Complex Manifolds training loop, used for Bayesian networks.

This document is intended to explain exactly how the contour integration loop I am working on, which is designed to work with Bayesian networks to reduce the training time and improve the accuracy involve in these, actually works and why I am using the particular methods that I am doing to implement this.

What does it do?

It calculates partial derivatives in $\mathbb{C}^n$*, where n can be as many dimensions as required.*

How does it work?

This algorithm works by using a Logarithmic derivative lemma, which unlike alternatives such as the Wirtinger derivative does not require a closed form for a derivative.

For example a closed form derivative may look like this:

$$\frac{\partial}{\partial x}\left(x^2 y\right) = 2\,xy, \textit{ where as the one produced } \textbf{from} \textit{ this algorithm may be more numerical.}$$

A Wirtinger derivative is more like the above, where as the algorithm I am creating relies on using contour integration around a branch cut to do it. for example:

$$\oint_C \frac{1}{2}\cdot Log\left(\sum_{j=0}^{n}\left|\psi_j\right|^2\right) - \frac{1}{2} Log\left(\sum_{i=k}^{n}\left|\psi_j(0)\right|^2\right) dt, \textit{ where t is the parameter used } \textbf{to do} \textit{ this integral } \textbf{and} \textit{ C is a smooth path around the branch cut, taken along the negative real axis.}$$

The argument in this case is defined as the principal argument, i.e. -

$\pi \le \pi$*, although this can be done with other argument values* **to** *get different results, depending on what result is desired* $\cdot$ *li*

References:

M.Schneider, Yum-Tong Siu, Several Complex Variables (37).
https://en.wikipedia.org/wiki/Wirtinger_derivatives [online] (accessed 03/06/2022)
.