

1 Data Volume Class

The data volume class is the 'backbone' of the STMA code, because it provides a template for data storage in the code.

The basic idea is that the data volume class is used to define all variable data in the code. Then, all 'number crunching' routines, such as derivative calculations, dissipation, fluxes, and message passing routines, will expect the data that they are working with to be in `DataVolumeClass` format.

To easily allow this, the `DataVolumeClass` has all internal data declared `PUBLIC`. The idea behind this is to allow data to be easily referred to as an object.

Next question: can I have *some* `PRIVATE` attributes in a data class and some `PUBLIC`?

It looks like I'll have to put an internal data class into the `DataVolumeClass` object to get the functionality I want. I've bookmarked some F2003 articles to read about (and try) putting procedures into derived data types. This way (I think) I can put a procedure `isInitialized` into the `DataVolumeClass` that acts like a `LOGICAL` – but actually calls a hidden routine using hidden data to find out if the DVC is initialized properly.

Let's try this tomorrow.