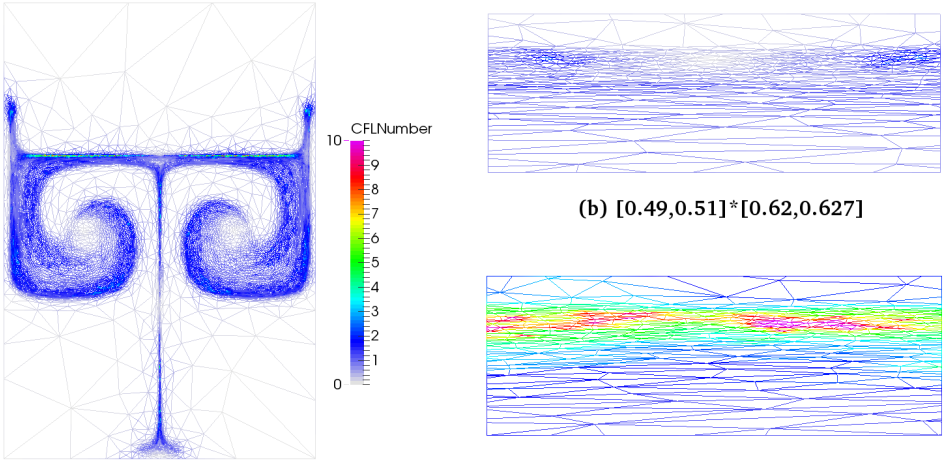


Figure 15. Case three – swirling deformation: the evolution of **(a)** number of nodes, **(b)** max local and **(c)** integral of CFL number for CV_Adapt_128 schemes.



(a) [0.24,0.76]*[0.12,0.88]

(b) [0.49,0.51]*[0.62,0.627]

(c) [0.365,0.385]*[0.62,0.627]

Figure 16. Case three – swirling deformation: the distribution of CFLNumber for CV_Adapt_128 scheme at $t = T/5$, where $T = 2.6376$.

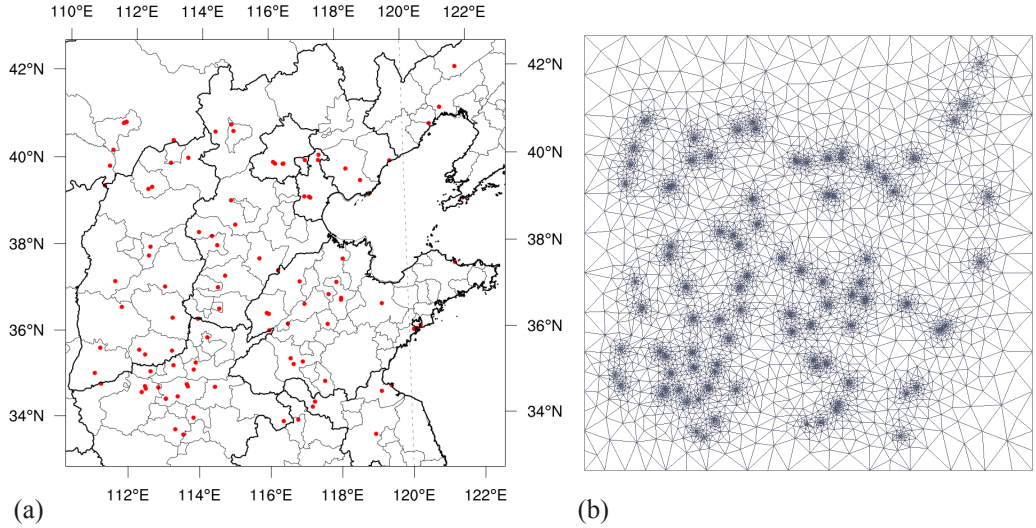


Figure 17. Case four – power plant plumes: **(a)** the distribution of power plants and **(b)** the corresponding initial mesh.

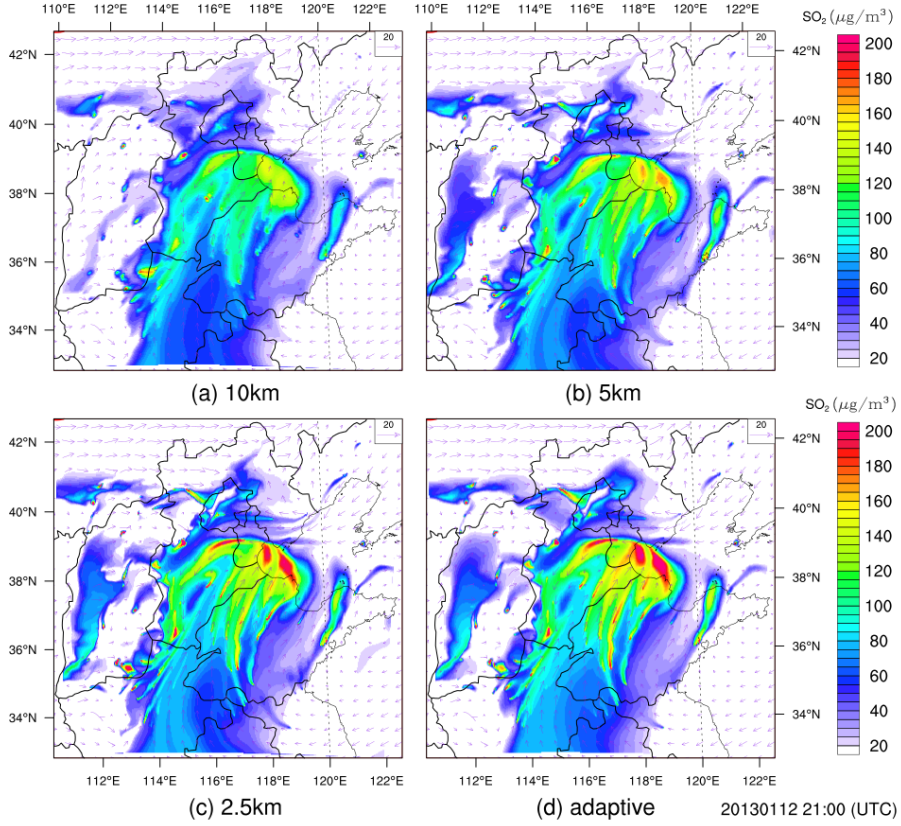
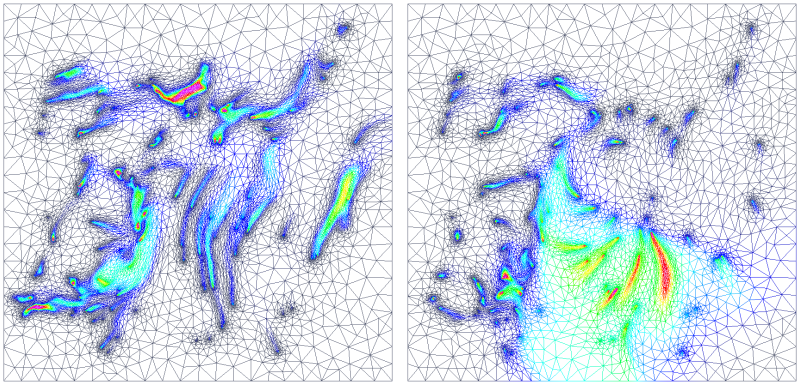
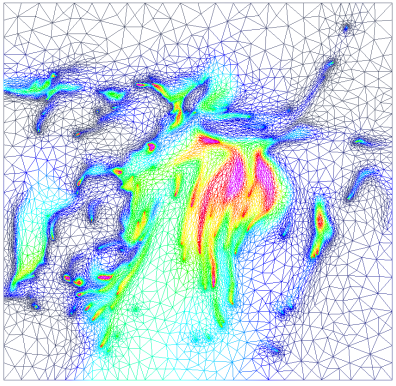


Figure 18. Case four – power plant plumes: simulated SO_2 concentrations ($\mu g/m^3$) at 21:00 UTC 12 January 2013 using the CV methods on different horizontal resolution of **(a)** 10km, **(b)** 5km, **(c)** 2.5km and on the **(d)** adaptive mesh.

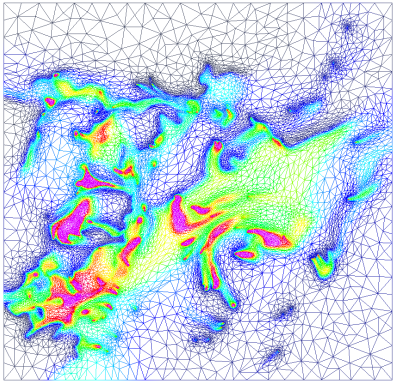


(a) 20130110 18:00(UTC)

(b) 20130111 18:00(UTC)

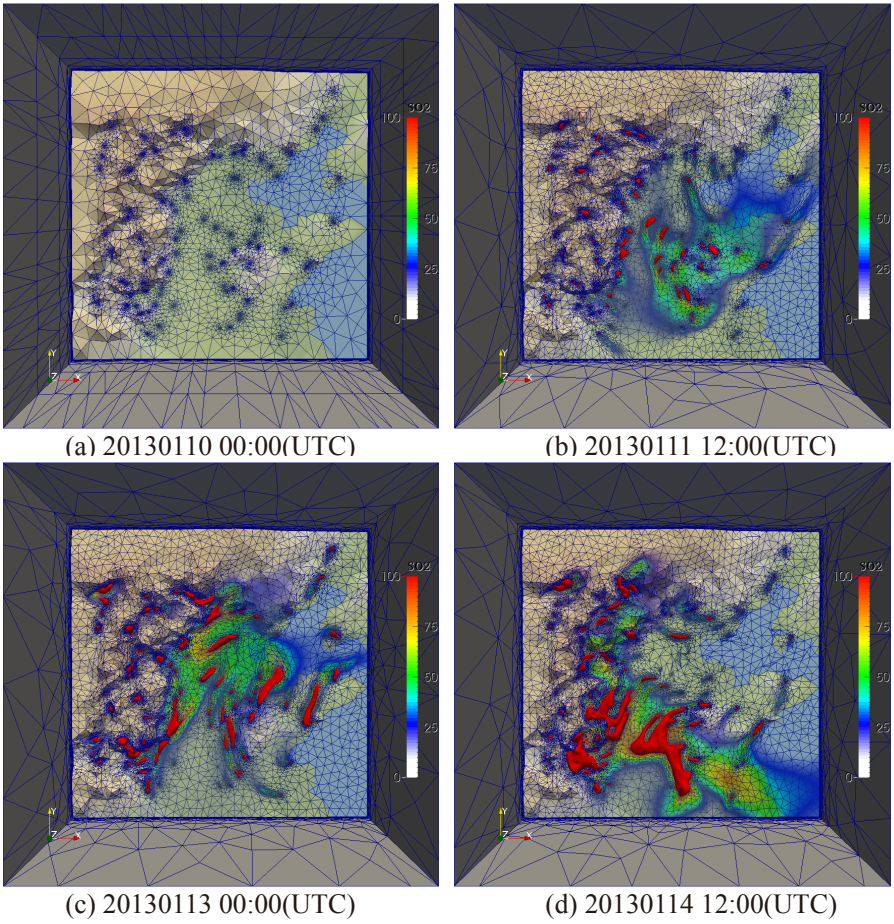


(c) 20130112 18:00(UTC)



(d) 20130113 18:00(UTC)

Figure 19. Case four – power plant plumes: the evolution of the adaptive mesh colored with SO_2 concentrations ($\mu g/m^3$), using the CV_Adapt scheme.



(a) 20130110 00:00(UTC) (b) 20130111 12:00(UTC)

(c) 20130113 00:00(UTC) (d) 20130114 12:00(UTC)

Figure 20. Case four – power plant plumes: the evolution of 3D plumes visualization, surface SO_2 concentrations ($\mu g/m^3$) and the corresponding adaptive mesh.