



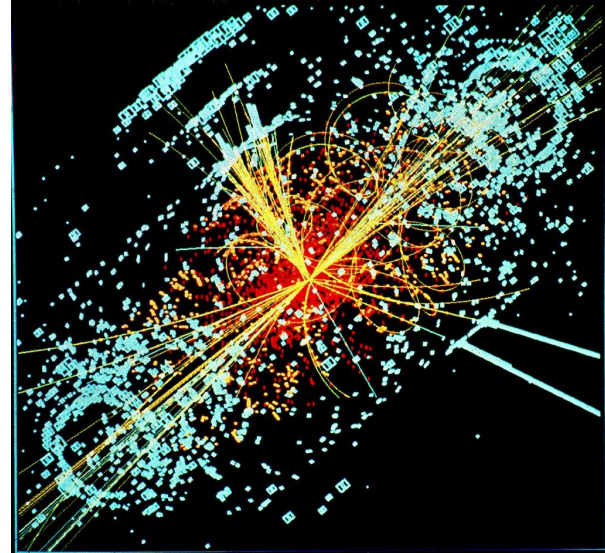
Оценка эффективности моделирования физических сигналов с помощью модифицированного метода морфинга

подготовил: Воробьев Г.
Научный руководитель: Беляев Н.
НИЯУ МИФИ, М21-115, 2021

Бозон Хиггса

Бозон Хиггса “завершает” СМ

Механизм Хиггса объясняет наблюдаемые массы калибровочных бозонов



Стандартная модель

[illegible]

теория всего

?

Новая физика?

EFT

(англ. Effective Field Theory)

$$\mathcal{L}_{EFT} = \mathcal{L}_{SM} + \sum_i \frac{c_i^{(5)}}{\Lambda} \mathcal{O}_i^{(5)} + \sum_i \frac{c_i^{(6)}}{\Lambda^2} \mathcal{O}_i^{(6)} + \sum_i \frac{c_i^{(7)}}{\Lambda^3} \mathcal{O}_i^{(7)} + \dots \quad (1)$$

где каждый член $\mathcal{O}_i^{(D)}$ разложения представляет собой $SU(3)_C \otimes SU(2)_L \otimes U(1)_Y$ -инвариантный оператор размерности D , а параметры $c_i^{(D)}$, являющиеся множителями операторов в лагранжиане, называются коэффициентами Вильсона.

Morphing

$$T_{\text{out}}(\vec{g}_{\text{target}}) = \sum w_i(\vec{g}_{\text{target}}; \vec{g}_i) T_{\text{in}}(\vec{g}_i), = \vec{P}(\vec{g}) \cdot A \vec{T}$$

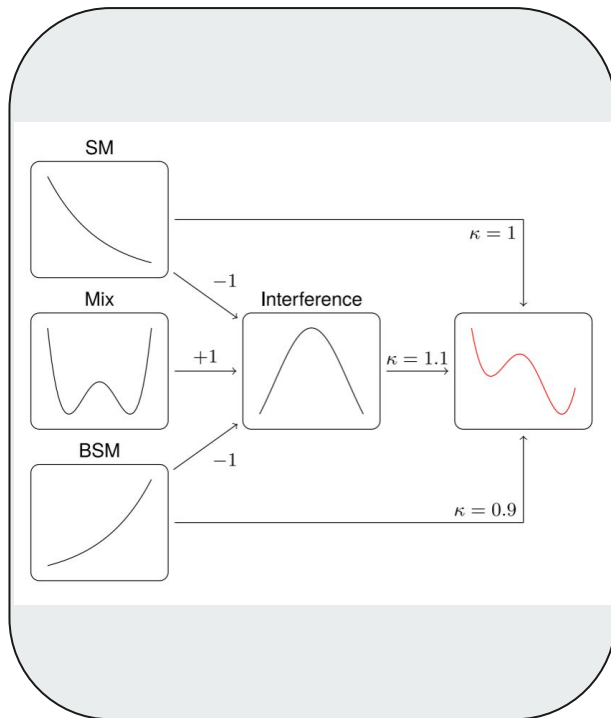
$$T_{\text{in},i}(g_{\text{SM},i}, g_{\text{BSM},i}) \propto$$

$$\propto g_{\text{BSM},i}^2 |\mathcal{O}_{\text{BSM}}|^2 + g_{\text{SM},i}^2 |\mathcal{O}_{\text{SM}}|^2 + 2g_{\text{SM},i}g_{\text{BSM},i} \mathcal{R}(\mathcal{O}_{\text{SM}}^* \mathcal{O}_{\text{BSM}}), i = 1, \dots, 3.$$

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \cdot \begin{pmatrix} g_{\text{SM},1}^2 & g_{\text{SM},2}^2 & g_{\text{SM},3}^2 \\ g_{\text{BSM},1}^2 & g_{\text{BSM},2}^2 & g_{\text{BSM},3}^2 \\ g_{\text{SM},1}g_{\text{BSM},1} & g_{\text{SM},2}g_{\text{BSM},2} & g_{\text{SM},3}g_{\text{BSM},3} \end{pmatrix} = \mathbf{I}$$

$$\Leftrightarrow A \cdot G = \mathbf{I}$$

- +: Из “базисных” сэмплов (распределений) получаем желаемое.
- -: “базис” надо оптимизировать
- Надо найти матрицу A



Overdetermined Morphing

Для неквадратных матриц и вырожденных матриц обратных матриц не существует:

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \\ a_{41} & a_{42} & a_{43} \end{pmatrix} \cdot \begin{pmatrix} g_{SM,1}^2 & g_{SM,2}^2 & g_{SM,3}^2 & g_{SM,4}^2 \\ g_{BSM,1}^2 & g_{BSM,2}^2 & g_{BSM,3}^2 & g_{BSM,4}^2 \\ g_{SM,1}g_{BSM,1} & g_{SM,2}g_{BSM,2} & g_{SM,3}g_{BSM,3} & g_{SM,4}g_{BSM,4} \end{pmatrix} =$$

Рассмотрим задачу $A\vec{x} = \vec{b}$, где $A = [m \times n], m \geq n$. Обозначим $E(\vec{x}) = \|\vec{b} - A\vec{x}\|^2, \|x_i\|^2 = \sum_i |x_i|^2$. Тогда:

$$= \mathbf{I} \Leftrightarrow A \cdot G = \mathbf{I} \quad (10)$$

$$E(\vec{x}) = (\vec{b} - A\vec{x})^T (\vec{b} - A\vec{x}) = (\vec{b})^T \vec{b} - 2(\vec{b})^T A\vec{x} + (\vec{x})^T A^T A\vec{x}$$

Методом наименьших квадратов:

$$\frac{\partial E(\vec{x})}{\partial \vec{x}} = -2A^T \vec{b} - 2A^T A\vec{x} = 0 \Rightarrow A^T A\vec{x} = A^T \vec{b} \Rightarrow \vec{x} = (A^T A)^{-1} A^T \vec{b}, \quad (11)$$

где $(A^T A)^{-1} A^T$ псевдо-инверсия A .



Поставим цель:

Усовершенствование метода для более широкого ряда условий
использования ->
-> Overdetermined Morphing

Структура Morphing Studies

1. `MorphingTools::MorphingBasisImpl<int nDimExternal
↔ int nDimInternal>`

2. Заполняем его методами описанными в MorphingTools.h и

т.д.

3. “Полиномизируем” матрицу G

4. Ищем G^{-1}

```
namespace MorphingTools{  
    template< Eigen::Matrix<float, 3,1> translateToInternal<3,1>(const std::array<float,1> & location){  
        Eigen::Matrix<float, 3,1> theVector;  
        theVector << 1 , location[0] , location[0]*location[0];  
        return theVector;  
    }
```

```
template <int nDimExternal, int nDimInternal>  
void MorphingBasisImpl<nDimExternal, nDimInternal>::initFromPoints(){  
    for (int col = 0; col < nDimInternal; ++col){  
        Eigen::Matrix<float,nDimInternal,1> inputSample =  
            =translateToInternal<nDimInternal,nDimExternal>(m_basisPoints[col].getLocation());  
        m_morphMatrix.col(col) = inputSample.col(0);  
    }  
    Eigen::JacobiSVD<Eigen::Matrix<float, nDimInternal, nDimInternal>> svd(m_morphMatrix);  
    m_condition = svd.singularValues()(0) / svd.singularValues()(svd.singularValues().size()-1);  
    m_weightMatrix = m_morphMatrix.inverse();  
}
```

[README](#) [Add LICENSE](#) [Add CHANGELOG](#) [Add CONTRIBUTING](#) [Add Kubernetes cluster](#) [Set up CI/CD](#)

[Configure Integrations](#)

Name	Last commit	Last update
MorphingStudies	Is impossible to do little changes even for on...	6 hours ago
Root	Is impossible to do little changes even for on...	6 hours ago
python	Add script to plot morphed shaped from a w...	9 months ago
util	Is impossible to do little changes even for on...	6 hours ago
CMakeLists.txt	VB3D 3D morphing and first steps towards ref...	2 weeks ago
Readme.md	Update Readme.md	2 weeks ago

[Readme.md](#)

Purpose of the package

This package is intended to provide EFT morphing functionality in plain ROOT, outside RooFit. It can for example be used to optimise morphing bases and to study signal predictions in a more accessible manner than RooFit allows for. It also avoids the tight coupling to a particular input file format that the ROOFit morphing class insists on. The package is designed to work with ATLAS analysis releases, and has a minimum set of dependencies - `NtupleAnalysisUtils` for some advanced plotting features and `ROOT / Eigen`.

Структура Morphing Studies

1. `MorphingTools::MorphingBasisImpl<int nDimExternal
↔ int nDimInternal>`

2. Заполняем его методами описанными в MorphingTools.h и

Т.д.

3. “Полиномиализуем” матрицу G

4. Ищем $G^{(-1)} \rightarrow (G^T G)^{(-1)} G^T$

```
template <int nDimExternal, int nDimInternal>
void MorphingBasisImpl<nDimExternal, nDimInternal>::initFromPoints(){
    for (int col = 0; col < nDimInternal; ++col){
        Eigen::Matrix<float, nDimInternal, 1> inputSample =
            =translateToInternal<nDimInternal, nDimExternal>(m_basisPoints[col].getLocation());
        m_morphMatrix.col(col) = inputSample.col(0);
    }
    Eigen::JacobiSVD<Eigen::Matrix<float, nDimInternal, nDimInternal>> svd(m_morphMatrix);
    m_condition = svd.singularValues()(0) / svd.singularValues()(svd.singularValues().size()-1);
    m_weightMatrix = m_morphMatrix.transpose()*m_morphMatrix.inverse()*m_morphMatrix.transpose();
}
```

```
namespace MorphingTools{
    template< Eigen::Matrix<float, 3, 1> translateToInternal<3, 1>(const std::array<float, 1> & location){
        Eigen::Matrix<float, 3, 1> theVector;
        theVector << 1, location[0], location[0]*location[0];
        return theVector;
    }
```

README Add LICENSE Add CHANGELOG Add CONTRIBUTING Add Kubernetes cluster Set up CI/CD

Configure Integrations

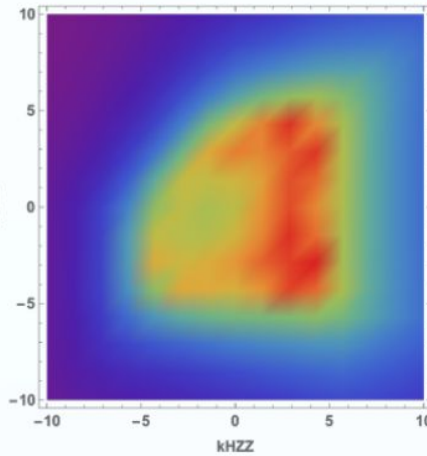
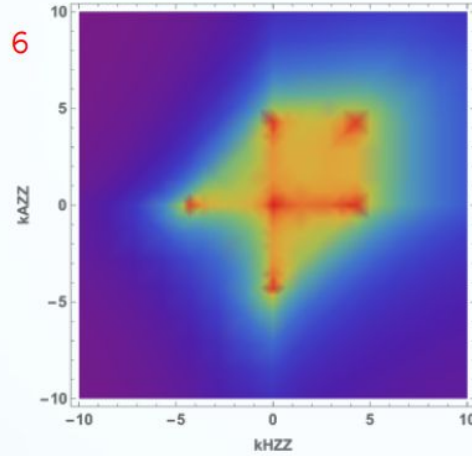
Name	Last commit	Last update
MorphingStudies	Is impossible to do little changes even for on...	6 hours ago
Root	Is impossible to do little changes even for on...	6 hours ago
python	Add script to plot morphed shaped from a w...	9 months ago
util	Is impossible to do little changes even for on...	6 hours ago
CMakeLists.txt	VB3D 3D morphing and first steps towards ref...	2 weeks ago
Readme.md	Update Readme.md	2 weeks ago

Readme.md

Purpose of the package

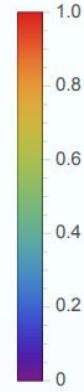
This package is intended to provide EFT morphing functionality in plain ROOT, outside RooFit. It can for example be used to optimise morphing bases and to study signal predictions in a more accessible manner than RooFit allows for. It also avoids the tight coupling to a particular input file format that the RooFit morphing class insists on. The package is designed to work with ATLAS analysis releases, and has a minimum set of dependencies - `NtupleAnalysisUtils` for some advanced plotting features and `ROOT / Eigen`.

6

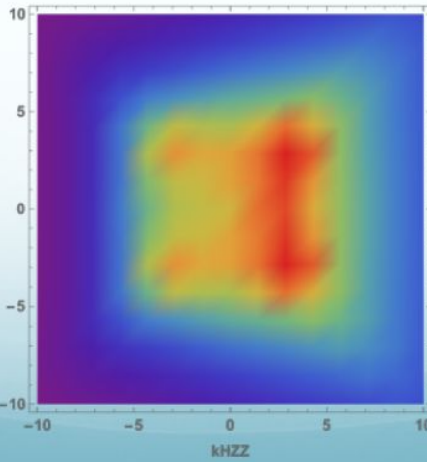
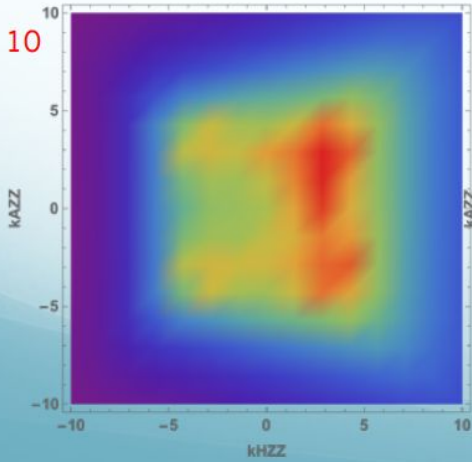


8

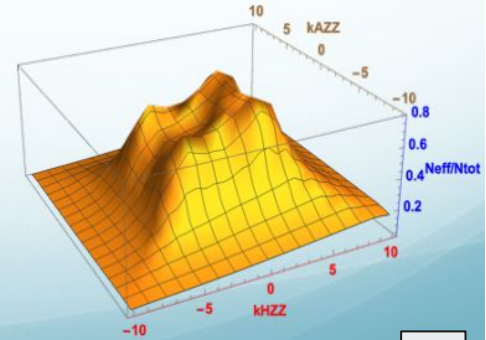
← Nbase



10



13





Заключение

В данной работе был проведён небольшой обзор на проблемы Стандартной модели и современной физики частиц, и в частности рассмотрено направление исследования бозона Хиггса, эффективная теория поля, как метод поиска новой физики, а также метод морфинга, разработкой и усовершенствованием которого занимается наша группа в ATLAS. Метод упрощает процедуру моделирования разных параметров их корреляцию.

В рамках научной работы были начаты разработка скрипта на языке C++ и усовершенствование для более общих случаев существующего модуля “MorphingStudies”, рассчитывающего матрицу морфинга, необходимую моделирования параметров (например распределений коэффициентов Вильсона).

В работе приведены частные косвенные подтверждения, что новый метод переопределённого морфинга должен улучшить достоверность моделирования.

Отдельно стоит отметить, что были улучшены навыки программирования на языке C++ с использованием математических библиотек, а также опыт работы с модулями стандарта ATLAS Analysis Release.