A look-up table method based on unstructured grids and its application to non-ideal compressible fluid dynamic simulations

Rubino, A.; Pini, M.; Kosec, M.; Vitale, S.; Colonna, P.

DOI
10.1016/j.jocs.2018.08.001

Publication date
2018

Document Version
Final published version

Published in
Journal of Computational Science

Citation (APA)

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright
Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy
Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.
A look-up table method based on unstructured grids and its application to non-ideal compressible fluid dynamic simulations

A. Rubino a, M. Pini a,∗, M. Kosec b, S. Vitale a, P. Colonna a

a Delft University of Technology, Aerospace Engineering Faculty, Propulsion and Power, Kruisplein 2, 2629 HS Delft, The Netherlands
b Stanford University, Aeronautics and Astronautics, Stanford, 94305 CA, United States

A R T I C L E   I N F O

Article history:
Received 28 May 2018
Received in revised form 18 July 2018
Accepted 1 August 2018
Available online 9 August 2018

Keywords:
Look-up table
Trapezoidal map
Unstructured mesh
Thermodynamic modeling
Interpolation
Real-gas flows
Non-ideal compressible flow
NICFD
CFD

A B S T R A C T

Fast and accurate computation of thermo-physical properties is essential in computationally expensive simulations involving fluid flows that significantly depart from the ideal gas or ideal liquid behavior. A look-up table algorithm based on unstructured grids is proposed and applied to non-ideal compressible fluid dynamics simulations. The algorithm grants the possibility of a fully automated generation of the tabulated thermodynamic region for any boundary and to use mesh refinement. Results show that the proposed algorithm leads to a computational cost reduction up to one order of magnitude, while retaining the same accuracy level compared to simulations based on more complex equation of state. Furthermore, a comparison of the LuT algorithm with a uniformly spaced quadrilateral tabulation method resulted in similar performance and accuracy.

© 2018 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

1. Introduction

The accurate estimation of thermo-physical properties of fluids is essential for many engineering and scientific applications, and it requires complex models in case the behavior of the fluid departs from that of the ideal gas or ideal liquid. Fluids exhibiting non-ideal behavior are involved in various technologies such as advanced power and propulsion systems, refrigeration and air conditioning systems, oil and gas processes, etc. [1–4]. In these cases, the evaluation of thermo-physical properties is often necessary for system design and performance evaluation or to simulate the flow behavior within components. Fluid dynamic simulations of vapors in non-ideal states are also employed in more fundamental research (see, e.g., Ref. [5]) and the branch of fluid mechanics dealing with this type of fluid flows was recently termed non-ideal compressible fluid dynamics (NICFD) [6].

The computational cost associated with fluid thermodynamic models expressed in terms of equations of state (EoS) can become a limiting factor if accurate estimations are needed in combination with expensive simulations. This is the case, e.g., in the design and optimization of industrial components [7] or in computational fluid dynamics (CFD) [8]. In order to decrease the computational time related to the calculation of thermo-physical fluid properties, while maintaining a satisfactory level of accuracy, look-up table (LuT) methods are convenient and widely adopted [9].

A LuT method consists of three basic parts: 1. The tabulation of a discrete set of values of thermodynamic properties pertaining to states generated with a given EoS-base model; 2. A search algorithm; 3. An interpolation method. Since the tabulation is performed only once, at preprocessing level, this way of evaluating fluid thermophysical properties can greatly reduce the computational effort if the models are based on complex EoS, provided that the associated search algorithm is efficient. Furthermore, the interpolation technique must be carefully chosen in order to achieve a satisfactory level of accuracy, which is a fundamental requirement to guarantee convergence in CFD simulations and accuracy of the final result. LuT methods based on a structured mesh of thermodynamic region of interest have been documented in the literature [10,11,9,12]. In order to increase the accuracy and decreasing the number of discretization points for thermodynamic region of interest, algorithms based on adaptive Cartesian mesh have been proposed [13,14]. However, the use of quadrilateral grids, in combination with local refinement, can lead to local discontinuities and poor interpolation accuracy of properties in the proximity of smooth boundaries [15,13]. This can especially occur for proper-
ties reconstruction or for interpolation close to the vapor–liquid saturation line.

The study described here resulted in ug-LUT, a new LuT method based on meshing thermodynamic region of interest by means of unstructured triangular grids. The generation of multidimensional thermodynamic tables is fully automated, even in case multiple fluid phases need to be computed. Unstructured grids allow for mesh refinements, a valuable feature in case of strong property variations, which occur in proximity of the vapor–liquid critical point, and of the saturation line in general. The ug-LUT method is applicable also to multi-component fluids. A search algorithm based on a trapezoidal map of the tabulated region contributes significantly to its computational efficiency. The ug-LUT method for thermo-physical property calculations is implemented within the open-source code SU2 [16–18] and its verification is presented by means of paradigmatic CFD test cases of increasing fidelity. Finally, the ug-LUT algorithm is compared to a structured-based LuT with the aim of assessing its computational cost and memory requirements.

2. Method

2.1. Generation of thermodynamic mesh

An unstructured mesh is generated for thermodynamic domain of interest. A 2D grid generator [19], based on the Advancing-Delaunay front method, is used in this study. The adopted grid generator allows for local refinements in selected regions of thermodynamic domain. Fig. 1 shows examples of thermodynamic meshes for siloxane MDM (octamethyltrisiloxane, C₈H₂₄O₂Si₃), generated by selecting T and log(ρ) as input state variables. The use of unstructured mesh in combination with local refinement is proposed here as an effective alternative to structured and quad-tree grids. Quad-tree based algorithms are efficient for controlling the mesh refinement level, but can suffer from the following issues [15,13]: (i) storing and retrieving the mesh connectivity associated to the recursive tree structure; (ii) hanging nodes at different sizes cells interfaces, if continuous property reconstruction is required; (iii) additional interpolation, triangulation or a curvilinear mesh system might be necessary to reconstruct smooth boundaries (e.g. vapor–liquid saturation line).

Once the mesh patches are generated using any suitable set of state variables (two if the fluid is pure), all other needed fluid thermo-physical properties are computed at each mesh node with an appropriate thermodynamic library. The current implementation of the ug-LUT method makes use of an external thermodynamic library [20], which embeds a large variety of models based on complex equations of state (EoS). As an example, Fig. 2 reports the pressure contour for siloxane MDM, obtained using a model based on an EoS in the Span–Wagner functional form [21].

2.2. Search algorithm

Once the set of thermo-physical properties for the selected fluid are stored in tabular form, a search algorithm is used to retrieve the best approximation of the query state or point. A point location algorithm based on a trapezoidal map has been adopted [22] because of the following considerations: (i) the same geometrical connectivity is used for all the search pairs. This is especially beneficial because the connectivity has not to be recomputed for each search pair and it can be used to search in highly skewed thermodynamic planes. As an example for this, if the initial thermodynamic mesh is built for the (P, ρ) plane, the resulting mesh on the (h, s) plane will be highly skewed; (ii) searching for the triangle containing the query vector by resorting to algorithms that do not use the mesh connectivity information (e.g. kd-tree) on an irregular and highly skewed grid can lead to inaccurate interpolation; (iii) trapezoidal maps work for general polyhedra. The search algorithm can be used to switch between different mesh zones, characterized by a polyhedron outline. This feature allows to avoid a mapping for grids not conforming with a rectangular thermodynamic domain [13].

Given the set S of n triangular mesh edges, the trapezoidal map T(S) is built according to the following steps (see Fig. 3):

1. a unique set of edges and the corresponding list of its x coordinate is created by filtering out duplicates;
2. the intersecting edges are associated to each band;
3. the edges in each band are sorted.

The trapezoidal map $T(S)$ is created in a preprocessing stage for each thermodynamic search pair, e.g., $(h, s), (\rho, v)$, etc. A more detailed description of the trapezoidal map algorithm can be found in Appendix A.

The mesh simplex, containing the query point $q$ within $T(S)$, is identified according to the following procedure:

1. the $x$ coordinate of the query point is obtained with a binary search for its containing band;
2. within the containing band, the edge above and the one below the query point are identified;
3. the simplex containing the query point is singled out by using the edge-to-face connectivity of the two edges selected during the previous step.

The query algorithm is also detailed in Appendix A.

2.3. Interpolation method

A two-dimensional linear interpolation problem can be written as

$$f(x, y) = \sum_{t=1}^{N} w_{t} G(x, y) = G^{T}(x, y) W,$$

where $G(x, y)$ is an interpolation basis which transforms the query coordinates $x, y$ (raw features) into $N$ linearly independent interpolation features. An example of a polynomial basis function $G(x, y)$ on a triangle (three points, $N=3$) is given by

$$G(x, y) = [1, x, y]^{T}.$$  

In the implemented thermodynamic look-up-table, the two-dimensional coordinates are thermodynamic pairs such as $(\rho, u), (P, T), (h, s)$. This choice of $G(x, y)$ was found to be sufficiently accurate, provided that a relatively fine mesh for the look-up-table is used. The vector of weights $W$ is found through

$$A = \begin{bmatrix} G^{T}(x_{1}, y_{1}) \\ \vdots \\ G^{T}(x_{N}, y_{N}) \end{bmatrix}, \quad F = \begin{bmatrix} f(x_{1}, y_{1}) \\ \vdots \\ f(x_{N}, y_{N}) \end{bmatrix}, \quad AW = F,$$

where $N$ is the number of sample points on which the interpolation is based and the $f(x_{i}, y_{i})$ are the known values. $F$ is an implicit function of the query point $(x_{q}, y_{q})$. The mapping from the query point to the forcing vector $F$ is given by the trapezoidal map algorithm, which provides the interpolation points.

The interpolation is computationally efficient when a single function is being interpolated at many different combinations of query point coordinates $(x_{q}, y_{q})$, as all the sample points can use the same weights.

However, the completion of thermodynamic look-up-table requires the interpolation of several functions (ten or more) for each query point. Thus, it is convenient to work out the dual form of the primal interpolation problem, such that the weights have to be computed only once for a given query point. The dual problem can be written as

$$G(x_{q}, y_{q})^{T} W = F^{T} V,$$

where the interpolation weights $V$ are now the adjoint solution of

$$A^{T} V = G(x_{q}, y_{q}).$$

The equivalence of the two formulations can be put into evidence by [23]

$$V^{T} F = V^{T} (A W) = (A^{T} V)^{T} W = G(x_{q}, y_{q})^{T} W.$$

The weights $V$ can be computed once for a given query point and reused to calculate the different thermodynamic properties of interest. For example, if twelve properties need to be interpolated, only one matrix-multiplication is required per each mesh triangle, instead of the twelve which would be needed with the primal interpolation method. Additionally, since the $(A^{T})^{-1}$ matrix depends only on a priori established values, it can be completely pre-computed without additional runtime cost. If the condition number of the matrix is high, a pseudo-inverse should be applied; in this case pre-computation is only possible with the primal interpolation.

3. Application to NICFD simulations

In order to verify and provide information on the performance of the ug-LUT method, three CFD test cases of increasing complexity level are discussed. The selected test cases feature expansions characterized by relevant non-ideal compressible flow effects, requiring the use of complex equations of state to accurately compute the flow behavior. The simulations are performed with SU2 [16], a code previously verified for non-ideal compressible fluid dynamics simulations [24]. For all test cases, the convective fluxes are discretized by a generalized Roe scheme [25,24], and second-order accuracy is achieved with the MUSCL approach [26]. Ref. [16] provides a more detailed description of the flow solver and the associated numerical methods.

Thermodynamic properties needed by the flow solver, whose value is interpolated from the values stored in the lut, are

$$\rho, P, T, c, e, h, s, \left( \frac{\partial \rho}{\partial \rho} \right)_{e}, \left( \frac{\partial e}{\partial e} \right)_{\rho}.$$

The partial derivatives of the pressure are necessary to calculate the convective fluxes in non-ideal compressible flow simulations, see, e.g., Ref. [18].

3.1. 2D supersonic nozzle

The geometry of the converging–diverging supersonic nozzle is depicted in Fig. 5a, together with the Mach contour resulting from the Euler simulation. The two-dimensional flow domain is discretized with approximately 15,000 triangular mesh elements.
and the working fluid is siloxane MM. Table 1 reports the main simulation parameters.

In order to assess the performance of the method, the computational cost of the nozzle simulation relying on the ug-LUT algorithm for the evaluation of fluid properties (ug-LUTsim) is compared with one in which the properties are provided to the flow solver by the external thermodynamic library based on the Span-Wagner EoS (SWsim), for an increasing number of thermodynamic mesh nodes. Fig. 4a shows the computational cost of ug-LUTsim as a function of the mesh nodes, normalized with the computational cost of SWsim. Fig. 4b reports the root-mean-square error (RMSE) between the flow field Mach number of ug-LUTsim with respect to SWsim for different thermodynamic mesh refinements. As expected, the RMSE value decreases with the number of thermodynamic mesh elements, whereas the computational cost shows the opposite trend.

Fig. 5b shows a comparison of the Mach number calculated at the nozzle center line with ug-LUTsim and simulation results in which fluid properties are evaluated with the ideal-gas model (IGsim) and with a model based on the Span-Wagner EoS (SWsim). The streamwise Mach distribution obtained with ug-LUTsim is well in agreement with the results of SWsim. As expected, both deviate from the distribution obtained with IGsim, considering that the inlet compressibility factor significantly departs from unity (Z = 0.64).

### 3.2. Turbulent transonic 2D turbine cascade

The turbine cascade reported in Fig. 6a was simulated under transonic conditions, in order to evaluate the performance of the ug-LUT method in case of RANS simulations. A hybrid mesh of approximately 40,000 elements was used to discretize the computational domain, with about 15,000 quads in the proximity of the blade surface to ensure y* ≈ 1. The simulation parameters are listed in Table 2. For this test case MDM is considered as working fluid.

Similarly to the previous test case, Fig. 7 reports the same trend in terms of computational cost and RMSE based on the Mach flow field. However, the computational gain provided in the RANS simulation is approximately 4 times higher as compared to the inviscid test case. Fig. 6b shows the comparison between the dimensionless static pressure, along the blade profile, from the IGsim, the SWsim, and the ug-LUTsim. The results achieved with the SWsim are well in agreement with ug-LUTsim, while both differ from the ones obtained with IGsim.

In order to further investigate the performance of the proposed tabular method, the same turbine configuration is simulated with the binary mixture MDM(85%)/MM(15%) as working fluid. For this test case, the cost reduction is approximately 5 times higher than the computation of the single-component working fluid (Fig. 8b), while retaining the same accuracy (Fig. 6a). These results show that...
the use of the LuT method is even more attractive when applied to flow problems involving mixtures.

### 3.3. Turbulent 3D supersonic turbine cascade

The supersonic stator of the Organic Rankine Cycle turbine, documented in Ref. [27], is finally considered to investigate the computational efficiency of the unstructured-based look-up table method for three-dimensional RANS simulations. The numerical parameters of the test case are provided in Table 3. The physical mesh consists of about one million cells and thermodynamic grid is composed by approximately 20,000 nodes.

Fig. 9a shows the contour of the density gradient: a complex flow pattern of both shock-waves and fans is present in the semi-bladed region, due to the post-expansion phenomena. Fig. 9b displays the density field relative error between the ug-LUTsim and the SWsim. The deviation in the order of 0.1% points out that, even with a relatively coarse thermodynamic grid, the tabular approach is accurate for three-dimensional problems involving complex flow phenomena. Furthermore, the computational cost reduction for the 3D
4. Performance and memory assessment

An equally spaced structured grid-based LuT method was implemented within SU2, in order to assess the performance of ug-LUT. The structured-based LuT algorithm was considered for carrying out a comparison in terms of computational cost and memory requirements because of its simple data structure and efficiency. The turbine cascade, described in Section 3.2, is selected as reference test case to perform this analysis. The nomenclature sg-LUT and ug-LUT is used hereinafter to refer to the structured-based and the unstructured-based LUT method, respectively.

4.1. Structured-grid LuT algorithm | sg-LUT

The structured grid LuT (sg-LUT) implementation features the same interpolation method of ug-LUT. Thermodynamic query vectors for the CFD application considered, are: \((P, T), (\rho, P), (\rho, s), (\rho, T), (h, s)\). By selecting the thermodynamic mesh as a function of pressure and density (Fig. 11), the searching algorithm is based on simple binary search for \((P, T), (\rho, P), (\rho, s), (\rho, T)\), because they have at least one common input with respect to thermodynamic mesh. A \(kd\)-tree is used for the \((h, s)\) pair. Fig. 11 shows an example of both structured and unstructured mesh of thermodynamic region of interest, generated using the same number of mesh nodes.

4.2. Comparison sg-LUT vs. ug-LUT

Fig. 12a and b reports the normalized CPU time associated with the sg-LUTsim and ug-LUTsim. The total time includes both the LuT pre-processing and the total CFD solver iteration time. As can be noticed (Fig. 12b), the pre-processing time becomes a relevant fraction of the total time for ug-LUT as opposed to sg-LUT. This is due to the trapezoidal map generation for each thermodynamic input pair. This operation is done just once for sg-LUT, when creating the \(kd\)-tree relative to the \((h, s)\) input pair.

The ratio between the total simulation time obtained by ug-LUT and sg-LUT (Fig. 13a) indicates that ug-LUT is faster than sg-LUT, for thermodynamic meshes that are approximately composed by a number of mesh nodes lower than 10,000. The sg-LUT is about 5% faster than ug-LUT for 25,000 thermodynamic mesh nodes. The ug-LUT performance is in agreement with the computational cost of LuT based on quad-tree data structures, whose computational cost has been found to be 10% higher than equally spaced structured tabulation methods [13]. Fig. 13b shows the comparison between the memory requirements of ug-LUT and sg-LUT. The ug-LUT memory

![Fig. 10. Summary of the computational time for the selected test-cases. Results are normalized using the computational time associated with the SWsim and based on a thermodynamic mesh of approximately 15,000 nodes.](image)

![Fig. 11. Mesh of thermodynamic region of interest for the turbine cascade test case, as a function of the reduced density \(\rho_s\) and reduced pressure \(P_r\). (a) Example of structured mesh used for sg-LUT. (b) Example of unstructured mesh used for ug-LUT.](image)
requirements are higher than sg-LUT mainly due to the following reasons: (i) in ug-LUT the trapezoidal maps are created for all thermodynamic input pairs; (ii) for ug-LUT the unstructured-mesh connectivity has to be stored. For practical applications, however, since thermodynamic meshes featuring around 10,000 elements are deemed sufficient for the required level of accuracy, the absolute memory associated never exceeded 200 Mb. Furthermore, for problems discretized on large domains both the preprocessing computational cost and the memory burden are expected to be a negligible fraction when compared with the CPU time and memory requirements of the CFD simulation.

Finally, a comparative assessment of the RMSE with respect to the SWsim is carried out for both ug-LUT and sg-LUT. Fig. 14 depicts the ratio between the RMSE obtained by ug-LUT and sg-LUT relative to the conservative variables \( \rho, p, \theta, \rho v_1, \rho v_2, \rho e \) (Fig. 14a), pressure and temperature (Fig. 14b). The RMSE is calculated, as shown in Section 3, with respect to SWsim. The ug-LUT algorithm is more accurate than sg-LUT, for the fluxes \( \rho v_1, \rho v_2, \rho e \) and the pressure while the opposite occurs with regard to the density and temperature. Without being exhaustive, these results indicate that unstructured tabular methods may be advantageous when it comes to accuracy as compared to structured grids characterized by the same number of nodes and level of refinement.

5. Conclusions

This paper documents the ug-LUT method, a novel look-up table method that can be used to improve the computational performance in non-ideal compressible fluid dynamics (NICFD) simulations. The method is based on an unstructured mesh in combination with a trapezoidal-map searching algorithm and a piece-wise interpolation method based on the duality approach. The algorithm was successfully implemented in the open-source code SU2 and its performance assessed in three paradigmatic NICFD cases: (1) an inviscid 2D supersonic nozzle; (2) a 2D RANS transonic turbine cascade; (3) a 3D RANS supersonic stator row. In all cases, thermodynamic property model of reference is based on a multi-parameter technical equation of state.

The outcome of this study can be summarized as follows:

- The ug-LUT method provides a computational cost reduction compared to simulations in which properties are calculated.
Appendix the Algorithm

1: edge simulations is input:
   Filter list
   output:
   list list
   return:
   for
   Map of intersects containing the 2D XBands y x band
   Build unique providing query edges from is given; in
   EdgeToFace which is unique and which extending
   Mueller, RANS to efficient calculation of the laboratory
   the future, (see Appendix 3).

Acknowledgements

This research has been supported by Robert Bosch GmbH and the Applied and Engineering Sciences Domain (TTW) of the Dutch Organization for Scientific Research (NWO), Technology Program of the Ministry of Economic Affairs, grant number 13385.

Appendix A.

Here, the pseudo-codes of the algorithms used for the Trapezoidal Map presented in Sec. 2.2 are reported.

Algorithm 1. Build Trapezoidal Map is.

1: input: list of unique edges: Edges;
2: list of edge to face connectivities: EdgeToFace;
3: list of the x-coordinates of the edges: x_edges;
4: list of the y-coordinates of the edges: y_edges;
5: output: list of unique x bands through which to search: X_edges;
6: list of connectivity of edges to a given unique x-band (ordered by y-value of edge at middle of band): Y_edges;
7: Filter out vertical edges from X_edges, creating unique list: X_edges;
8: for each b in X_edges do
9:   for each e in Edges do
10:      if e intersects b then Add e to Y_edges;
11: end if
12: end for
13: Sort the edges in Y_edges according to the y value of the edge in the center of the band
14: end for
15: return: X_edges, Y_edges;

Algorithm 2. Trapezoidal Map is.

1: input: 2D query point (what to search for): x, y;
2: list of edge to face connectivities: EdgeToFace;
3: list of unique x bands through which to search: X_edges;
4: list of connectivity of edges to a given unique x-band (ordered by y-value of edge at middle of band): Y_edges;
5: output: the index of the face in which the query point lies: CF;
6: 7: Use binary search to find band b in X_edges which contains x;
8: Within band b use a binary search to find edge e (below the query point) and e  above the query point;
9: Use interpolation to check if edge is below or above the query point;
10: The containing face CF is given by the intersection of the edge to face connectivities of e  and e );
11: return: CF;

References