`q1dcfd`: A fast dynamic simulation framework for axially-dominated thermofluid systems

Tom Reddell, Andrew Lock, Viv Bone, Kamel Hooman, Michael Kearney, Peter Jacobs, Ingo Jahn
The University of Queensland, Brisbane, Queensland 4072, Australia

September 16, 2021

Dynamic simulation tools

- Applications:
  - dynamic optimization
  - digital twins
  - control design

- Example systems:
  - thermal power plants
  - jet engines
  - HVAC systems

- Requirements: *fast*, *accurate*, and *flexible*

Quasi-1D thermofluid systems

- Many systems are 'axially dominated'
- Pressures and velocity modelling:
  - solve full (in)compressible flow equations
- Challenges:
  - numerical methods
  - source terms to capture local 2/3D flow

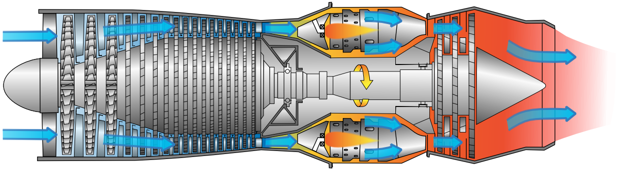


Figure 1: Jet engines consist of multiple axially-dominated gas paths

Quasi-1D flow

- Quasi-1D flow equations with source terms:

$$\frac{\partial}{\partial t}\begin{bmatrix} A\rho \\ A\rho u \\ A\rho E \end{bmatrix} + \frac{\partial}{\partial x}\begin{bmatrix} A\rho u \\ A\rho u^2 + Ap \\ A\rho u H \end{bmatrix} = \begin{bmatrix} S_{mass} \\ S_{mom} \\ S_{energy} \end{bmatrix} + \begin{bmatrix} 0 \\ p\frac{\partial A}{\partial x} \\ 0 \end{bmatrix}$$

- Characteristics: u, $u \pm a$

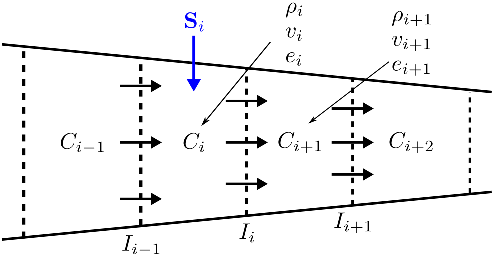
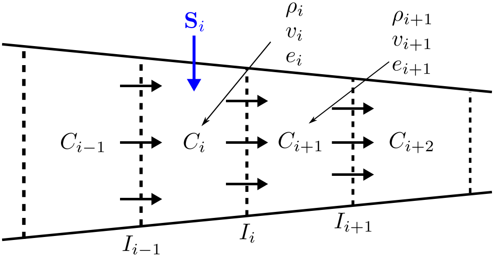


Figure 2: Quasi-1D flow schematic

Compressible flow: numerical methods

- Finite-volume solution method:
  - Reconstruct to interfaces (3rd-order MUSCL [4])
  - Compute interface fluxes (AUSMDV [5])
  - Integrate conserved variables (RK45 Cash-Karp [2])
  - Flux limiter (Van Albada [3])
- Tabular fluid properties with bicubic interpolation, or `CoolProp` [1]



Compressible flow: boundary model

- Inlet: fluid accelerates isentropically from reservoir into domain
- Modulate reservoir pressure to achieve target mass flow
- Outlet: all kinetic energy converted to heat

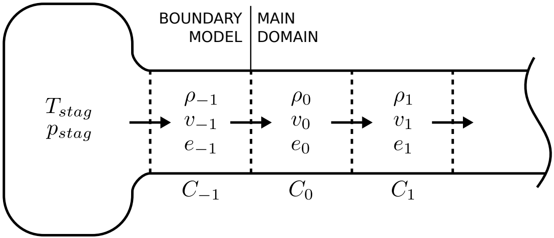


Figure 3: Compressible boundary model

Incompressible flow

- Continuity is a constraint on velocity field
- Solve with PISO algorithm (semi-implicit)
- Tolerates larger Δt: substepping

Heat transfer

- Wall thermal dynamics:

$$A_w \, \rho_w \, C_{p,\,w} \, \frac{dT_w}{dt} = q_h'' + q_c'',$$

- Nu correlation captures 2D/3D heat transfer:

$$q_h'' = N_{chans} U_h P_h (T_h - T_w), \qquad U_h = \mathsf{Nu}_h \, k / L_{C,\,h}$$

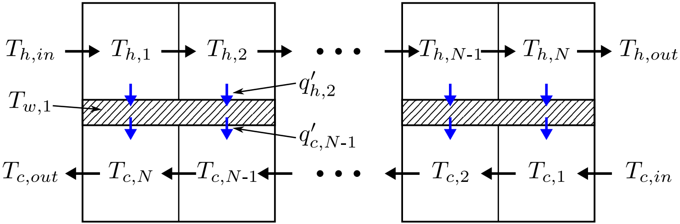


Figure 4: Heat transfer modelling — use correlation to capture multidimensional flow

Turbomachinery

- Momentum and energy discontinuity over interface I_C

$$T_{out},\, \dot{m}_{tb} = f'_{tb}(p_{in},\, p_{out},\, T_{in},\, N_s),$$

- Rotordynamics:

$$J\frac{dN_s}{dt} = T_{external} - T_{load}$$

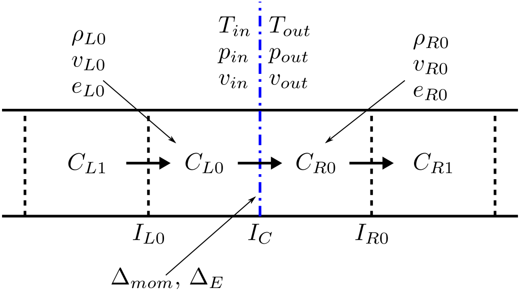


Figure 5: Turbomachinery model

Turbomachinery maps

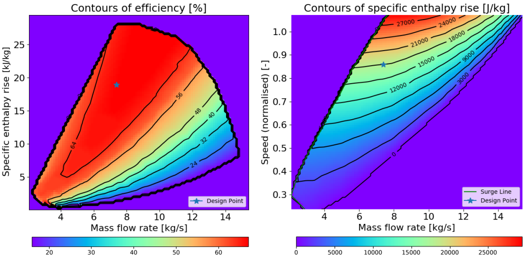


Figure 6: Compressor performance maps.

A systematic method correlation feature selection

- Often we wish to derive a correlation (such as heat transfer) from a data set, but the choice of correlation terms is unclear

- We've used a method which combines dimensional analysis with sparse (Lasso) regression to select correlation terms

- The process involves:
  1. Identify many dimensionless groups through standard Buckingham Pi analysis
  2. Linearise the correlation form (if necessary), and normalise the regressors
  3. Use Lasso sparse regression to identify the most correlated terms
  4. Use a standard regression (or other method) to determine final correlation coefficients

An example - sCO_2 heat transfer correlation

- Buckingham Pi theorem can be used to identify dimensionless groups. Carefully selected "repeated" properties are combined one at a time with the remaining properties, and the exponents are solved so all Π_j are dimensionless.

- The primary Π was Nusselt number, $Nu = \frac{hD}{k} = f(\Pi_1, \Pi_2 ... \Pi_m)$

- In our heat transfer experiment, we have four repeated variables L,k,ρ, and μ. We had many other variables, including v, c_p, g, β, $\Delta\rho$, ΔT, f, and t, and many properties could be evaluated at T_b, T_f or T_w, for $m \approx 200$.

- After selection of repeated variables, creating dimensionless groups can be automated simply by assigning each property a vector of primary dimensions, and using

$$\Pi_j \text{ exponents} = \text{null}\left(\begin{bmatrix} L_1 & L_2 & ... \\ m_1 & m_2 & ... \\ T_1 & T_2 & ... \\ t_1 & t_2 & ... \end{bmatrix}\right)^T \tag{1}$$

Linearising correlation form and normalising Π

- We assumed a correlation of the form $Nu = \beta_0 {\Pi_1}^{\beta_1} {\Pi_2}^{\beta_2} ... {\Pi_j}^{\beta_m} ...$
- In linearised form, it is $\ln(Nu) = \ln(\beta_0) + \sum_{j=1}^{m} \beta_j \ln(\Pi_j)$
- For sparse regression techniques, it is necessary to normalise regressors and the regressand

$$X_{ij}^* = \frac{X_{ij} - \mu_j}{\sigma_j} \tag{2}$$

where $X_{ij} = \ln(\Pi_{ij})$, and i refers to each data point. The regression equation then becomes

$$Y_i* = \ln(\beta_0^*) + \sum_{j=1}^{m} \beta_j^* \ln(\Pi_{ij}^*) + \varepsilon \tag{3}$$

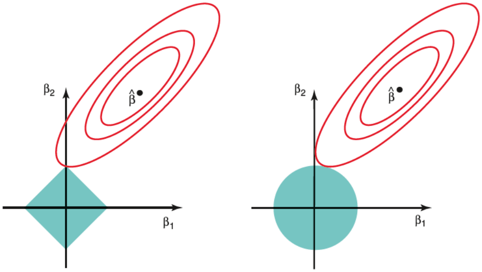
where $\beta_j^* = \beta_j \frac{\sigma_{X_j}}{\sigma_Y}$ for $i = 1...m$.

Lasso sparse regression

- Lasso (Least Absolute Shrinkage and Selection Operator) promotes sparsity in the regression coefficient vector.
- Consider two cases of regularised regression

$$\min_{\boldsymbol{\beta}} \left\{ \boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta}^T \right\} \tag{4}$$

1. subject to $||\boldsymbol{\beta}||_1 < k$ (Lasso)
2. subject to $||\boldsymbol{\beta}||_2^2 < k$ (Ridge)



Lasso sparse regression

- The standard implementation of Lasso is of the form

$$\min_{\boldsymbol{\beta}} \left\{ \frac{1}{n} ||\mathbf{Y} - \beta_0 - \mathbf{X}\boldsymbol{\beta}^{\mathrm{T}}||_2^2 + \lambda ||\boldsymbol{\beta}||_1 \right\} \tag{5}$$

- The regulariser $||\boldsymbol{\beta}||_1$ promotes sparsity in the result, and λ can be varied.

  The result looks like $\boldsymbol{\beta} = \begin{bmatrix} 0 \\ \beta_3 \\ 0 \\ ... \\ \beta_m \end{bmatrix}$

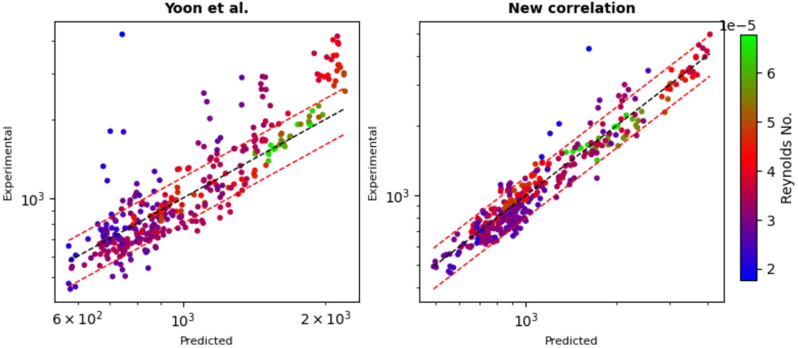
- This identifies the dimensionless groups most correlated with the data set.

- However, once the subset $\boldsymbol{\Pi}_{\mathrm{sub}}$ have been identified through sparse regression, it is useful to then use standard least squares regression, or another fit method for determining final $\boldsymbol{\beta}_{\mathrm{sub}}$.

- Using this process, four dominant Π were identified for four heat transfer process:

$$\Pi_1 = Re_{\mathrm{b}}, \quad \Pi_2 = Pr_{\mathrm{b}} \quad \Pi_3 = \frac{\rho_w - \rho_b}{\rho_b}, \quad \Pi_4 = \frac{g d^3 \rho_b^2}{\mu_b^2} \tag{6}$$

- These terms provided a strong correlation to the data set compared to existing correlations, while also being simpler with generally less terms.



Case study: control of a supercritical CO_2 cycle

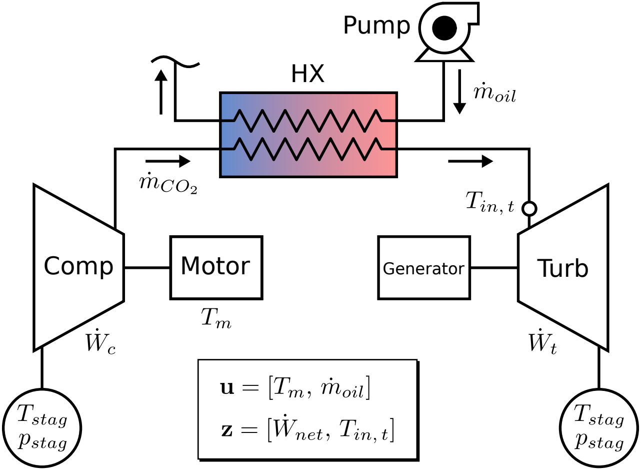


Figure 7: High-pressure side of simple sCO_2 cycle. Manipulate T_m and $\dot{m}_{oil}$ to achieve target output power

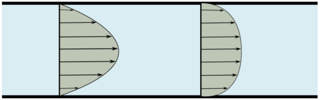
Configuring the model

```
Simulation: sco2_mpc {
    end_time        : 120.0;
    max_CFL         : 1.2;
}

Define: {
    hx_flow_area    : 7.853e-07;
    hx_perimeter    : 0.00314;
    hx_length       : 1.0;
    hx_chans        : 4000;
}

ChannelCrossSection: hx_cross_section {
    cross_area           : hx_flow_area;
    heat_circumference   : hx_perimeter;
}

HeatExchanger: hx {
    orientation                   : counterflow;
    channel[0]_cross_section      : hx_cross_section;
    channel[1]_cross_section      : hx_cross_section;
    wall_cross_section            : hx_wall_cross_section;
    cells                         : 100;
    length                        : hx_length;
    channel[0]_heat_transfer      : Ngo;
    channel[0]_initial_data       : FromData(...);
    ...
}
```

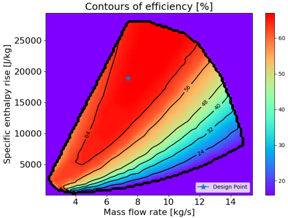
Configuring the model

```
Inflow: inflow {
    inflow_model        : InflowFromStagnation;
    inflow_area         : pipe_flow_area;
    temp_transient      : 700; // K
    massflow_transient  : CubicRamp(0.0, 550.0, 4.0, 595.0); // t0, T0, t1, T1
}

MapCompressor: compressor {
    plenum_length : 0.04;
    map_data      : sandia-compressor-data;
    rotor_inertia : 0.7;
}

Stream: co2_flow_path {
    fluid_data : sCO2;
    inflow -> inflow_pipe -> compressor -> compressor_hx_pipe
        -> hx[1] -> hx_turbine_pipe -> turbine -> outflow_pipe
        -> outflow;
}

Controller: controller {
    control_model       : MPC;
    controller_dt       : 0.3;
    ref_traj_0          : DoubleStep(-55_000, -40_000, -55_000, 20.0, 80.0);
    ref_traj_1          : Polynomial(565.0);
}

sensor-input: {
```

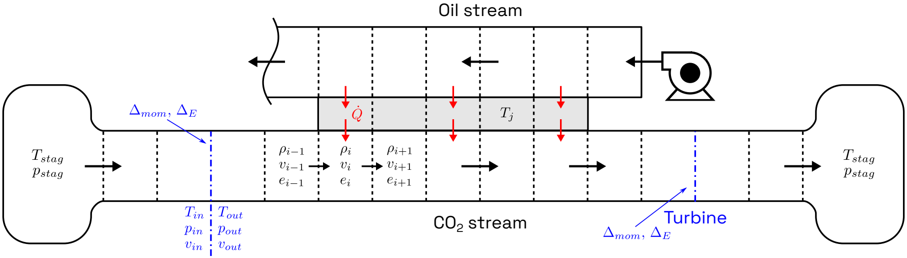
Model schematic



Boundary layer profiles

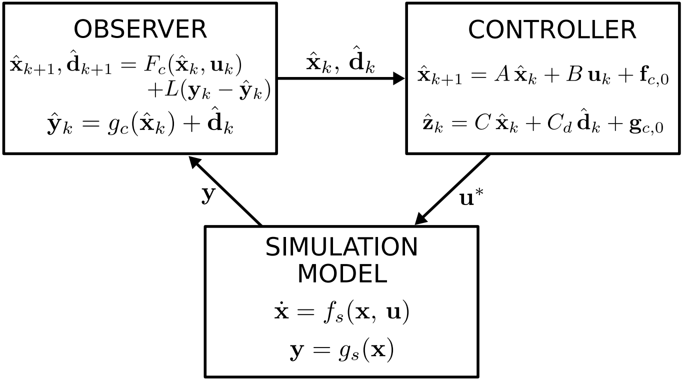


Compressor map



Simulation architecture

Closed-loop simulations



Model predictive control (MPC)

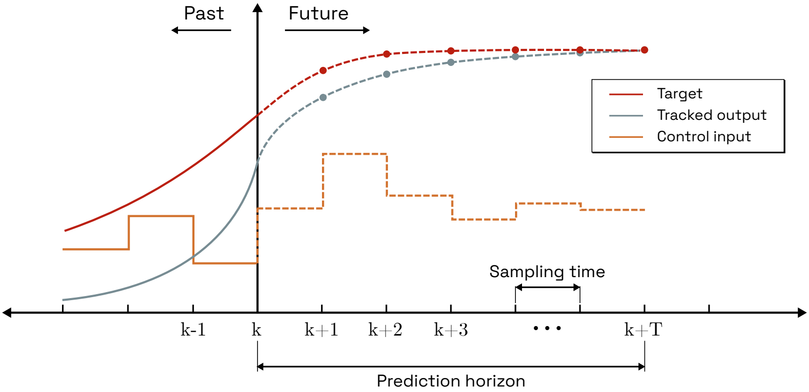


Figure 8: MPC concept — solve a constrained optimization problem looking T steps ahead to compute the control inputs

Results: design-point

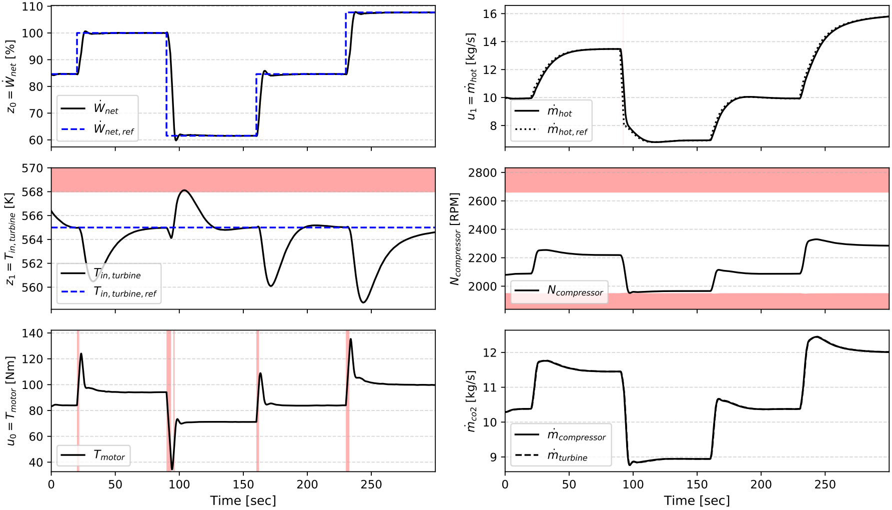
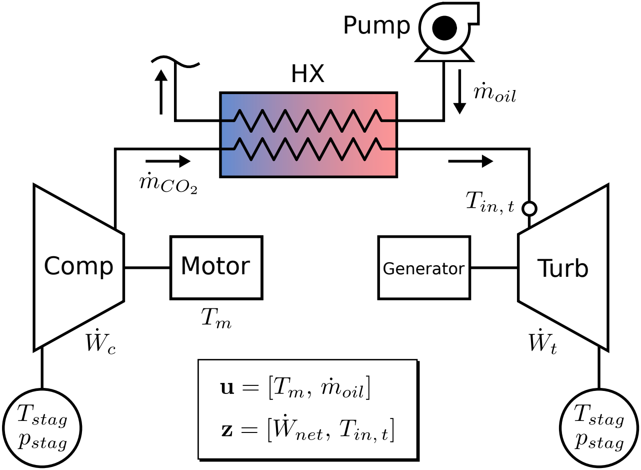


Figure 9: Design-point load change, sampling time = 0.3 sec, horizon = 15 sec

System layout



$$\mathbf{u} = [T_m,\, \dot{m}_{oil}]$$

$$\mathbf{z} = [\dot{W}_{net},\, T_{in,\,t}]$$

Results: off-design

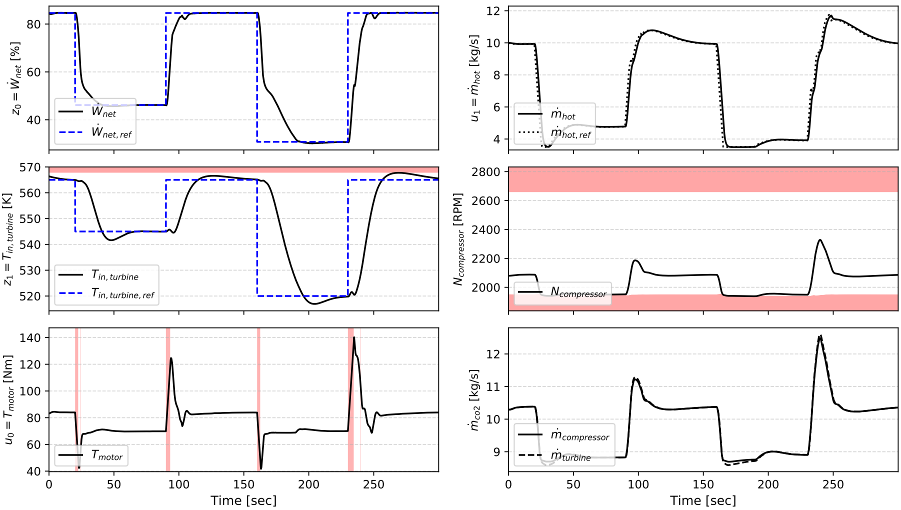


Figure 10: High turndown load change, sampling time = 0.3 sec, horizon = 15 sec

Conclusions and outlook

- Example runs 5% of real-time (nice!)
- Very robust
- Accuracy depends on source terms
- Current work:
  - Gas path branching
  - Combustion modelling
  - Interfaces with external models (e.g., external aerodynamics model for wall heating)

References

Ian H Bell et al. “Pure and pseudo-pure fluid thermophysical property evaluation and the open-source thermophysical property library CoolProp”. In: Industrial & engineering chemistry research 53.6 (2014), pp. 2498–2508.

Jeff R Cash and Alan H Karp. “A variable order Runge-Kutta method for initial value problems with rapidly varying right-hand sides”. In: ACM Transactions on Mathematical Software (TOMS) 16.3 (1990), pp. 201–222.

Gale Dick Van Albada, Bram Van Leer, and WWjun Roberts. “A comparative study of computational methods in cosmic gas dynamics”. In: Upwind and high-resolution schemes. Springer, 1997, pp. 95–103.

Bram Van Leer. “Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov’s method”. In: Journal of computational Physics 32.1 (1979), pp. 101–136.

Yasuhiro Wada and Meng-Sing Liou. “An accurate and robust flux splitting scheme for shock and contact discontinuities”. In: SIAM Journal on Scientific Computing 18.3 (1997), pp. 633–657.