EOS: a Bayesian package for flavour physics

Progress in algorithms and numerical tools for QCD

Orsay – 19/06/2024

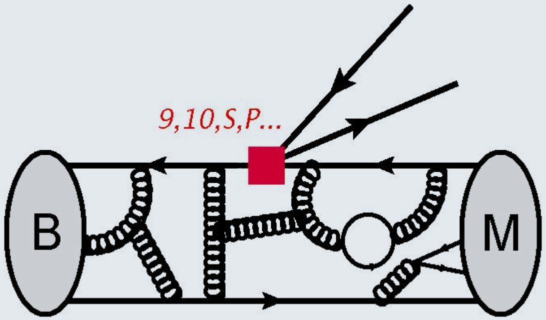
Méril Reboud

On behalf of the collaboration of EOS authors

Raison d'être with an example

- Let us consider B → K*μμ decays:
  - Dominant source of uncertainties are the **form-factors**

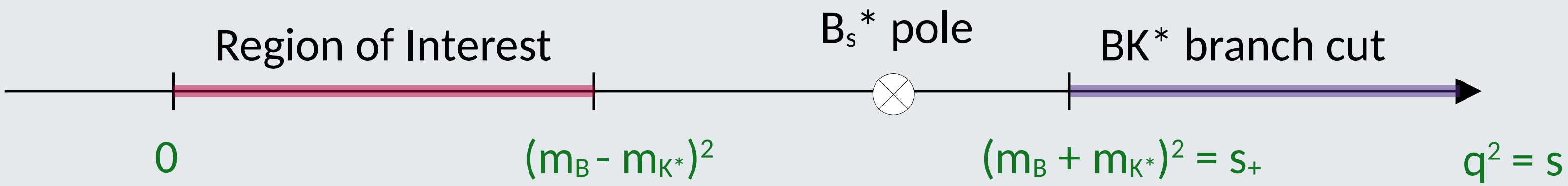
$$\mathcal{F}_\mu(k,q) = \langle \bar{M}(k) | \bar{s}\gamma_\mu b_L | \bar{B}(q+k) \rangle$$



+ other diagrams...

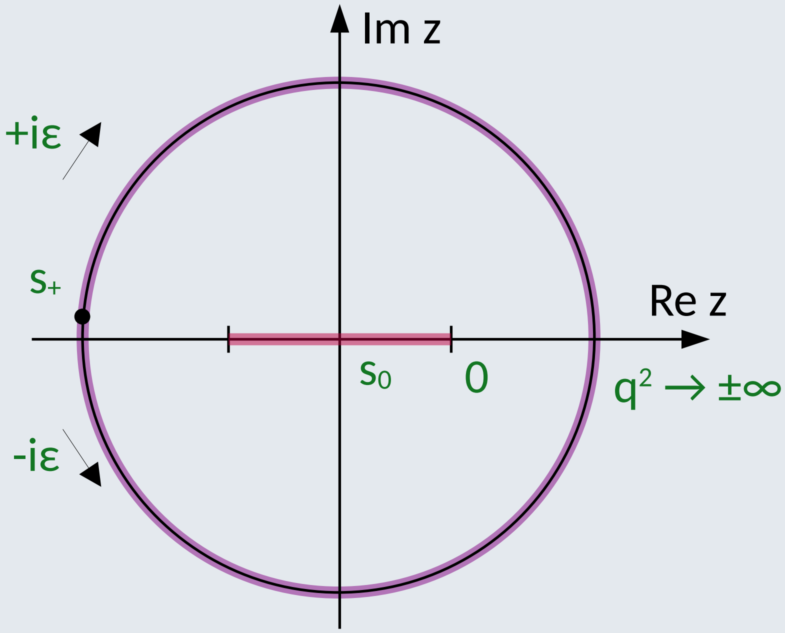
  - The best estimates come from a **combined analysis**: [Gubernari, MR, van Dyk, Virto '23]
    - Light-cone Sum Rules (LCSR) → Highly correlated estimate at a few negative q^2 points
    - Lattice QCD → Assumes a parametrization

Raison d'être with an example (II)



Conformal mapping [Boyd, Grinstein, Lebed '97]

$$z(s) \equiv \frac{\sqrt{s_+ - s} - \sqrt{s_+ - s_0}}{\sqrt{s_+ - s} + \sqrt{s_+ - s_0}}$$



Simplified Series expansion [Bourrely, Caprini, Lellouch, '08; Bharucha, Feldmann, Wick '10]

$$\mathcal{F}_\lambda^{(T)}(q^2) = \frac{1}{q^2 - m_{B_s^*}^2} \sum_{k=0}^{N} \alpha_{\lambda,k} z^k$$

On the lattice, extrapolations (continuum, physical points, ...) are performed on the $\alpha_{\lambda,k}$

Raison d'être with an example (III)

- We need to:
  - **Combine** q^2 constraints and parametric constraints
  - Potentially add **further constraints** (experimental, dispersive bounds, ...)
  - Potentially change the parametrization
  - Use this combination to **predict observables**

→ Big analysis with many varied parameters (~ 100, most of them are nuisance parameters), **proper samples** are needed for further analyses

→ **Bayesian framework** is absolutely essential

The EOS software

- **~1500 (pseudo-)observables**
- For each observable, one can:
  - **Select** different **models** (SM vs Effective theory, form-factor parametrization,…)
  - **Vary** all the **parameters**
- Fast evaluation is ensured by the **C++ back-end**, multi-threading, and observable caching
- EOS also implements a **python front-end**
- Current version: **EOS v1.0.11**

https://eos.github.io/

The EOS software

- **~1500 (pseudo-)observables covering:**
  - (semi)leptonic charged-current B meson decays (e.g. $B \to D^* \tau\nu$)
  - semileptonic charged-current Λ_b baryon decays (e.g. $\Lambda_b \to \Lambda_c(\to \Lambda\pi)\mu\nu$)
  - rare (semi)leptonic and radiative neutral-current B meson decays (e.g. $B \to K^*\mu\mu$)
  - rare semileptonic and radiative neutral-current Λ_b baryon decays (e.g. $\Lambda_b \to \Lambda(\to p\pi)\mu\mu$)
  - B-meson mixing observables (e.g. Δm_s)
  - hadronic tree-level B meson decays (e.g. $B \to DK$)
  - a handful of charm decays (e.g. $D \to K\mu\nu$)

Example: $b \to u\ell\nu_\ell$ decays [Leljak, MR, *et al.* 2302.05268]

- Goal: **Extract V_{ub}** and study **new physics** from all available $b \to u\ell\nu_\ell$ decays:

$$\mathcal{H}^{ub\ell\nu} = -\frac{4G_F}{\sqrt{2}}\tilde{V}_{ub}\sum_i \mathcal{C}_i^\ell \mathcal{O}_i^\ell + \cdots + \text{h.c.}$$

with e.g. $\mathcal{O}_{V,L}^\ell = \left[\bar{u}\gamma^\mu P_L b\right]\left[\bar{\ell}\gamma_\mu P_L \nu\right]$

  - $B \to \pi\,\{e,\mu\}\,\bar{\nu}$, $B \to \rho\,\{e,\mu\}\,\bar{\nu}$, $B \to \omega\,\{e,\mu\}\,\bar{\nu}$, mostly from BaBar and Belle
  - Form-factors from LQCD ($B \to \pi$) and QCD sum rules

Example: $b \to u \ell \nu_\ell$ decays [Leljak, MR, *et al.* 2302.05268]

- The analysis setup is written in a **yaml file** → Reproducibility and easy sharing

```
likelihoods:
  - name: TH-pi
    constraints:
      - 'B->pi::f_++f_0+f_T@FNAL+MILC:2015C;form-factors=BCL2008-4'
      - ...
priors:
  - name: CKM
    parameters:
     - { 'parameter': 'CKM::abs(V_ub)', 'min': 3.0e-3, 'max': 4.0e-3, 'type': 'uniform' }
...
```

- Tasks (optimize, sample, plot…) can be run from command line or from python

```
eos-analysis sample-nested -f analysis.yaml CKMfit
```

Example: $b \to u\ell\nu_\ell$ decays [Leljak, MR, *et al.* 2302.05268]

- Global fits

	Goodness of fit			
Data set	χ^2	d.o.f.	p value [%]	$\lvert V_{ub}\rvert \times 10^3$
$\bar{B} \to \pi\ell\nu$	27.83	31	62.98	$3.79^{+0.15}_{-0.15}$
$\bar{B} \to \rho\ell\nu$	4.05	10	94.49	$2.92^{+0.28}_{-0.25}$
$\bar{B} \to \omega\ell\nu$	4.20	4	37.90	$3.00^{+0.38}_{-0.32}$
all data	43.75	47	60.78	$3.59^{+0.13}_{-0.12}$

- MCMC or nested sampling (based on the public code dynesty: https://dynesty.readthedocs.io/)

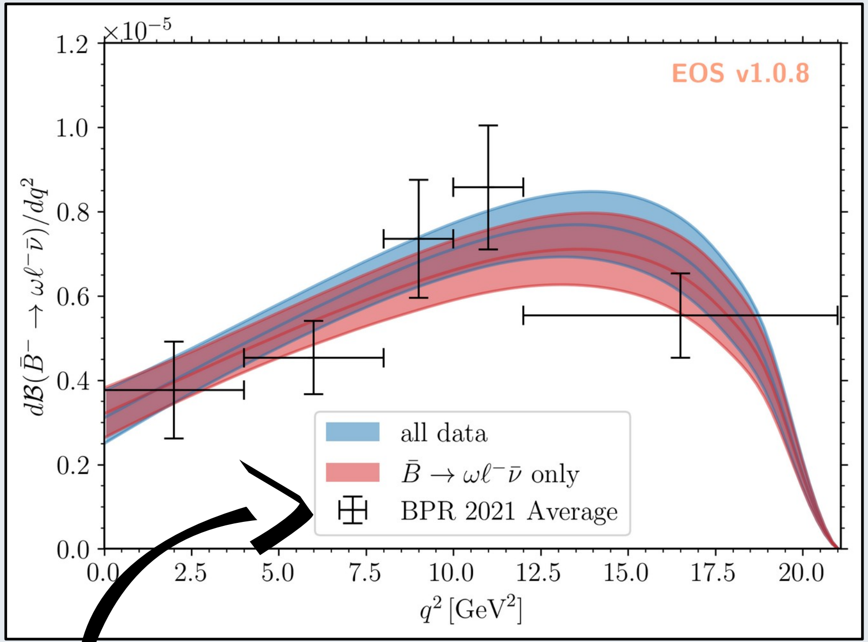
- Use the sample to predict observables

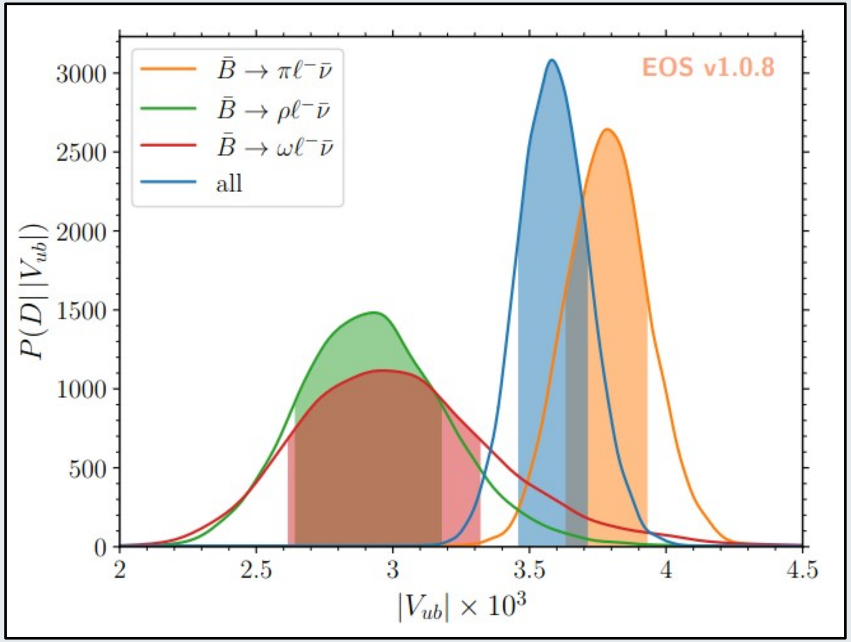
$$\mathcal{B}(\bar{B}^- \to \tau^-\bar{\nu}) = \left(8.28^{+0.61}_{-0.57}\Big|_{|V_{ub}|} \pm 0.13\Big|_{f_B}\right) \times 10^{-5},$$

$$\mathcal{B}(\bar{B}^- \to \mu^-\bar{\nu}) = \left(3.72^{+0.27}_{-0.25}\Big|_{|V_{ub}|} \pm 0.06\Big|_{f_B}\right) \times 10^{-7},$$

$$\mathcal{B}(\bar{B}^- \to e^-\bar{\nu}) = \left(8.71^{+0.64}_{-0.60}\Big|_{|V_{ub}|} \pm 0.14\Big|_{f_B}\right) \times 10^{-12}.$$

- Versatile plotting framework based on matplotlib

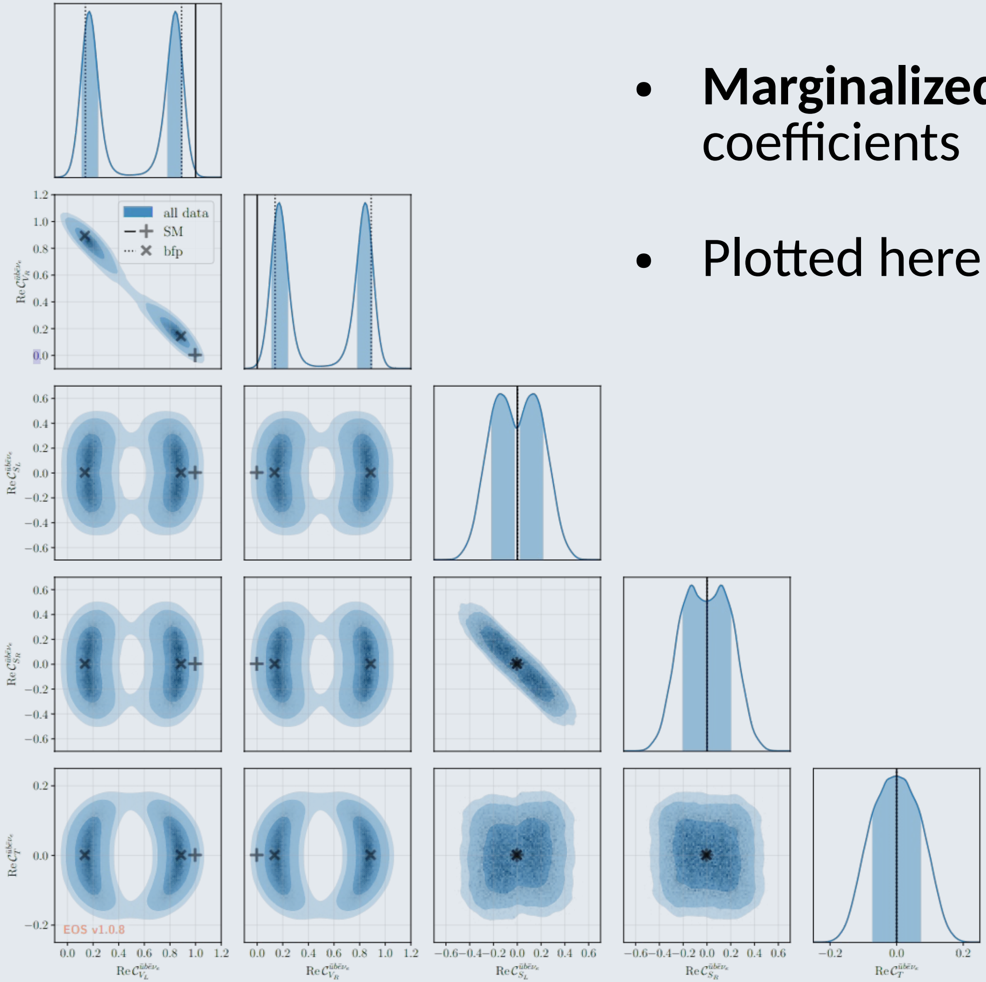




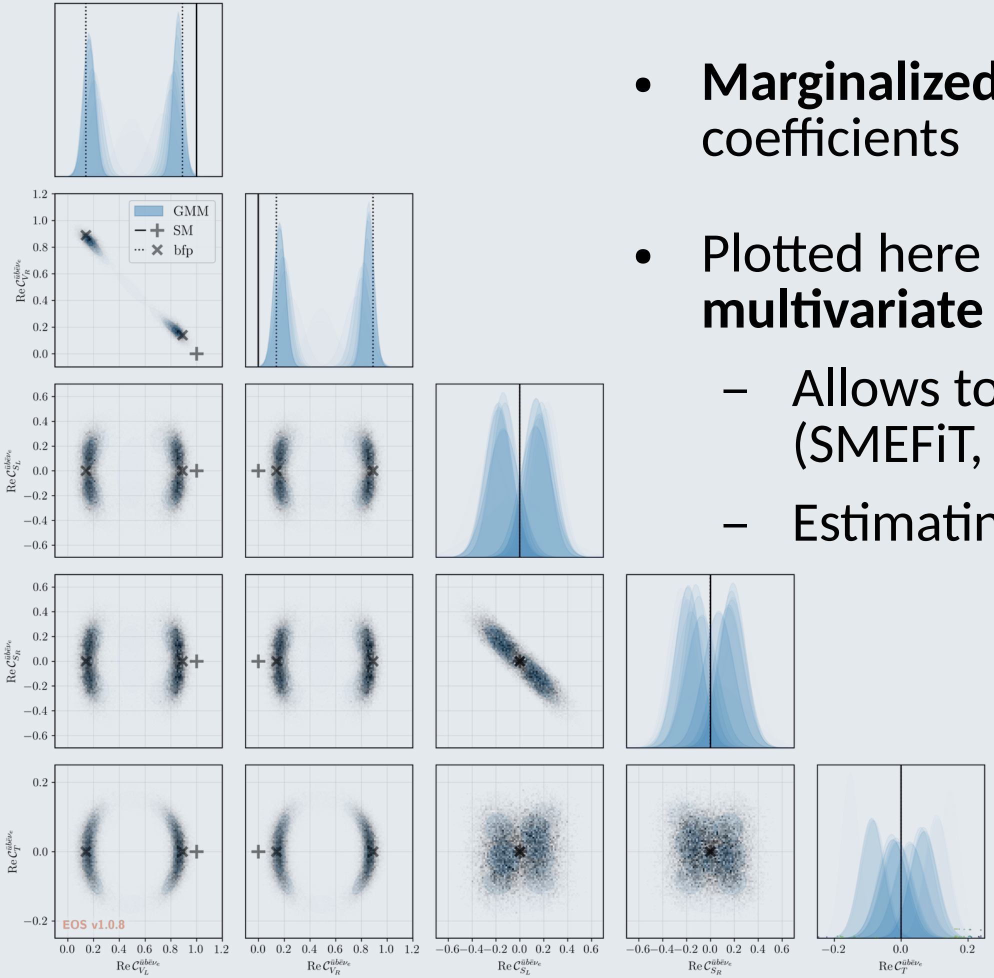
Database of references

[BPR:2021A] $B \to \rho \ell \bar{\nu}$ and $\omega \to \ell \bar{\nu}$ in and beyond the Standard Model: Improved predictions and \abs$V_{ub}$ arXiv 2104.05739

Example: $b \to u\ell\nu_\ell$ decays [Leljak, MR, *et al.* 2302.05268]



- **Marginalized likelihood** in the space of BSM Wilson coefficients
- Plotted here with KDE



- **Marginalized likelihood** in the space of BSM Wilson coefficients

- Plotted here with KDE, but can also be described with a **multivariate Gaussian mixture density** (scipy's GMM):
  - Allows to export likelihood → **contact with other tools** (SMEFiT, Flavio, ...)
  - Estimating the test statistics requires some work

Conclusion

- **EOS is open source**, the current version is **v1.0.11**
  - **~1500 (pseudo-)observables**, **~1500 parameters**, **~750 constraints**
  - Development on **github** https://github.com/eos/eos/
  - Used in **~40 theory papers** and many experimental papers; part of Belle II external software
- **Online documentation** and tutorials: https://eos.github.io/doc/
- We are happy to discuss how to **add further observables**

Backup

Installation

- **Installation is very simple** on Linux:

  pip3 install --user eoshep

- Then open a **Jupyter Notebook** and just run

  import eos

- For other platforms and **further details:**
  https://eos.github.io/doc/installation.html

- **Support** can be found here:
  https://discord.com/invite/hyPu7f7K6W