Experiences in solving nonlinear eigenvalue problems with SLEPc

Jose E. Roman

D. Sistemes Informàtics i Computació
Universitat Politècnica de València, Spain

PETSc Annual Meeting – June, 2023

DSIC

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

SLEPc: Scalable Library for Eigenvalue Problem Computations

A general library for solving large-scale sparse eigenproblems on parallel computers

- Linear eigenproblems (standard or generalized, real or complex, Hermitian or non-Hermitian) and related problems

$$Ax = \lambda Bx \qquad Av_i = \sigma_i u_i \qquad T(\lambda)x = 0 \qquad y = f(A)b$$

- Parallel: MPI and some support for GPU

Authors: J. E. Roman, C. Campos, L. Dalcin, E. Romero, A. Tomas

`http://slepc.upv.es`

Current version: 3.19 (released March 2023)

PETSc

Nonlinear Systems		
Line Search	Trust Region	. . .

Time Steppers				
Euler	Backward Euler	RK	BDF	. . .

Krylov Subspace Methods							
GMRES	CG	CGS	Bi-CGStab	TFQMR	Richardson	Chebychev	. . .

Preconditioners						
Additive Schwarz	Block Jacobi	Jacobi	ILU	ICC	LU	. . .

Matrices					
Compressed Sparse Row	Block CSR	Symmetric Block CSR	Dense	CUSPARSE	. . .

Vectors		
Standard	CUDA	ViennaCL

Index Sets		
General	Block	Stride

SLEPc

Nonlinear Eigensolver					
SLP	RII	N-Arnoldi	Interp.	CISS	NLEIGS

M. Function	
Krylov	Expokit

Polynomial Eigensolver			
TOAR	Q-Arnoldi	Linear-ization	JD

SVD Solver		
Cross Product	Cyclic Matrix	Thick R. Lanczos

Linear Eigensolver						
Krylov-Schur	Subspace	GD	JD	LOBPCG	CISS	. . .

Spectral Transformation			
Shift	Shift-invert	Cayley	Precond.

BV	DS	RG	FN
. . .	. . .	. . .	. . .

Outline

General Nonlinear Eigenproblems

NEP:
$$T(\lambda)x = 0, \qquad x \neq 0$$

$T : \Omega \to \mathbb{C}^{n\times n}$ is a matrix-valued function analytic on $\Omega \subset \mathbb{C}$

Example: Rational eigenproblem arising in the study of free vibration of plates with elastically attached masses

$$-Kx + \lambda Mx + \sum_{j=1}^{k} \frac{\lambda}{\sigma_j - \lambda} C_j x = 0$$

Example: Discretization of parabolic PDE with time delay τ

$$(-\lambda I + A + e^{-\tau\lambda} B)x = 0$$

User interface: callback functions or split form $T(\lambda) = \sum_{i=0}^{\ell-1} A_i f_i(\lambda)$

Currently Available NEP Solvers

C. Campos, J. E. Roman (2021), "NEP: A Module for the Parallel Solution of Nonlinear Eigenvalue Problems in SLEPc," ACM TOMS 47(3)

Single eigenvalue

1. Successive linear problems (SLP) [Ruhe 1973]
2. Residual inverse iteration (RII) [Neumaier 1985]
3. Nonlinear Arnoldi [Voss 2004]

Several eigenvalues

4. Contour Integral (CISS) [Asakura et al. 2009]
5. Polynomial Interpolation [Effenberger/Kressner 2012]
6. Rational Interpolation (NLEIGS) [Güttel et al. 2014]

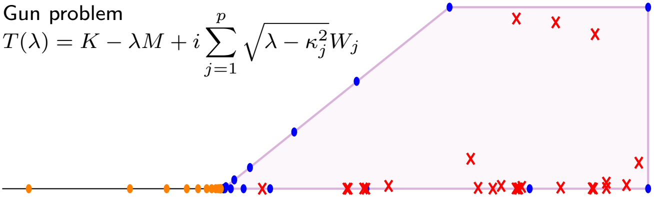
We have added deflation [Effenberger 2013] in the first three solvers

NEP Solver: NLEIGS

NEP: $T(\lambda)x = 0$ $\quad\Rightarrow\quad$ $$T(\lambda) \approx R_d(\lambda) = \sum_{j=0}^{d} b_j(\lambda) D_j$$

with $D_j = \frac{T(\sigma_j) - R_{j-1}(\sigma_j)}{b_j(\sigma_j)}$ and $b_j(\lambda) = \frac{1}{\beta_0} \prod_{k=1}^{j} \frac{\lambda - \sigma_{k-1}}{\beta_k (1 - \lambda/\xi_k)}$

Interpolation nodes and poles $\{(\sigma_i, \xi_i)\}$ are Leja-Bagby points from discretized Σ and Ξ. $\{\beta_i\}$ are scaling factors



Singularity points determined automatically (REP poles or w/AAA)

NLEIGS Details

Linearization: $L(\lambda)y = 0$ with $L(\lambda) = \mathcal{L}_0 - \lambda\mathcal{L}_1$,

$$\mathcal{L}_0 = \begin{bmatrix} D_0 & D_1 & \cdots & D_{d-1} \\ \sigma_0 I & \beta_1 I & & \\ & \ddots & \ddots & \\ & & \sigma_{d-2} I & \beta_{d-1} I \end{bmatrix} \quad \mathcal{L}_1 = \begin{bmatrix} 0 & & & \\ I & \frac{\beta_1}{\xi_1} I & & \\ & \ddots & \ddots & \\ & & I & \frac{\beta_{d-1}}{\xi_{d-1}} I \end{bmatrix} \quad y = \begin{bmatrix} b_0(\lambda)x \\ b_1(\lambda)x \\ \vdots \\ b_{d-1}(\lambda)x \end{bmatrix}$$

Compute an eigenpair (y, λ) of $L(\lambda)$, then extract x from y

---

SLEPc implementation:

- R_d may have many terms. If in split form $T(\lambda) = \sum A_i f_i(\lambda)$, divided differences D_j are not explicly computed
- Implicit block LU of $L(\sigma) \to$ LU factorizaton of $R_d(\sigma)$
- TOAR-like version, with Krylov-Schur restart and deflation

Nanophotonic Resonances

Joint work with:

- Carmen Campos
- Researchers at Institute Fresnel, France (A. Nicolet, G. Demésy, F. Zolla)
- Christophe Geuzaine (GetDP)

A. Nicolet, G. Demésy, F. Zolla, C. Campos, J. E. Roman, C. Geuzaine (2022), "Physically agnostic quasi normal mode expansion in time dispersive structures: From mechanical vibrations to nanophotonic resonances," European J. Mechanics - A/Solids. DOI: 10.1016/j.euromechsol.2022.104809

Highlights:

- Compute also left eigenvectors
- Use resolvent-based expansion for faster computation

Frequency-Dispersive Photonic Open Structures

Source-free Helmholtz equation in frequency-dispersive media

$$\boldsymbol{\varepsilon}_r(\mathbf{r},\omega)^{-1}\,\mathbf{curl}\,\left[\boldsymbol{\mu}_r^{-1}(\mathbf{r})\,\mathbf{curl}\,\mathbf{E}\right]=\frac{\omega^2}{c^2}\,\mathbf{E}$$

Relative permittivity is a rational function of the frequency

- Drude model:

$$\varepsilon_r(\omega)=\varepsilon_\infty-\frac{\omega_d^2}{\omega(\omega+i\gamma_d)}$$

  Resulting NEP: 1 rational term (1 pole) + cubic polynomial

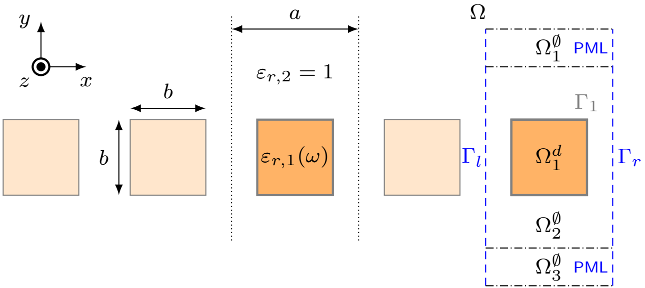
- Drude-Lorentz model:

$$\varepsilon_r(\omega):=\varepsilon_\infty+\sum_{j=0}^{N_p}\frac{f_j\omega_p^2}{\omega_j^2-\omega^2-i\omega\gamma_j}$$

  Example: in the case of gold, $N_p=5$ (12 poles)

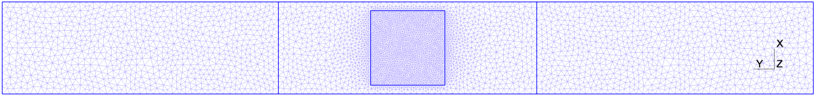
Computational Domain

1-D grating (periodic in x, invariant in z)



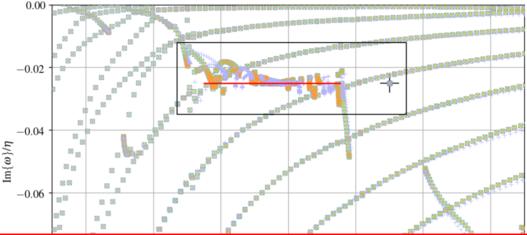
Perfectly Matched Layers used to truncate the infinite domain in y

Discretization mesh (33346 DOFs):

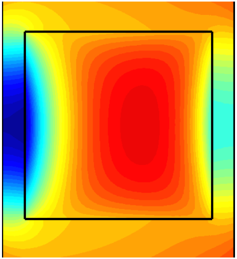


Solution

Computed eigenvalues



Mode profiles



Next objective:

Absorption spectrum: Losses as a function of source frequency

- Strategy 1 Loop over direct problems
- Strategy 2 Modal expansion, need also an expansion for adjoint problem (left eigenvectors)
- Strategy 3 Use explicit representation of resolvent

Left Eigenvectors and the Resolvent

$$T(\lambda)x = 0, \quad x \neq 0$$

$$y^* T(\lambda) = 0^*, \quad y \neq 0$$

- Compute left eigenvector y as right eigenvector of T^*
- Compute eigen-triples $\{(x_i, y_i, \lambda_i)\}_{i=1}^k$ with two-sided method

---

Keldysh's theorem provides an explicit formula for the resolvent

$$T^{-1}(z) = \sum_{i=1}^{k} (z - \lambda_i)^{-1} x_i y_i^* + R(z)$$

for $\lambda_i \in \Omega$, x_i, y_i normalized so that $y_i^* T'(\lambda_i) x_i = 1$

In SLEPc: `NEPApplyResolvent()` computes the action $T^{-1}(\lambda_s) u_j$

Summary of the Two Approaches

Physical approach

Explicit computation of all the integrals

- Solve one NEP to get the eigenpair $(\lambda_k, \mathbf{E}_{\mathsf{r}k})$
- Solve adjoint NEP to get the eigenpair $(\lambda_k, \mathbf{E}_{\mathsf{l}k})$
- Save all on disk

For each source $\mathbf{j}$:
→ Assemble $\mathbf{j}$
→ For each eigentriplet:
→ Compute

$$\int_\Omega \overline{\mathbf{E}_{\mathsf{l}k}} \cdot \mathbf{j}\, \mathrm{d}\Omega$$

→ Compute

$$\int_\Omega \overline{\mathbf{E}_{\mathsf{l}k}} \cdot \left(\mathsf{T}'_{\varepsilon,\mu}(\lambda_k)\mathbf{E}_{\mathsf{r}k}\right) \mathrm{d}\Omega$$

$$\sum_{k=1}^{N} \dots$$

Post-process relevant quantities

Expansion of the solution of a problem driven by a given source $\mathbf{j}$ at real frequency ω_s

- Pre-assemble all sources $\mathbf{j} \rightarrow \mathbf{u}_{\mathsf{j}}$
- Solve the NEP (full basis variant) and get the eigentriplet $(\lambda_k, \langle \mathbf{u}_{\mathsf{l}k}|, |\mathbf{u}_{\mathsf{r}k}\rangle)$

Compute the resolvent matrix directly:

$$\mathsf{T}^{-1}_{\varepsilon,\mu}(z) \approx \sum_{k=1}^{N} \frac{1}{(z-\lambda_k)} \frac{|\mathbf{u}_{\mathsf{r}k}\rangle \langle \mathbf{u}_{\mathsf{l}k}|}{\langle \mathbf{u}_{\mathsf{l}k}|\mathsf{T}'_{\varepsilon,\mu}(\lambda_k)\mathbf{u}_{\mathsf{r}k}\rangle}$$

For each source $\mathbf{u}_{\mathsf{j}}$:
→ Apply resolvent : $\mathsf{T}^{-1}_{\varepsilon,\mu}(z)\, \mathbf{u}_{\mathsf{j}}$

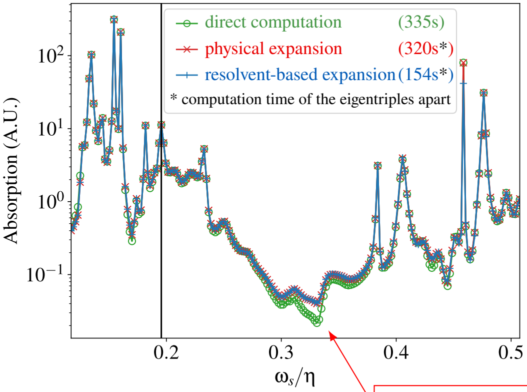
Post-process relevant quantities

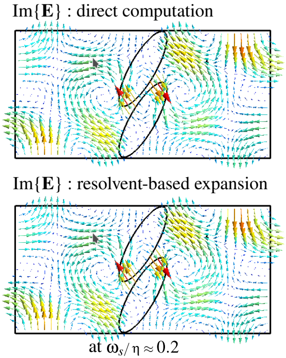
Matrix based

Physically agnostic approach

Validation

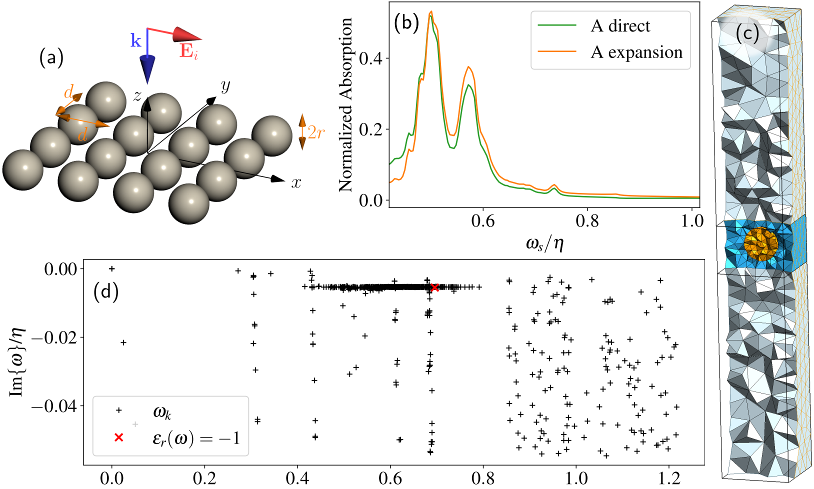
Absorption spectrum of two dispersive ellipses in a vacuum
(200 driving frequencies, 12000 dof's, 150 eigenvalues in 600s)





Small discrepancy due to accumulation point

Square Bi-Periodic 3D Structure



(120 driving frequencies, 40000 dof's, times: 686 / 5863 / 2470)

Aero-Acoustics Modal Analysis

Joint work with:

- Varun Hiremath, formerly at Siemens PLM

V. Hiremath, J. E. Roman (2022), "Acoustic modal analysis with heat release fluctuations using nonlinear eigensolvers," arXiv:2208.08717

Highlights:

- Custom preconditioner built from first-order approximation of problem matrices

Thermoacoustic Instabilities

Can cause problems (noise, vibrations, flame extinction, . . .)
$\rightarrow$ Goal: early identification and mitigation of instabilities

Approach: implement Helmholtz equation solver in multiphysics CFD software STAR-CCM+ using SLEPc for eigenmodes

Acoustic wave equation:

$$\frac{\partial^2 p'}{\partial t^2} - \nabla \cdot (\bar{c}^2 \nabla p') = (\gamma - 1)\frac{\partial \dot{q}'}{\partial t}$$

p': acoustic pressure
$\dot{q}'$: heat release fluctuation
$\bar{c} = \sqrt{\gamma \bar{p}/\bar{\rho}}$: mean speed-of-sound
γ: specific heat ratio

In frequency domain: $\omega^2 \hat{p} + \nabla \cdot (\bar{c}^2 \nabla \hat{p}) = S(\mathbf{x}) e^{\iota\omega\tau}$
$\rightarrow$ need a model to express $\hat{q}$ in terms of $\hat{p}$: $n - \tau$ model

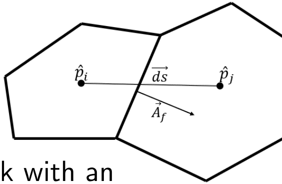
Algebraic Eigenvalue Problem

Discretization with the Finite Volume method results in

$$\left[A + \omega B + \omega^2 C\right] \hat{P} = D(\omega)\hat{P}$$

Depending on the simulation type (reacting/non-reacting) and boundary conditions, can be solved as linear, polynomial or NEP

A comes from the wave propagation term $\bar{c}^2 \nabla \hat{p}$
$\rightarrow$ need to compute gradient $\nabla \hat{p}$ at cell faces



- 2nd order approx.: accurate but needs to work with an implicit matrix for A (shell matrix)
- 1st order approx.: inaccurate but can build explicit matrix $\tilde{A}$

SLEPc

Split Preconditioner

When solving a polynomial eigenproblem $P(\lambda)x = 0$, the most expensive operation is $y = P(\sigma)^{-1}v$ for a given σ

1. Build matrix $M = P(\sigma) = A_0 + A_1\sigma + \cdots + A_d\sigma^d$
2. Default is LU factorization

`STSetSplitPreconditioner` allows passing matrices $\tilde{A}_0, \tilde{A}_1, \ldots, \tilde{A}_d$ to build the preconditioner from $\tilde{M} = \tilde{A}_0 + \tilde{A}_1\sigma + \cdots + \tilde{A}_d\sigma^d$ for any σ
$\longrightarrow$ `KSPSetOperators(ksp,` M `,` $\tilde{M}$`)`

Similarly for $T(\lambda)x = 0$, NLEIGS needs to solve $y = R_d(\sigma)^{-1}v$

- R_d built from linear combinations of $T(\sigma_j) = \sum_{i=0}^{\ell-1} A_i f_i(\sigma_j)$

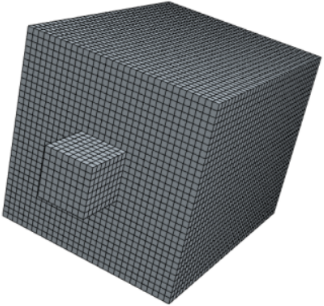
`NEPSetSplitPreconditioner` to pass $\tilde{A}_0, \tilde{A}_1, \ldots, \tilde{A}_d$ to build the preconditioner from $T(\sigma_j) = \sum_{i=0}^{\ell-1} \tilde{A}_i f_i(\sigma_j)$

Validation: Helmholtz Resonator

Classical Helmholtz resonator: a big cavity connected to a small open neck

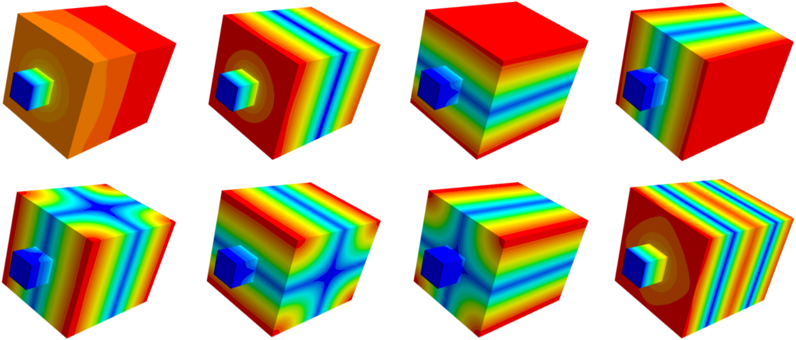
- Non-reacting case with no impedance boundary condition $\longrightarrow$ linear eigenproblem

	STAR-CCM+ using SLEPc	AVSP [Nicoud 2007]	Theoretical [Nicoud 2007]
Mode	Frequency, Hz	Frequency, Hz	Frequency, Hz
1	258	263	233
2	1772	1774	1781
3	2174	2176	2169
4	2174	2176	2169
5	2785	2787	2778
6	2785	2787	2778
7	3069	3069	3068
8	3479	3479	3483



All eigenvalues are real, zero growth rate (standing waves)

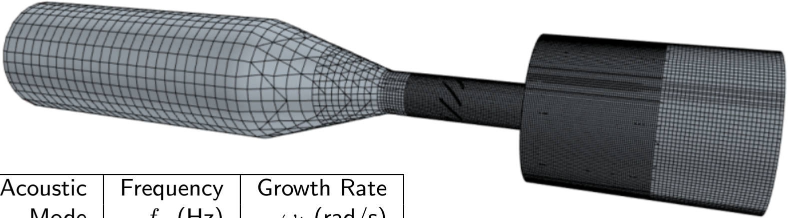
Validation: Helmholtz Resonator - Computed Modes



EM2C Burner

Turbulent swirled combustor designed and studied at EM2C Lab

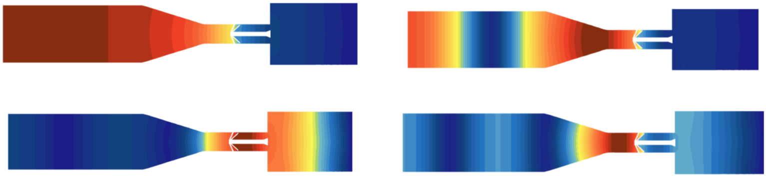
- variable length upstream manifold + axisymmetric duct + variable length cylindrical combustion chamber



Acoustic Mode	Frequency f_r (Hz)	Growth Rate ω_i (rad/s)
1	116.6	-37.73
2	907.5	15.31
3	1334.9	336.01
4	1648.0	-173.22

unstable modes

EM2C Burner - Computed Modes



With 500,000 cells, `NEPSolve` time $\approx$ 4 minutes

Concluding Remarks

Nonlinear eigensolvers in SLEPc

- Several eigensolvers, NLEIGS is generally good
- Left eigenvectors needed in some applications, e.g., resolvent
- Flexible way to pass preconditioner matrices

Further development

- Improve other solvers (contour integral)

Acknowledgement:

Grant PID2019-107379RB-I00 funded by:

GOBIERNO DE ESPAÑA

MINISTERIO DE CIENCIA E INNOVACIÓN

AGENCIA ESTATAL DE INVESTIGACIÓN