`MPSpack` user manual

Alex Barnett[*]and Timo Betcke[†]

code revision 16: July 20, 2008

Abstract

`MPSpack` is a fully object-oriented MATLAB toolbox that implements the Method of Particular Solutions and related methods (*e.g.* fundamental solutions) for efficient solution of Laplace eigenvalue, interior/exterior Helmholtz problems (*e.g.* wave scattering), and related problems, on piecewise-constant 2D domains. It is designed to be simple to use, and to enable highly-accurate solutions. This is the user manual.

1 Installation

Requirements:

- MATLAB version 7.6 (2008a) or newer.

We make use of the object-oriented programming ability of MATLAB which unfortunately means the toolbox cannot work with versions earlier than 7.6.

The project is hosted at `http://code.google.com/p/mpspack`

Anonymous check out of the package is via the subversion command:

```
svn checkout http://mpspack.googlecode.com/svn/trunk/ mpspack-read-only
```

This creates a directory `mpspack-read-only` in which you will find the `mpspack` subdirectory.

Setting up MATLAB path...

[*]Department of Mathematics, Dartmouth College, Hanover, NH, 03755, USA

[†]School of Mathematics, The University of Manchester, Manchester, M13 9PL, UK

2 Overview

`MPSpack` is a MATLAB toolbox to solve Helmholtz boundary-value problems (BVP) and eigenvalue problems (EVP) with particular and fundamental solution methods.

Define BVP, EVP

2.1 Boundary-value problems

Let $\Omega \subset \mathbb{R}^2$

$$\begin{aligned} (\Delta + k^2)u &= 0 \qquad \text{in } \Omega & (1) \\ au + bu_n &= f \qquad \text{on } \partial\Omega & (2) \end{aligned}$$

In our implementation we restrict to functions a, b that are constant on segments used to define the boundary $\partial\Omega$.

If Ω is an exterior domain, we may wish to impose additional boundary conditions at infinity, such as Sommerfeld's radiation condition. This may be achieved by choosing basis sets satisfying this condition.

See [1] for an example of using the Method of Fundamental Solutions to solve an interior Helmholtz problem.

To solve a BVP the usual flow is as follows:

1. define piecewise-analytic segments forming all boundaries
2. define domains using various of these segments as their boundaries
3. choose MPS basis set(s) within each domain
4. set up inhomogeneous boundary or matching conditions on each segment
5. solve a linear system to get the basis coefficient vector
6. evaluate solution on desired points
7. plot solution or compute error estimates

...

2.2 Eigenvalue problems

2.3 Geometry

...

When domains are built from segments, each segment stores information about which domain, if any, it is connected to on each side. If segments are to be reused to construct new domains, while erasing any old domains, the current way to do this is via the `segment.disconnect(s)` command, where `s` is an array of segment handles. This clears these segments of any connections with domains.

...

2.4 Boundary conditions

A segment Γ may contain a boundary condition on only one of its sides, of the form

$$au(s) + bu_n(s) = f(s) \qquad s \in \Gamma, \quad a, b \in \mathbb{C}$$

or, alternatively matching conditions of the form

$$\begin{aligned} a^+u^+(s) + a^-u^-(s) &= f(s) \qquad (3) \\ b^+u_n^+(s) + b^-u_n^-(s) &= g(s) \qquad (4) \end{aligned}$$

where $a^+, a^-, b^+, b^- \in \mathbb{C}$. In the above f and g are functions on the boundary. In our implementation the user may supply these as handles to functions of segment parameter $t \in [0, 1]$, or as a data column vector on the segment quadrature points.

3 In-depth examples

4 Test routines

The code `testbvp.m` shows the main steps for solution of a BVP on a variety of domains.

5 Limitations

- Checking of whether inside a domain is approximate, based on approximating polygons.

6 List of improvements to make

List to be prioritized:

- make `segment.bdryfunc` which creates u, u_n data given a field function and its gradient field function, useful for choosing BC data corresponding to exact fields
- make `domain.setbc` which uses one BC data function on all segments
- eigenvalue problems: MPS, scaling method
- segment methods to create analytic interpolant function from boundary point data, enabling user to specify a segment using points on a curve.
- add better automated ways to choose MFS charge points, based on JCP paper
- add layer-potential bases, and associated periodic quadrature methods
- write `segment.bdrysolution` which evaluates u, u_n on one or other side of a boundary. For layer potentials this would take into account jump relations. This should then be used in `problem.fillbcmatrix`
- keep discrepancy and evaluation matrices, for efficiency in multiple right-hand sides.
- classes which symmetrize basis sets for single reflection, C_4, etc symmetry, by wrapping the calls to basis evaluations using reflection points
- dielectric constants, overall frequency, automatically changing k in domains
- periodic boundary conditions connecting values and derivatives on given sides of two different segments, needed for photonic crystals

7 Known bugs

Please alert the authors to any bugs you discover, including a description of how to reproduce the behavior, using the interface at

`http://code.google.com/p/mpspack/issues/list`

References

[1] A. H. Barnett and T. Betcke, *Stability and convergence of the Method of Fundamental Solutions for Helmholtz problems on analytic domains*, J. Comput. Phys., *in press* (2008). `arXiv:0708.3533 [math.NA]`.