TECHNISCHE UNIVERSITÄT WIEN

INSTITUT FÜR MECHANIK UND MECHATRONIK
Forschungsbereich Technische Akustik

New features from FEAST v4 Eigensolver

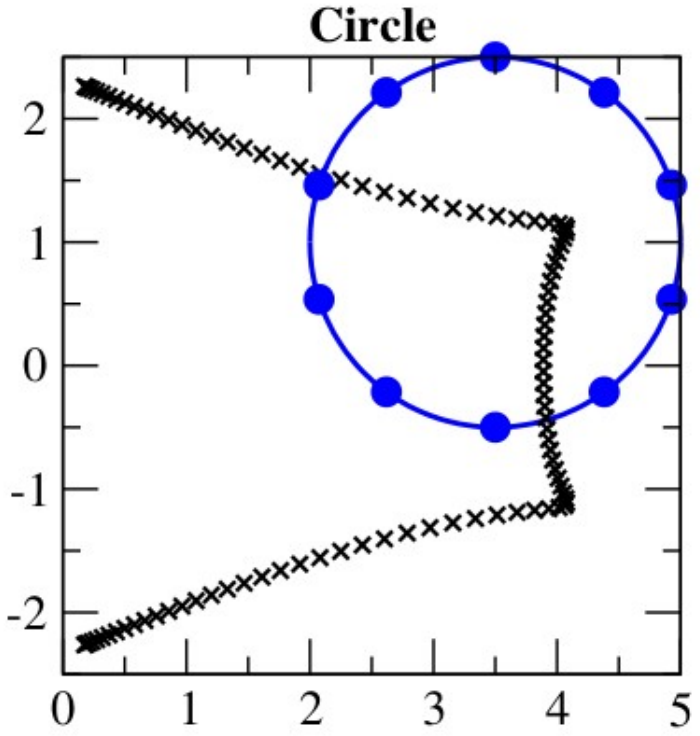
Heinrich, Ben

Scharner, Andreas

Strondl, Manuel

FEAST Eigenvalue Solver

- General purpose eigenvalue solver
- Calculates EVs in „search contour”
- Version 3.0 previously implemented in openCFS (for standard and generalised EVPs)
- Version 4.0: better performance, support for Polynomial EVP and Stochastic Estimate
- Goal: Implement Version 4.0, add Interface for Quadratic EVP and Stochastic Estimate



$$\sum_{i=0}^{p} \lambda^i A_i x = 0$$

Overview

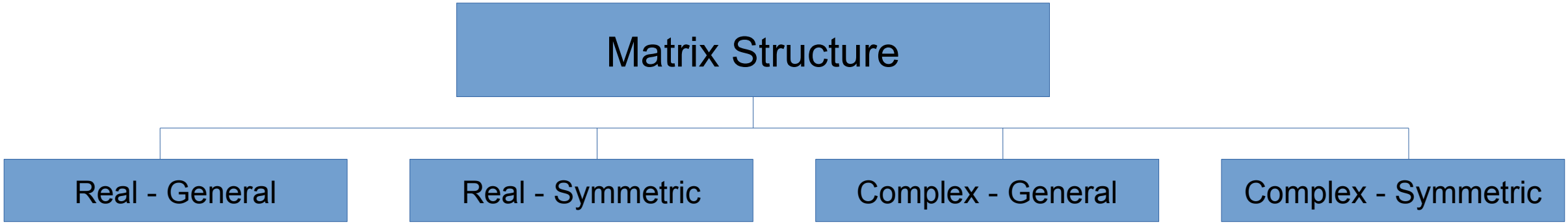
- Implementation FEAST v4 and quadratic EVP
- Testcases and Performance
- Stochastic Estimate Implementation an Test

Implementation - Build process

- Download mirror used for source
  - Download from FEAST website requires questionnaire
- Fortran compiler must be specified for FEAST v4
  - GNU (gfortran)
  - Intel (ifort)
- Type mismatch issue for gfortran
  - Fix with compiler flags via Makefile.in
  - Different flags depending on gfortran version (changed from version 10)

Implementation - FEAST interface

- Only sparse interface included in openCFS
- Different interface calls based on:
  - System matrix entry type (real, complex)
  - Eigenvalue problem type (standard, generalized, polynomial)
  - Structure type of system matrices:



Implementation - FEAST Polynomial example

- Real General Polynomial Eigenvalue Problem (CSR storage)
- External call (C, not C++!)

dfeast_gcsrpev(SystemMatrices, SolverSettings, Intervall, Results)

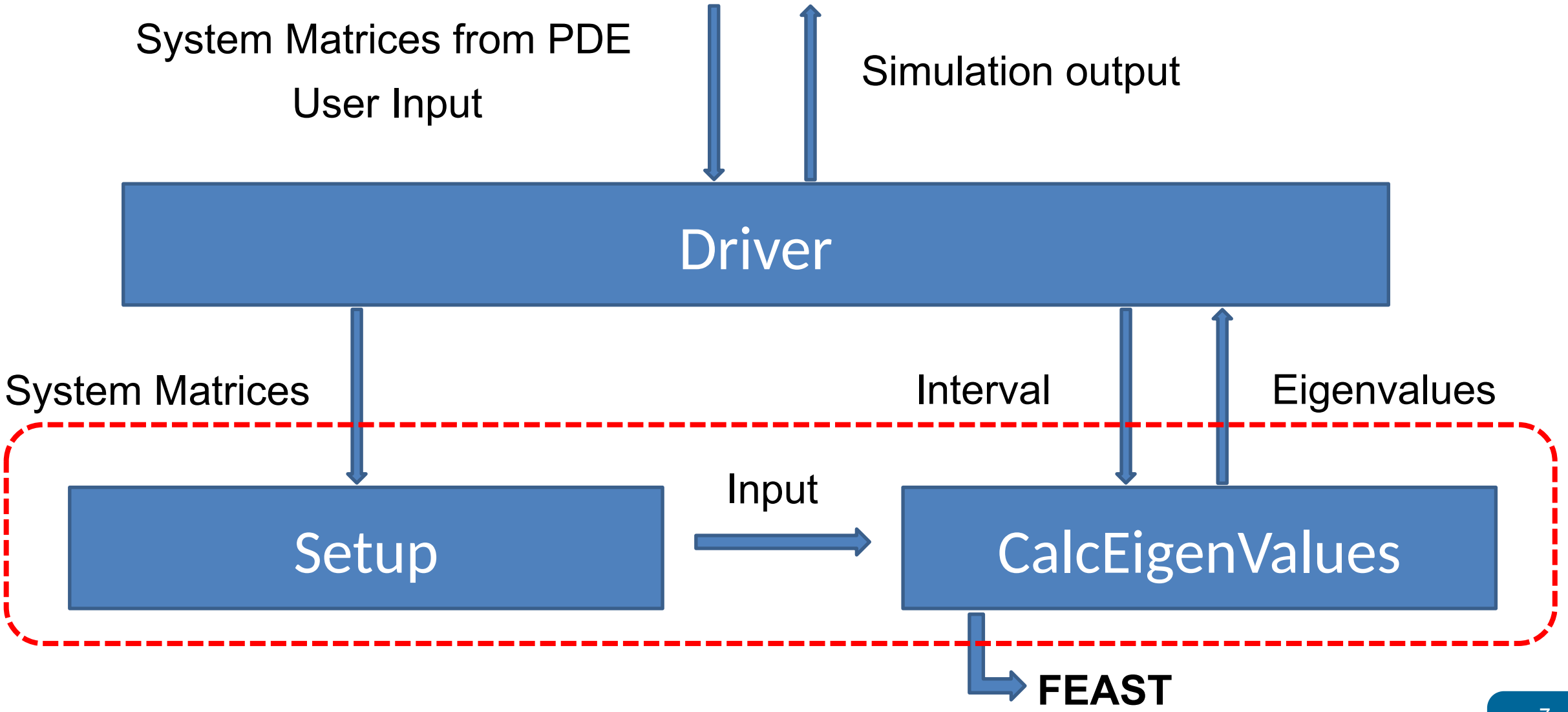
- SystemMatrices:
  - Polynomial degree p=2
  - System size N
  - 1 CSR arrays for 3 system matrices

[values0, values1, values2] **values** (3xNNZ)

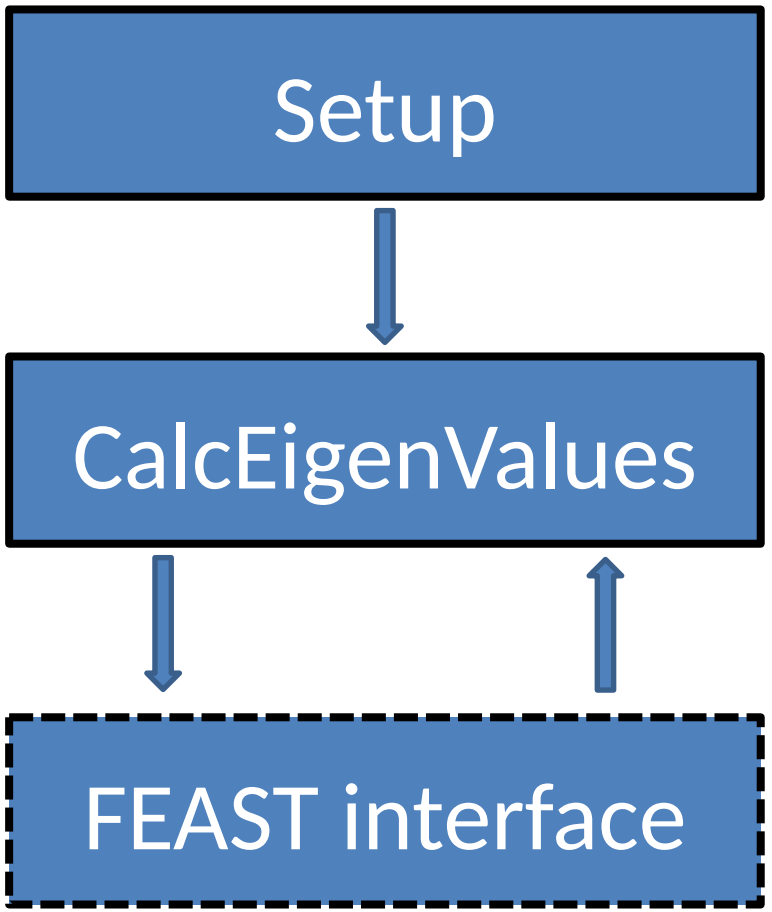
[I0,I1,I2] rows (3xN)

[J0,J1,J2] columns (3xNNZ) **coordinates**

Implementation - FeastEigenSolver class



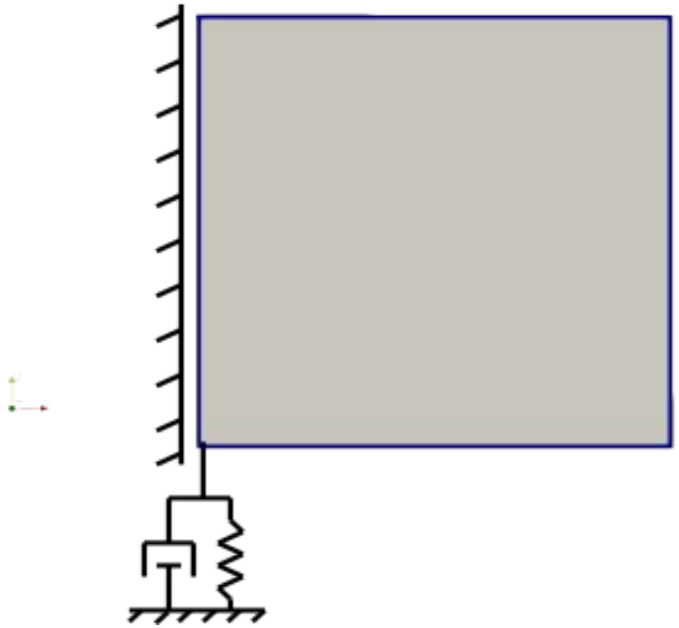
Implementation - Changes for quadratic EVP



- ✓ Add additional system matrix
- ✓ Stack system matrices for input
- ✓ New cases for quadratic problems
- ✓ Add polynomial calls

Testcases - 1D Oscillator

- Adapted existing Testcase for Quadratic Eigenvalue Solver

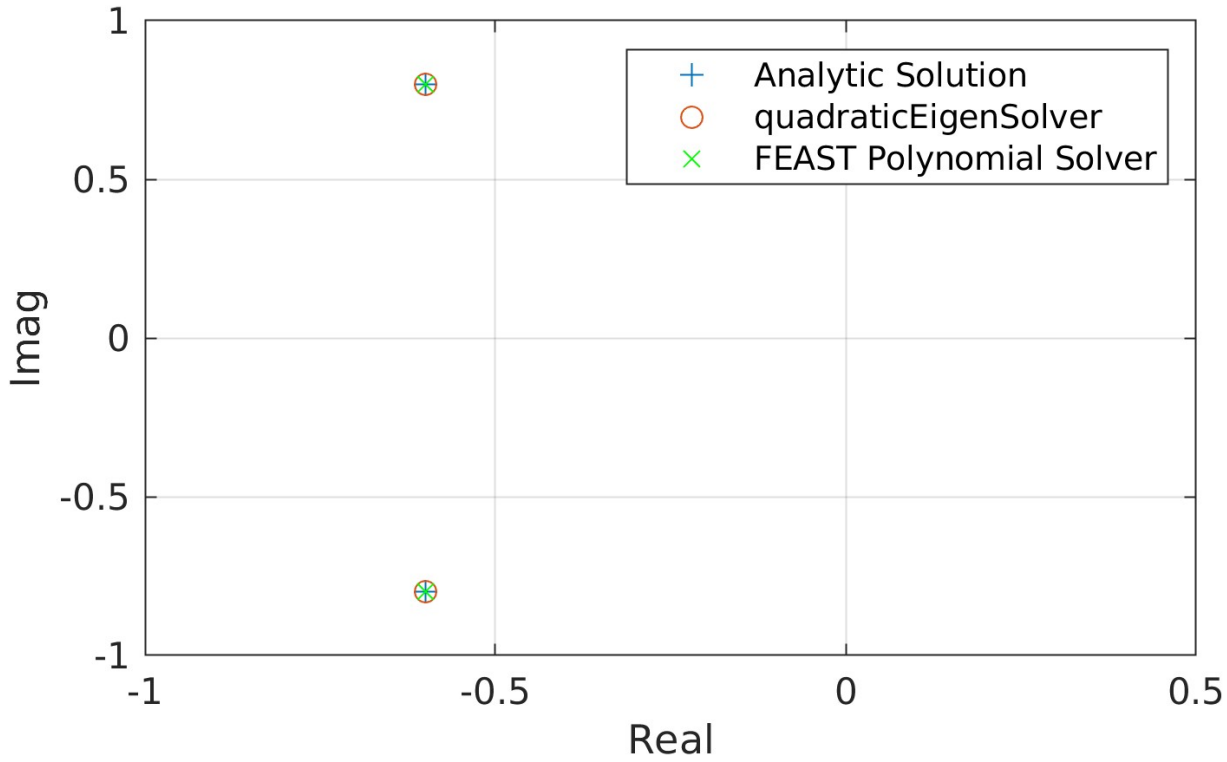


Analytic Solution:

$$\lambda = -\xi w_0 \pm w_0 \sqrt{\xi^2 - 1}$$

$$w_0 = \sqrt{\frac{k}{m}}$$

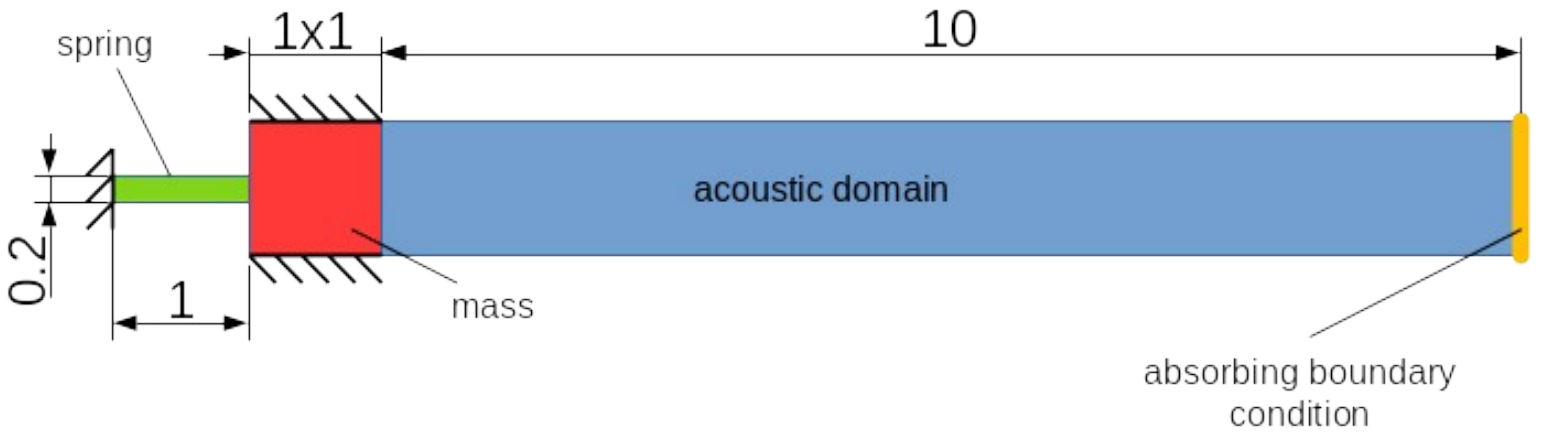
- Comparison shows exact agreement with Quadratic EV Solver (which solves linearised system)



Testcases - New Physical Testcase

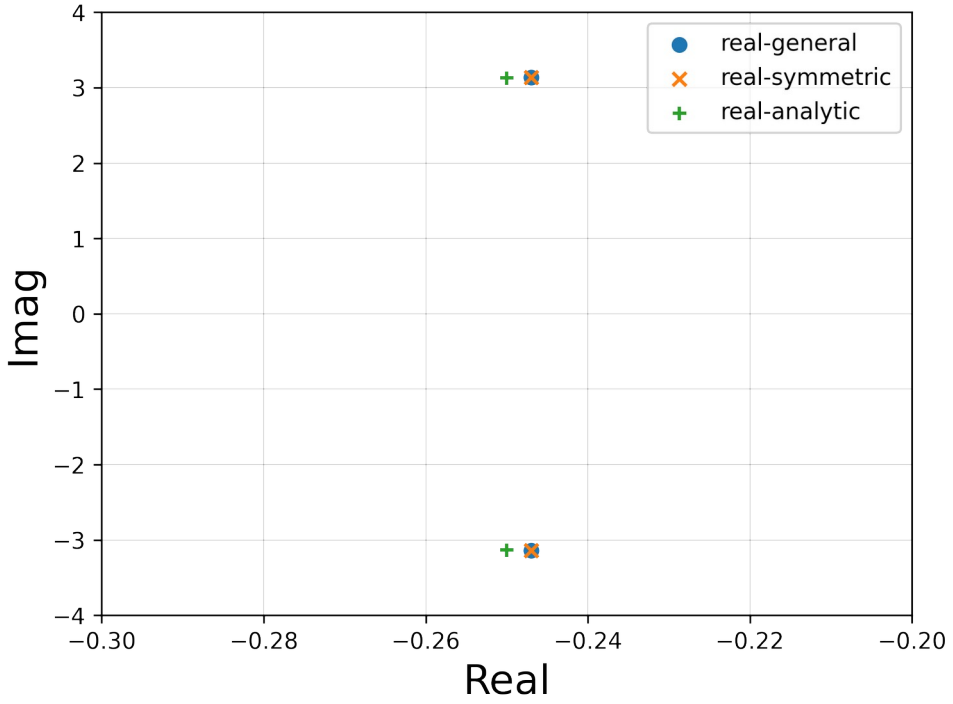
- One mass oscillator with acoustic radiation
- Enforce different storage types naturally via acoustic formulation and spring material
- Pressure/Potential acoustic formulation → Unsymmetric/Symmetric matrices
- Elastic/Viscoelastic spring material → Real/Complex storage type
- Gives one dominating oscillation mode + eigenvalue
- Compare to analytical solution

$$\lambda_{1,2} = -\frac{c + A\sqrt{\rho_f K_f}}{2m} \pm j\frac{\sqrt{4km - (c + A\sqrt{\rho_f K_f})^2}}{2m}$$

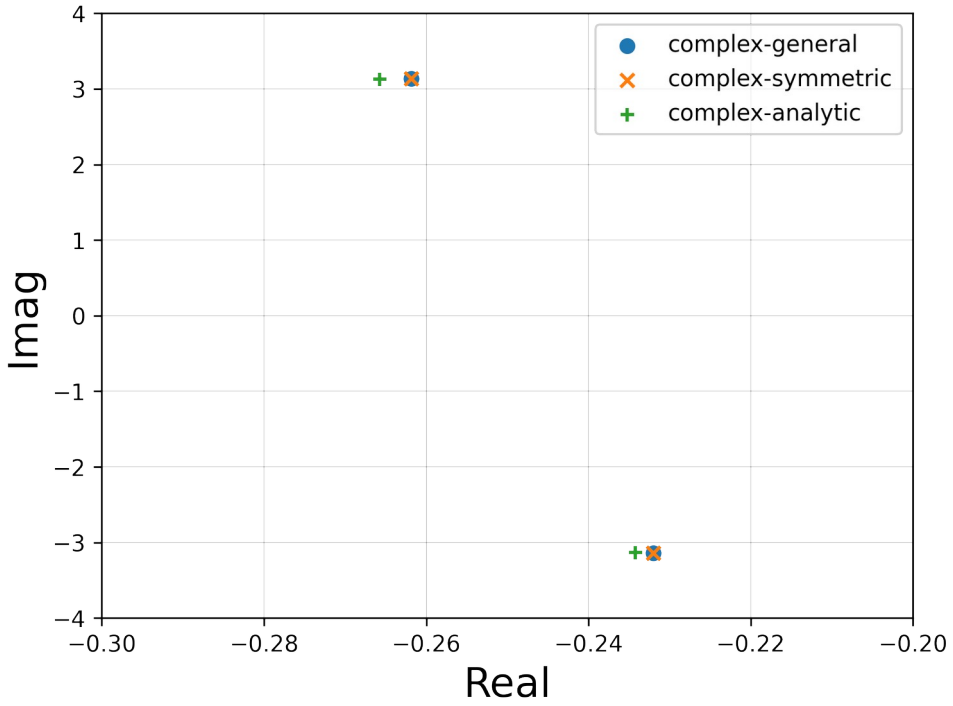


Testcases – New Physical Testcase

- Real spring parameters:

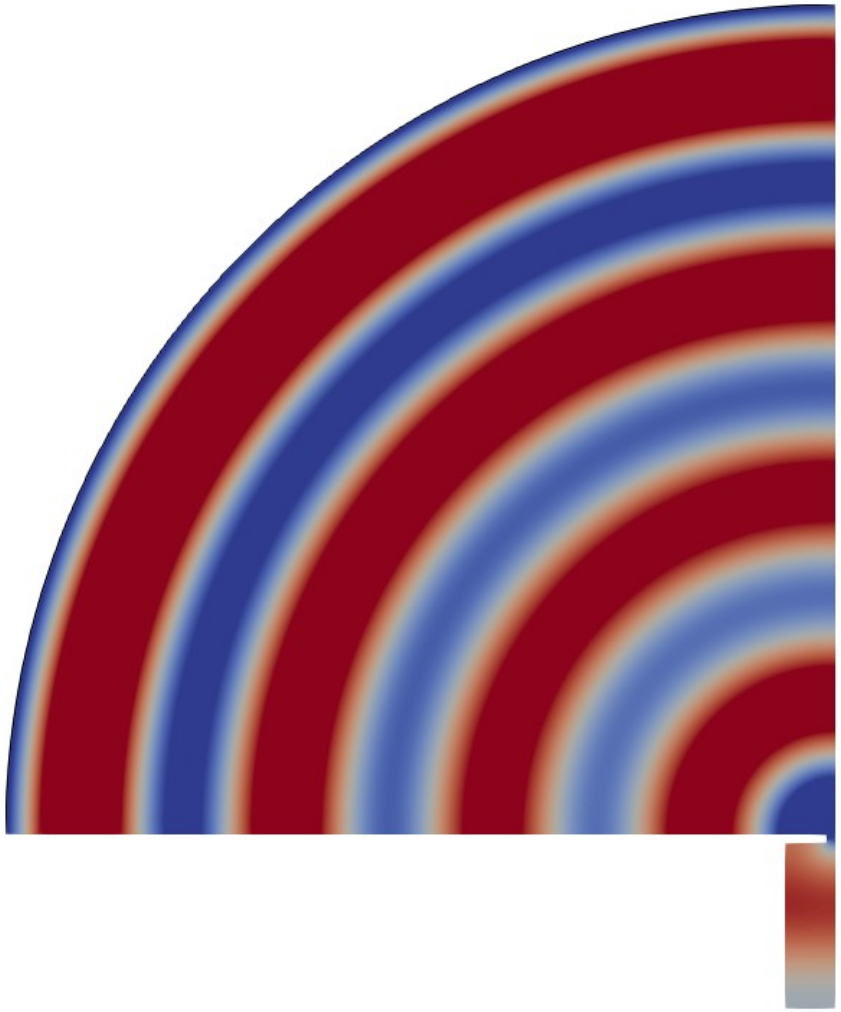


- Complex spring parameters:



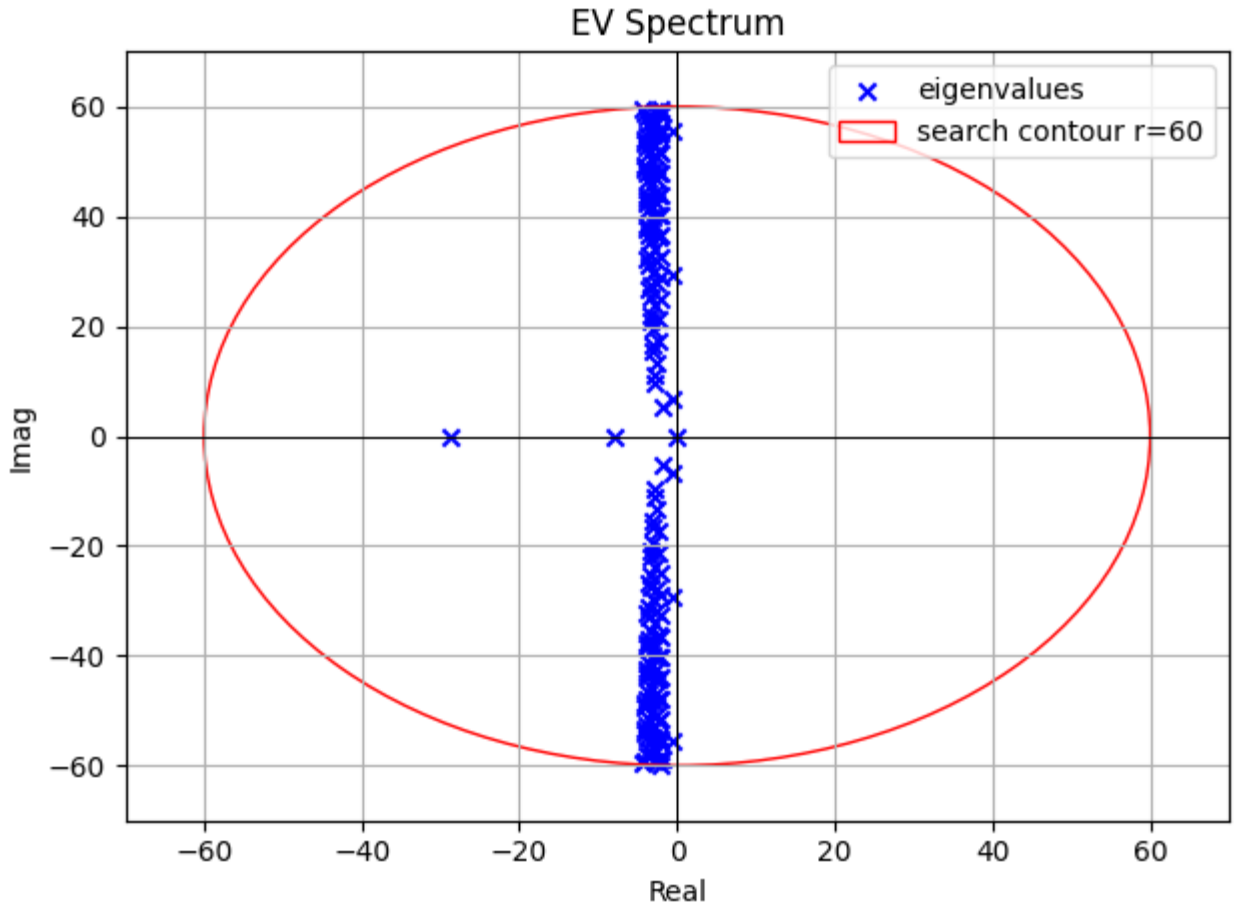
Performance Tests – Setup

- Acoustic Helmholtz Resonator
- Absorbing B.C. at circumference
- Damped by radiation
- QEVP with 32576 DoF



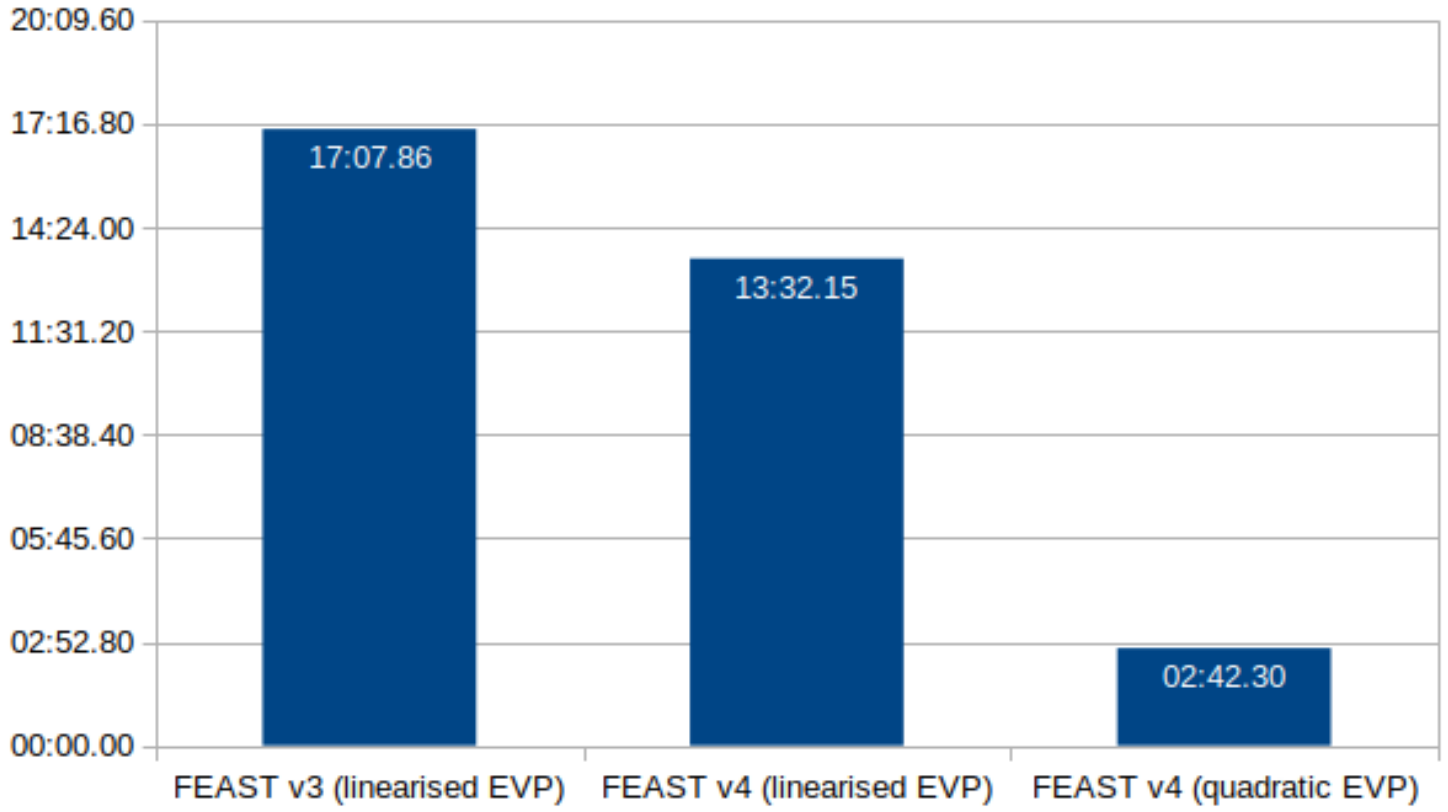
Performance Tests – Solver Settings

- Search contour with r=60 (9.55 Hz)
- 161 EVs in search contour



Performance Tests – CPU Time

- FEAST v3 (linearised EVP):
  - Iterations: 6
  - CPU time: 17:07.86
- FEAST v4 (linearised EVP):
  - Iterations: 6
  - CPU time: 13:32.15
- FEAST v4 (quadratic EVP):
  - Iterations: 4
  - CPU time: 2:42.30



Performance Tests – Interpretation

- Linearised EVP:

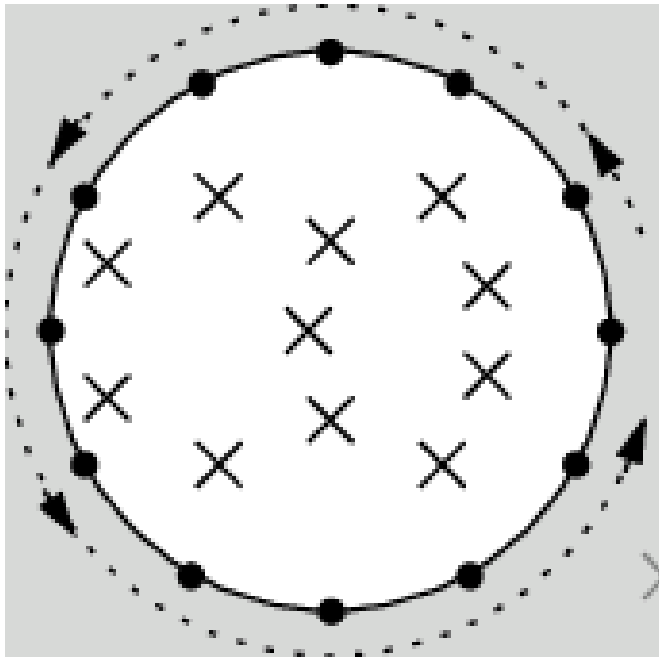
$$\left(\lambda^2 \boldsymbol{M} + \lambda \boldsymbol{C} + \boldsymbol{K}\right)\boldsymbol{\phi} = \boldsymbol{0} \quad \rightarrow \quad \left(\lambda \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{M} \end{bmatrix} + \begin{bmatrix} \boldsymbol{0} & -\boldsymbol{I} \\ \boldsymbol{K} & \boldsymbol{C} \end{bmatrix}\right) \begin{bmatrix} \boldsymbol{\phi} \\ \lambda\boldsymbol{\phi} \end{bmatrix} = \boldsymbol{0}$$

➔ 2x DoF!

- v4 generally faster than v3 (according to user manual)

Implementation - Stochastic Estimate

- Goal: improved workflow
  - Estimate for amount
  - of EVs in search contour
- Required changes
  - Change array of solver settings
  - Add console output for result
  - Store result in info XML

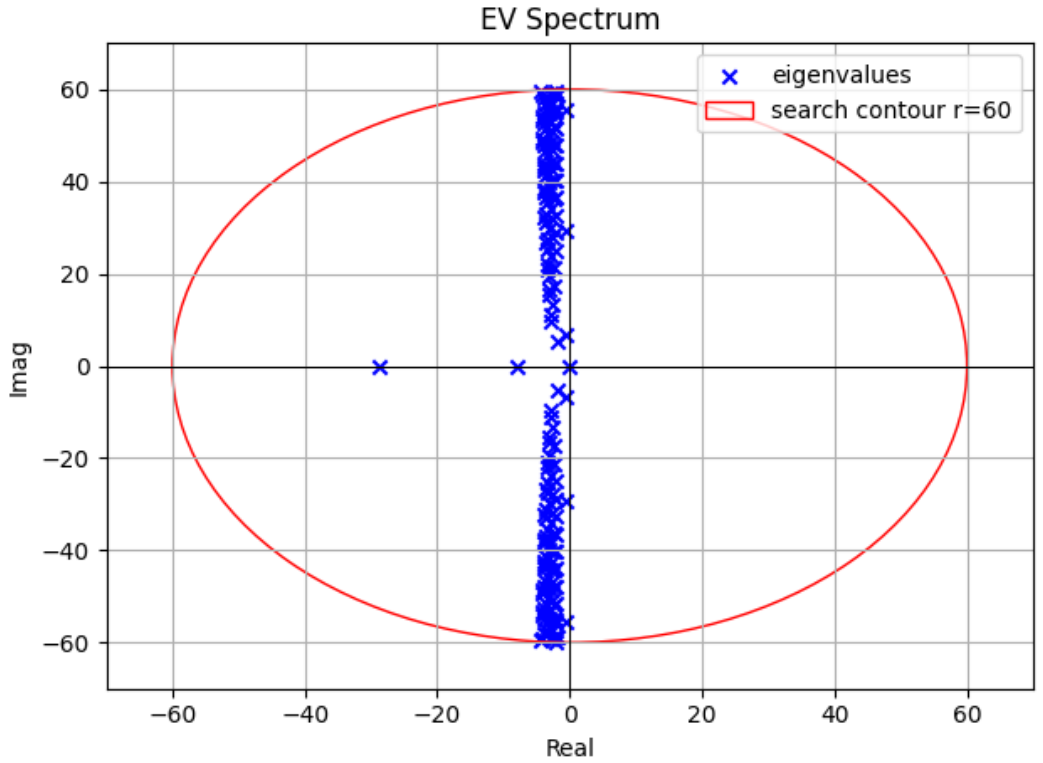


Performance Tests – Workflow

- FEAST calculates EVs by iterations:
  - Initial guess of EVs (M0) in defined search contour
  - M0 usually unknown ➔ trial and error
  - Very time consuming

- Subspace guess by FEAST v4:
  - M0 within 7.23 ± 0.2 seconds

➔ **Huge workflow improvement**



Summary

- FEAST v4 shows good performance improvement for solving quadratic EVPs
- Stochastic Estimate improves workflow
- For explanation of usage check out the Testcases in
  - /TESTSUIT/Solver/feast
  - /TESTSUIT/Coupledfield/MechAcou/ComplexSpringOscillator_EV2Dpressure