Chebfun – lab

September 24, 2019

1 Stabilizing a bicycle

Consider the paper [2], where the authors describe a model for the dynamics of a bicycle moving at a certain velocity v, and subject to external forces.

The model is given by a second order ODE

$$M\ddot{x} + C(v)\dot{x} + K(v) = f,$$

where as usual M is the mass matrix, C the damping-like term, and K the stiffness matrices. The matrices C and K depend on the velocity; recall that a linear system as the one above is stable if and only if the eigenvalues of the quadratic eigenvalue problem

$$\det(M\lambda^2 + C\lambda + K) = 0$$

are all contained in the left half plane, that is, they have negative real part. Look at the model described in [2], and try to compute the eigenvalues using `polyeig` in MATLAB — the model is very simple, these are 2×2 matrices. Then, use Chebfun to determine for which values of the velocity v the system is stable; compare your findings with the ones of the authors.

Possible hints:

- The eigenvalues depend analytically on the entries of the matrices, except at at a few exceptional points; Chebfun has a special options to automatically split the domain into parts where the function is smooth, which you can enable by calling: `chebfun(..., 'splitting', 'on')`.
- The paper is the file `bicycle.pdf`.

2 Finding the material characteristics of a portal

Download the file `portal.mat`; this contains a vector called `frequencies`, and two cell arrays `K` and `M`. These describe the parametric mass and stiffness matrices of an undamped portal, of which the mass density and the Young's modulus of one of the two pillars are unknown (this a simplified model taken from [1])

If we call ρ and E the mass density and the Young's modulus of this material, then the stiffness and mass matrices are given by

$$K(\rho, E) = K_1 + EK_1 + \rho K_2, \qquad M(\rho, E) = M_1 + EM_1 + \rho M_2.$$

The vector `frequencies` contains the first 5 natural frequencies of the structure, given as $f_j = \sqrt{\lambda_j}/(2\pi)$, where λ_j are the eigenvalues of the pencil $K - \lambda M$. Consider the objective function

$$\Phi(E, \rho) = \|\mathbf{f}(E, \rho) - \hat{\mathbf{f}}\|_2^2,$$

where $\mathbf{f}(E, \rho)$ are the frequencies computed at some value of the parameters, and $\hat{\mathbf{f}}$ the reference ones. It is known that the parameters lie in the box $10^9 \leq E \leq 10^10$, and $10^3 \leq \rho \leq 10^4$. Use Chebfun2 to construct a model of the objective function, and find its minimum.

Hints:

- You can use `eigs` in MATLAB to solve the eigenvalue problem, and only compute the first 5 eigenvalues. For this problem, the smallest eigenvalues are of interest!
- As you will sound find out, solving a large scale eigenvalue problem at every Chebyshev points takes a considerable amount of time; therefore, we should come up with some decent model reduction idea to reduce the size of the problem before feeding it into chebfun. One possibility is to compute a few eigenvectors of the smallest eigenvalues of the problem for a few values of the parameters x_j, y_j, put them together in a matrix side by side; then take an SVD (economy-size!), and construct a basis of the column space by dropping singular values relatively smaller than, say, 10^{-3}. Then, project everything setting $\tilde{K}_j = U^T K_j U$ and $\tilde{M}_j := U^T M_j U$, and use this reduced model in Chebfun.
- Once you find the first minimum, you may want to construct a second chebfun object on a much smaller domain around the minimum, and try to refine the approximation.

3 Transient behavior of an ODE

Load the matrix A from the transient MAT file. This defined an ODE

$$\begin{cases} \dot{x} = Ax, \\ x(0) = x_0 \end{cases},$$

for different values of x_0. We are interested in x_0 of Euclidean norm 1, obtained by combining and normalizing the vectors e_6 and e_7.

- Verify that this matrix is stable (the real part of its eigenvalues is strictly negative), and therefore $\lim_{t\to\infty} \|x(t)\|_2 = 0$.

- Use chebfun to verify what the maximum transient norm is. That is, compute
$$\max_{\substack{x_0=\alpha e_6+\beta e_7\\ \|x_0\|_2=1}} \max_{t\geq 0} \|e^{tA}x_0\|_2.$$

- Hint: Since $\lim_{t\to\infty}\|x(t)\|_2 = 0$, it is sufficient to consider a finite interval for the time.

- Concerning the previous points, explicitly give the starting vector x_0 (inside the considered set) and the time t, at which the maximum possible norm is reached.

- Find the starting point x_0, inside the considered set, which minimizes the norm in the transient state. That is, compute
$$\min_{\substack{x_0=\alpha e_6+\beta e_7\\ \|x_0\|_2=1}} \max_{t\geq 0} \|e^{tA}x_0\|_2.$$

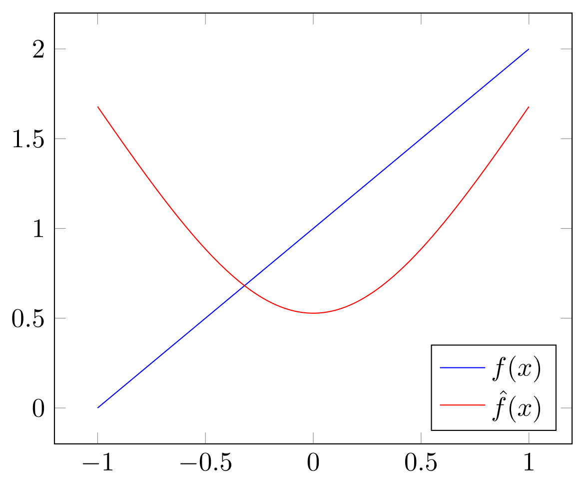
The matrix reported has been considered in [3].

4 Fast multiplication by a kernel function

Consider the following operation. Given $f(x)$, we want to compute the integral transform
$$\hat{f}(x) := \int_{-1}^{1} \log\left(1 + x^2 + y^2\right) f(y)\, dy.$$

- Use chebfun2 to implement this transform; hint: define the bivariate integral and integrate out one variable using the `sum` command.

- Try your implementation on a few test functions. For instance, for $f(x) = x + 1$ you should get the following result:

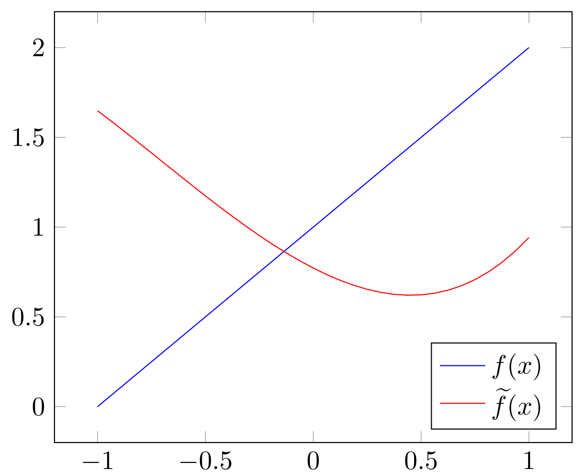


- Modify your procedure to compute the integral transform

$$\widetilde{f}(x) := \int_{-1}^{1} \log\left(1 + |x - y|\right) f(y)\ dy.$$

You will see that this cannot be extended so easily. Why is that?

- Find a way around the problem, and test it on some function. For instance, the plot above should now look as follows:



- Hint for developing the scheme:
  - You can split the kernel $\kappa(x, y)$ as

$$\kappa(x, y) = \kappa_{11}(x, y) + \kappa_{12}(x, y) + \kappa_{21}(x, y) + \kappa_{22}(x, y),$$

    where each splitting has support in one of the four parts of the domain $[-1, 1]^2$ obtained by splitting both x and y in two equal parts. On two of these domains, $\kappa(x, y)$ is smooth, so the previous approach works with no problems, on the other, we may call our procedure recursively.
  - When the domain gets small enough (say, the width is smaller than $1/10$), use some crude approximation of the integral. Here $x \approx y$, and therefore the contribution will be close to 0.

- The above scheme will not be super-effective, as it is — but it should at least complete in a bunch of seconds.

- A clever observation of self-similarities inside the above decomposition might lead to a more efficient technique!

References

[1] Maria Girardi, Cristina Padovani, Daniele Pellegrini, Margherita Porcelli, and Leonardo Robol. Finite element model updating for structural applications. *arXiv preprint arXiv:1801.09122*, 2018.

[2] Jaap P Meijaard, Jim M Papadopoulos, Andy Ruina, and Arend L Schwab. Linearized dynamics equations for the balance and steer of a bicycle: a benchmark and review. *Proceedings of the Royal society A: mathematical, physical and engineering sciences*, 463(2084):1955–1982, 2007.

[3] Elmar Plischke and Fabian Wirth. Stabilization of linear systems with prescribed transient bounds. In *Proceedings of the 16th International Symposium on Mathematical Theory of Networks and Systems (MTNS2004), Leuven, Belgium*, 2004.