Notes: A Lagrangian Spray Model for Airfoil Icing

Anthony DeGennaro

July 2014

1 Introduction and Motivation

The purpose of these notes is to describe the ongoing development of a wing ice accretion code which applies techniques from the liquid spray community to improve on existing methods.

It is well-known that individual droplets in the air can significantly affect ice accretion on wings. As an example, an ATR-72 crashed in Roselawn, Indiana in 1994; it is speculated that this incident was largely due to Supercooled-Liquid Droplets (SLDs) in the air. This crash has led to a revamping of the FAA regulations on flight conditions to explicitly consider freezing drizzle/rain.

Existing computational codes for icing generally do *not* track each individual droplet impinging on the wing throughout the course of a simulation. The reason for this is simple: there are too many droplets for this to be computationally feasible. We can demonstrate this easily with a back of the envelope calculation: assuming the liquid water content (LWC) of the air to be 0.3 g/m^3, the airspeed to be 100 m/s, and the mean volumetric diameter (MVD) of the liquid droplets to be 50 μm, this gives a free-stream flux of about 500 million particles per square meter per second. Since wing icing simulations can frequently last 5-10 minutes or longer, one is faced with the proposition of tracking (potentially) billions of particles over the course of one simulation (this is not even accounting for new droplets created through the process of droplet splashing and breakup).

Existing methods deal with this problem in a few ways. First, many methods do not employ a Lagrangian approach; rather, they treat the liquid/air as a two-phase continuum and use Eulerian formulations, which automatically circumvents the need to track all particles. Regardless, most methods assume that the liquid mass impinging on the wing surface can be modeled as a continuum (the so-called "collection efficiency"). Because of this assumption, there is no need to advect the billions of particles one would do in a fully time-resolved Lagrangian simulation; instead, one need only advect an initial "sheet" of particles, observe how that sheet impinges on the wing surface, and calculate a mass flux ratio that way. Of course, another notable assumption here is that the collection efficiency does not vary over the course of the simulation, even though the droplet properties may be randomly distributed.

Both of these solutions lead to fast simulations at the expense of ignoring (1) the statistical effect of having droplet properties which are randomly distributed and (2) the effect on the ice shape of individual droplets. This is where we hope to make a contribution. Borrowing from methods used in the spray community, we formulate the icing problem probabilistically, by equipping the freestream droplet properties with a probability density function. We then formulate a Lagrangian in which the advected particles are actually "clumps" of droplets. This keeps the number of parcels tracked in each simulation relatively low. In order to investigate the statistical effects of the droplet distribution, one need only generate a multirealization ensemble of simulations. We also hope to develop a thermodynamic model that governs the behavior of the individual droplets as they impinge and freeze on the surface; this would potentially help towards investigating the effect of individual droplets on ice shape.

2 Brief Sketch of Method

Writing the interpolation matrix as $\mathbf{\Phi_X}$, and the vector of response surface values as $\mathbf{Y}$, we have the regression problem $\mathbf{\Phi_X^T c} = \mathbf{Y}$. Note that $\mathbf{\Phi_X} \in \mathbb{R}^{K\times Q}$ and $\mathbf{Y} \in \mathbb{R}^{Q\times 1}$.

A least-squares solution of this system would involve calculating the psuedo-inverse: $\mathbf{c} = \mathbf{Y}(\mathbf{\Phi_X^T})^{\dagger}$.

First, we show how to reduce the dimensionality of the calculation through a QR factorization. Define:

$$\mathbf{\Phi_{XY}} = \begin{bmatrix} \mathbf{\Phi_X} \\ \mathbf{Y^T} \end{bmatrix} \in \mathbb{R}^{(K+1)\times Q} \tag{1}$$

We use the QR factorization to write $\mathbf{\Phi_{XY}^T} = \mathbf{QR} = \mathbf{QL^T}$ with:

$$\mathbf{L} = \begin{bmatrix} \mathbf{L_{11}} & \mathbf{0} \\ \mathbf{L_{21}} & \mathbf{L_{22}} \end{bmatrix} \tag{2}$$

This gives us $(\mathbf{\Phi_X})^\mathbf{T} = \mathbf{Q}[\mathbf{L_{11}} \quad \mathbf{0}]^\mathbf{T}$ and $\mathbf{Y} = \mathbf{Q}[\mathbf{L_{21}} \quad \mathbf{L_{22}}]^\mathbf{T}$. Thus:

$$\begin{aligned} \mathbf{c} = \mathbf{Y}(\mathbf{\Phi_X^T})^\dagger &= \left\{ \mathbf{Q} \begin{bmatrix} \mathbf{L_{21}^T} \\ \mathbf{L_{22}^T} \end{bmatrix} \right\} \left\{ \mathbf{Q} \begin{bmatrix} \mathbf{L_{11}^T} \\ \mathbf{0} \end{bmatrix} \right\}^\dagger \\ &= \left\{ \mathbf{Q} \begin{bmatrix} \mathbf{L_{21}^T} \\ \mathbf{L_{22}^T} \end{bmatrix} \right\} \left\{ \begin{bmatrix} \mathbf{L_{11}^{-T}} & \mathbf{0} \end{bmatrix} \mathbf{Q^T} \right\} \\ &= \mathbf{Q} \begin{bmatrix} \mathbf{L_{21}^T L_{11}^{-T}} & \mathbf{0} \\ \mathbf{L_{22}^T L_{11}^{-T}} & \mathbf{0} \end{bmatrix} \mathbf{Q^T} \end{aligned} \tag{3}$$

This matrix has a transpose which is upper block triangular, and so its eigenvalues are given as the union of those of the transposed diagonal blocks, which are zeros together with the eigenvalues of $(\mathbf{L_{11}^{-1}})\mathbf{L_{21}}$.