Current state of errors in geometric coordinate calculation

Aaron Bader

June 25, 2018

1 Introduction

The goal is to calculate coefficients for various geometric coordinates internally in STELLOPT. These geometric coefficients will be used as input for the PTSM3D code, which will calculate a maximum coupling coefficient between unstable and stable modes. The goal is to use this metric as an optimizing parameter.

Ideally the geometric components computed by STELLOPT would agree exactly with external calculations using the GIST code. The first attempt, an adaptation of the stellopt_txport code, written by Sam Lazerson, had significant errors in the L_2 coefficient. The alternative approach currently used, was written by Matt Landreman. The code is renamed to "vmec2gs2", since it converts to coefficients using normalizations favored by GS2. An extra step to convert to the GIST coefficients is done after. The conversion was calculated by Ben Faber and is included at the end of this document.

All in all there are three pathways to calculate the same final quantity. These are:

- Use STELLOPT to create a wout file in .txt format. Use GIST to calculate the geometric quantities. Use standalone PTSM3D to calculate the coupling coefficient.
- Use STELLOPT to create a wout file in .nc format. Use internal "vmec2gs2" and conversion equations to calculate geometric quantities. Use standalone PTSM3D to calculate the coupling coefficient.
- Use STELLOPT to create a wout file in .nc format. Use internal "vmec2gs2" and conversion equations to calculate geometric quantities. Use PTSM3D_opt in STELLOPT to calculate the coupling coefficients.

The second and third methods agree exactly. That is, there is no difference between the standalone PTSM3D code and the PTSM3D_opt code in STELLOPT. However, there is a difference in the geometric components. These differences produce about a 4% difference in the "usm" calculation in the PTSM3D code.

This difference is the same for normalized flux values of s=0.3, 0.5 and 0.7, however it is not true that one method always calculates a larger value than the other. Furthermore, the error does not improve with an increase in the number of flux surfaces in the VMEC wout file. No improvement is seen between 101 and 201 surfaces, implying that the error does not arise from a simple difference in interpolation schemes.

Components between the two calculations, i.e. the data that gets fed into PTSM3D show some differences. The rest of this writeup will be an attempt to catalog these differences.

There are some questions to ask on this topic. Is the 4% error good enough, or do we need to track down where it's coming from? If 4% is not good enough, we may need to think more about the PTSM3D algorithm, because it is quite sensitive. Is there a way to formulate a new metric based on the coupling coefficients that is less sensitive to precision calculation of the geometric quantities?

2 Cataloguing Errors

For the calculations below, we calculate over 100 poloidal turns, i.e. from $\theta = -100\pi$ to 100π with 12800 total points, for 128 points per poloidal turn. This distance and resolution was found previously to be sufficient to give a stable answer in PTSM3D for a given geometry.

We call a generic geometric quantity, Q, if calculated by vmec2gs2, Q_v and if calculated by GIST, Q_g.

Calculating the relative errors in the terms is somewhat difficult. Many of the terms oscillate around zero. Some contain secular terms and have an envelope that increases monotonically with the poloidal angle, θ. A simple metric of Q_v/Q_g will spike when $Q_g \approx 0$, provided that Q_v is nonzero at that point. A metric like $|Q_v - Q_g|/\max(Q_v)$ also has a problem in that it can miss important errors near $\theta = 0$ where the secular contribution is small. The method chosen here is to plot $|Q_v - Q_g|/\overline{Q_v}$, where $\overline{Q_v}$ is an average of $|Q_v|$ over a range of θ that is much less than a single oscillation (in this case between 5 and 20 points in either direction). All references to "error" in the text will refer to this quantity.

The parameters $g11, g12, g22$ and $Bhat$ all show very good agreement between the two codes. The errors for the parameters for $s = 0.5$ are plotted below.

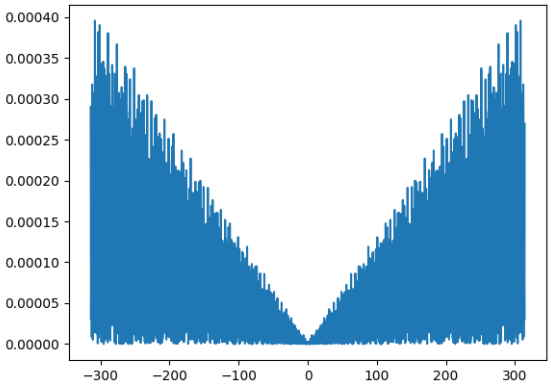

Figure 1: Error for the $g11$ term

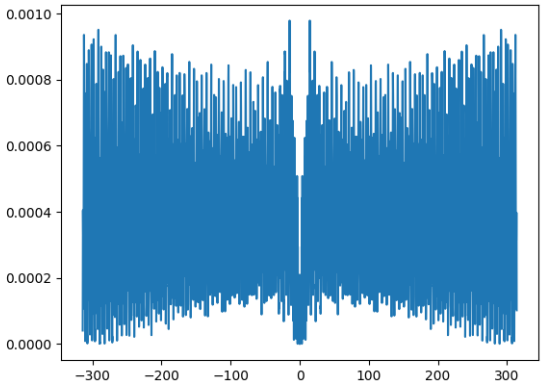

Figure 2: Error for the $g12$ term

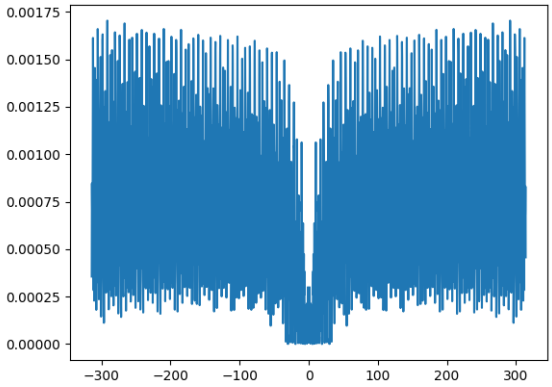

Figure 3: Error for the $g22$ term

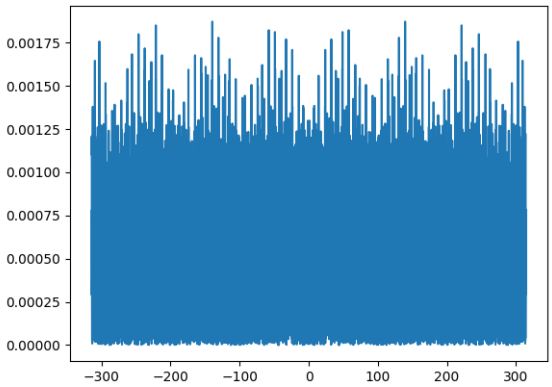

Figure 4: Error for the $Bhat$ term

The errors are larger for the jac, $L1$ and $L2$ terms. First the errors for jac are given for three values of s, 0.3, 0.5 and 0.7.

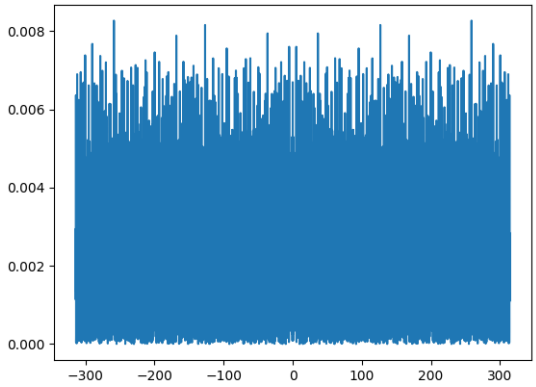

Figure 5: Error for the jac term for $s = 0.3$

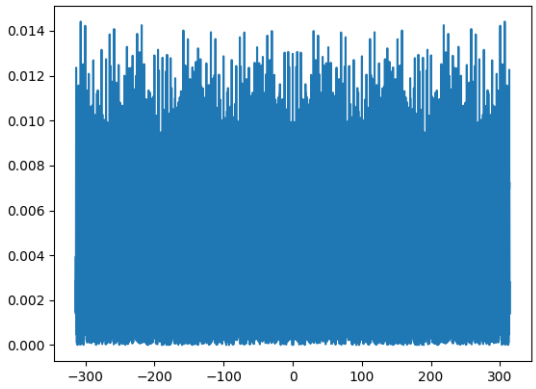

Figure 6: Error for the jac term for $s = 0.5$

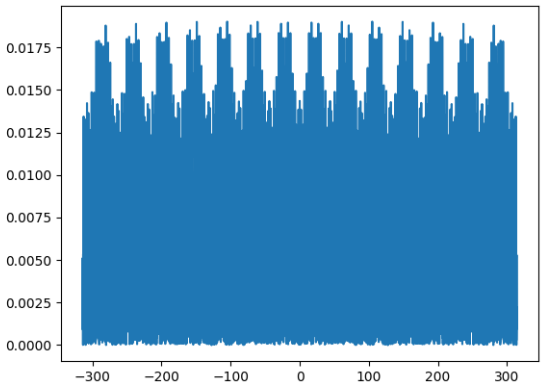

Figure 7: Error for the jac term for $s = 0.7$

We can also plot a zoom in of jac calculated by both STELLOPT and GIST, to see where the errors are. The following plot is for $s = 0.5$.

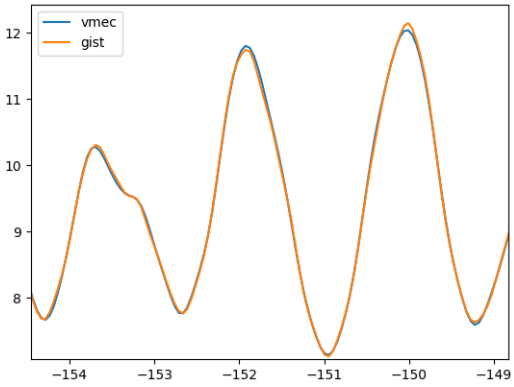

Figure 8: jac_v and jac_g

We do the same now for $L1$. First we provide the three errors for $s = 0.3, 0.5$ and 0.7. Then we provide an example of the difference for $s = 0.5$

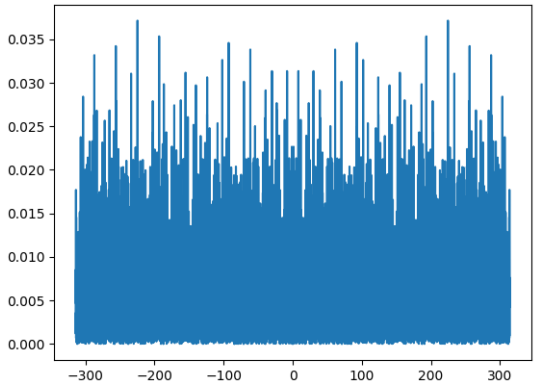

Figure 9: Error for the $L1$ term for $s = 0.3$

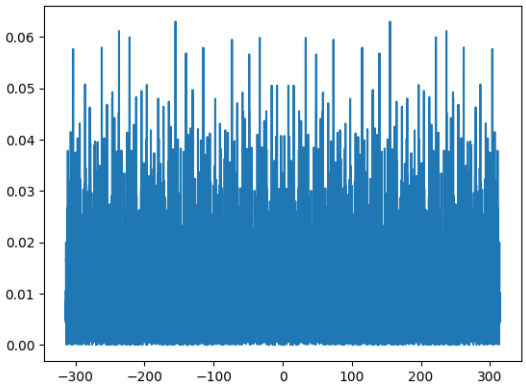

Figure 10: Error for the $L1$ term for $s = 0.5$

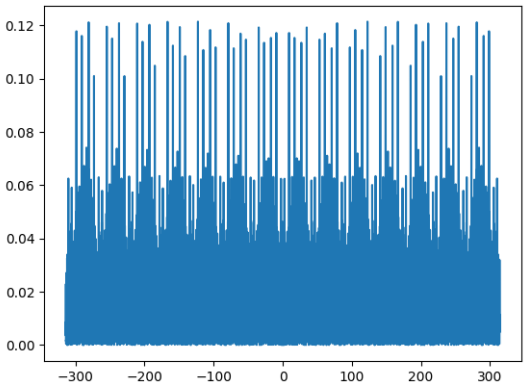

Figure 11: Error for the $L1$ term for $s = 0.7$

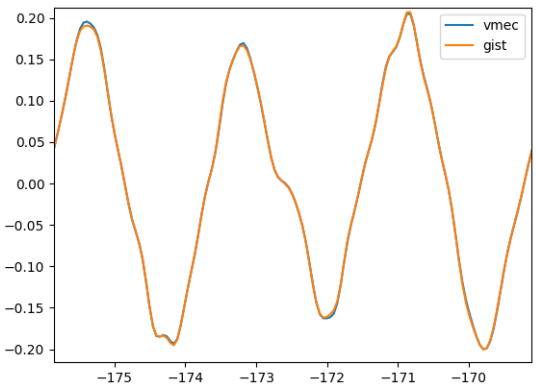

Figure 12: $L1_v$ and $L1_g$

Finally the $L2$ term.

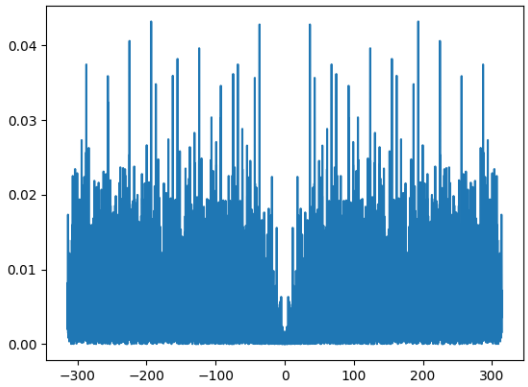

Figure 13: Error for the $L2$ term for $s = 0.3$

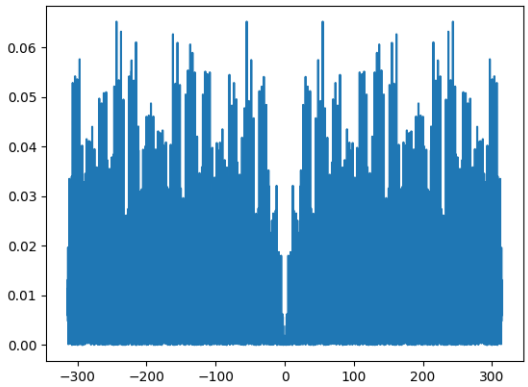

Figure 14: Error for the $L2$ term for $s = 0.5$

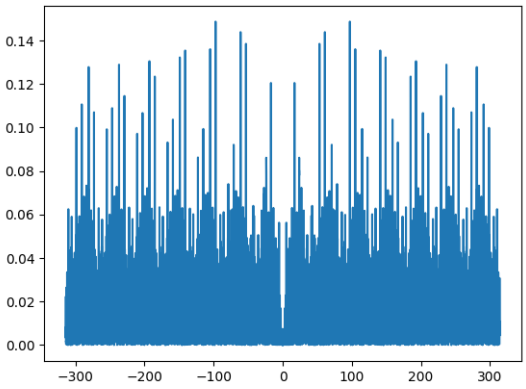

Figure 15: Error for the $L2$ term for $s = 0.7$

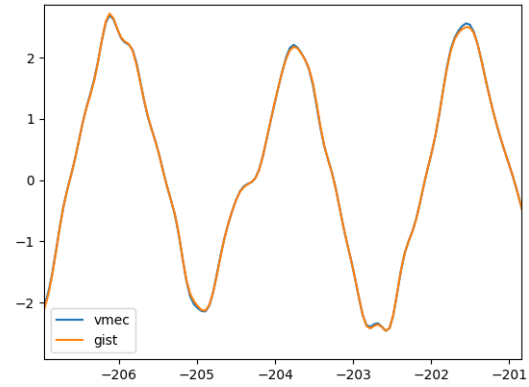

Figure 16: $L2_v$ and $L2_g$

As a final analysis. We zoom in to the $L2$ parameter near one of the locations where a $\approx 7\%$ error is seen.

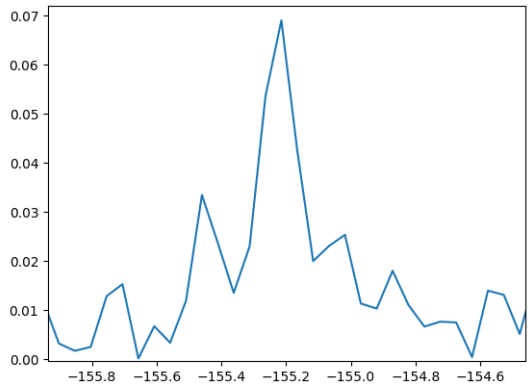

Figure 17: Error for the $L2$ term for $s = 0.5$, zoomed into region of a large error

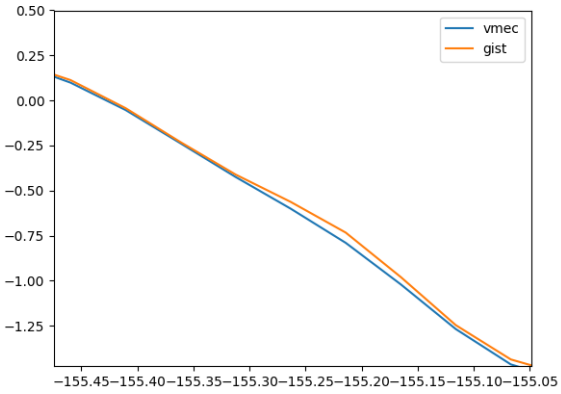

Figure 18: $L2_v$ and $L2_g$, zoomed towards a region of large error

It's clear that there is actually a finite divergence of the two parameters in this region. The error is real, and not an artifact of a location where $L2 \approx 0$.

3 Calculation of PTSM3D quantities

The calculation for the PTSM3D coupling coefficient for non-zonal coupling are shown below for three fields lines at $s = 0.3$, 0.5 and 0.7

s	VMEC	GIST	**ratio**
0.3	0.7702	0.7984	**0.965**
0.5	0.8129	0.8472	**0.959**
0.7	0.8087	0.7868	**1.0278**

4 Converting between GS2 and GIST

The following are the conversions implemented in STELLOPT to convert from the output of vmec2gs2 to the GIST coordinate system. The calculations were done by Ben Faber, and implemented by me.

$$
\begin{aligned}
g11 &= gds22/\hat{s}^2 \\
g12 &= gds21/\hat{s} \\
g22 &= gds2 \\
Bhat &= Bmag \\
jac &= 2q/(Lref^3(1 + d\Lambda/d\theta)\sqrt{g}) \\
L2 &= Bhat * cvdrift/2 \\
L1 &= -Bhat * cvdrift0/(2\hat{s})
\end{aligned}
$$

5 Next steps

There are a few steps that should be done. The first is to re-implement the number of poloidal turns used. The required range of θ is inversely proportional to q'. This is mostly a trivial task, but there may need to be some careful consideration for calculations at which $q' \approx 0$, because in this case the number of turns required is very large and this can lead to a memory overflow crash in STELLOPT.

Another step, optional now, is to load the STELLOPT parameters directly from memory rather than reading them from the "wout" file. Hopefully there should be absolute no difference between these two methods, but that needs to be checked as well.

It is also possible to attempt some basic optimization with the calculation in its current state (provided the implementation of the proper number of turns). This is probably a priority, so that we can get something to present on turbulence optimization. However, I would like some confirmation that the errors we see currently are acceptable.

Current plan is to implement the theta range and begin some optimization calculations this week.