title: Perturbation methods using backward error

abstract:
Numerical analysts are very familiar with backward error, especially for linear algebra but really for all aspects of scientific computing. It doesn't always work (for example, there is no strict backward error analysis generally possible for an outer product because there are n^2 errors and only $2n$ pieces of data to reflect them into) but it works in a very large number of cases, and is quite useful. One needs to know the conditioning of the problem, of course, but then one needs to know that even if one has the exact reference solution to the model problem, because it is (generally speaking) only a model anyway, and certain aspects of the true situation will have been neglected in order to create the model in the first place, and one will need to know the impact of the neglected aspects.

Backward error analysis is also useful for detecting blunders. In the talk I will cite several researchers (including some famous names indeed) who have published erroneous results that would (probably) have been caught had they computed at least a residual. Of course I also make blunders, even if I do compute residuals, but for sure I make fewer blunders when I compute residuals than I when I don't.

We believe therefore that the use of (and the teaching of!) perturbation methods can, like numerical methods, be improved by paying attention to backward error (residuals, modified equations, and optimal backward error). This approach costs more: the residual could instead be used (typically almost trivially) to get "one more term" in the expansion, and that final step is typically the most expensive in terms of human labour. But with computer algebra systems to help, this is less of a barrier than it used to be.

I will also talk (with the voice of forty years experience) about using computer algebra for perturbation methods. It's not as easy as it should be, although in part that's a good thing, because there is a lot of room remaining for artistry. The tech setup at the conference willing, I will show Maple in a Jupyter notebook environment being used to solve several classic problems, and demonstrate quite high-order analyses using the relatively modern Renormalization Group method, which has displaced the method of multiple scales as my favourite.

I intend the talk to spark discussion, in order to improve the new book that Nic Fillion and I are writing at this moment, currently entitled just the same as this talk.