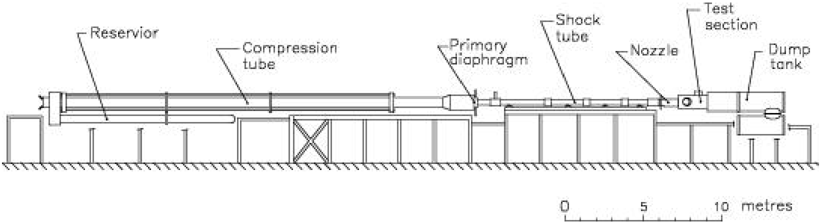
Still using NENZF? That's so 1960s.

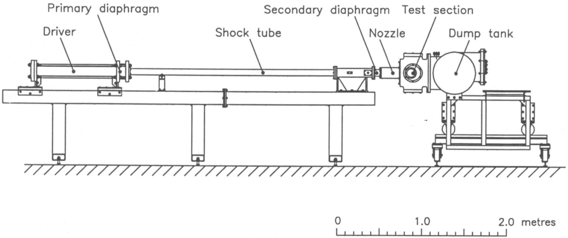
PA Jacobs, RJ Gollan, F Zander
School of Mechanical and Mining Engineering, Uni of Qld

23 March 2011

Contents:

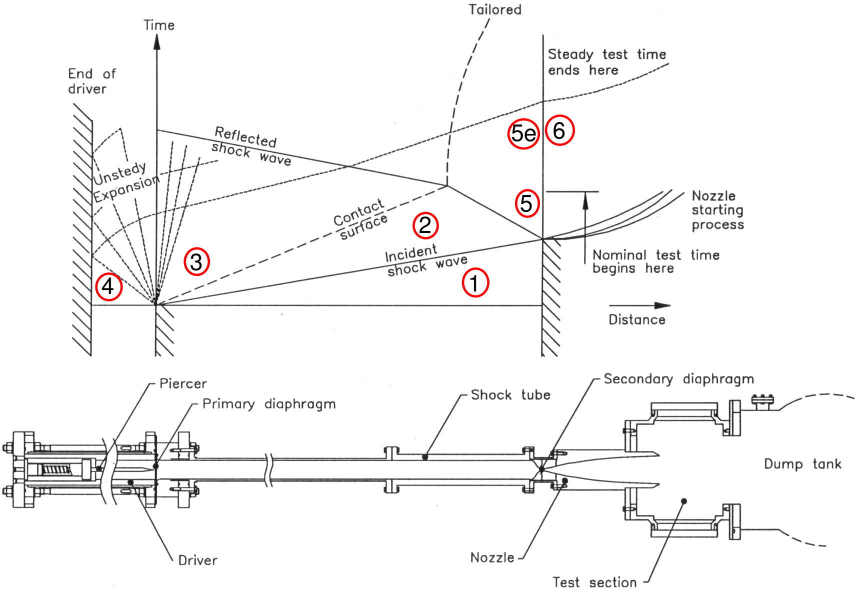
Shock tunnels at UQ





- T4: 6 km/s
- Drummond Tunnel: 2 km/s

Internal view and operation with flow states



Recipe for estimating flow conditions - 1

- Measure State 1 and State 4.
  Ideally, we could compute everything from this but it turns out that practice does not match theory and it takes months of supercomputer time to do this with any useful precision. So...
- Run the shot and measure V_s and p_e.
  Should also measure as much as we can reasonably (p_2, p_5, etc) noting that, for machines like T4, there is no one-true-value for V_s.
- Compute State 2 from State 1 and V_s, assuming a normal shock moving into quiescent gas. Although the gas will react chemically, we assume that it stays in thermochemical equilibrium.

Recipe for estimating flow conditions - 2

- Compute State 5 from State 2 assuming that the reflected shock brings the test gas to rest, as described in JD Anderson's text book. However, the pressure after the reflected shock is nothing like that expected from an ideal computation so...
- Assume an isentropic relaxation from p_5 down to the observed equilibrium value p_e to get the *nozzle-supply condition 5e*.
- Take State 5e as the stagnation conditions and expand isentropically to sonic conditions (State 6) at the throat and further to nozzle-exit conditions. It's probably OK to assume thermochemical equilibrium down through the throat but not necessarily further into the low-density and cooler parts of the nozzle expansion so use a finite-rate thermochemistry model in this last part.

Codes for estimating shock-tunnel flow conditions

Codes that have done the rounds at UQ:

- ESTC – Equilibrium Shock Tube Conditions – McIntosh 1968
- NENZF – Nonequilibrium Nozzle Flow – Lordi *et al.* 1965
- STUBE – Nonequilibrium chemistry in tube and nozzle – Vardavas 1984
- Sharc – Axisymmetric, Space-marching finite-volume – Brescianini and Morgan 1992
- Surf – Axisymmetric method of characteristics, fast chemistry, Martin Rein 1992
- STN – Shock Tube and Nozzle – Krek and Jacobs 1993
  A reimplementation of the interesting bits of ESTC extended to do the nozzle, but only for equilibrium air or ideal air/nitrogen.
- NENZFr – this presentation – ESTC+NENZF *reloaded*.

ESTC – Equilibrium Shock Tube Conditions

- Malcolm K. McIntosh (1968)
  "Computer program for the numerical calculation of frozen and equilibrium conditions in shock tunnels.",
  Unpublished Technical Report from the Department of Physics, Australian National University.
- Good, but old-school, Fortran code that is difficult to maintain.
- **You** are responsible for the thermo model. Have seen some dodgy behaviour at moderate enthalpies, presumably because of incompatible polynomial pieces.
- Small code and the parts that are needed can be rebuilt simply in Python...

ESTC – 1960s FORTRAN code

```
C INITIAL GUESSES ROR REFLECTED SHOCK (EQUILIBRIUM ONLY)
      PRES=.3826087E+02+SM*(-.5251812E+02+SM*(.1387908E+02+SM*.3781703E+
     100))
      PRES=PRES*PRESA
      CT=-.1111801E+01+SM*(.2046454E+01+SM*(.6856574E-01-SM*.321882E-02)
     1)
      ITHERM=1
      II=9
      GO TO 1001
 3008 WRITE(3,309)IRUN,CTP,RHAP,APRES,CHA,SEN,CM,VELYI,AMACH,(HP(I),GJA(
     1I),I=1,ISS)
      IF (ISW5A.NE.2) GO TO 1000
      E=SEN
C INITIAL GUESSES FOR EXPANSION TO STAGNATION CONDITIONS
      PRES=STGPR/CPRES
      ISHOCK=6
      II=10
      GO TO 1007
 3009 WRITE(3,310)IRUN,CTP,RHAP,APRES*0.1,CHA*1.0e-4,SEN,VELYI*1.0e-2,
     1AMACH,CM,(HP(I),GJA(I),I=1,ISS)
 1000 IF (ISW1A.NE.IZERO) GO TO 1
 1006 CALL EXIT
 1001 IF(ISW6A.EQ.IZERO) GO TO 1007
      READ(1,100)PRES,CT
      IF (ISW6A.EQ.2) PRES=PRES*PRESA
      IF (ISW6A.EQ.1) CT=CT/CTAP
 1007 PRESI=PRES /PRESA
      TEMPI=CT
      ICALL=0
      IER1=0
      IER2=1
 1005 CONTINUE
      CALL NRAFH(PRES,CT)
      RPRES=PRES/PRESA
      GO TO (3000,3001,3002,3003,3004,3005,3006,3007,3008,3009),II
      END
```

ESTC – 1960s spaghetti code

```
C INITIAL GUESSES ROR REFLECTED SHOCK (EQUILIBRIUM ONLY)
      PRES=.3826087E+02+SM*(-.5251812E+02+SM*(.1387908E+02+SM*.3781703E+
     100))
      PRES=PRES*PRESA
      CT=-.1111801E+01+SM*(.2046454E+01+SM*(.6856574E-01-SM*.321882E-02)
     1)
      ITHERM=1
      II=9
      GO TO 1001
 3008 WRITE(3,309)IRUN,CTP,RHAP,APRES,CHA,SEN,CM,VELYI,AMACH,(HP(I),GJA(
     1I),I=1,ISS)
      IF (ISW5A.NE.2) GO TO 1000
      E=SEN
C INITIAL GUESSES FOR EXPANSION TO STAGNATION CONDITIONS
      PRES=STGPR/CPRES
      ISHOCK=6
      II=10
      GO TO 1007
 3009 WRITE(3,310)IRUN,CTP,RHAP,APRES*0.1,CHA*1.0e-4,SEN,VELYI*1.0e-2,
     1AMACH,CM,(HP(I),GJA(I),I=1,ISS)
 1000 IF (ISW1A.NE.IZERO) GO TO 1
 1006 CALL EXIT
 1001 IF(ISW6A.EQ.IZERO) GO TO 1007
      READ(1,100)PRES,CT
      IF (ISW6A.EQ.2) PRES=PRES*PRESA
      IF (ISW6A.EQ.1) CT=CT/CTAP
 1007 PRESI=PRES /PRESA
      TEMPI=CT
      ICALL=0
      IER1=0
      IER2=1
 1005 CONTINUE
      CALL NRAFH(PRES,CT)
      RPRES=PRES/PRESA
      GO TO (3000,3001,3002,3003,3004,3005,3006,3007,3008,3009),II
      END
```

NENZF – Nonequilibrium Nozzle Flow

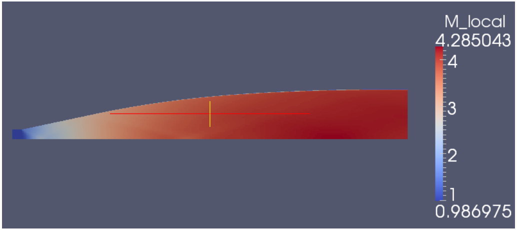
- J.A. Lordi, R.E. Mates, and J.R. Moselle (1965) "Computer program for the numerical solution of non-equilibrium expansions of reacting gas mixtures.", NASA Contractor Report 472.
- Fast steady-state analysis produces simple (single number) values for nozzle-exit flow properties.
- Same thermo model as for ESTC; same responsibilities.
- Has trouble producing answers for low enthalpies.
- Several versions of the code floating around, even within the UQ group. It's essentially unmaintained.

NENZF thermo – who's fiddled with *my* polynomials

```
11111111
  2.5      E+00
  0.0      E+00  -1.1735   E+01  0.0      E+00              E-
    E- 1.0      E+00-1.492823 E+01 0.0      E+00 0.0      E+00 1
  2   0.0     E+00
  3.451483 E+00  3.088332 E-04 -4.251428 E-08  2.739295 E-12
 -5.46832  E-17  3.071269 E+00  0.0      E+00              N2
    N2 2.0      E+00-4.2163   E-01 3.35324  E+03 0.0      E+00 4
  1   0.0     E+00  3   1.43685 E+05  6   1.70475 E+05  1   1.754   E+05
  3.249473 E+00  4.963449 E-04 -6.701753 E-08  4.443339 E-12
 -1.000281 E-16  5.915022 E+00  0.0      E+00              O2
    O2 2.0      E+00 1.0745   E-01 2.23897  E+03 0.0      E+00 5
  3   0.0     E+00  2   2.2037  E+04  1   3.7725  E+04  3   1.03198 E+05
  3   1.4239  E+05
  2.563282 E+00 -3.59177  E-05  7.469208 E-09 -6.747034 E-13
  2.234019 E-17  4.000939 E+00  0.0      E+00              AR
    AR 1.0      E+00 1.86557  E+00 0.0      E+00 0.0      E+00 3
  1   0.0     E+00  5   2.66307 E+05  3   2.68042 E+05
  3.008922 E+00 -3.134625 E-04  6.311813 E-08 -4.165203 E-12
  9.334886 E-17  1.303476 E+00  1.125906 E+05               N
     N 1.0      E+00 2.9868   E-01 0.0      E+00 1.125906 E+05 5
  4   0.0     E+00  6   5.4974  E+04  4   5.5125  E+04  6   8.2455  E+04
 12   2.3821  E+05
  2.594143 E+00 -5.008914 E-05  1.199502 E-08 -8.681611 E-13
  2.1481   E-17  4.600615 E+00  5.898    E+04               O
     O 1.0      E+00 4.932    E-01 0.0      E+00 5.898    E+04 6
  5   0.0     E+00  3   4.5462  E+02  1   6.4898  E+02  5   4.5368  E+04
  1   9.6615  E+04  5   2.10907 E+05
  3.756216 E+00  2.083961 E-04 -2.639548 E-08  1.690332 E-12
 -3.611523 E-17  3.611167 E+00  2.1477   E+04              NO
    NO 2.0      E+00 5.3941   E-01 2.69918  E+03 2.1477   E+04 3
  4   0.0     E+00  2   1.257   E+05  4   1.31283 E+05
  3.397385 E+00  3.749384 E-04 -6.06203  E-08  4.637506 E-12
 -1.107704 E-16  4.200563 E+00  2.3533   E+05             NO+
   NO+ 2.0      E+00 3.7861   E-01 3.37295  E+03 2.3533   E+05 6
  1   0.0     E+00  6   1.15232 E+05  3   1.68987 E+05  6   2.08732 E+05
  2   2.0959  E+05
```

NENZFr = ESTCj + CEA2 + Eilmer3

- NENZFr – Nonequilibrium Nozzle Flow *reloaded*
- It's actually a coordinating script in Python that uses:
  - CEA2 – Chemical Equilibrium Analysis 2
    equilibrium-thermochemistry "module" for ESTCj and Eilmer3
    Someone else does the maintenance.
  - ESTCj – Equilibrium Shock Tube Conditions (Junior)
    One-dimensional, quasi-steady, decoupled wave-processing
    (just the interesting bits from McIntosh's ESTC)
  - Eilmer3 – Axisymmetric nozzle flow
    Space-marched with full nonequilibrium thermochemistry.



NENZFr – Sample input to Eilmer3

```
from estcj import reflected_shock_tube_calculation as rstc
gasName = 'air'
# Katsu's T4 shot 9378
T1 = 300.0   # test-gas fill temperature, K
p1 = 160.0e3 # test-gas fill pressure, Pa
Vs = 2707.0  # incident shock speed, m/s
pe = 38.0e6  # equilibrium pressure in nozzle supply region, Pa
estc_result = rstc(gasName, p1, T1, Vs, pe, area_ratio=10.0, task='stn')
throat = estc_result['state6']
throat_V = estc_result['V6']
print "Flow condition at nozzle throat:"
throat.write_state(sys.stdout)

# Estimate turbulence quantities for free stream
# by specifying the intensity as 0.05 and estimating the
# turbulence viscosity as 100 times the laminar viscosity.
throat_tke = 1.5 * (throat_V * 0.05)**2
throat_mu_t = 100.0 * throat.mu
throat_omega = throat.rho * throat_tke / throat_mu_t
print "Inflow turbulence: tke=", throat_tke, "omega=", throat_omega

select_gas_model(fname="cea-lut-air.lua.gz" )
inflow = FlowCondition(p=throat.p, u=throat_V, v=0.0, T=throat.T,
                       massf=[1.,], tke=throat_tke, omega=throat_omega)
initial = FlowCondition(p=130.0, u=0.0, v=0.0, T=300.0,
                        massf=[1.,], tke=0.0, omega=1.0)
```

NENZFr – Sample output

File: /home3/peterj/work/eilmer3/2D/t4-nozzle/LOGFILE_STATS

```
Nozzle-exit statistics:
  variable    mean-value          minus        plus
-------------------------------------------------
       rho        0.2365       -0.00417     0.00983
     vel.x          3140          -7.39        2.92
     vel.y        -20.61          -7.75        19.8
         p     1.017e+05      -2.34e+03    5.52e+03
         a         748.6          -1.92        4.31
        mu     5.717e-05      -2.09e-07    4.71e-07
      k[0]       0.09289      -0.000455     0.00103
      mu_t             0              0           0
       k_t             0              0           0
       tke             0              0           0
     omega             1              0           0
  massf[0]             1              0           0
      e[0]     1.208e+06       -7.7e+03    1.74e+04
      T[0]          1497          -8.18        18.4
   M_local         4.194        -0.0339      0.0146
   pitot_p     2.231e+06      -3.58e+04    8.31e+04
   total_p     2.699e+07      -8.11e+04    1.67e+05
   total_h     6.567e+06      -1.07e+03    1.27e+03
-------------------------------------------------
Done.
```