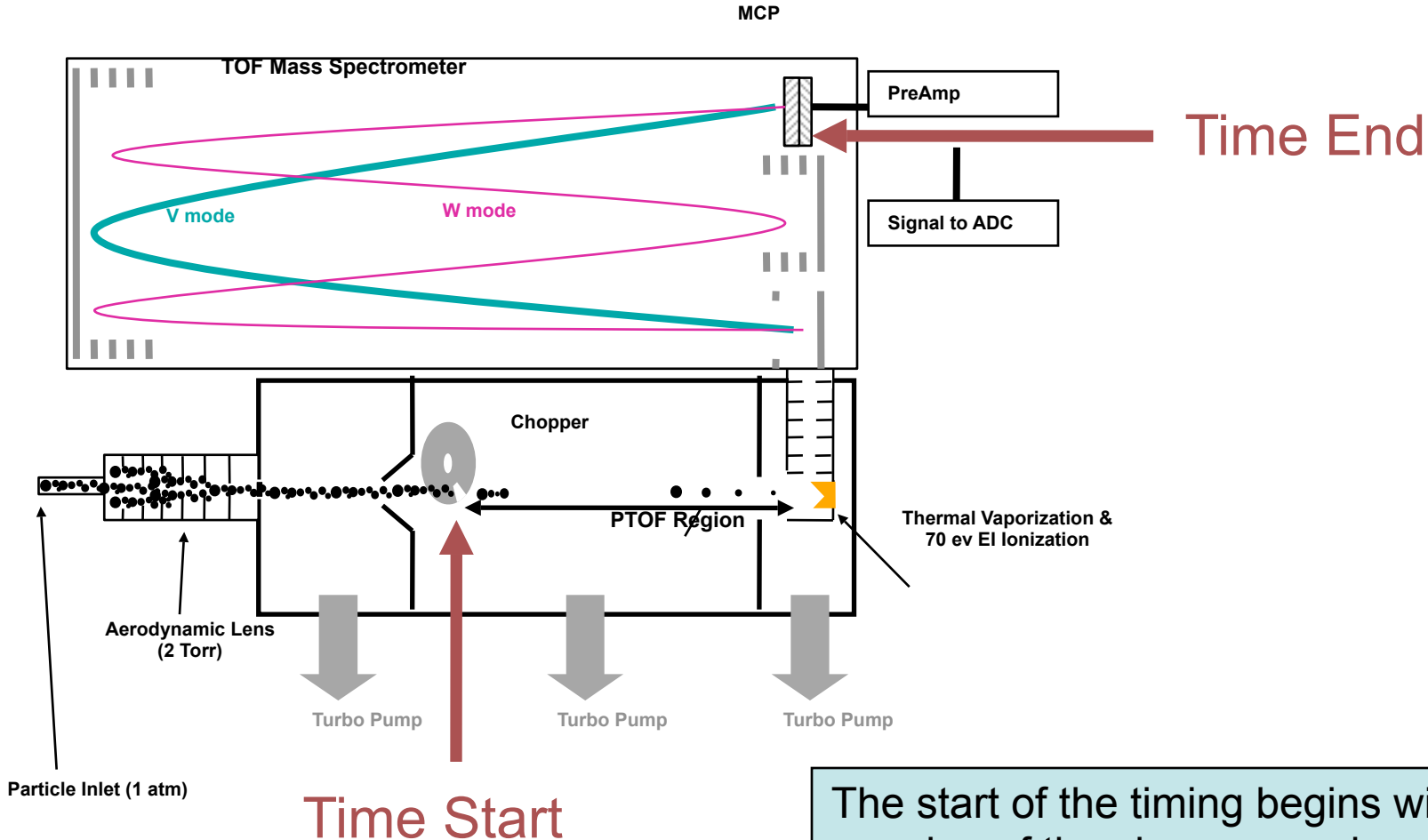


Size calibrations with the ToF-AMS

Pete DeCarlo

11-June-2008

Basics of the timing



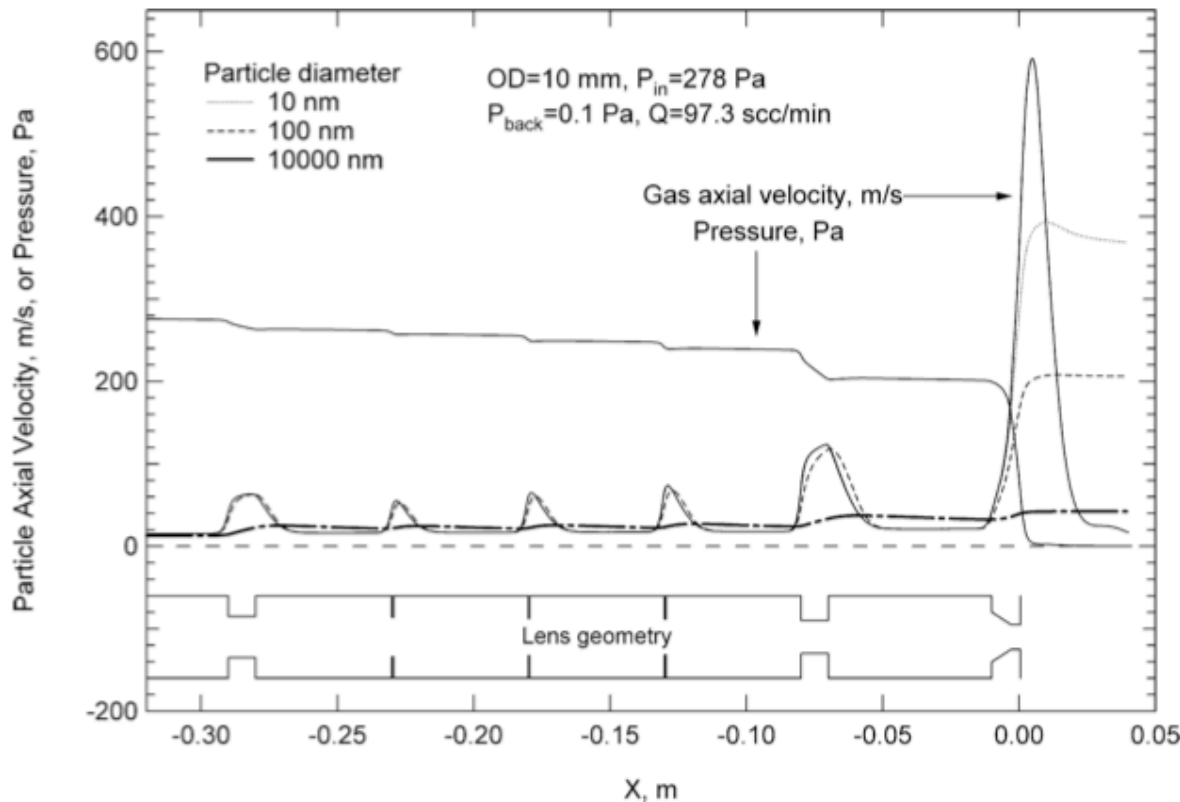
The start of the timing begins with the opening of the chopper, and ends with the detection of ions. The Particle Time of Flight timescale is milliseconds, while the ion time of flight is microseconds, which is why we can use the ion detection as our "time end"

Empirical Sizing Equation and “realistic” values

$$V_{lens} + \frac{V_{gas} - V_{lens}}{1 + \left(\frac{d_{va}}{D^*}\right)^b} = V_{PSL} = \frac{FlightLength}{FlightTime}$$

- V_{lens} = velocity of particle in lens (see next slide)
- V_{gas} = velocity of gas (see next slide)
- d_{va} = vacuum aerodynamic diameter of PSL
- D^* = empirical parameter
- b = empirical parameter (typically ~0.5)
- Flight Length = 0.293 m (short chamber)
- Flight Time = Flight time of PSL

Equation is not well constrained, but...



- Fluent modeling of the lens, gives indication of “reasonable” values.
- $V_{gas} \sim 300$ or so
- $V_{lens} \sim 10$ or so
- These can vary quite a bit...think order of magnitude

Figure 2. The same inlet as in Figure 1, plot of axial velocity of particles ($D_p = 10, 100,$ and 10000 nm) and of gas; gas pressure is also plotted.

Zhang, X., et al., A Numerical Characterization of Particle Beam Collimation by an Aerodynamic Lens-Nozzle System. Part I: An Individual Lens or Nozzle, *Aerosol Science and Technology*, 36: 617–631 (2002).

Zhang, X, et al. Numerical Characterization of Particle Beam Collimation: Part II Integrated Aerodynamic Lens-Nozzle System, *Aerosol Science and Technology*, 38(6): 619-638, 2004.

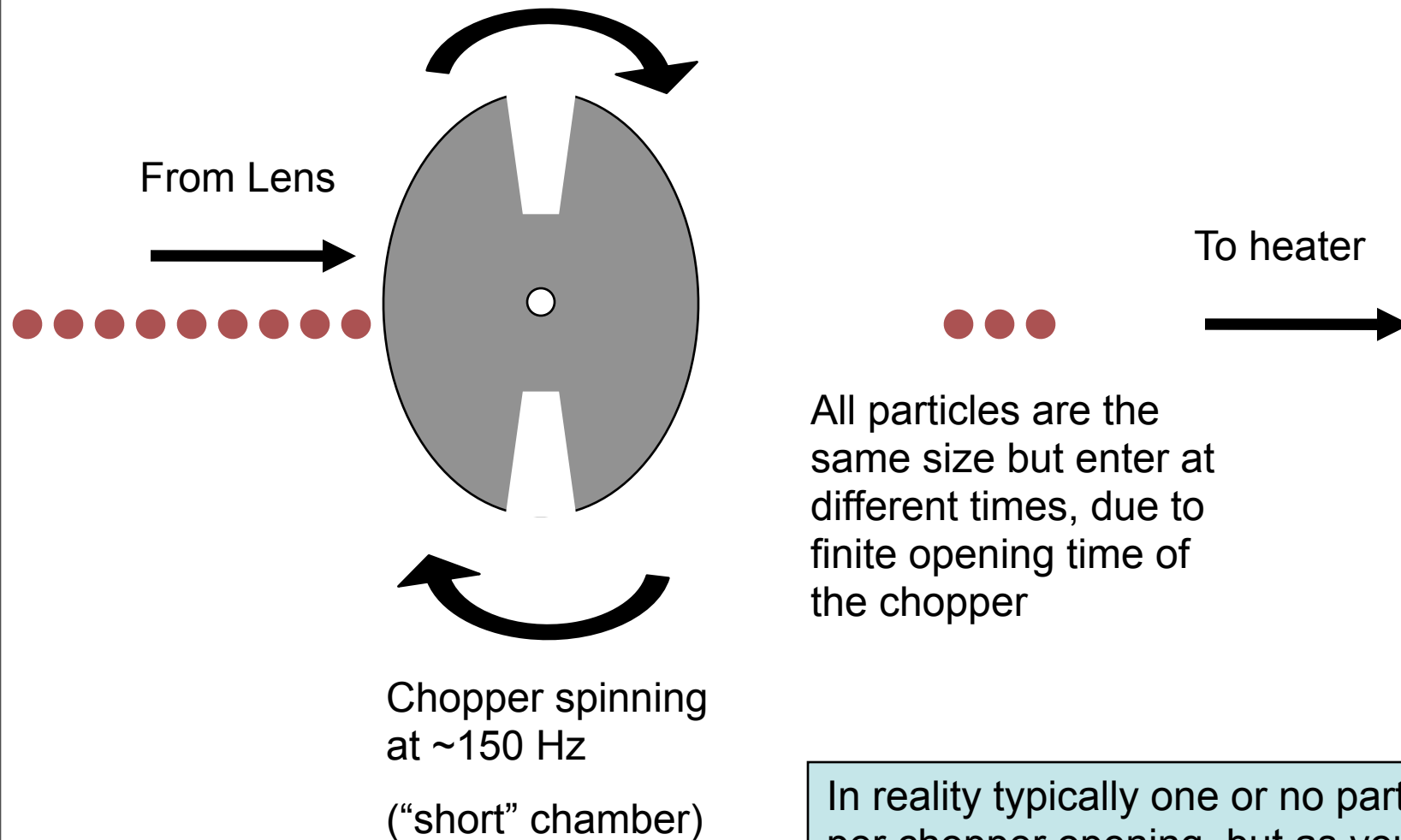
Taking the Data

- For the most accurate size calibration the time resolution of the Particle Time of Flight data is key.
- No co-adding, and pulse as fast as possible.
- One can push V-mode to ~ 25 μs pulser period with a m/z range up to ~ 200 .
- C-ToFs can pulse at $\sim 12\mu\text{s}$ resolution and this is fine.
- Coadding reduces time resolution by a factor equal to the number of co-adds, avoid this

How to calculate the Particle-time-of-Flight (PTOF) - Velocity

- Some effects to keep in mind when calculating the PTOF.
- Chopper broadening
 - Particles can enter the sizing region at anytime with the chopper slit open
 - Larger effect with larger chopper Duty cycle
- Evaporative Broadening
 - PSLs are plastic
 - Heater at 800 C
 - “slow” evaporation

Chopper Broadening



All particles are the same size but enter at different times, due to finite opening time of the chopper

In reality typically one or no particles enter per chopper opening, but as you average many chopper cycles these particles average together to form a "square wave" pulse of particles.

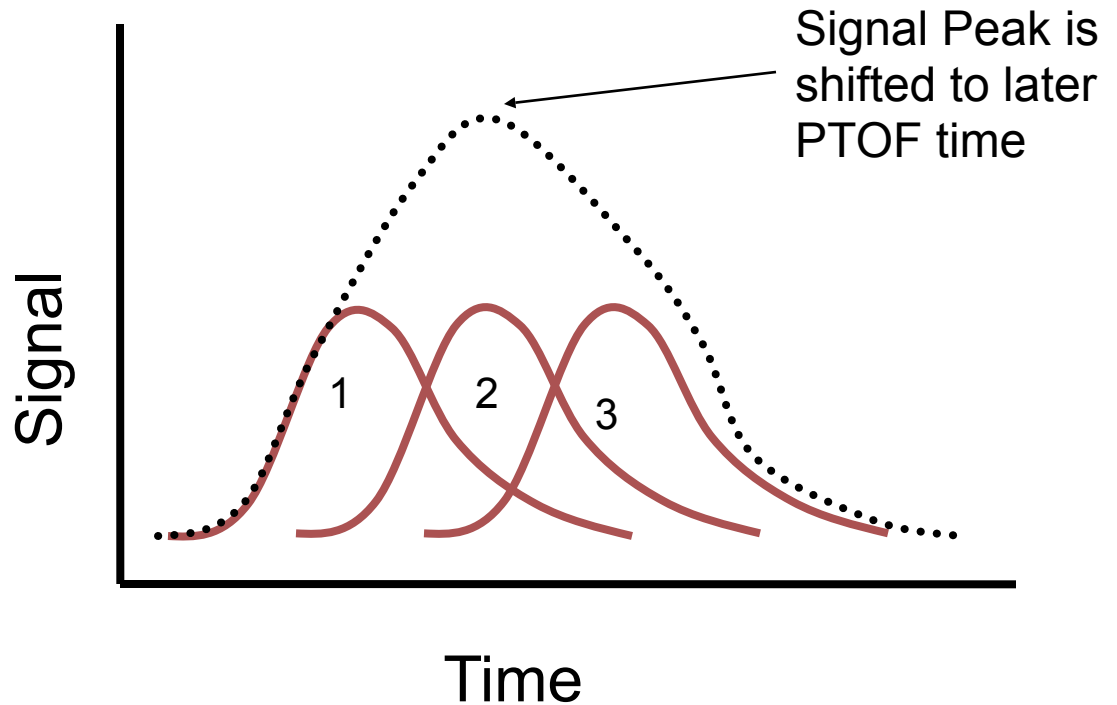
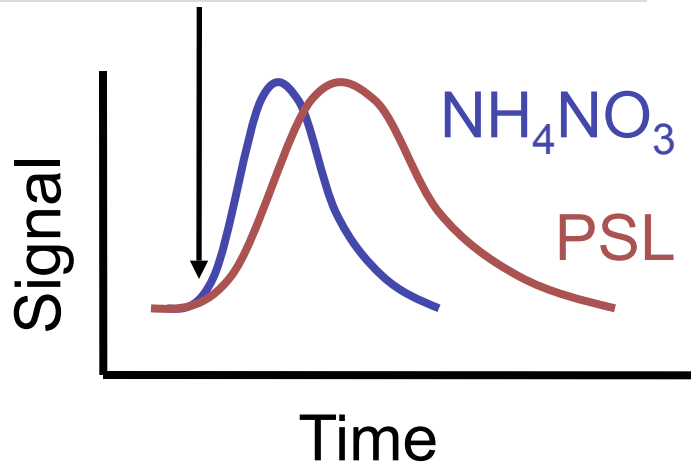
Evaporative Broadening

1 2 3



Heater at ~ 800 C
(We're melting plastic)

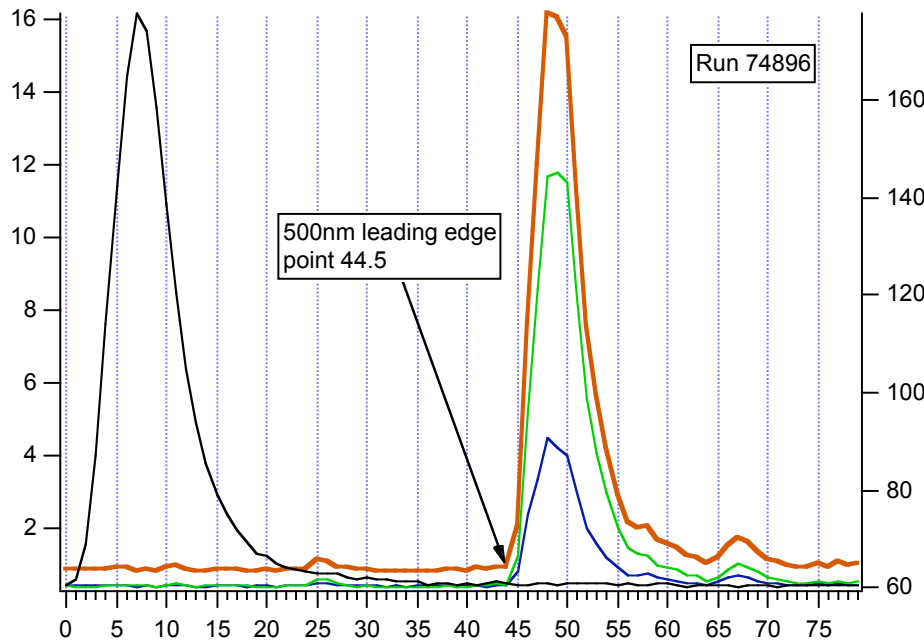
Initial signal rise occurs at same
time, even though evap takes longer



Evaporation of PSLs takes a long
time, even at 800 C. This “smears”
the signal to longer times, and
makes using the “peak” of the signal
undesirable

Determining the Particle arrival time

- The point which we know best is the rising edge of the signal.
- We can then calculate the middle of the chopper opening by adding half of the chopper open time.



$$T_{PSL} = T_{edge} + \frac{1}{2} \frac{ChopperDC}{ChopperHz}$$

Fitting the Data

- A X-array/wave of dva values for the PSLs
 - Assume spheres, density = 1.04
- A Y-array/wave of velocity values.
 - Divide flight length by Particle time of flight

```
////////////////////////////////////
// Fit Function for PSL Size Calibration - P. DeCarlo //
////////////////////////////////////
Function PSL(w,dva) : FitFunc
    Wave w
    Variable dva

    //Original parameters:
    // Vgas = 795.62
    // Vlens = 57.23
    // D* = 3.758
    // b = 0.6398

    //CurveFitDialog/ These comments were created by the Curve Fitting dialog. Altering them will
    //CurveFitDialog/ make the function less convenient to work with in the Curve Fitting dialog.
    //CurveFitDialog/ Equation:
    //CurveFitDialog/ f(dva) = 0.293 / (Vlens+ (( Vgas - Vlens)/(1+(dva/Dstar)^B)))
    //CurveFitDialog/ End of Equation
    //CurveFitDialog/ Independent Variables 1
    //CurveFitDialog/ dva
    //CurveFitDialog/ Coefficients 4
    //CurveFitDialog/ w[0] = Vlens
    //CurveFitDialog/ w[1] = Vgas
    //CurveFitDialog/ w[2] = Dstar
    //CurveFitDialog/ w[3] = B

    return (w[0]+ (( w[1] - w[0])/(1+(dva/w[2])^w[3])))
```

End

Final Result

