Oxygen Enrichment for High Mach Number Scramjets

Nick N. Gibbons and Vincent Wheatley

The University of Queensland

Thursday, 24th September

About Me!

I am Nick:

- PhD in supersonic combustion, 2019
- Started as postdoctoral fellow @ UQ in 2020
- CFD code dev (Eilmer) and HPC sims expert
- This work sponsored by ARC grant DP230102601

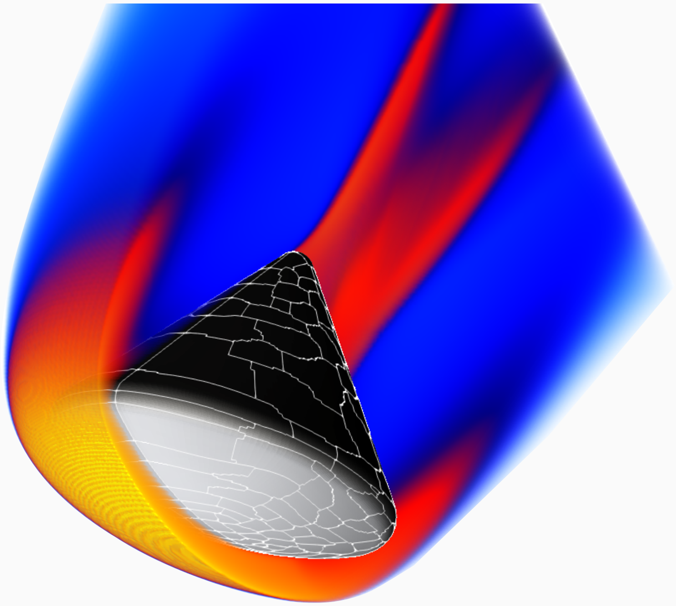
n.gibbons@uq.edu.au

nicholas-gibbons-67342555

GDTk is a collection of software tools for analysing hypersonic flow:

- Includes our flagship compressible flow code Eilmer [1]
- Specialised tools for facility design and more
- Maintained at UQ and UniSQ
- Free and Open-Source

Project Website:



Apollo capsule at 18.6° angle of attack

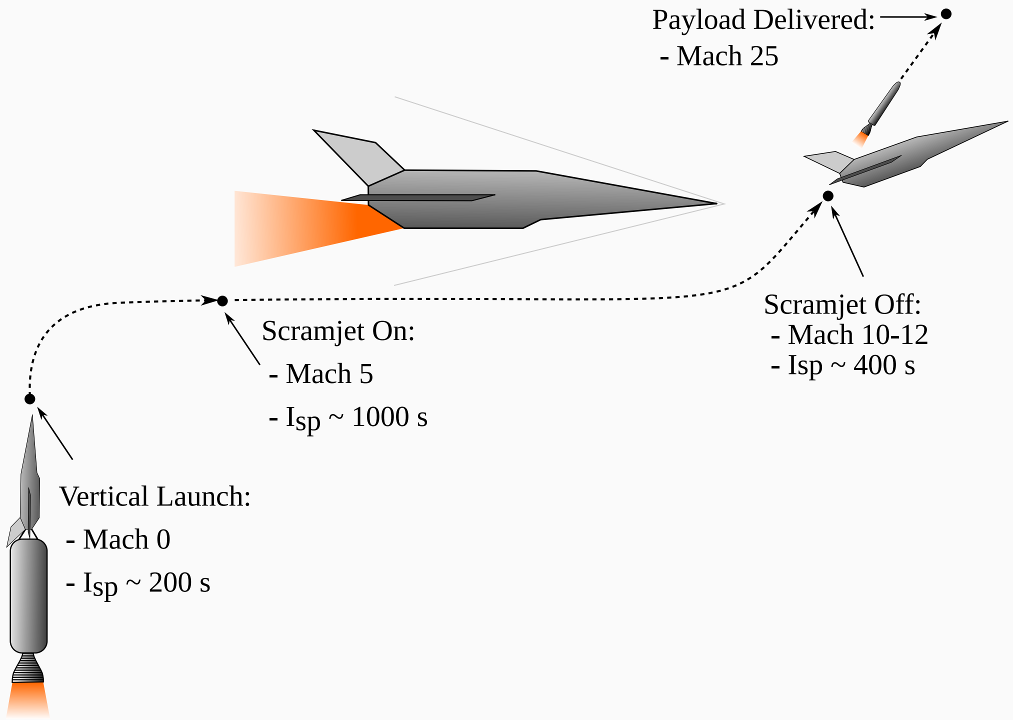
An Airbreathing 3-Stage to Orbit System for Launching Small Satellites:

Developed by Michael Smart and Matthew Tetlow [2]

- Thomas Jazra [3]
- Dawid Preller [4]
- Sholto Forbes-Spyratos [5]
- Alex Ward [6]

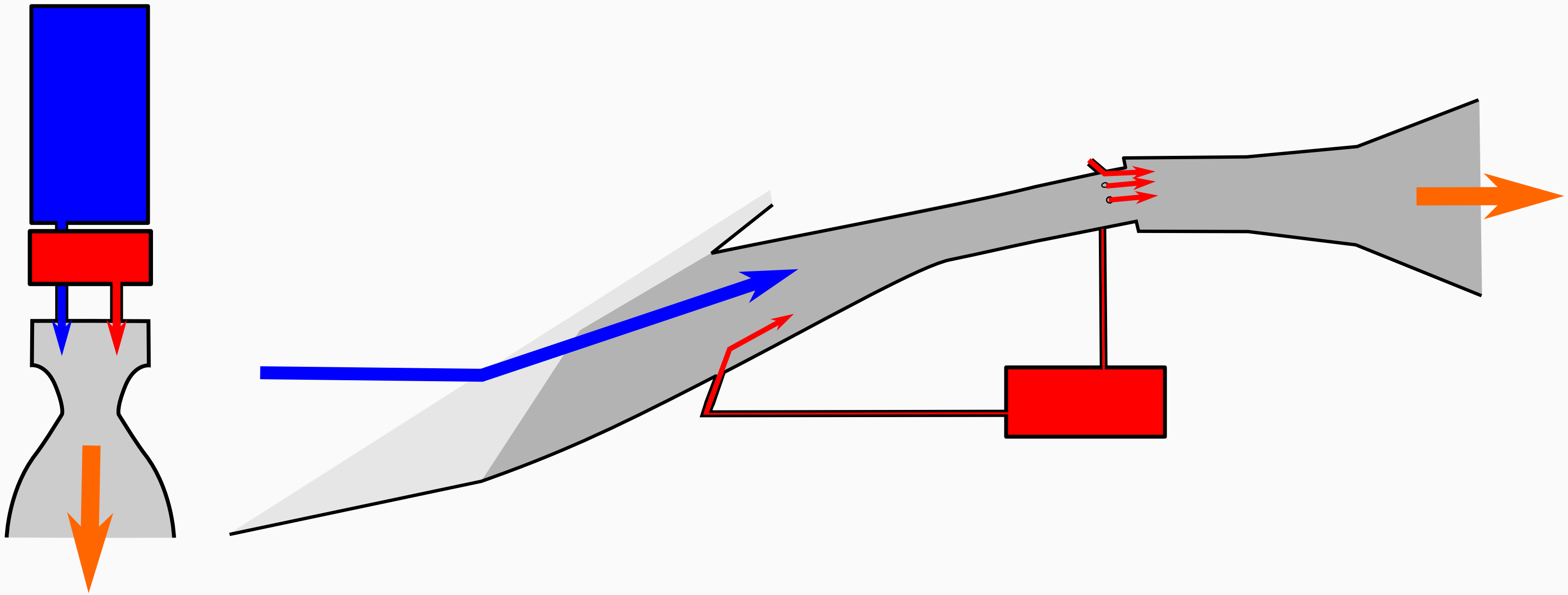
Scramjet Off point is important:

- Marginal thrust/drag performance
- Uncertainty about best Mach number
- Room for improvement with added O2?



How do scramjets work again?

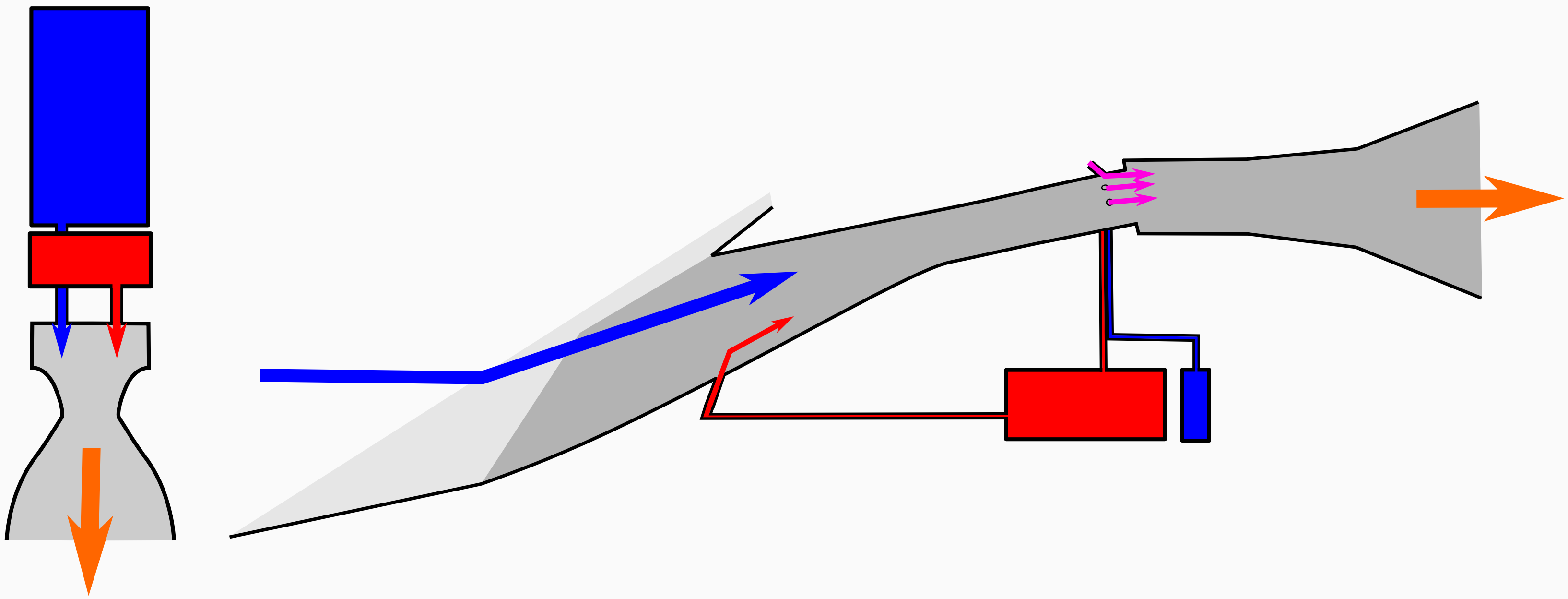
- Scramjet engines get high performance by harvesting oxygen from the atmosphere
- Significantly better Isp and size compared to a rocket



Why Oxygen Enrichment?

Accelerator scramjets start to struggle at high altitude/Mach numbers:

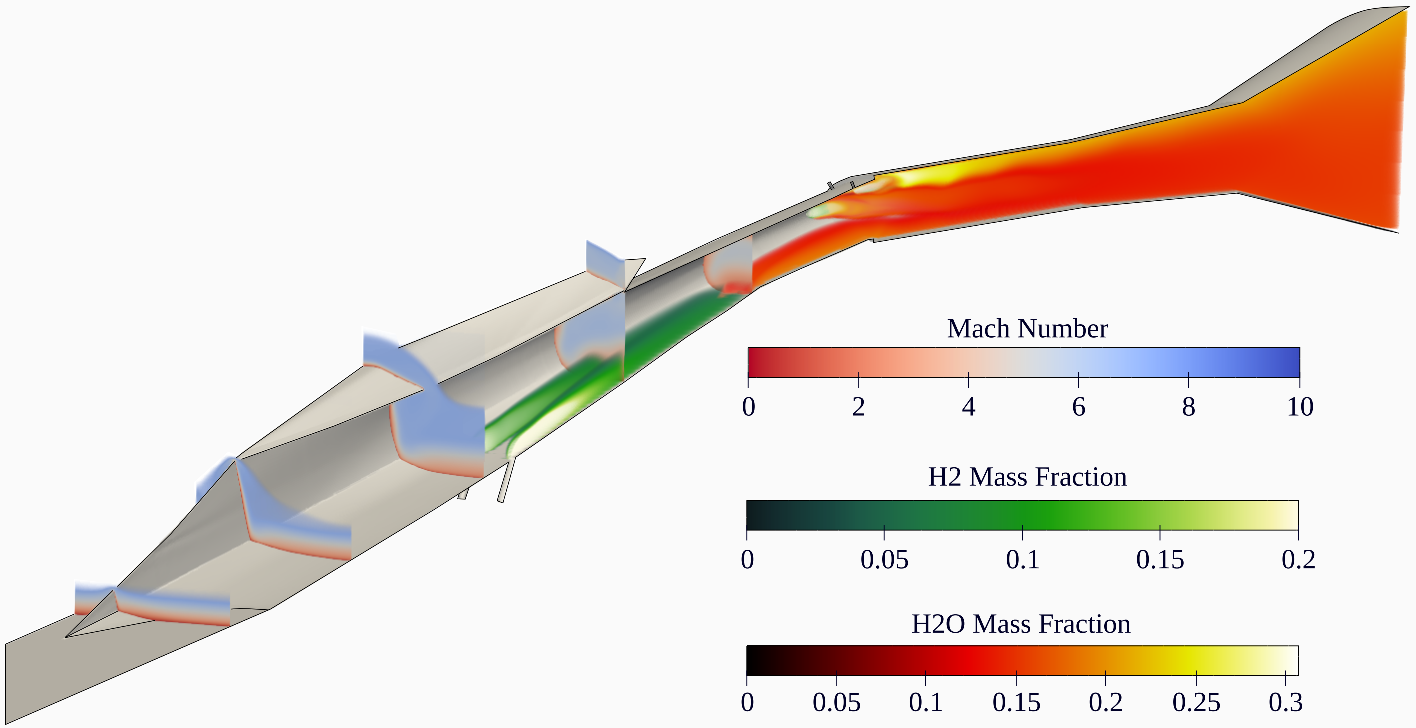
- Can we carry a *little* bit of onboard oxygen to help at the end?
- First let's simulate this problem with Eilmer



2024 work: Oxygen Enrichment for High Mach Number Scramjets

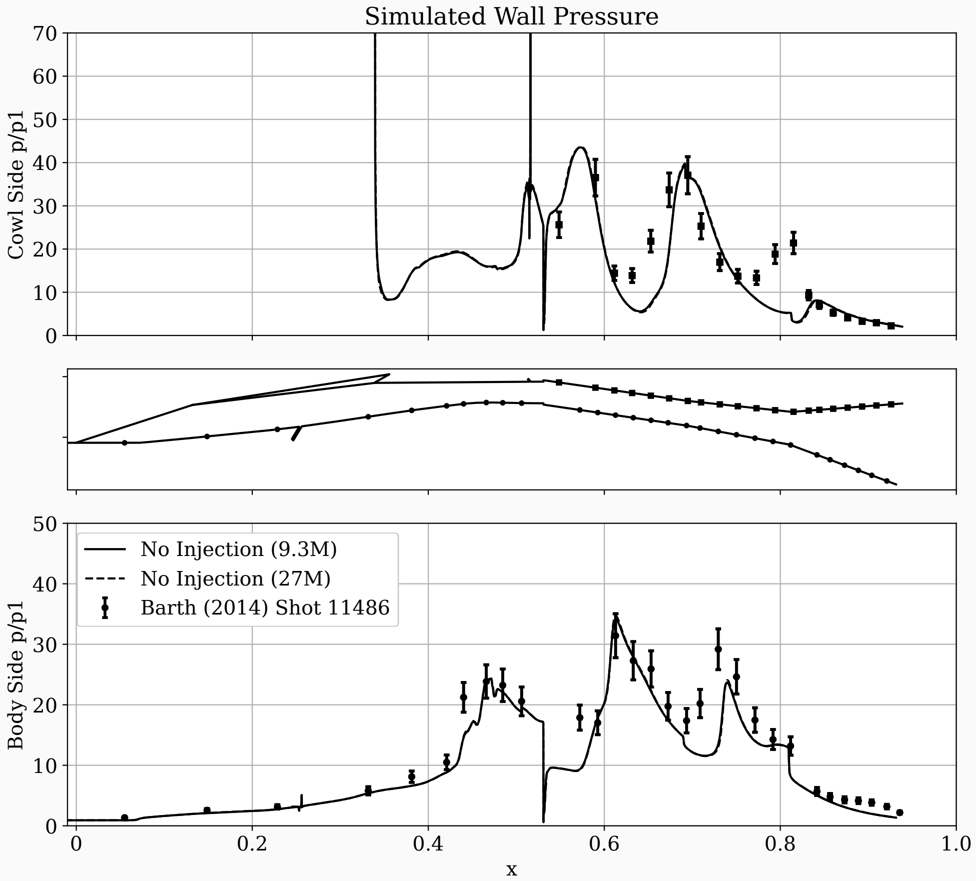
Eilmer simulations of the M12REST flowpath by me:

- Originally developed by Suraweera and Smart [7], Barth and Wheatley [8], Wise [9]:



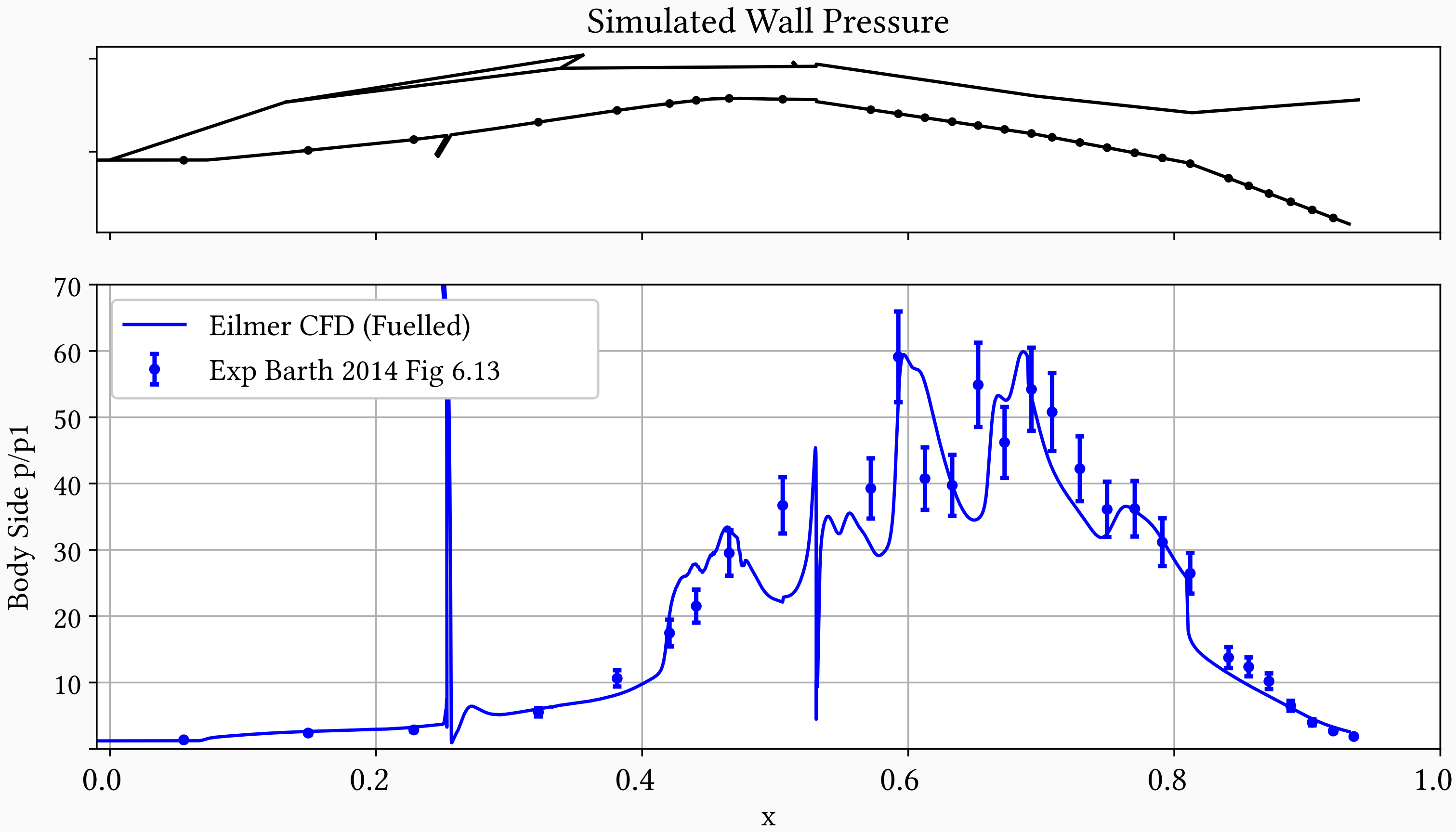
2024 work: Validation Against Experimental Data

Unfuelled RANS CFD with 1 eqn. Spalart-Allmaras-Edwards:



2024 work: Validation Against Experimental Data

Fuelled RANS CFD with Jachimowski (1992) [10]:



2024 work: Presentation at Australasian Fluid Mechanics Conference

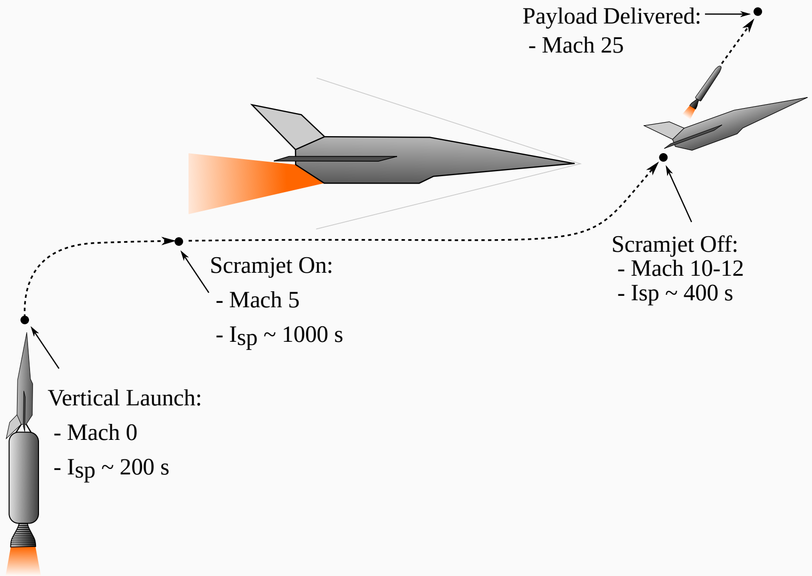
How do you guys know that this is even worth it?

- Mass penalty associated with carrying on-board oxygen
- Tank also adds a volume penalty that takes away from payload

2024 work: Presentation at Australasian Fluid Mechanics Conference

How do you guys know that this is even worth it?

- Mass penalty associated with carrying on-board oxygen
- Tank also adds a volume penalty that takes away from payload

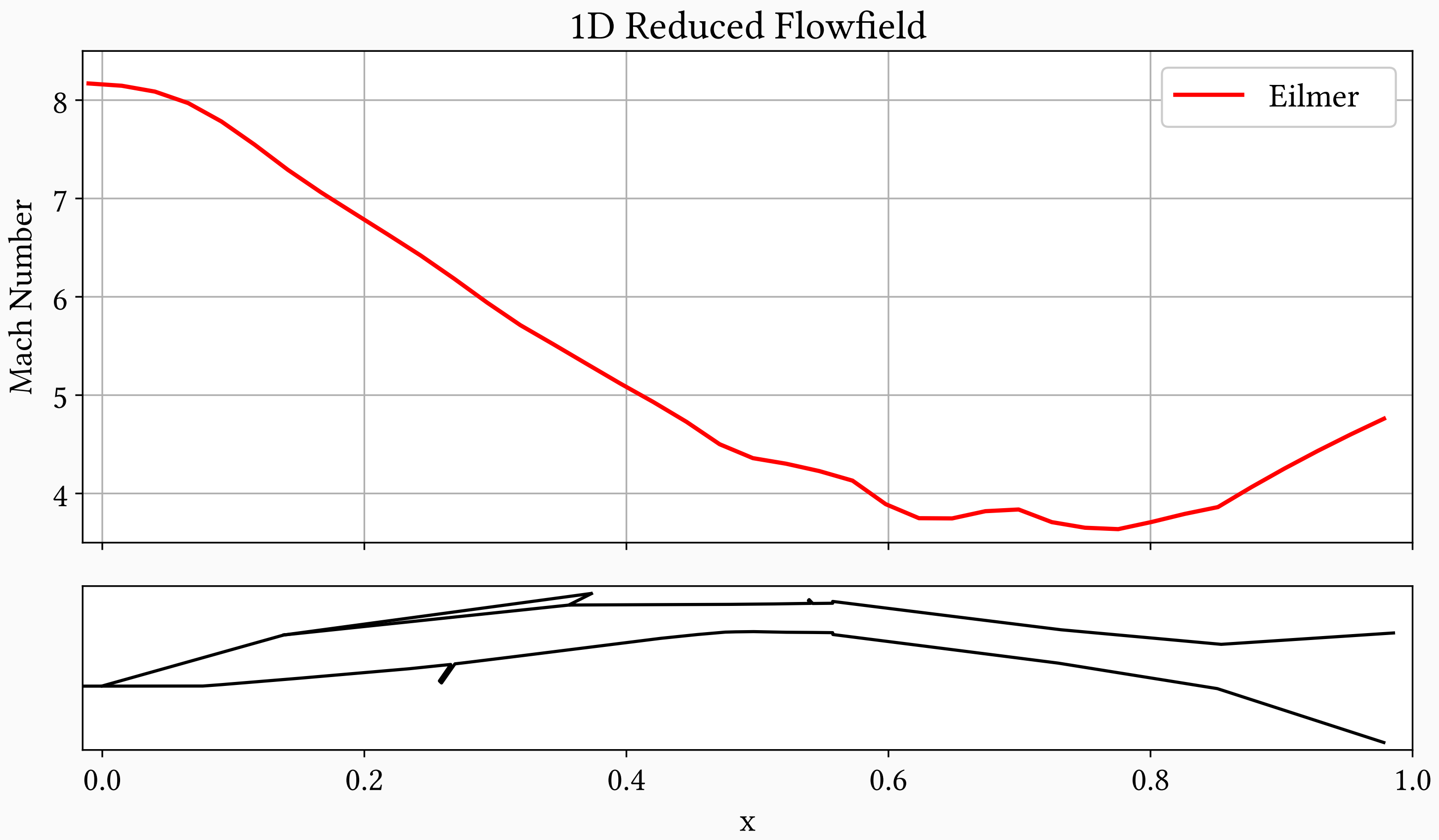


Systems-level modelling needed:

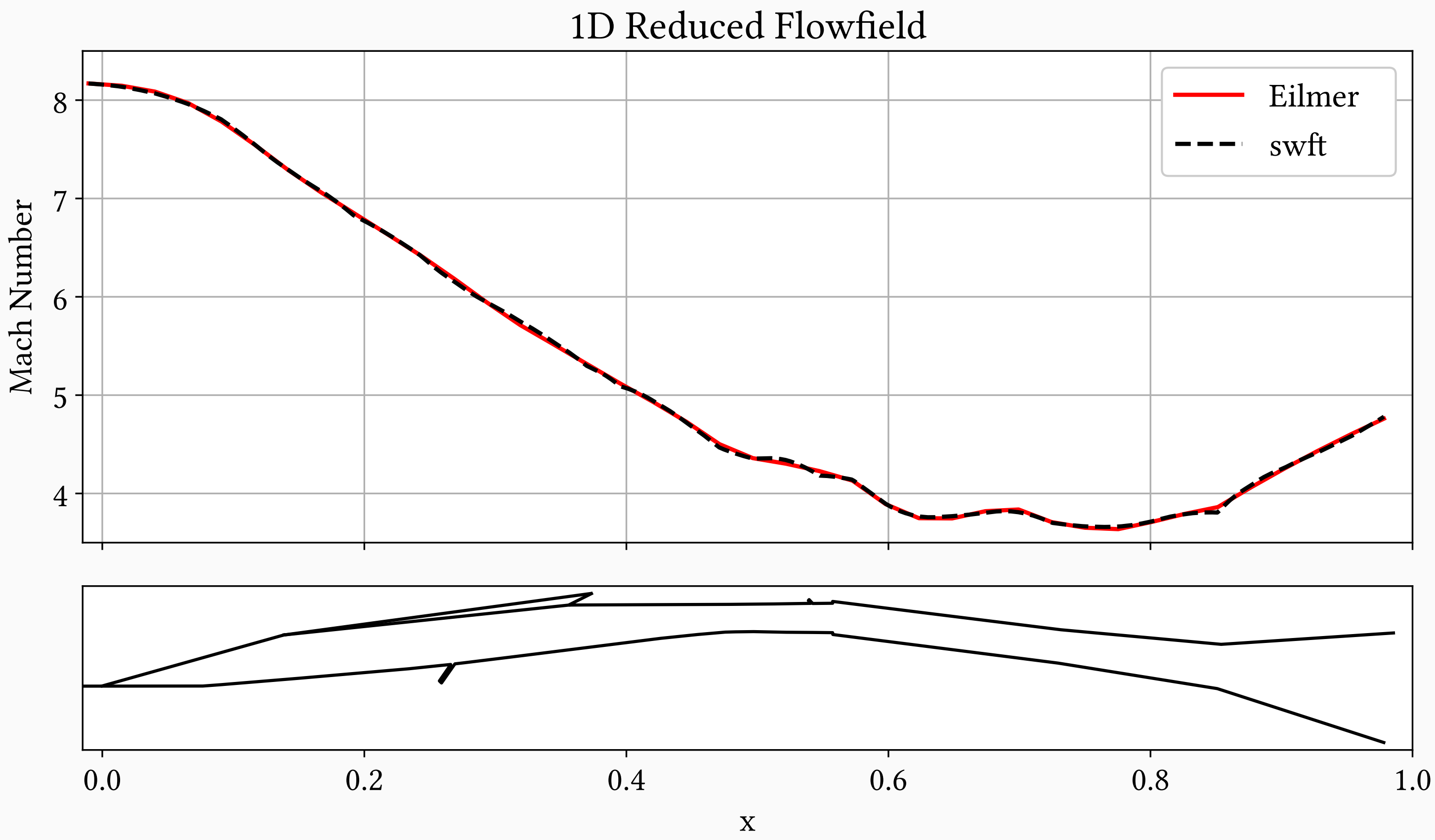
- Jack Williamson and Ingo Jahn @ UniSQ
- Some kind of CFD meta-model for the engine
- That gives me an idea...

This work: Engine metamodelling for systems-level design

Here is a 1D-ified output from the unfuelled CFD simulation:



And here is the output from my new Q1D solver, `swft`:



Aviary Envy and Swift Execution

`swft` is a Q1D supersonic solver with source terms for viscous effects:

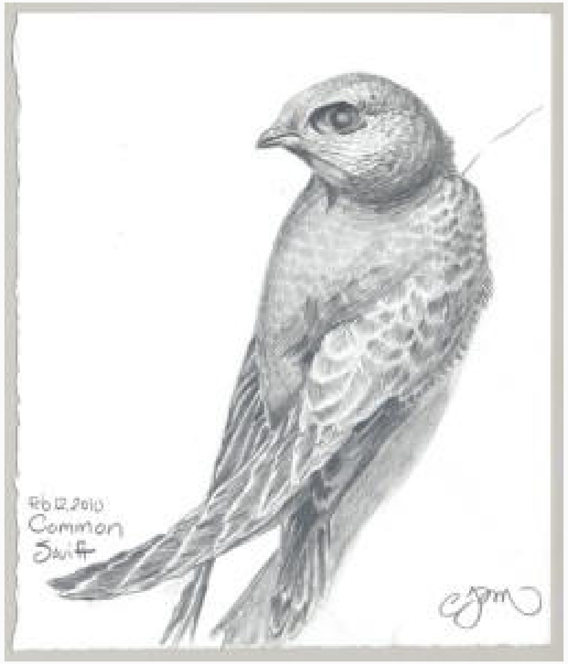
- Written in D and hooked into GDTk gas and linalg libraries
- Includes Fanno friction factor and heat loss by convection coefficient
- Space marching using symbolically inverted backward Euler step (Maxima)

Aviary Envy and Swift Execution

`swft` is a Q1D supersonic solver with source terms for viscous effects:

- Written in D and hooked into GDTk gas and linalg libraries
- Includes Fanno friction factor and heat loss by convection coefficient
- Space marching using symbolically inverted backward Euler step (Maxima)

"The Common Swift" by Claire Meyer:



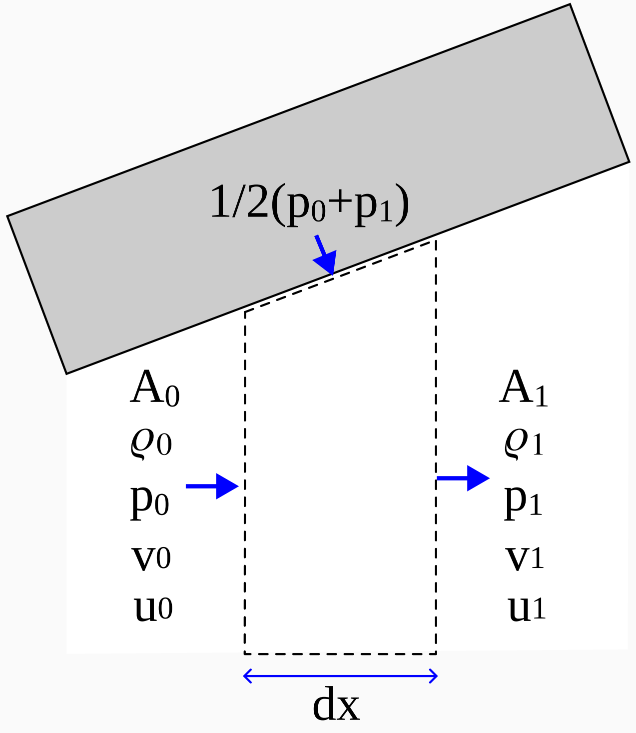
Aviary Envy and Swift Execution

```
uqngibbo@melchior:~/source/swft/tests (gdtk->opt) (git->214b3f3)

> time swft friction.yaml
swft: A Q1D Flow Analysis Tool
Init State: v=3.623000000000e+03 (m/s) T=3.610000000000e+02 (K) p=9.680000000000e+02 Pa
            M=9.511120063378e+00 rho=9.339638203028e+00 (g/m3) A=7.853980000000e-03 (m2)
            a=3.809225386556e+02 (m/s) massf: [air:1.000e+00]

Timestep 5.000000e-07, Reserving space for 1104 simdatas
Running...
[------------------------------------------------------------] 100%
Done in 553 iters

 End State: v=3.496045396594e+03 (m/s)   T=8.106970567036e+02 (K)    p=2.252770868502e+03 (Pa)
            M=6.124416193859e+00       rho=9.678773410504e+00 (g/m3) A=7.853980000000e-03 (m2)
            a=5.708373314177e+02 (m/s) massf=[air:1.000e+00]
Writing solution to file friction.bin...

real    0m0.007s
user    0m0.003s
sys     0m0.004s
```

How does it work?

Conservation of mass:

$$\rho_0 v_0 A_0 = \rho_1 v_1 A_1$$



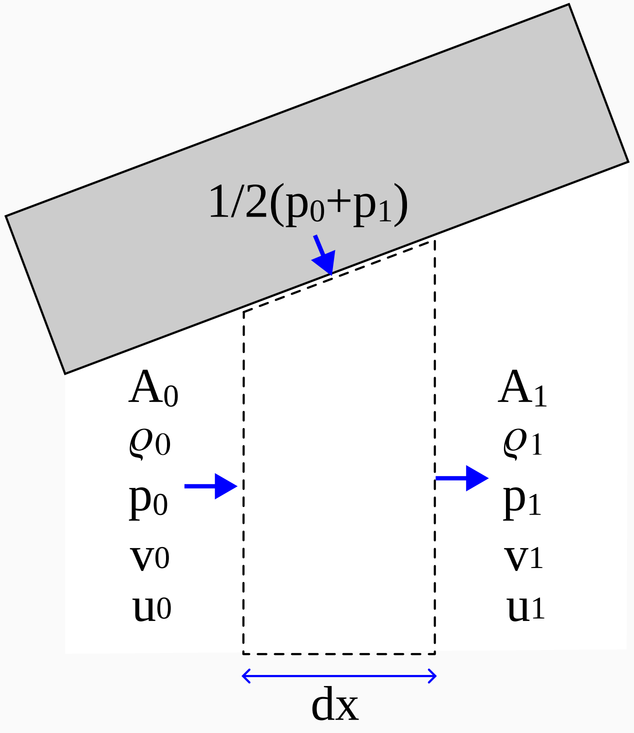
How does it work?

Conservation of mass:

$$\rho_0 v_0 A_0 = \rho_1 v_1 A_1$$

Over a small control volume, we can linearise:

$$\rho_0 v_0 A_0 = (\rho_0 + d\rho)(v_0 + dv)(A_0 + dA)$$



How does it work?

Conservation of mass:

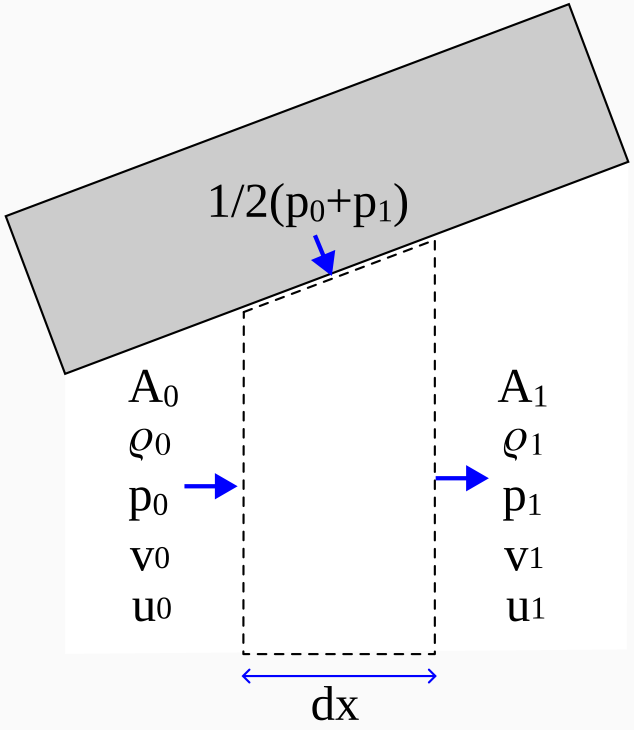
$$\rho_0 v_0 A_0 = \rho_1 v_1 A_1$$

Over a small control volume, we can linearise:

$$\rho_0 v_0 A_0 = (\rho_0 + d\rho)(v_0 + dv)(A_0 + dA)$$

Cancel higher order small terms:

$$\rho v dA + A v d\rho + A \rho dv = 0$$



How does it work?

Mass:

$$\rho v dA + Av d\rho + A\rho dv = 0$$

Momentum:

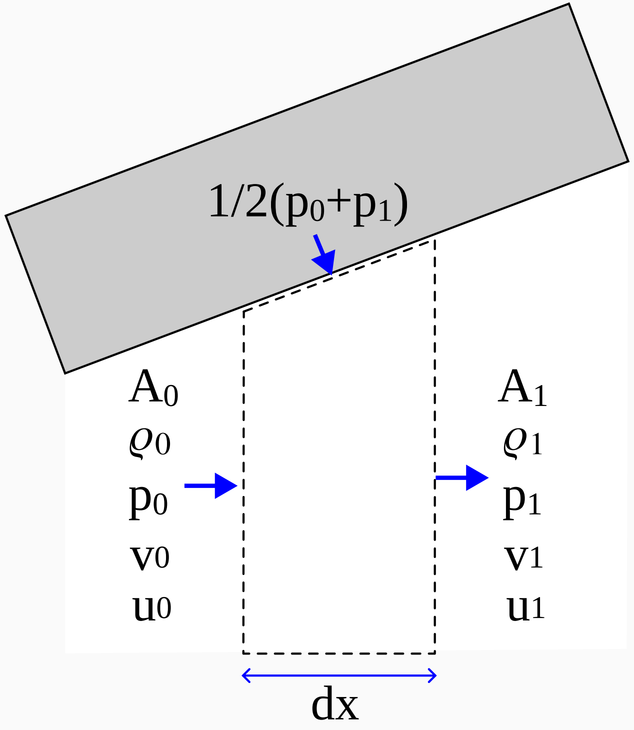
$$\rho v^2 dA + Av^2 d\rho + 2A\rho v dv + \tau\pi D dx + A dp = 0$$

Energy:

$$A dv \rho v^2 + ((A du + E dA)\rho + dA p + AE d\rho + A dp)v + AE dv \rho + A dv p - 2q_w\sqrt{\pi A} dx = 0$$

Equation of State:

$$\frac{\rho R}{c_v} du + RT d\rho - dp = 0$$



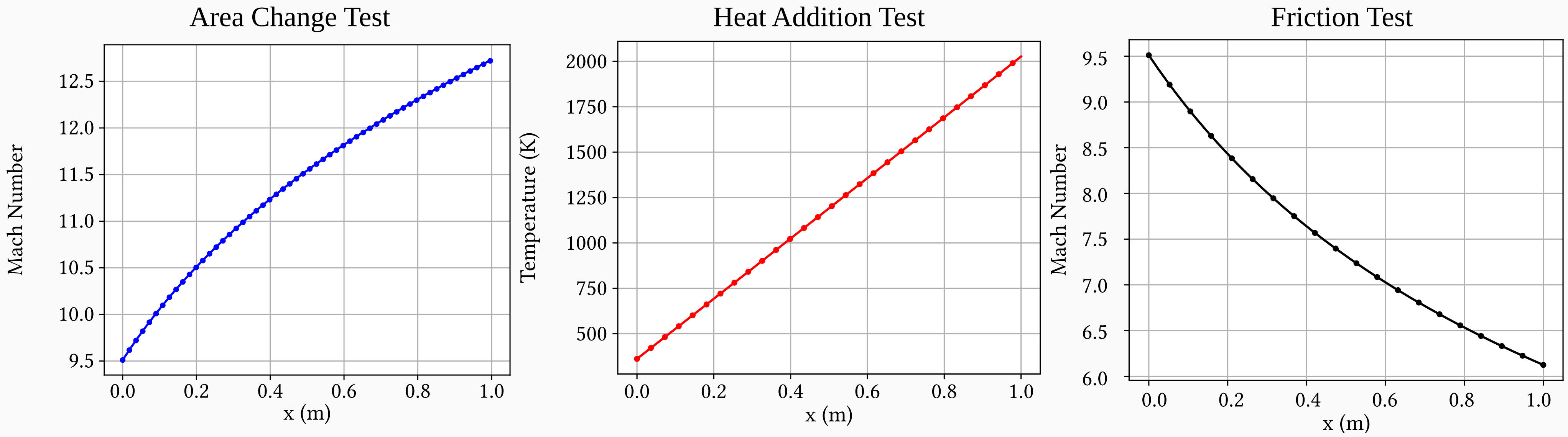
How Does It Work?

Collect increments into a linear system and solve with Maxima:

```
double denom = A*rho^^2*v^^2 - A*dfdr*rho^^2 - A*dfdu*p;

double drho = -(dA*rho^^3*v^^3+((-rho^^2)-dfdu*rho)*taupiDdx*v-Qdot*dfdu*rho)/(denom*v);

double dv = -(((rho+dfdu)*taupiDdx-dA*dfdr*rho^^2-dA*dfdu*p)*v+Qdot*dfdu)/denom;

double dp = ((dfdu*rho*taupiDdx-dA*dfdr*rho^^3-A*dp_chem*rho^^2-dA*dfdu*p*rho)*v^^2
                 +Qdot*dfdu*rho*v+(dfdr*rho^^2+dfdu*p)*taupiDdx+A*dfdr*dp_chem*rho^^2
                 +A*dfdu*dp_chem*p)/denom;

double du = ((rho*taupiDdx-dA*p*rho)*v^^3+Qdot*rho*v^^2+(p-dfdr*rho)*taupiDdx*v
                 -Qdot*dfdr*rho)/(denom*v);
```

Verification: Compressible flow relations from a textbook

Tests are good:
- Automated integration tests helped us A LOT with Eilmer
- swft was developed using a ‘test-driven’ approach



How did I match that Scramjet Result?

swft is designed to be a differentiable CFD code:

- Compute the field solution AND its derivatives w.r.t. input parameters:

How did I match that Scramjet Result?

swft is designed to be a differentiable CFD code:

- Compute the field solution AND its derivatives w.r.t. input parameters:
- We have two parameters of interest, friction factor f, and convection coefficient C_H

$$\tau = \frac{1}{8} f \rho v^2 \qquad\qquad q_w = \rho v C_H \left(u + \sqrt{\mathrm{Pr}} \rho \frac{v^2}{2} - u_w \right)$$

How did I match that Scramjet Result?

swft is designed to be a differentiable CFD code:

- Compute the field solution AND its derivatives w.r.t. input parameters:
- We have two parameters of interest, friction factor f, and convection coefficient C_H

$$\tau = \frac{1}{8} f \rho v^2 \qquad q_w = \rho v C_H \left(u + \sqrt{\mathrm{Pr}} \rho \frac{v^2}{2} - u_w \right)$$

Given $\boldsymbol{P} = [\rho, v, p, u]$, compute at each location: $\frac{\partial \boldsymbol{P}}{\partial f}$ and $\frac{\partial \boldsymbol{P}}{\partial C_H}$

The Method of Total Derivatives

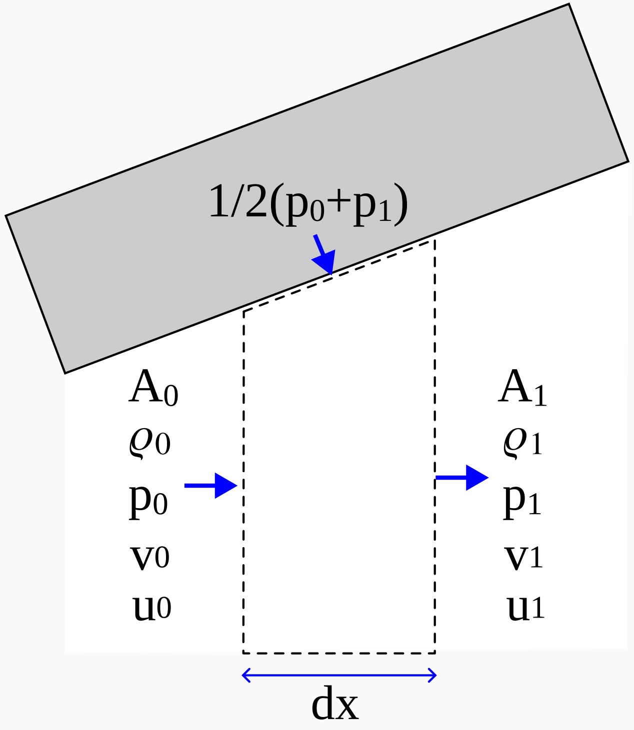
Given a vector of primitive variables $\boldsymbol{P} = [\rho, v, p, u]$, define conservation equations $\boldsymbol{R}$:

$$\boldsymbol{R}(\boldsymbol{P}) = 0$$

$$d\boldsymbol{R} = \frac{\partial \boldsymbol{R}}{\partial \boldsymbol{P}_1} d\boldsymbol{P}_1 + \frac{\partial \boldsymbol{R}}{\partial f} df$$

$$d\boldsymbol{R} = 0$$

$$\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{P}_1} \frac{d\boldsymbol{P}_1}{df} = -\frac{\partial \boldsymbol{R}}{\partial f}$$



Futher Derivatives from the Method of Total Derivatives

Unfortunately things are starting to get complicated:

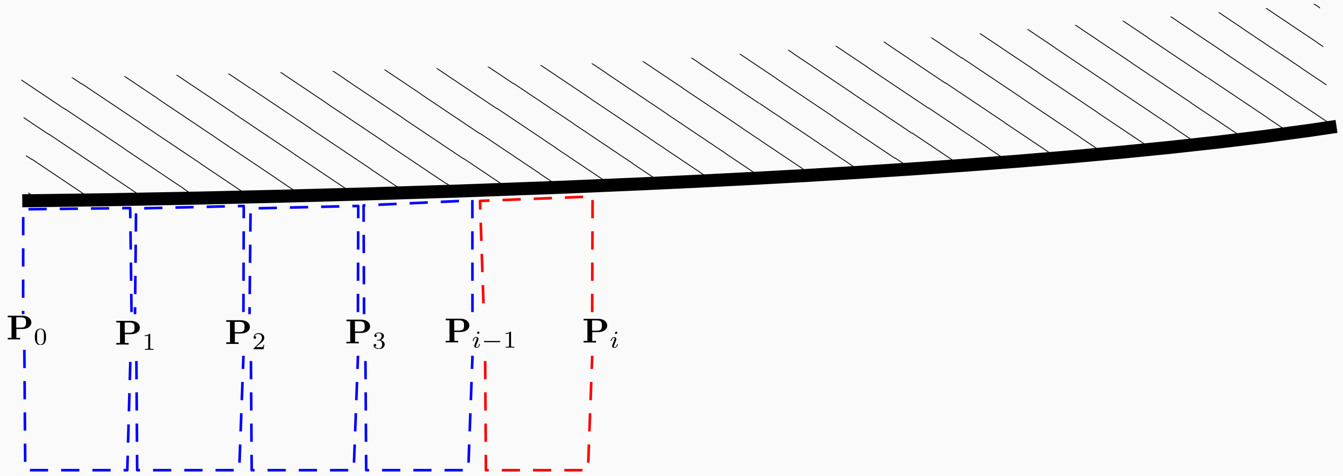
- Once we go past the first control volume, $\partial \boldsymbol{P}_0 \,/\, \partial f$ is no longer zero
- Also, $\boldsymbol{P}_1$ depends not just on f in this control volume, but all previous ones!

Futher Derivatives from the Method of Total Derivatives

Unfortunately things are starting to get complicated:

- Once we go past the first control volume, $\partial P_0 \,/\, \partial f$ is no longer zero
- Also, P_1 depends not just on f in this control volume, but all previous ones!

$$\boldsymbol{dR_i} = \frac{\partial \boldsymbol{R}_i}{\partial \boldsymbol{P}_{i-1}} \left[\sum_j^{i-1} \frac{\boldsymbol{dP}_{i-1}}{d f_j} d f_j \right] + \frac{\partial \boldsymbol{R}_i}{\partial \boldsymbol{P}_i} \boldsymbol{dP}_i + \frac{\partial \boldsymbol{R}_i}{\partial f_{i-1}} d f_{i-1} = 0$$

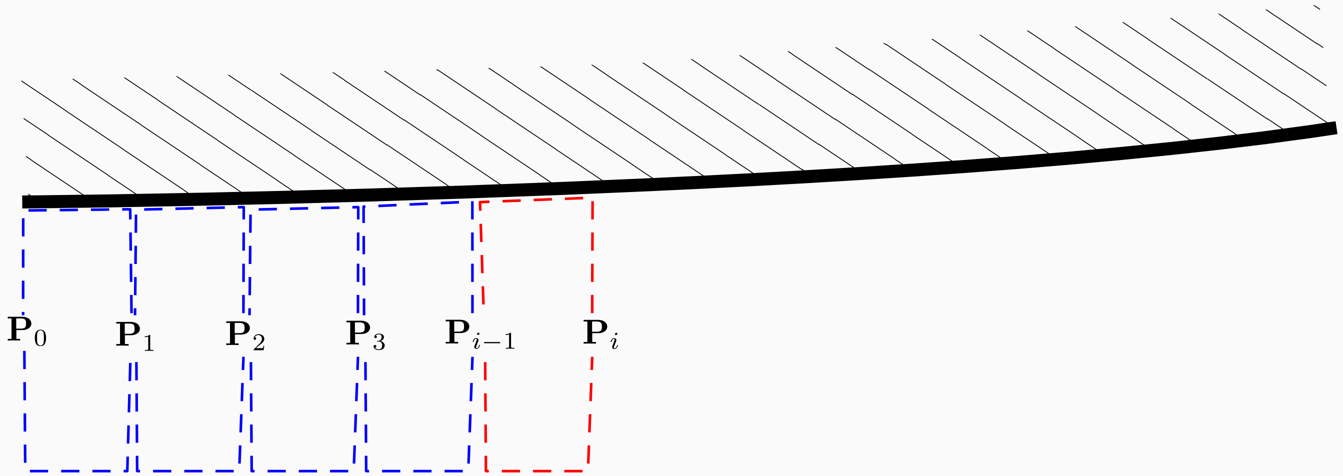


Futher Derivatives from the Method of Total Derivatives

First we consider the partial derivative of each blue df_j up to j=i-1, all others being zero:

$$0 = \frac{\partial R_i}{\partial P_{i-1}} \left[\sum_j^{i-1} \frac{dP_{i-1}}{df_j} df_j \right] + \frac{\partial R_i}{\partial P_i} dP_i + \frac{\partial R_i}{\partial f_{i-1}} df_{i-1}$$

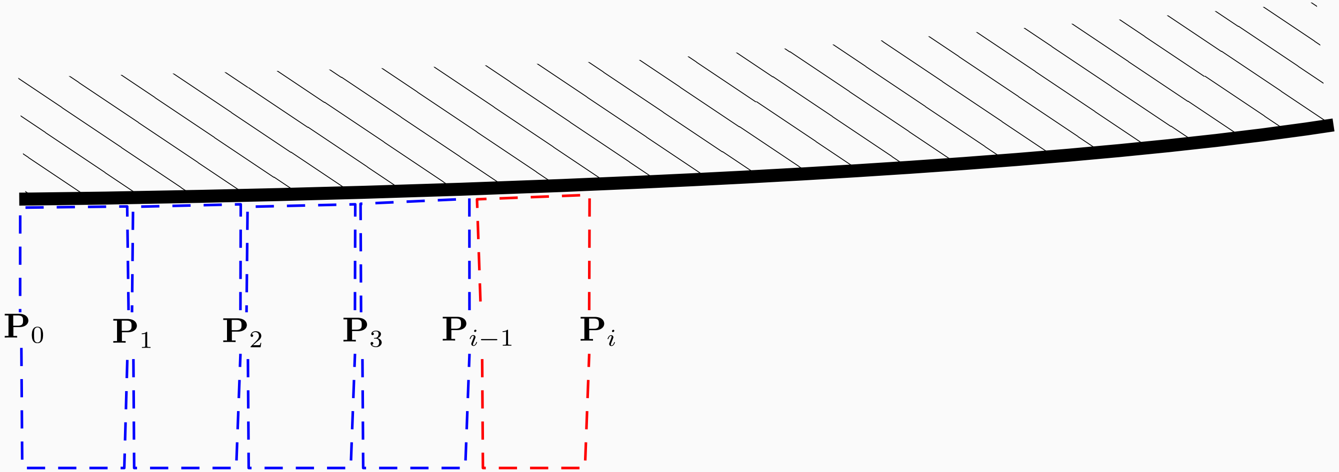
$$\frac{\partial R_i}{\partial P_i} \frac{dP_i}{df_j} = -\frac{\partial R_i}{\partial P_{i-1}} \frac{dP_{i-1}}{df_j}$$



Then we consider the final, red, control volume $df_i \neq 0$:

$$0 = \frac{\partial \boldsymbol{R}_i}{\partial \boldsymbol{P}_{i-1}}\left[\sum_{j}^{i-1} \frac{\boldsymbol{dP}_{i-1}}{df_j} df_j\right] + \frac{\partial \boldsymbol{R}_i}{\partial \boldsymbol{P}_i}\boldsymbol{dP}_i + \frac{\partial \boldsymbol{R}_i}{\partial f_{i-1}} df_{i-1}$$

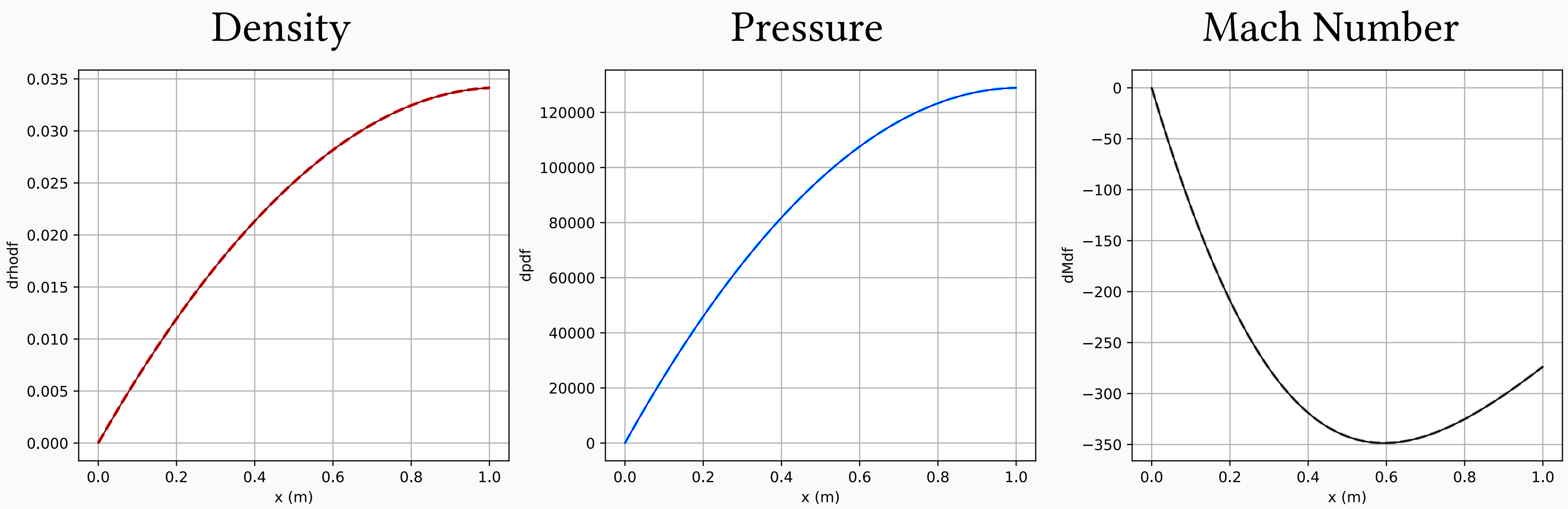
$$\frac{\partial \boldsymbol{R}_i}{\partial \boldsymbol{P}_i}\frac{\boldsymbol{dP}_i}{df_{i-1}} = -\frac{\partial \boldsymbol{R}_i}{\partial f_{i-1}}$$



Check with Finite Differencing of f:

Compute a separate simulation with a perturbed $f[0]$:

- L2 Errors: $|\rho|$=1.58e-6, $|p|$=3.763, $|v|$=0.123



Futher Derivatives from the Method of Total Derivatives

This builds a lower triangular matrix of sensitivities of the flow to each f_j:

- Actual f distributions are not fully discrete
- In practice, use a linear schedule of

$$f_{k(x)} : x_{k-1} < x \leq x_k$$

- The actual optimiser then needs to use the chain rule to compute:

Futher Derivatives from the Method of Total Derivatives

This builds a lower triangular matrix of sensitivities of the flow to each f_j:

- Actual f distributions are not fully discrete
- In practice, use a linear schedule of

$$f_{k(x)} : x_{k-1} < x \leq x_k$$

- The actual optimiser then needs to use the chain rule to compute:

$$\frac{\partial \boldsymbol{P}_i}{\partial f_k} = \sum_j \frac{\partial \boldsymbol{P}_i}{\partial f_j} \frac{\partial f_j}{\partial f_k}$$

Futher Derivatives from the Method of Total Derivatives

This builds a lower triangular matrix of sensitivities of the flow to each f_j:

- Actual f distributions are not fully discrete
- In practice, use a linear schedule of

$$f_{k(x)} : x_{k-1} < x \leq x_k$$

- The actual optimiser then needs to use the chain rule to compute:

$$\frac{\partial \boldsymbol{P}_i}{\partial f_k} = \sum_j \frac{\partial \boldsymbol{P}_i}{\partial f_j} \frac{\partial f_j}{\partial f_k}$$

Then *another* chain rule to get the objective function F derivatives w.r.t f_k:

$$\frac{\partial F}{\partial f_k} = \sum_i \frac{\partial F}{\partial \boldsymbol{P}_i} \frac{\partial \boldsymbol{P}_i}{\partial f_k}$$

Fitting a model to the Scramjet Data

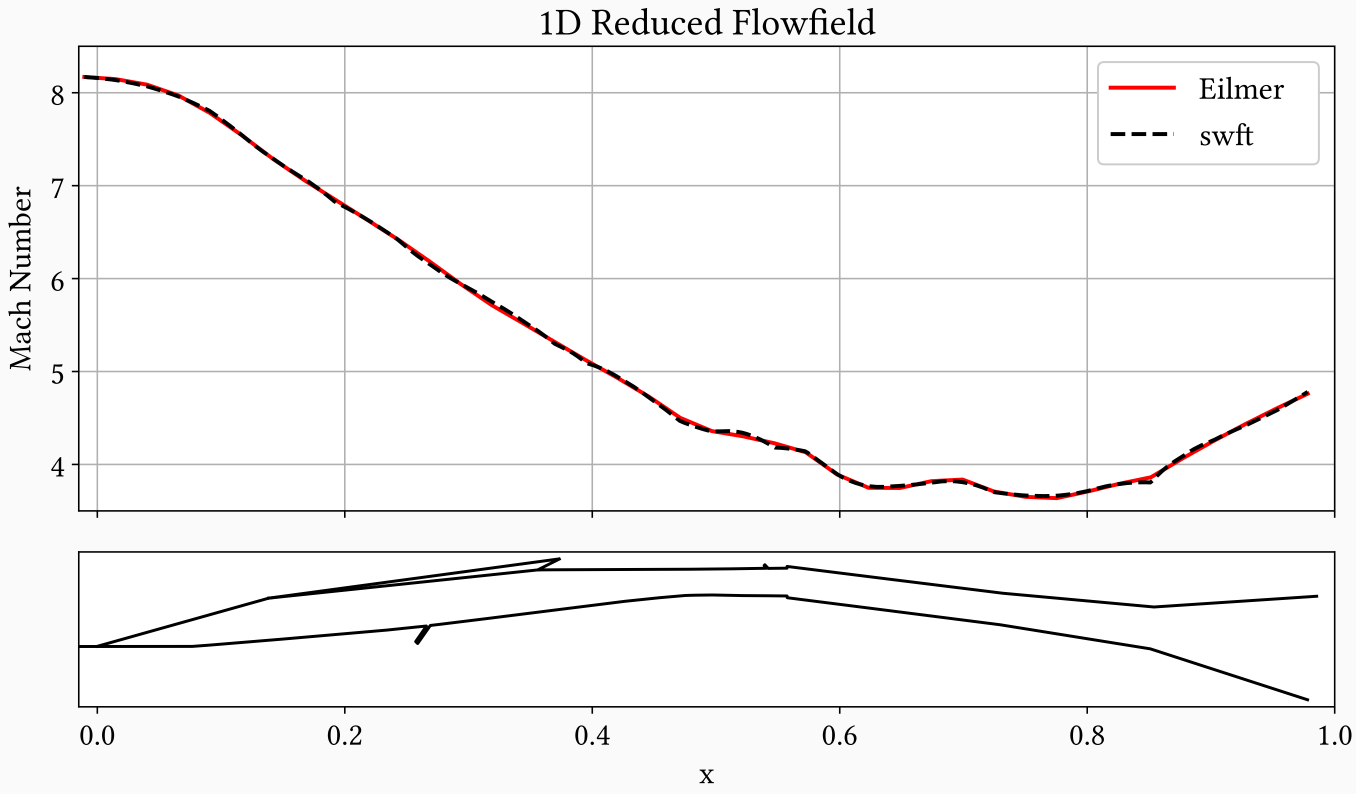
Use scipy's `minimize` in SLSQP mode with bounds:

- 24 point linear interpolation for friction factor f and single constant C_H
- Fit to the L2 norm of the Mach number error

```
nick@deepwinter:~/sims/nightmare/swft$ time python3 fit_uf.py
Optimization terminated successfully    (Exit mode 0)
            Current function value: 0.0016966193196047555
            Iterations: 161
            Function evaluations: 197
            Gradient evaluations: 161
     fun: 0.0016966193196047555
     jac: array([-4.81187900e-05,  2.82611811e-05, ... 4.72567173e-04])
 message: 'Optimization terminated successfully'
    nfev: 197
     nit: 161
    njev: 161
  status: 0
 success: True

real    0m8.112s
user    0m5.881s
sys     0m3.532s
```

Fitting a model to the Scramjet Data

Here's an animation of what this process actually looks like:



Final Result: Does this generalise to other conditions?

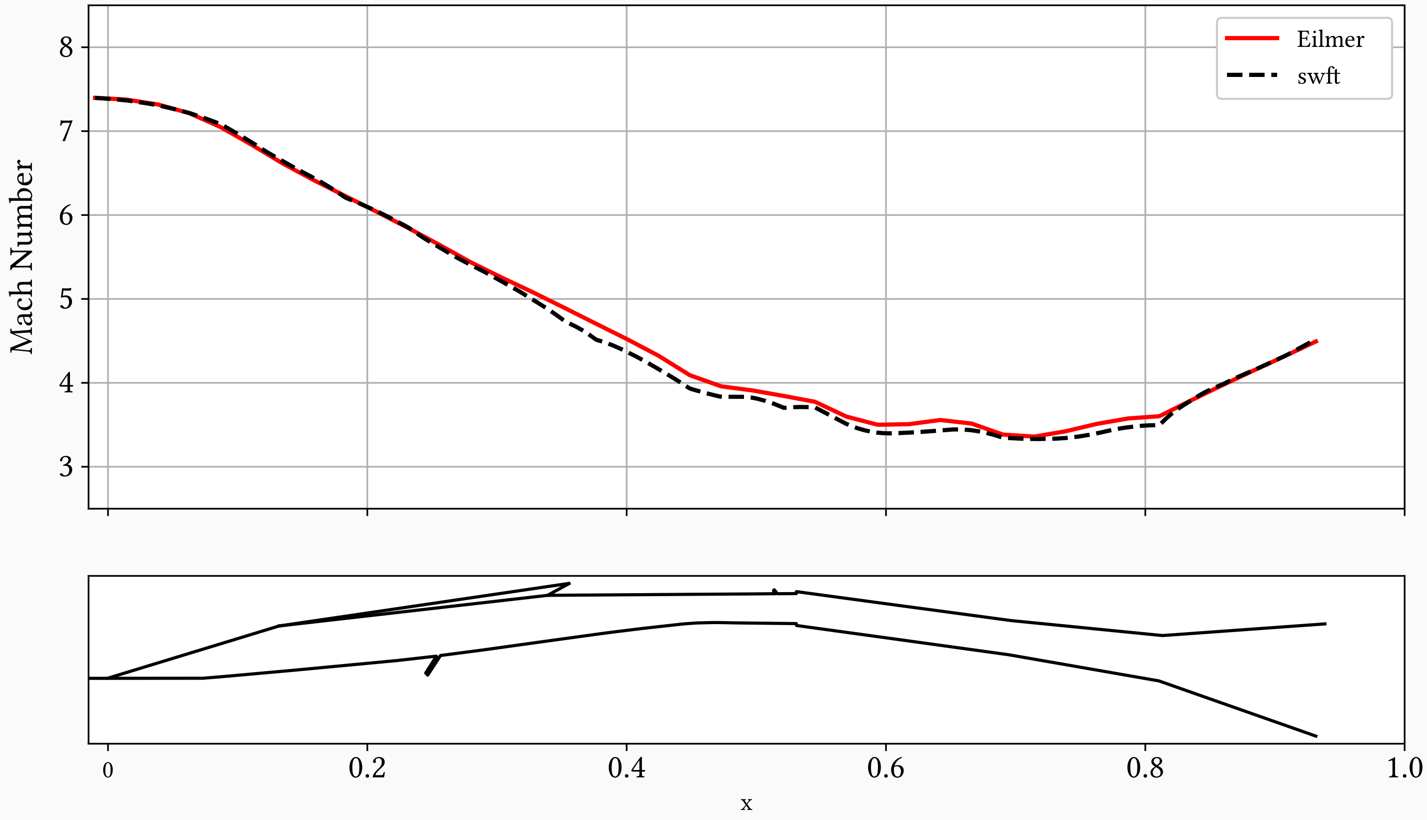
Let's try using the fitted result on a different flight condition:

- Mach 8.2 -> Mach 7.4, other properties the same

Final Result: Does this generalise to other conditions?

Let’s try using the fitted result on a different flight condition:

- Mach 8.2 -> Mach 7.4, other properties the same



Conclusions and Outlook:

Reduced order modelling vs. pure surrogate modelling:

- swft shows these two approaches can be combined
- Requires less training data and better at staying on the rails
- Gradient based optimisation is important even in small-scale

Conclusions and Outlook:

Reduced order modelling vs. pure surrogate modelling:

- swft shows these two approaches can be combined
- Requires less training data and better at staying on the rails
- Gradient based optimisation is important even in small-scale

Differentiable CFD codes open up interesting new research directions:

- Eilmer can technically do this as well
- Method of Total Derivatives vs. Adjoint
- Ideas from the audience on new things to try?

Thanks!

- The GDTk Team and Vince Wheatley
- Typst
- Much help was taken from Peter Jacobs' 2020 CfH Seminar:

Calculation of Test Flow Conditions for Reflected-Shock Tunnels

Peter Jacobs, Wilson Chan, Tamara Sopek, Fabian Zander, Kyle Damm, Nick Gibbons, Rowan Gollan

The University of Queensland (at one time or another)

03 Dec 2020

Link:

Bibliography

[1] N. N. Gibbons, K. A. Damm, P. A. Jacobs, and R. J. Gollan, "Eilmer: An open-source multi-physics hypersonic flow solver," *Computer Physics Communications*, vol. 282, no. 108551, 2023, doi: doi.org/10.1016/j.cpc.2022.108551.

[2] M. K. Smart and M. R. Tetlow, "Orbital Delivery of Small Payloads Using Hypersonic Airbreathing Propulsion," *Journal of Spacecraft and Rockets*, vol. 46, no. 1, Jan. 2019, doi: 10.2514/1.38784.

[3] T. Jazra, D. Preller, and M. K. Smart, "Design of an Airbreathing Second Stage for a Rocket-Scramjet-Rocket Launch Vehicle," *Journal of Spacecraft and Rockets*, vol. 50, no. 2, Mar. 2013, doi: 10.2514/1.A32381.

[4] D. Preller and M. K. Smart, "Reusable Launch of Small Satellites Using Scramjets," *Journal of Spacecraft and Rockets*, vol. 54, no. 6, 2017, doi: 10.2514/1.A33610.

[5] S. O. Forbes-Spyratos, "Trajectory Optimisation of a Partially-Reusable Rocket-Scramjet-Rocket Small Satellite Launch System," St Lucia, QLD 4072, 2020. doi: 10.14264/31e3e84.

[6] A. D. T. Ward, "Aftbody Design of Winged-Cone Derived Hypersonic Vehicles," St Lucia, QLD 4072, 2021. doi: 10.14264/ee04424.

[7] M. V. Suraweera and M. K. Smart, "Shock-Tunnel Experiments with a Mach 12 Rectangular-to-Elliptical Shape-Transition Scramjet at Offdesign Conditions," *Journal of Propulsion and Power*, vol. 25, no. 3, pp. 555–564, 2009.

[8] J. E. Barth, "Mixing and Combustion Enhancement in a Mach 12 Shape-Transitioning Scramjet Engine," St Lucia, QLD 4072, 2014. doi: 10.14264/uql.2014.614.

[9] D. Wise, "Experimental investigation of a three dimensional scramjet engine at hypervelocity conditions," St Lucia, QLD 4072, 2015. doi: 10.14264/uql.2015.465.

[10] C. J. Jachimowski, "An analysis of Combustion Studies in Shock Expansion Tunnels," Hampton, Virginia, Jul. 1992.