

8th. World Congress on Computational Mechanics (WCCM8)  
5th European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS 2008)  
June 30 – July 5, 2008  
Venice, Italy

## TOLERANCE AND UQ4SIM: NIMBLE UNCERTAINTY DOCUMENTATION AND ANALYSIS SOFTWARE

**Bil Kleb**

NASA Langley Research Center  
Mailstop 408a, Hampton, Virginia 23681 USA  
Webpage: [nasarb.rubyforge.org](http://nasarb.rubyforge.org)  
Email: [Bil.Kleb@nasa.gov](mailto:Bil.Kleb@nasa.gov)

**Key Words:** *Monte Carlo, Sensitivity, Uncertainty, Error Propagation.*

### ABSTRACT

Ultimately, scientific numerical models need quantified output uncertainties so that modeling can evolve to better match reality. Documenting model input uncertainties and variabilities is a necessary first step toward that goal. Without known input parameter uncertainties, model sensitivities are all one can determine, and without code verification, output uncertainties are simply not reliable.

The basic premise of uncertainty markup is to craft a “tolerance” and “tagging” mini-language that offers a natural, unobtrusive presentation and does not depend on parsing each type of input file format. Each file is marked up with tolerances and optionally, associated tags that serve to label the parameters and their uncertainties.

The evolution of such a language, often called a Domain Specific Language or DSL, is given in [1], but in final form it parallels tolerances specified on an engineering drawing, e.g.,

```
1 +/- 0.5  
5 +/- 10%  
2 +/- 1o
```

where % signifies percent and o signifies order of magnitude. Tags, necessary for error propagation, can be added by placing a quotation-mark-delimited tag after the tolerance, e.g.,

```
0.7 +/- 20% 'T_effective'
```

In addition, tolerances might have different underlying distributions, e.g., Uniform, Normal, or Triangular, or the tolerances may merely be intervals due to lack of knowledge (uncertainty).

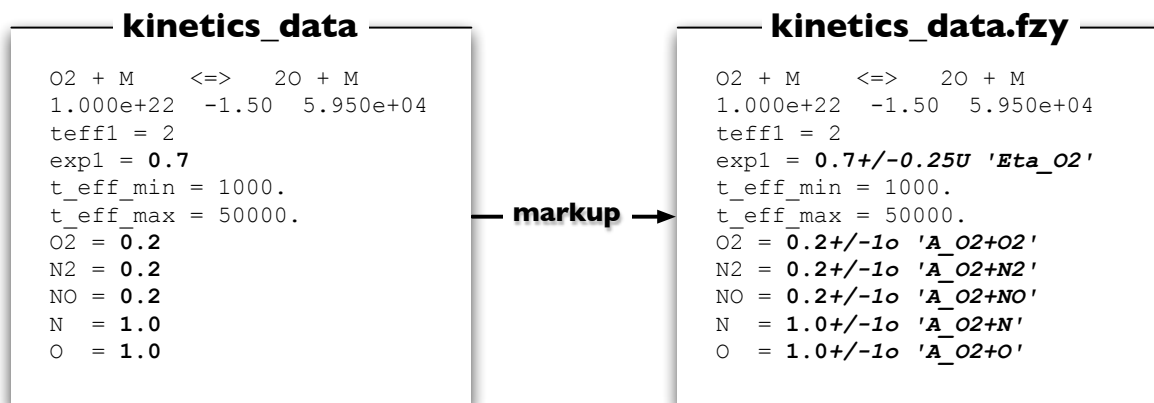


Figure 1: Sample tolerance and tagging markup[1].

Finally, to address pragmatic considerations such as older models that require specific number-field formats, C-style format specifiers can be appended to the tolerance like so,

```
1.35 +/- 10U_3.2f
```

As an example of use, consider figure 1, where a chemical reaction input file is has been marked up to include tolerances and tags per table 1.

Not only does the technique provide a natural method of specifying tolerances, but it also servers as in situ documentation of model uncertainties. This tolerance language comes with a utility to strip the tolerances (and tags), to provide a path to the nominal model parameter file. And, as shown in [1], having the ability to quickly mark and identify model parameter uncertainties facilitates error propagation, which in turn yield output uncertainties.

To get started with model parameter tolerance/uncertainty markup, roll your own tool based on the grammar given here, or download one from [NASArb.rubyforge.org](http://NASArb.rubyforge.org).

Table 1: LAURA model parameter uncertainties[1].

Reaction rates	±1 order
Vibration-dissociation	0.7 ± 0.25*
Thermal-relaxation	±10%
Collision integrals	±30%

\* With bounded distribution to assure samples fall in [0, 1].

## REFERENCES

- [1] B. Kleb, C.O. Johnston, and R.A. Thompson. “Blurring the Inputs: A Natural Language Approach to Sensitivity Analysis”. *AIAA Paper*. **2007-4206**, 2007.