110914

Jordi:

I am nearsighted so small print and large screens are a must for me. I used the "nedit" editor and "Preferences"->"Wrap"->”auto newline” so the code is easy for me to read. If you can’t turn off line wrapping in your editor then you could just deleted all long lines. These lines are only for debugging and the remaining lines will be less than 128 characters.

When I sent the original files in a few days ago, I was assuming that the best solution was to copy "residue" to "residuez" and then make the required modifications in "residuez" to fix the four issues. I have changed my mind -- all changes can and should be made in "residue". No changes are necessary in "residuez".

In the original code, the major problem that I addressed was how to add a “switch” statement so you could obtain the correct results for discrete-time problems using "residuez". I added a new input parameter to "residue" to allow this statement. Sunday, I realized that this new parameter is really unnecessary. Basically, if you examine the poles and they are all very large then it is a continuous-time problem using "residue" and if they are very small then it is a discrete-time problem using "residuez". Remember "residuez" effectively inverts the poles so that large poles become small poles.

Here are some answers to your questions in “Follow-up Comment #8, bug #34266 (project octave)”:

Xxxxx I have confirmed with Sage that indeed the polynomials returned by residue are
Xxxxx not monic as they should be. While we investigate this problem, I have pushed
Xxxxx the following change on the stable branch:
Xxxxx http://hg.savannah.gnu.org/hgweb/octave/rev/d049192e5d15

I am not sure who Sage is or what the stable branch is – but it sounds good. Of the three examples that I sent to you, the third case is the worst case – it returns [0](which is certainly not monic) rather than a four element vector.

Xxxxx While indeed arbitrarily dividing small by 1e3 solves the non-monic polynomial
Xxxxx problem for this test, I am sure that such absolute measures are not correct,
Xxxxx and we should pick small in a smarter way that takes relative size into
Xxxxx account. What do you suggest?

At the present time, I believe that trimming of the polynomial coefficients should be eliminated from the Octave code. The user should decide if he wants to reduce the polynomials in his code.

Xxxxx Further, even with that suggested change, your test fails because the
Xxxxx tolerance is too low for br and b. Is this also a bug with residue or should
Xxxxx your test allow for higher tolerance?

The "residue" code employs linear algebra (with emphasis on the word **linear)** so that most calculations are only limited by round off errors. For a 64-bit system, the machine accuracy (eps) is about 2.22x10-16. For most poles, the tolerance should be better than 1e-12. But, there are outliers so that the assert test should be done pole-by-pole with variable tolerances. This is rather complicated so I used just 1e-8 as a compromise.

Bernie