

An eigensolver for BOUT++

D. Dickinson and B. Dudson

York Plasma
Institute



This work has been carried out within the framework of the EUROfusion Consortium and has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement number 633053. The views and opinions expressed herein do not necessarily reflect those of the European Commission.

Acknowledgements



D. Dickinson* and B. Dudson

*d.dickinson@york.ac.uk

York Plasma Institute, Department of Physics, University of York, YO5 10DQ, UK

This work was part funded by the RCUK Energy programme, EURATOM, Enabling research project ER15-CCFE-03 and by a EUROfusion fusion researcher fellowship (WP14-FRF-CCFE/Dickinson).

York Plasma
Institute



- Why is an eigensolver useful?
- Brief discussion of implementation details
- Example: Wave in a box
- Example: ELM_pb model
- Summary

Why an eigenvalue solver for BOUT++?



BOUT++ is a nonlinear initial value code, so why develop an eigenvalue capability?

- Although nonlinear runs are useful, a lot of physics understanding comes from the linear behaviour of a system: mode frequencies and growth rates.
- For many studies, in particular ELM physics, most of the BOUT++ runs performed are linear simulations, where the aim is to find the growth rate. This requires long simulations in order to identify an eigenmode, and can only find the fastest growing mode.
- Allows more detailed comparison / benchmarking against linear codes such as ELITE or GATO
- Allows exploration of linear modes in a range of fluid and gyro-fluid models

How (1): Overview of implementation



- Want use of eigensolver to be easy for the user ***and*** developer → Implement as a type of time solver.
- Don't write our own eigenvalue library → Use the SLEPc library → Builds on top of PETSc (which BOUT++ already supports).
- Result is essentially that once BOUT++ has been compiled with SLEPc support user can just select `solver:type=slepc` in *any* of their ***linear*** physics models.
- In practice some knowledge of what eigensolver is doing is useful!

What (1): Eigenvalues of what?!



- We're finding eigenvalues + eigenvectors (eigenpairs) of some matrix operator, but what is the operator?
- Initially designed solver to find eigenvalues of time advance operator, M , acting on system state, f , i.e.

$$Mf(t) = \lambda f(t) = f(t + \delta t) \quad \text{with} \quad \lambda = e^{-i\Omega\delta t}$$

- More recently added option to find eigenvalues of ddt operator, N , i.e.

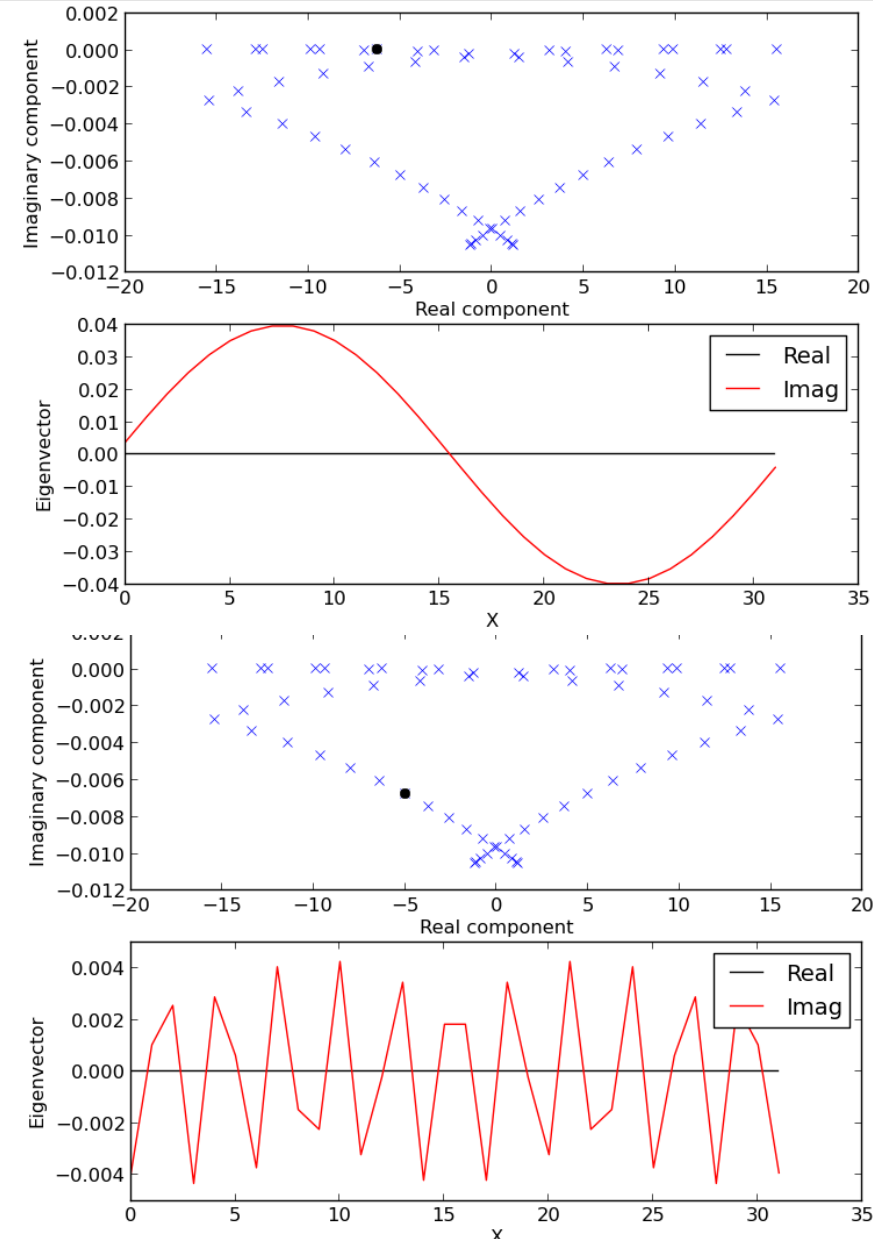
$$Nf(t) = \lambda f(t) = df(t)/dt \quad \text{with} \quad \lambda = -i\Omega$$

- Advantages + disadvantages to each which will be discussed in a few slides time.

Practical application: eigen-box



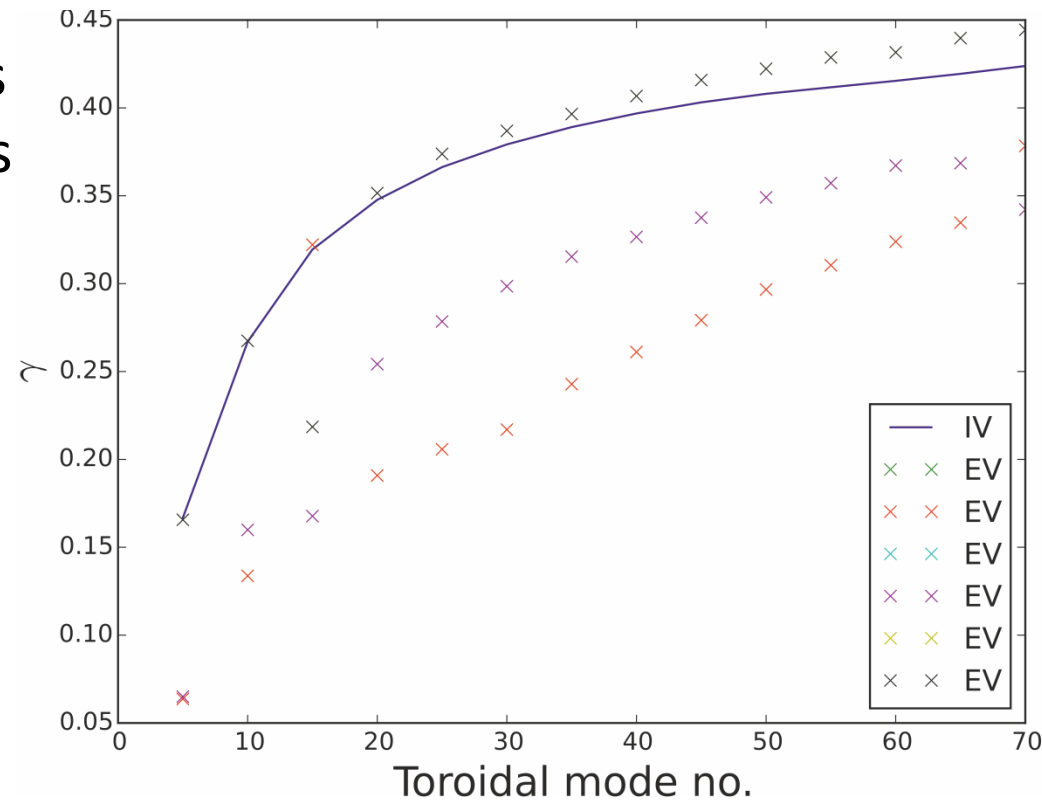
- Wave equation in a 1D box with fixed boundaries
 $dg/dx = D2DX2(f)$; $df/dx = g$
- Clearly identifies physical mode spectrum with zero growth-rate
- Numerical (unresolved) modes also found, with negative growth-rates
- Example includes python script to plot spectrum, click on eigenvalue to show eigenvector.



Practical application: elm-pb



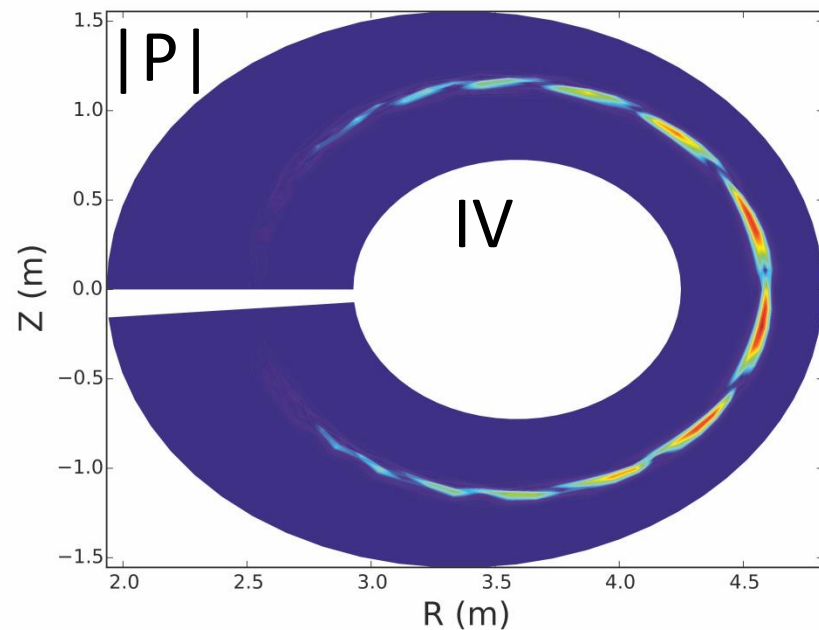
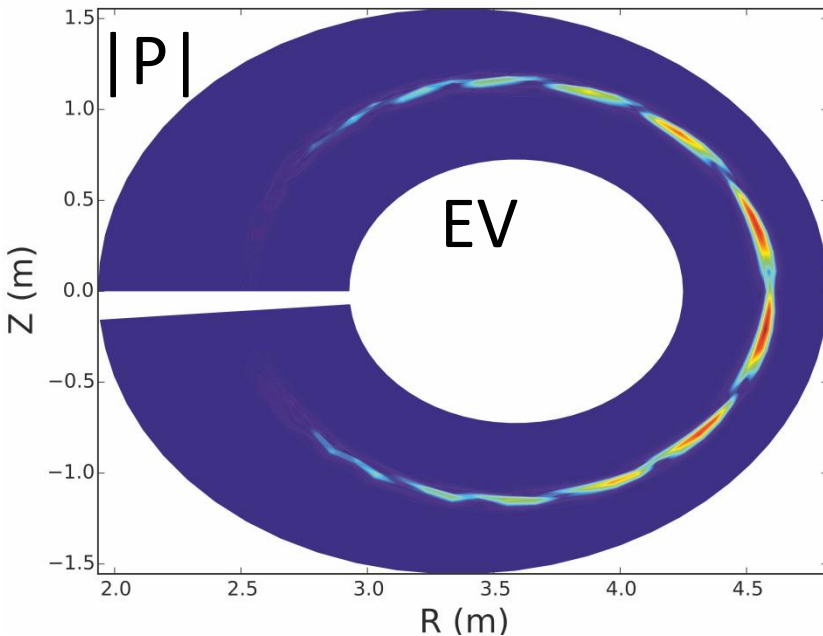
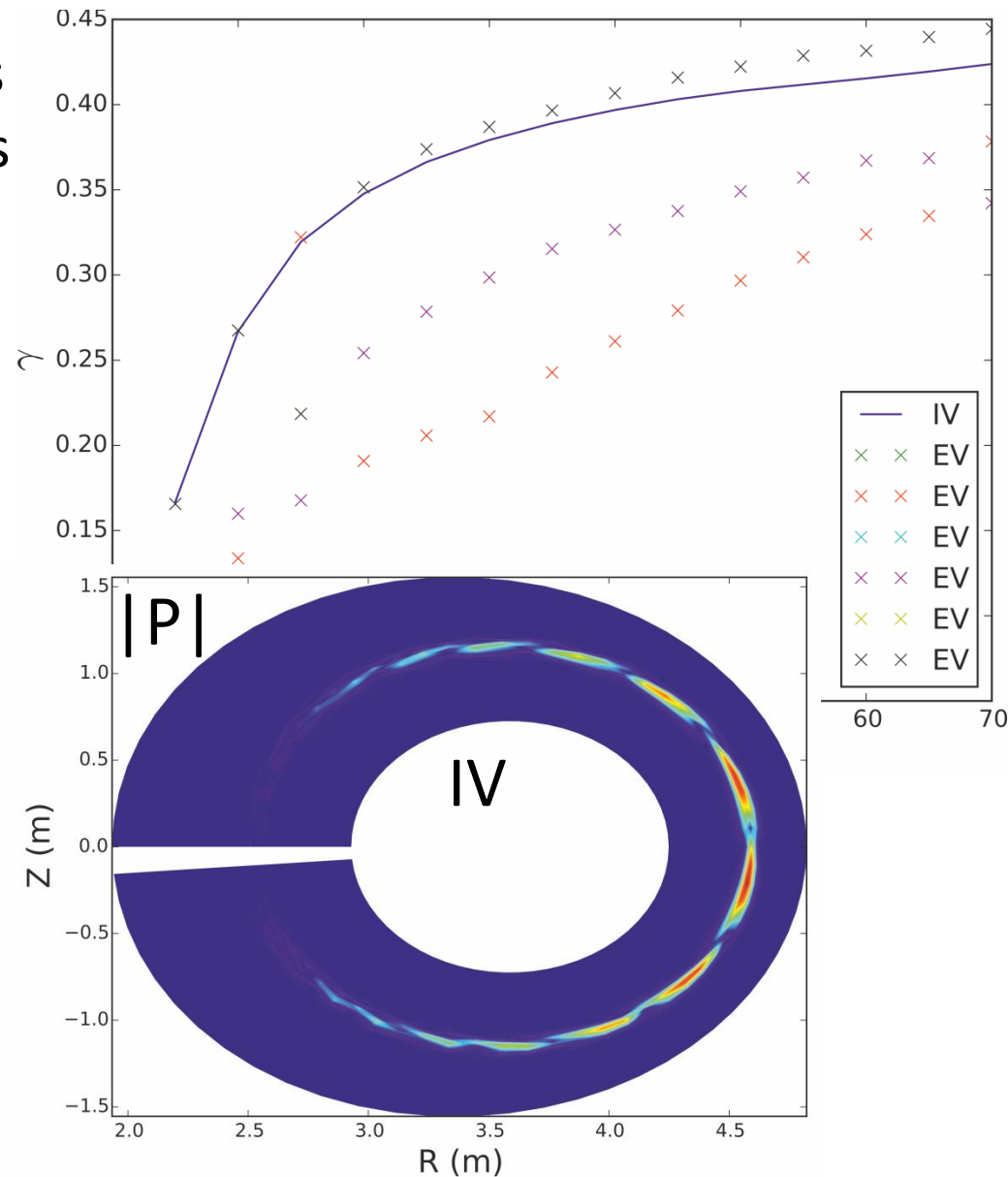
- Comparison of growth rates from linear initial value runs (green line), and SLEPc eigenvalues (crosses).



Practical application: elm-pb



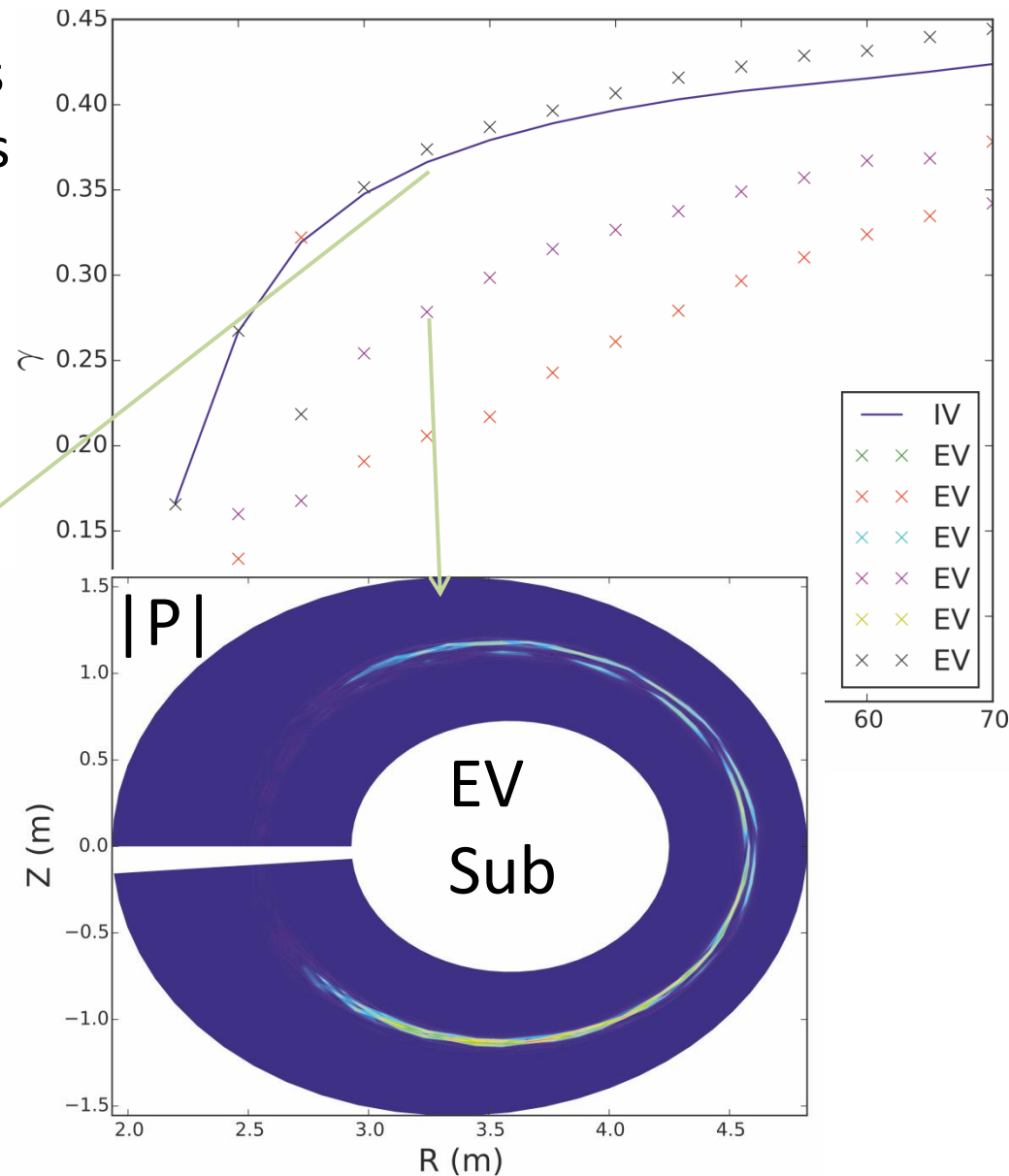
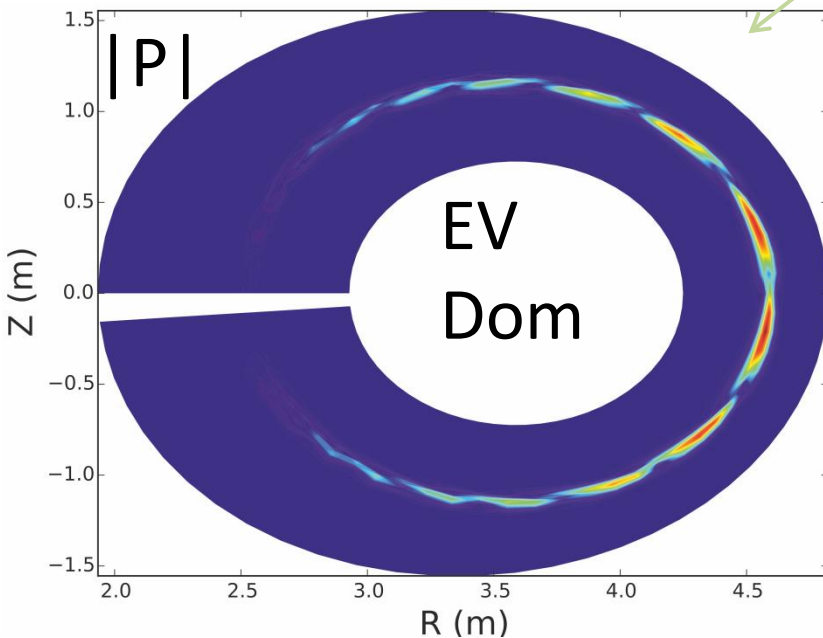
- Comparison of growth rates from linear initial value runs (green line), and SLEPc eigenvalues (crosses).
- Mode structure matches.



Practical application: elm-pb



- Comparison of growth rates from linear initial value runs (green line), and SLEPc eigenvalues (crosses).
- Mode structure matches.
- Sub-dominant, peaks off midplane?





- SLEPc solver implemented in BOUT++ can use existing (linear) physics models.
- Can find a subset of the eigenvalues of either the time advance or time derivative operators.
- Many (SLEPc) options can be tweaked to change the numerics, can have a large impact on speed **and** which eigenvalues are found → Not always the most interesting modes are found first.
- Requires some user experimentation to find best operating parameters for your case.
- Not well documented in the manuals yet, I aim to improve this but feel free to email me for guidance/help etc. in the meantime.

Any questions?



D. Dickinson* and B. Dudson

*d.dickinson@york.ac.uk

York Plasma Institute, Department of Physics, University of York, YO5 10DQ, UK

This work was part funded by the RCUK Energy programme, EURATOM, Enabling research project ER15-CCFE-03 and by a EUROfusion fusion researcher fellowship (WP14-FRF-CCFE/Dickinson).

York Plasma
Institute