

A Decoupled Method for the Roe FDS Scheme in the Reacting Gas Path of FUN3D

Kyle B. Thompson* and Peter A. Gnoffo†

NASA Langley Research Center, Hampton, VA, 23681-2199

In this paper, we describe an approach for decoupling the species continuity equations from the mixture continuity, momentum, and total energy equations for the Roe flux difference splitting scheme. Utilizing aspects of the implicit system, the flow solver can be made significantly more efficient, with very little penalty on overall scheme robustness. Most importantly, the computational cost of the point implicit relaxation is shown to scale linearly with the number of species for the decoupled system, whereas the fully coupled approach scales quadratically. We show that solving the implicit system in a decoupled fashion significantly reduces the cost in wall time and memory in comparison to the fully coupled approach.

Nomenclature

| | |
|---|---|
| A, A_d, A_m | Jacobian Matrices |
| a | Speed of sound m/s |
| \mathbf{b} | Residual vector |
| c_s | Species s mass fraction |
| C | Decoupled scheme chemical source term jacobian |
| D | Decomposed diagonal jacobian matrix |
| dv_1, dv_2, dv_3 | Eigenvector components |
| e_s | Internal energy of species s |
| E | Total Energy |
| F' | Decoupled scheme mixture mass flux |
| F'_ρ | Decoupled scheme mass flux of species s |
| F'_{ρ_s} | |
| $\mathbf{F}, \mathbf{F}', \hat{\mathbf{F}}$ | Flux vectors |
| N_{nodes} | Number of nodes |
| N_{nz} | Number of non-zero off-diagonal entries in jacobian |
| O | Decomposed off-diagonal jacobian matrix |
| p | Pressure, N/m^2 |
| \mathbf{R}, \mathbf{L} | Right and left eigenvectors |
| R_ρ | Decoupled scheme constraint |
| \mathbf{S} | Face normal vector, m^2 |
| $\mathbf{U}, \mathbf{U}', \hat{\mathbf{U}}$ | Conservative variable vectors |
| \bar{U} | Normal velocity, m/s^2 |
| u, v, w | Components of velocity, m/s |
| V | Decoupled variables vector |
| w | Roe scheme weighting factor |
| \hat{V} | Cell volume, m^3 |
| \mathbf{W} | Chemical source term vector |
| ρ | Mixture density kg/m^3 |

*NASA Pathways Intern, Aerothermodynamics Branch, AIAA Student Member.

†Senior Research Engineer, Aerothermodynamics Branch, AIAA Fellow.

| | |
|-----------------------------------|-------------------------------------|
| ρ_s | Species s density kg/m^3 |
| $\lambda_1, \lambda_2, \lambda_3$ | Acoustic and convective eigenvalues |
| λ^-, λ^+ | Species flux effective eigenvalues |
| Λ | Diagonal eigenvalue matrix |

Superscript

| | |
|------------|---------------------------------|
| $n, n + 1$ | Time level |
| R, L | Right and left state quantities |

I. Introduction

The usability of hypersonic solvers on complex geometries is often limited by the extreme problem size associated with high energy physics. The additional equations required in reacting gas simulations lead to large jacobians that scale quadratically in size to the number of governing equations. This leads to a significant increase in memory required to store the flux linearizations and the computational cost of the point solver. As reacting gas CFD solvers are used to solve increasingly more complex problems, this onerous quadratic scaling of computational cost and jacobian size will ultimately surpass the limits of hardware and time constraints on achieving a flow solution.

The proposed method is based heavily upon the work of Candler et al.,² where the quadratic scaling between the cost of solving the implicit system and the addition of species mass equations can be reduced to that of a linear scaling by decoupling the species mass equations from the mixture mass, momentum, and energy equations and solving the two decoupled systems sequentially. In the aforementioned work, the scheme was derived for a modified form of the Steger-Warming flux vector splitting method,⁵ whereas the work presented here is derived for Roe flux difference splitting (FDS) scheme.

A primary motivator for the presented work is to prepare for the implementation of an adjoint capability in a reacting gas solver. This is an extremely challenging problem, and a significant hardship is deriving exact jacobians for a reacting gas system. A beneficial side effect of decoupled the system is the simplification of deriving exact jacobians for the Roe flux difference splitting scheme. By decoupling the species equations, the mixture mass, moment, and energy flux jacobians can be easily derived,⁶ using a weighting scheme to correct for the non-uniqueness of the pressure linearization.⁷ For the decoupled species fluxes we derive an exact linearization in the presented work.

II. Background: Fully-Coupled Point Implicit Method

All work presented here is for the inviscid conservation equations, but can be extended to include viscous terms. For an inviscid, multi-species mixture, the governing equations in vector form are:

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \mathbf{F} = \mathbf{W} \quad (1)$$

or, in finite volume form:

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{1}{V} \sum_f (\mathbf{F} \cdot \mathbf{S}) = \mathbf{W} \quad (2)$$

summing over all faces in the domain, with V being the cell volume and \mathbf{S} being the face outward normal vector. The vectors of conserved variables and fluxes are:

$$\mathbf{U} = \begin{pmatrix} \rho_1 \\ \vdots \\ \rho_{ns} \\ \rho u \\ \rho v \\ \rho w \\ \rho E \end{pmatrix}, \quad \mathbf{F} = \begin{pmatrix} \rho_1 \bar{U} \\ \vdots \\ \rho_{ns} \bar{U} \\ \rho u \bar{U} + p s_x \\ \rho u \bar{U} + p s_y \\ \rho u \bar{U} + p s_z \\ (\rho E + p) \bar{U} \end{pmatrix} \quad (3)$$

Where \bar{U} is the outward pointing normal velocity and ρE is the total energy of the mixture, defined as:

$$E = \sum \rho_s e_s + \frac{u^2 + v^2 + w^2}{2} \quad (4)$$

where e_s is the internal energy of species s . Using the Roe FDS scheme the numerical fluxes can be expressed as follows:

$$\mathbf{F}^f = \frac{\mathbf{F}(\mathbf{U}^L) + \mathbf{F}(\mathbf{U}^R)}{2} - \frac{1}{2} \mathbf{R} \Lambda \mathbf{L} \quad (5)$$

Where Λ is the diagonal eigenvalue matrix, and \mathbf{R} and \mathbf{L} are the right and left eigenvectors, respectively. We now linearize the flux and the source term vectors as:

$$\mathbf{F}^{n+1} \approx \mathbf{F}^n + \frac{\partial \mathbf{F}}{\partial \mathbf{U}} \delta \mathbf{U}^n \quad (6)$$

$$\mathbf{W}^{n+1} \approx \mathbf{W}^n + \frac{\partial \mathbf{W}}{\partial \mathbf{U}} \delta \mathbf{U}^n$$

where $\delta \mathbf{U}^n = \mathbf{U}^{n+1} - \mathbf{U}^n$. Using an implicit time integration, the implicit scheme becomes:

$$\frac{\delta \mathbf{U}^n}{\Delta t} + \frac{1}{V} \sum_f \left(\frac{\partial \mathbf{F}}{\partial \mathbf{U}^L} \delta \mathbf{U}^L + \frac{\partial \mathbf{F}}{\partial \mathbf{U}^R} \delta \mathbf{U}^R \right)^n - \frac{\partial \mathbf{W}}{\partial \mathbf{U}} \delta \mathbf{U}^n = -\frac{1}{V} \sum_f (\mathbf{F} \cdot \mathbf{S})^n + \mathbf{W}^n \quad (7)$$

or, put more simply:

$$A \delta \mathbf{U}^n = \mathbf{b} \quad (8)$$

where A is the jacobian matrix, and \mathbf{b} is the residual vector. For a point implicit relaxation scheme the jacobian matrix can be split into its diagonal and off-diagonal elements, with the latter moved to the RHS as such:

$$A = O + D \quad (9)$$

Each matrix element is a square $(ns+4) \times (ns+4)$ matrix. One method of solving this system is a Red-Black Gauss-Seidel scheme, where matrix coefficients with even indices are updated first and, subsequently, the coefficients with odd indices are updated. This enables better vectorization in solving the linear system. A key point here is the matrix multiplication of the elements of the decomposed A matrix is a quadratic operation; thus, the cost of solving the implicit system scales quadratically with the number of equations being solved.

The addition of viscous terms does not change the form of the system and results in dense matrices being added to the jacobian matrix, A .

III. Background: Decoupled Point Implicit Method

By replacing the species mass equations with a single mixture equation and separating the mixture equations from the species mass equations, the conserved variables become:

$$\mathbf{U}' = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho E \end{pmatrix} \quad \hat{\mathbf{U}} = \begin{pmatrix} \rho_1 \\ \vdots \\ \rho_{ns} \end{pmatrix} \quad (10)$$

Solving for the flux is now done in two sequential steps. The mixture fluxes are first solved as:

$$\frac{\partial \mathbf{U}'}{\partial t} + \frac{1}{V} \sum_f (\mathbf{F}' \cdot \mathbf{S}) = 0 \quad (11)$$

Followed by the species fluxes

$$\frac{\partial \hat{\mathbf{U}}}{\partial t} + \frac{1}{V} \sum_f (\hat{\mathbf{F}} \cdot \mathbf{S}) = \hat{\mathbf{W}} \quad (12)$$

The point relaxation proceeds using Red-Black Gauss-Seidel to update the conserved variables in \mathbf{U}' and all associated auxiliary variables, such as temperature, pressure, speed of sound, etc. This is done holding the thermo-chemical state constant, and will always result in the relaxation of a 5-equation system.

The solution of the species mass equations takes a different form. Based on the work of Candler et al.,² the decoupled variables can be rewritten in terms of mass fraction, as follows:

$$\delta \hat{\mathbf{U}}^n = \rho^{n+1} \hat{\mathbf{V}}^{n+1} - \rho^n \hat{\mathbf{V}}^n = \rho^{n+1} \delta \hat{\mathbf{V}}^n + \hat{\mathbf{V}}^n \delta \rho^n \quad (13)$$

where $\hat{\mathbf{V}} = (c_1, \dots, c_{ns})^T$, and $c_s = \rho_s / \rho$ the mass fraction of species s . While the derivation of the species mass equations is different from that of the Steger-Warming FVS scheme proposed by Candler et al.,² the final result takes a similar form:

$$\hat{F}_{\rho_s} = c_s F'_\rho + (c_s^L - \tilde{c}_s) \rho^L \lambda^+ + (c_s^R - \tilde{c}_s) \rho^R \lambda^- \quad (14)$$

where F'_ρ is the total mass flux computed previously using all \mathbf{U}' variables. Likewise, linearizing the species mass fluxes with respect to the $\hat{\mathbf{V}}$ variables yields:

$$\hat{\mathbf{F}}^{n+1} = \hat{\mathbf{F}}^n + \frac{\partial \hat{\mathbf{F}}}{\partial \hat{\mathbf{V}}^L} \delta \hat{\mathbf{V}}^L + \frac{\partial \hat{\mathbf{F}}}{\partial \hat{\mathbf{V}}^R} \delta \hat{\mathbf{V}}^R \quad (15)$$

$$\frac{\partial \hat{\mathbf{F}}}{\partial \hat{\mathbf{V}}^L} = w F_\rho + (1 - w) \rho^L \lambda^+ - w \rho^R \lambda^- \quad (16)$$

$$\frac{\partial \hat{\mathbf{F}}}{\partial \hat{\mathbf{V}}^R} = (1 - w) F_\rho + (w - 1) \rho^L \lambda^+ + w \rho^R \lambda^- \quad (17)$$

A full derivation of Eq.s (14-17) is included in Appendix A. The chemical source term is evaluated with the \mathbf{U}' variables, and the linearization is carried out as:

$$\hat{\mathbf{W}}^{n+1} = \hat{\mathbf{W}}^n + \frac{\partial \hat{\mathbf{W}}}{\partial \mathbf{U}} \bigg|_{\mathbf{U}'} \frac{\partial \mathbf{U}}{\partial \hat{\mathbf{V}}} \quad (18)$$

For simplicity of notation, we define:

$$C = \frac{\partial \hat{\mathbf{W}}}{\partial \mathbf{U}} \bigg|_{\mathbf{U}'} \frac{\partial \mathbf{U}}{\partial \hat{\mathbf{V}}} \quad (19)$$

The decoupled system to be solved becomes:

$$\rho^{n+1} \frac{\delta \hat{\mathbf{V}}^n}{\Delta t} + \frac{1}{V} \sum_f \left(\frac{\partial \hat{\mathbf{F}}}{\partial \hat{\mathbf{V}}^L} \delta \hat{\mathbf{V}}^L + \frac{\partial \hat{\mathbf{F}}}{\partial \hat{\mathbf{V}}^R} \delta \hat{\mathbf{V}}^R \right)^{n,n+1} - C^{n,n+1} \delta \mathbf{V}^n \quad (20)$$

$$= -\frac{1}{V} \sum_f (\hat{\mathbf{F}}^{n,n+1} \cdot \mathbf{S}) + \mathbf{W}^{n,n+1} - \hat{\mathbf{V}}^n \frac{\delta \rho^n}{\Delta t} - R_\rho \quad (21)$$

$$R_\rho = -\frac{1}{V} \sum_f \sum_s (\hat{F}_{\rho_s}^{n,n+1} \cdot \mathbf{S})$$

where R_ρ is included to preserve the constraint that the mass fractions sum to unity, i.e. $\sum_s c_s = 1$, $\sum_s \delta c_s = 1$.

IV. Cost and Memory Savings of the Decoupled Implicit Problem

In decoupling the species equations, the most significant savings comes from the source term linearization being purely node-based.⁴ Solving the mean flow equations is conducted in the same manner as the fully coupled system, where all entries in the jacobian A_m are 5×5 matrices. For the decoupled implicit system, all entries in the jacobian A_d are $ns \times ns$ matrices. Because there is no interdependence of species, except through the chemical source term, all contributions due to linearizing the convective flux are purely diagonal

$ns \times ns$ matrices. Via Eq. (9), we decompose A_d into its diagonal and off-diagonal elements, resulting in the following linear system:

$$\begin{pmatrix} \square & & \\ & \ddots & \\ & & \square \end{pmatrix} \begin{pmatrix} \delta \hat{V}_1 \\ \vdots \\ \delta \hat{V}_{nodes} \end{pmatrix} = b - \begin{pmatrix} 0 & & \diagdown \\ & \ddots & \\ \diagup & & 0 \end{pmatrix} \begin{pmatrix} \delta \hat{V}_1 \\ \vdots \\ \delta \hat{V}_{nodes} \end{pmatrix} \quad (22)$$

where \square represents a dense $ns \times ns$ matrix, and \diagup represents a diagonal matrix. Thus, the non-zero entries in the off-diagonal matrix can be reduced from diagonal matrices to vectors, reducing the rank of the overall matrix by 1. This results in significant savings in both cost and memory, as the only quadratic operation left in solving the implicit system is dealing with the diagonal entries in the jacobian. Because the off-diagonal entries significantly outnumber of the diagonal entries, we can expect nearly linear scaling in cost with the number of species. Utilizing compressed row storage³ for the off-diagonal entries, the relative memory savings in the limit of a large number of species for the jacobian is given by:

$$\begin{aligned} \text{Relative Memory Cost} &= \frac{\text{size}(A_d)}{\text{size}(A_m)} \\ &= \lim_{ns \rightarrow \infty} \frac{(ns^2 + 5^2)(N_{nodes}) + (ns + 5^2)(N_{nz})}{(ns + 4)^2(N_{nodes} + N_{nz})} \\ &= \frac{N_{nodes}}{N_{nodes} + N_{nz}} \end{aligned} \quad (23)$$

Where N_{nodes} is the number of nodes, and N_{nz} is the number of non-zero off-diagonal entries stored using compressed row storage. For a structured grid, each node has 6 neighbors in 3D, i.e. $N_{nz} = 6N_{nodes}$; therefore, we can expect the jacobian memory required to decrease by a factor of 7 using this decoupled scheme. Interestingly, for a grid that is not purely hexahedral, $N_{nz} > 6N_{nodes}$, making this decoupled scheme even more efficient for unstructured grids than structured grids.

V. Results

The proposed decoupled scheme has been implemented in the NASA Langley unstructured grid solver FUN3D,⁷ as well as the traditional, fully coupled approach. To demonstrate the improved efficiency in cost and memory required utilizing the decoupled scheme, and that both the fully coupled and decoupled approaches converge to the same result, a grid convergence study was conducted on a simple hemisphere geometry (radius 0.5 m). A 50 x 50, 100 x 100, and 200 x 200 family of grids were adapted using the line adaptation capability in FUN3D to produce well-aligned grids that approximate a true coarsening of the finest grid level. The free stream conditions used were $V_\infty = 5000 \text{ m/s}$, $\rho_\infty = 0.001 \text{ kg/m}^3$, $T_\infty = 200 \text{ K}$. Several chemical kinetics models were used, including a 5-species model with 5 reactions, an 11-species model with 22 reactions, and an 18-species model with 29 reactions.

A. Comparison of Solution

Most important is that the decoupled and fully coupled scheme give nearly identical converged solutions. To quantitatively assess this, we compare the predicted surface pressure, surface temperature, the species composition on the stagnation line for both schemes. Figures 1 through 3 show that the predicted quantities on the on the 100 x 100 grid. All results are indeed nearly identical, with temperature and pressure matching discretely to 8 digits and the species mass fractions on the stagnation line matching to 4 digits. This difference was observed to be reduced slightly more on the finest grid level of 200 x 200.

B. Memory Cost

In order to determine the required memory of the decoupled scheme in comparison to the fully coupled scheme, a convergence study was conducted using Valgrind¹ to determined the memory actually allocated by FUN3D for an increasing number of species. Figure 4 shows the relative memory cost converges asymptotically to ~ 0.25 , which is nearly twice the predicted value of $1/7$. For the implementation of FUN3D this is actually correct, because the off-diagonal entries are reduced from double precision to single precision.

Starting from Eq. (23), for a structured grid (as is the case here) where each node has 6 neighbors, the relative memory cost come to:

$$\text{Relative Memory Cost} = \frac{N_{nodes}}{N_{nodes} + N_{nz}} = \frac{N_{nodes}}{N_{nodes} + (6N_{nodes}/2)} = \frac{1}{4} \quad (24)$$

thus, the relative memory saved by using the decoupled scheme correctly approaches a factor of 1/4. It should be noted that the additional species in the 35-species and 64-species cases were chosen arbitrarily, since jacobian size is only dependent on the number of species, not the mixture composition.

C. Computational Cost

As stated before, the cost of solving the decoupled implicit system should scale approximately linearly with the number of species, whereas the fully coupled problem should scale quadratically. Figure 5 shows this to be true for the hemisphere test case. Figure 6 shows the total speed up of the problem to be less than that of the just the linear solve. This is to be expected, as there are many other factors that scale with the number of species, especially calculating the species source term and its linearization.

VI. Conclusion

The motivation for this work was to greatly increase the efficiency of the reacting gas flow solver path in FUN3D to aid in the development of an adjoint development for this path. Because of the complexity of the high-energy physics and the large problem size associated with additional equations needed to preserve species mass, and this increased efficiency is extremely beneficial towards an adjoint formulation. Most importantly, the decoupling of the species equations from the mixture equations is imperative in deriving sufficiently accurate jacobians for the Roe FDS scheme. The 400% decrease in memory required and 300% speedup for the 18-species, at little to no cost of accuracy or stability, case clearly illustrates significant advantages of this method, at little to no cost of accuracy or stability.

A. Decoupled Flux Derivation

For the Roe Flux Difference Splitting scheme, the species mass fluxes are given by:

$$F_{\rho_s} = \frac{\rho_s^L \bar{U}^L + \rho_s^R \bar{U}^R}{2} - \frac{\tilde{c}_s(\lambda_1 dv_1 + \lambda_2 dv_2) + \lambda_3 dv_{3_s}}{2} \quad (25)$$

$$dv_1 = \frac{p^R - p^L + \tilde{\rho} \tilde{a}(\bar{U}^R - \bar{U}^L)}{\tilde{a}^2} \quad (26)$$

$$dv_2 = \frac{p^R - p^L - \tilde{\rho} \tilde{a}(\bar{U}^R - \bar{U}^L)}{\tilde{a}^2} \quad (27)$$

$$dv_{3_s} = \frac{\tilde{a}^2(\rho_s^R - \rho_s^L) - \tilde{c}_s(p^R - p^L)}{\tilde{a}^2} \quad (28)$$

$$\lambda_1 = |\bar{U} + \tilde{a}|, \quad \lambda_2 = |\bar{U} - \tilde{a}|, \quad \lambda_3 = |\bar{U}| \quad (29)$$

where the $\tilde{\cdot}$ notation signifies a roe-averaged quantity, given by:

$$\tilde{U} = w \tilde{U}^L + (1 - w) \tilde{U}^R \quad (30)$$

$$w = \frac{\tilde{\rho}}{\tilde{\rho} + \rho^R} \quad (31)$$

The species mass fluxes must sum to the total mass flux; thus, the total mixture mass flux is given as:

$$F_\rho = \sum_s F_{\rho_s} = \frac{\rho^L \bar{U}^L + \rho^R \bar{U}^R}{2} - \frac{\tilde{c}_s(\lambda_1 dv_1 + \lambda_2 dv_2) + \lambda_3 dv_3}{2} \quad (32)$$

$$dv_3 = \frac{\tilde{a}^2(\rho^R - \rho^L) - (p^R - p^L)}{\tilde{a}^2} \quad (33)$$

Multiplying eq. (32) by the roe-averaged mass fraction and substituting into eq. (25) results in:

$$F_{\rho_s} = \tilde{c}_s F_\rho + \frac{(c_s^L - \tilde{c}_s)\rho^L(\bar{U}^L + |\tilde{U}|)}{2} + \frac{(c_s^R - \tilde{c}_s)\rho^R(\bar{U}^R - |\tilde{U}|)}{2} \quad (34)$$

The notation can be further simplified by defining the normal velocities as follows:

$$\lambda^+ = \frac{\bar{U}^L + |\tilde{U}|}{2}, \quad \lambda^- = \frac{\bar{U}^R - |\tilde{U}|}{2} \quad (35)$$

Finally, substituting Eq. (35) into Eq. (34) yields the final result for calculating the species flux in decoupled system:

$$F_{\rho_s} = \tilde{c}_s F_\rho + (c_s^L - \tilde{c}_s)\rho^L\lambda^+ + (c_s^R - \tilde{c}_s)\rho^R\lambda^- \quad (36)$$

Form the convective contributions to the jacobians is straightforward. Because the \mathbf{U}' level variables are constant, only the left, right, and roe-averaged state mass fractions vary. Differentiating Eq. (36) with respect to the mass fraction, c_s , the left and right state contributions are:

$$\frac{\partial F_{\rho_s}}{\partial c_s^L} = w F_\rho + (1 - w)\rho^L\lambda^+ - w\rho^R\lambda^- \quad (37)$$

$$\frac{\partial F_{\rho_s}}{\partial c_s^R} = (1 - w)F_\rho + (w - 1)\rho^L\lambda^+ + w\rho^R\lambda^- \quad (38)$$

Because there is no dependence between species in decoupled convective formulation, the jacobian block elements are purely diagonal for the convective contributions, of the form:

$$\begin{pmatrix} \frac{\partial F_{\rho_1}}{\partial c_1} & & 0 \\ & \ddots & \\ 0 & & \frac{\partial F_{\rho_{ns}}}{\partial c_{ns}} \end{pmatrix} \quad (39)$$

Acknowledgments

The authors would like to recognize the FUN3D team at NASA Langley Research Center, for their support in integrating aspects of the compressible gas path into the reacting gas path of FUN3D.

References

- ¹ACM SIGPLAN 2007 Conference on Programming Language Design and Implementation. *How to Shadow Every Byte of Memory Used by a Program*, San Diego, CA, June 2007.
- ²Graham V. Candler, Pramod K. Subbareddy, and Ioannis Nompelis. Decoupled implicit method for aerothermodynamics and reacting flows. *AIAA Journal*, 51(5):1245–1254, 2015/04/23 2013.
- ³Alan George and Joseph W. Liu. *Computer Solution of Large Sparse Positive Definite*. Prentice Hall Professional Technical Reference, 1981.
- ⁴Peter A. Gnoffo Roop N. Gupta and Judy L. Shinn. Conservation equations and physical models for hypersonic air flows in thermal and chemical nonequilibrium. Technical Paper 2867, NASA, 1989.
- ⁵Robert W. McCormack and Graham V. Candler. The solution of the navier-stokes equations using gauss-seidel line relaxation. *Computers and Fluids*, 17(1):135 – 150, 1989.
- ⁶Hiroaki Nishikawa. *Implementing a Real-Gas Roe Solver into Implementing a Real-Gas Roe Solver into CFL3D*, April 2004.
- ⁷Jian-Shun Shuen, Meng-Sing Liou, and Bram Van Leer. Inviscid flux-splitting algorithms for real gases with non-equilibrium chemistry. *Journal of Computational Physics*, 90(2):371 – 395, 1990.

List of Figures

| | | |
|---|--|----|
| 1 | Pressure on Hemisphere Surface | 9 |
| 2 | Temperature on Hemisphere Surface | 10 |
| 3 | Species Mass Fractions on the Hemisphere Stagnation Line | 11 |
| 4 | Memory Required Convergence Study | 12 |
| 5 | Relative Speedup of Implicit Solve for the Decoupled Scheme wrt the Fully Coupled Scheme . | 13 |
| 6 | Overall Relative Speedup of the Decoupled Scheme wrt the Fully Coupled Scheme | 14 |

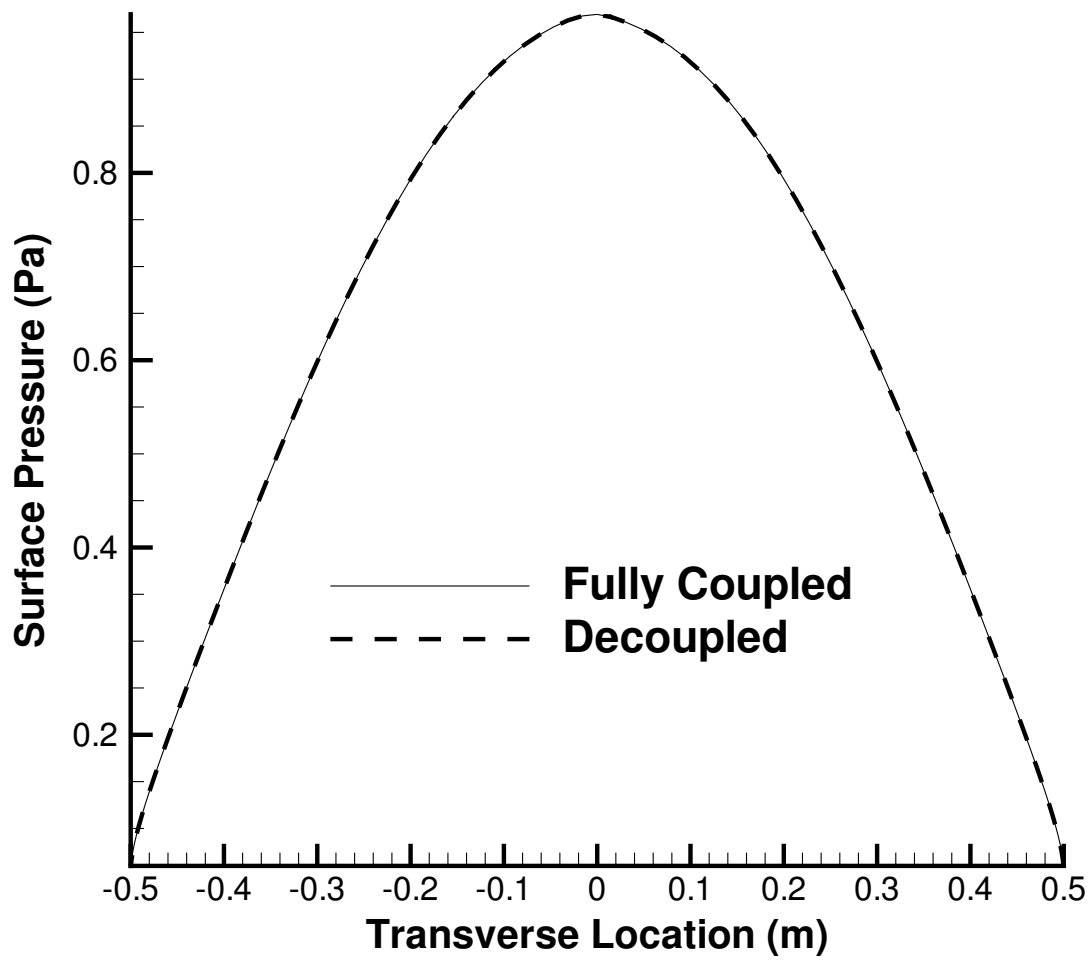


Figure 1. Pressure on Hemisphere Surface

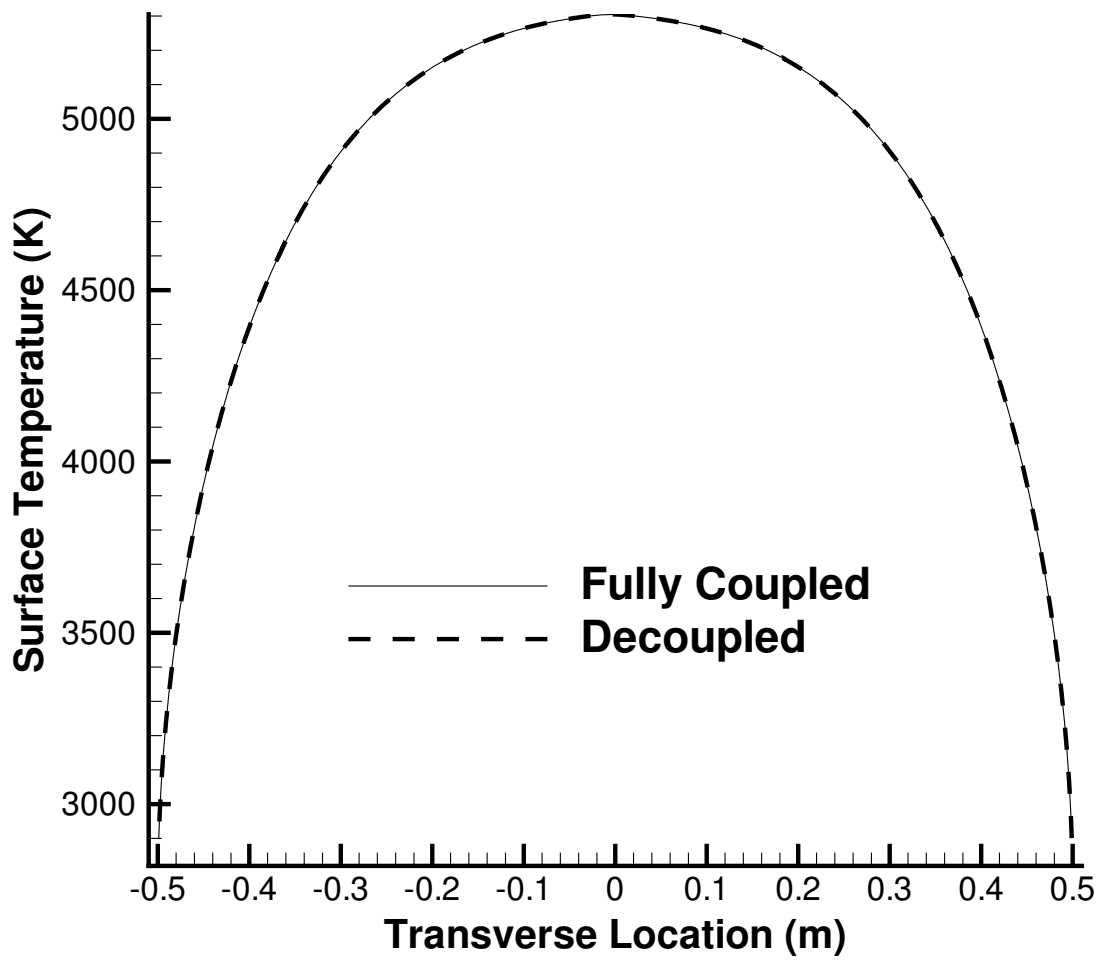


Figure 2. Temperature on Hemisphere Surface

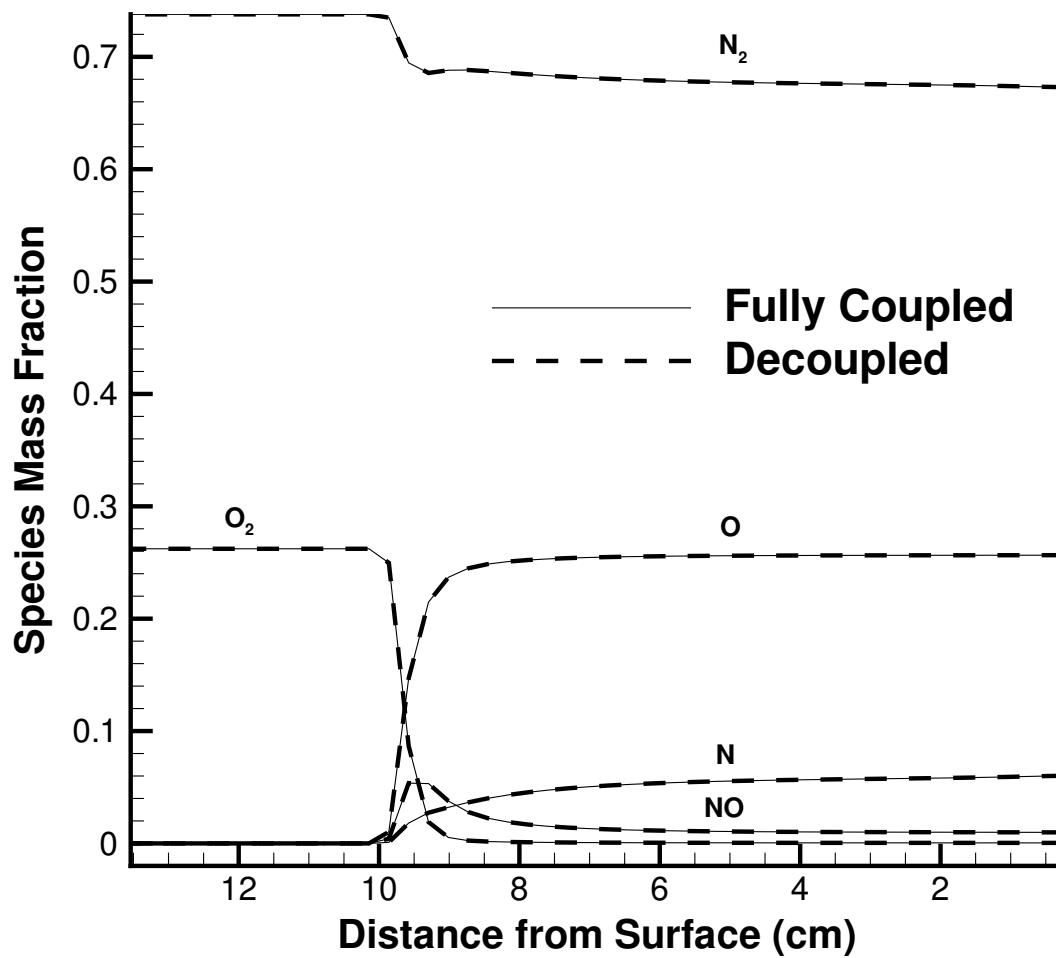


Figure 3. Species Mass Fractions on the Hemisphere Stagnation Line

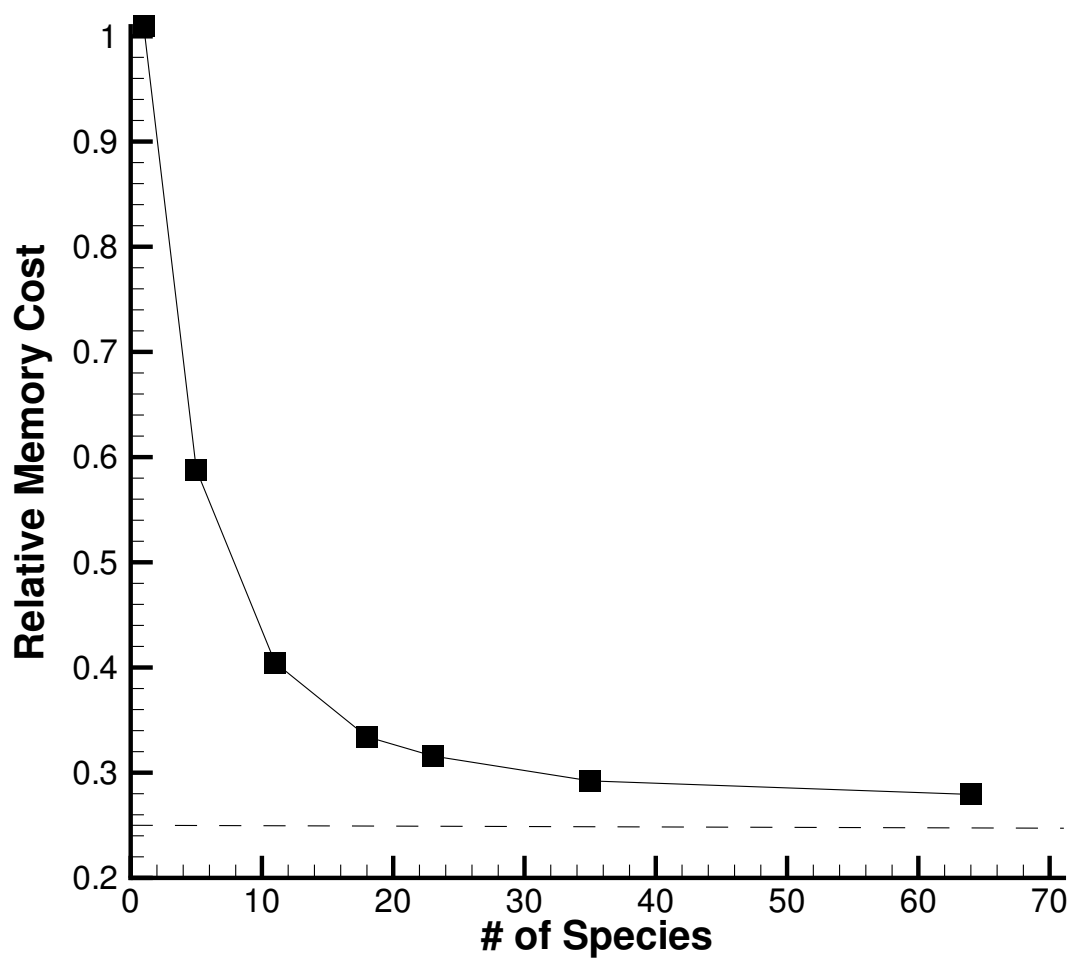


Figure 4. Memory Required Convergence Study

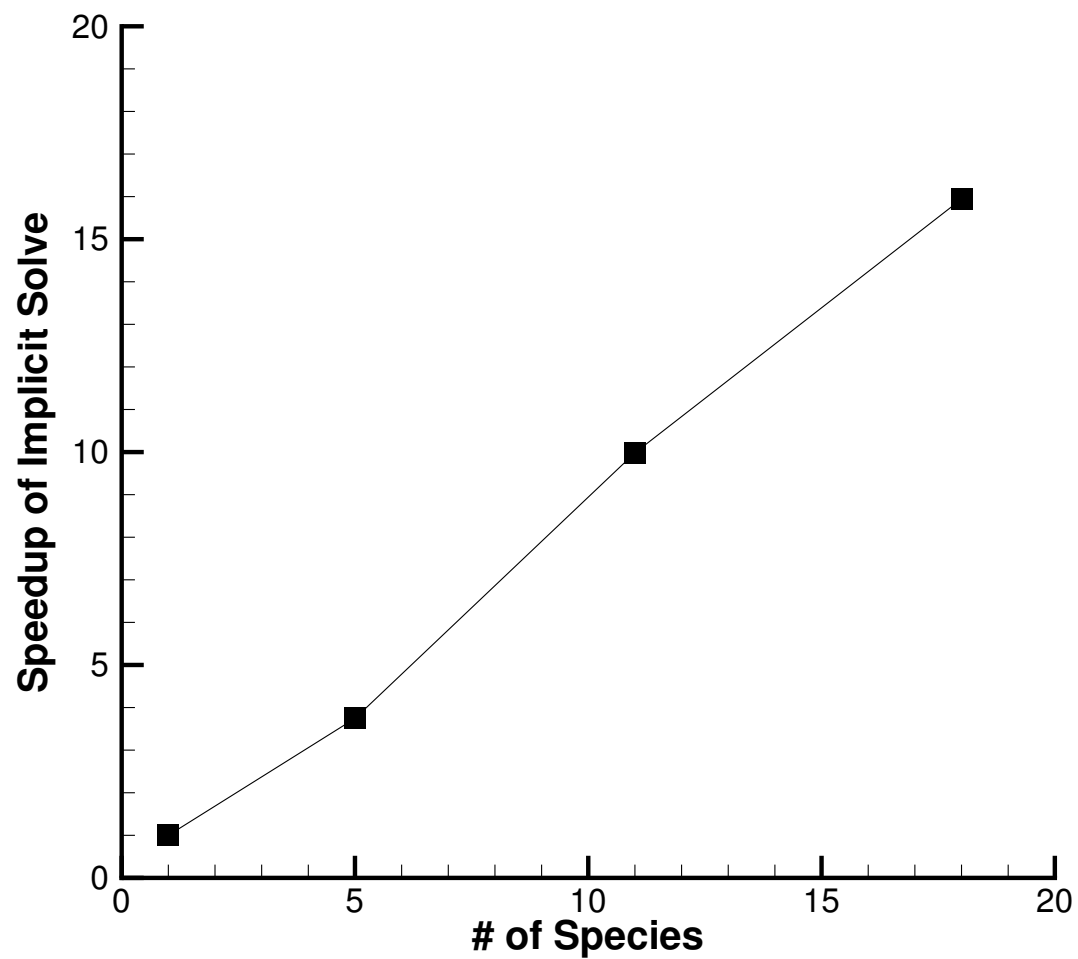


Figure 5. Relative Speedup of Implicit Solve for the Decoupled Scheme wrt the Fully Coupled Scheme

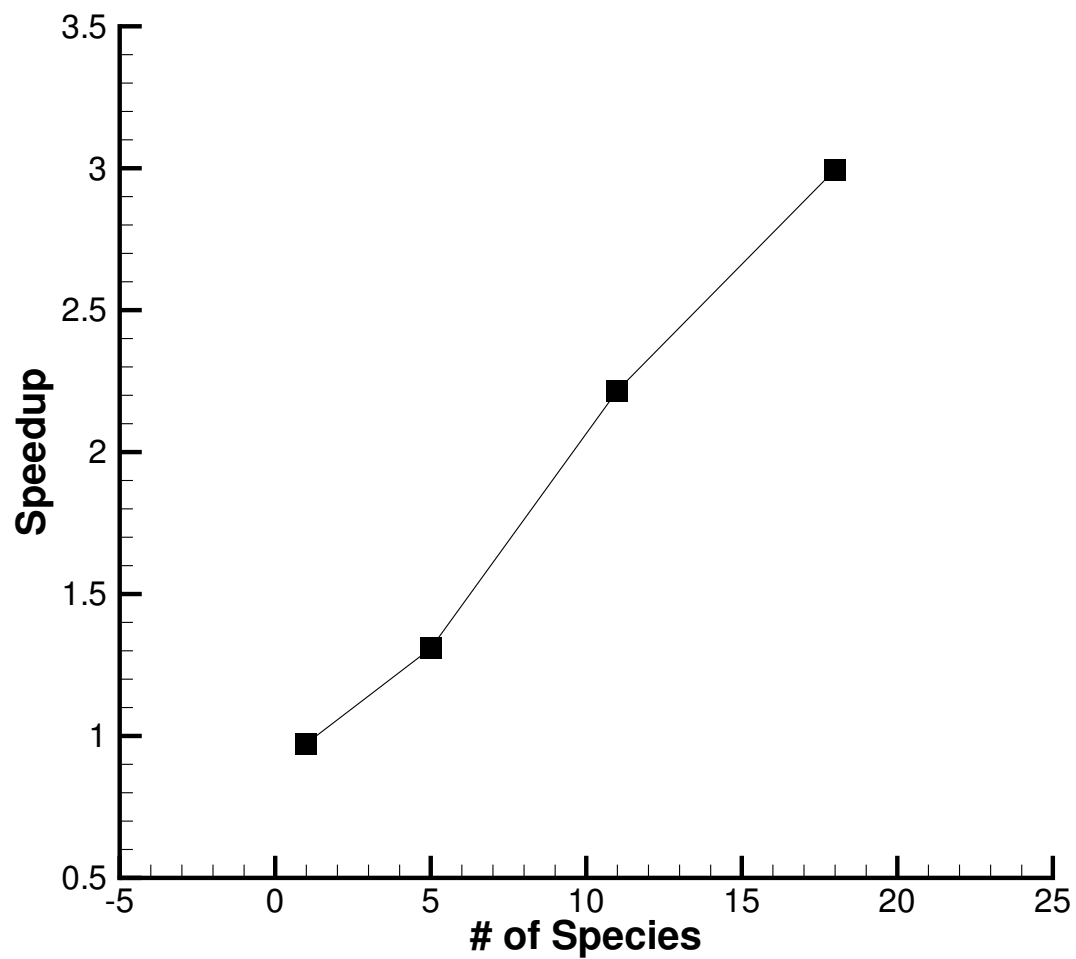


Figure 6. Overall Relative Speedup of the Decoupled Scheme wrt the Fully Coupled Scheme