DefinitelyNotReduze 2.0

Matthew Schiavi[1,*]

[1] *Department of Physics, University at Buffalo*
The State University of New York, Buffalo, NY 14260-1500, USA

DefinitelyNotReduze is a program which reduces loop integrals using the Integration by parts method (IBP).

Contents

*Email: mmschiav@buffalo.edu

I. INTRODUCTION

II. THE ALGORITHM

A. IBP Generation

An important part of using the Laporta algorithm is generating IBPs and it is important that this is fully automated. Luckily it is a fairly simple process given the propagators, external momenta, and internal momenta are given.

1. First we can organize the coefficients of the expanded propagators and place them in a matrix. The rows will correspond to propagators and the columns correspond to the specific term. For example, if the propagator expands out to $k^2 + p_1^2 - 2kp_1$ the corresponding row for this propagator would be (1, -2, 0) if we define the columns to contain terms k^2, kp_1, kp_2 respectively. At the moment it is important that we only care about terms which contain factors of the internal momenta in them.

2. At this point we take the matrix which was found in 1 and dot it with a vector of the squares. The order of this vector is important and is in the same order as the columns for the previous matrix are organized.

3. Now subtracting the vector we had just found with the propagators (again these are in a vector and the order is kept with the order which was in the original matrix) and what is left should be anything that does not include any factors of internal momenta.

4. Take the matrix from step 1 and append the vector to the end to add an additional column. Also append a row at the bottom full of 0's except for the last column where there should be a 1. Now the inverse of that matrix is the last thing that needs to be done. It is important to note at this point this matrix can be used to generate any IBP for the Feynman diagram that was specified in step 1.

5. To begin generating the IBP two variables need to be specified. For instance if you want the IBP which relates to the internal momenta one would input "k,k" (the first term will be referred to as x and the second y). Looping through it takes the derivative of the first propagator with respect to x and then that line will be multiplied by y. This term is stored in the variable called input.

6. In a temporary variable (called temp) the squares of terms which contain units of momentum squared are stored respecting the order which was specified in step 1.

7. Earlier on in step 1 a variable which stored each term with units of momentum squared in a vector was defined and called squares. At this point we append to the temp variable which was defined in step 6 to input - temp(dot)squares.

8. Now with the temp from step 7 dot this vector with the matrix form step 1. At this point is a matter of piecing everything together. All terms before the last term in this vector will be annihilation operators which effect the propagator with respect to the index. For instance if we are left with $(1, 0, 1, -p_1^2)$ the first term would be a annihilation operator for the first propagator and the third for the third propagator. Now we sum over this vector and multiply the entire thing by a factor of the exponent of the propagator which we took the derivative of in step 5. Also this entire thing also gets multiplied by a creation operator for the same index. (So if this was for the first propagator it would get a creation operator for index 1 and a factor of the exponent a(1)).

9. This is done for every propagator which each output is summed together. Lastly taking the derivative of the two inputs (y with respect to x) and multiplying with by d (for the number of dimensions) is the last bit which is added on.