**Negative Jump Algorithm**

When calculating the ticks associated with a UTC, add a before or after jump flag if the UTC value is within the ambiguous range. A value of 0 indicates before the jump and 1 indicates after the jump. We need to add the flag to the createTicks function (and make sure the createTicks function is always used – there are times when the tick struct is loaded directly). The flag will be unnecessary, except for the 0.1 second and 0.2 second ambiguous intervals.

Will there ever be a situation where we have a tick count that was not created from a UTC, so don’t know the value of the before or after flag. This can occur if we allow the addition or subtraction of a relative tick count to a UTC. For the time being, we want to allow this. In the future, when we have a relative CalCoords type, we can restrict the addition or subtraction to these types. We could still create a relative tick by subtracting two UTCs, but not allow a relative tick to be added to a UTC. For the time being, we could adopt the following approach:

When a relative tick count is added to or subtracted from a UTC and the resulting tick count is in the ambiguous region, do the following:

If the UTC is before the start of the ambiguous region, set the flag to before

If the UTC is after the end of the ambiguous region, set the flag to after

If the UTC is within the ambiguous region, preserve the value of the flag

When subtracting raw tick counts, not as a result of a UTC subtraction, ignore the before or after flag. This will produce the intuitive result.

When subtracting tick counts in support of a UTC subtraction, we need to respect the before or after flags to produce an intuitive result.

If neither is within the ambiguous region, there is no issue.

If both values are within the ambiguous range:

If both values have the same before or after flag, just subtract

If the flags are different, add .05 to the absolute value of the difference