COMPUTATIONAL MECHANICS 2015
31st conference with international participation
Špičák
November 9 - 11, 2015

The root-finding of dispersion curves in a bar impact problem.

Petr Hora[a]

[a] Institute of Thermomechanics, Czech Academy of Sciences, Veleslavínova 11, 301 14 Plzeň, Czech Republic

To calculate the stress wave propagation in a bar impact problem it is used the integration along the dispersion curves. This dispersion relations $f(x, \gamma a)$ is defined as

$$\left(2-x^2\right)^2 \mathrm{J}_0\left(\gamma a A\right) \mathrm{J}_1\left(\gamma a B\right)+4AB\mathrm{J}_1\left(\gamma a A\right) \mathrm{J}_0\left(\gamma a B\right)-\frac{2x^2}{\gamma a}A\mathrm{J}_1\left(\gamma a A\right) \mathrm{J}_1\left(\gamma a B\right)=0,$$

where a is radius of the semi-infinite bar, γ is wavenumber, x is the ratio of the phase velocity and the shear wave velocity, κ means the ratio of the squares of the phase velocities for the bar's material, $A=\sqrt{\kappa x^2-1}$, $B=\sqrt{x^2-1}$ and J is the Bessel function of the first kind.

Summary of the Chebyshev expansion algorithm [1]:

1. Choose the following:

   (i) γa

   (ii) Search interval, $x \in [a, b]$.
   The search interval must be chosen by physical and mathematical analysis of the individual problem. The choice of the search interval $[a, b]$ depends on the user's knowledge of the physics of his/her problem, and no general rules are possible.

   (iii) The number of grid points, N.
   N may be chosen by setting $N = 1 + 2^m$ and the increasing N until the Chebyshev series displays satisfactory convergence. To determine when N is sufficiently high, we can examine the Chebyshev coefficients a_j, which decrease exponentially fast with j.

2. Compute a Chebyshev series, including terms up to and including T_N, on the interval $x \in [a, b]$.

   (i) Create the interpolation points (Lobatto grid):

   $$x_k \equiv \frac{b-a}{2}\cos\left(\pi\frac{k}{N}\right)+\frac{b+a}{2}, \quad k=0,1,2,\ldots,N.$$

   (ii) Compute the elements of the $(N+1)\times(N+1)$ interpolation matrix.
   Define $p_j = 2$ if $j = 0$ or $j = N$ and $p_j = 1, j \in [1, N-1]$. Then the elements of the interpolation matrix are

   $$I_{jk}=\frac{2}{p_j p_k N}\cos\left(j\pi\frac{k}{N}\right).$$

(iii) Compute the grid-point values of $f(x)$, the function to be approximated:

$$f_k \equiv f(x_k), \quad k = 0, 1, \ldots, N.$$

(iv) Compute the coefficients through a vector-matrix multiply:

$$a_j = \sum_{k=0}^{N} I_{jk} f_k, \quad j = 0, 1, 2, \ldots..., N.$$

The approximation is

$$f_N \approx \sum_{j=0}^{N} a_j T_j \left(\frac{2x - (b+a)}{b-a} \right) = \sum_{j=0}^{N} a_j \cos \left\{ j \arccos \left(\frac{2x - (b+a)}{b-a} \right) \right\}.$$

3. Compute the roots of f_N as eigenvalues of the ChebyshevFrobenius matrix

   Frobenius showed that the roots of a polynomial in monomial form are also the eigenvalues of the matrix which is now called the *Frobenius companion matrix*. Day and Romero [2] developed a general formalism for deriving the *Frobenius matrix* for any set of orthogonal polynomials.

4. Refine the roots by a Newton iteration with $f(x)$ itself.

   Once a good approximation to a root is known, it is common to *polish* the root to close to machine precision by one or two Newton iterations.

Computations were performed with the normalized Bessel functions that eliminate the large fluctuations in magnitude. For numerical experiments were used MATLAB's toolbox CHEBFUN [3] and Julia's package ApproxFun [4].

Acknowledgements
The work was supported by the institutional support RVO:61388998.

References

[1] Boyd, J.P., A Chebyshev Polynomial Interval-Searching Method (”Lanczos Economization”) for Solving a Nonlinear Equation with Application to the Nonlinear Eigenvalue Problem, Journal of Computational Physics 118 (1995) 1–8.

[2] Day, D.M., Romero, L., Roots of polynomials expressed in terms of orthogonal polynomials, SIAM J. Numer. Anal. 43 (2005) 1969–1987.

[3] Driscoll, T.A., Hale, N., Trefethen, L.N., editors, Chebfun Guide, Pafnuty Publications, Oxford, 2014.

[4] URL: github.com/ApproxFun/ApproxFun.jl.git