Boundary Integral Solver Algorithm Summary

February 16, 2017

Surface definition

We start with the description of the overlapping-chart surface abstraction used in the code, which is based on (Ying, Biros, Zorin, 2006); however, we make some slight changes in notation, to be more consistent with the code.

The surface is viewed as an abstract domain M. The overlapping charts domains U_k are equipped with bijective maps $\xi_k : P_k \to U_k$, where P_k is a subset of M; U_k cover all of M, and may overlap.

For each chart, there is a geometry map $g^k : U_k \to \mathbf{R}^3$, and a partition-of-unity blending function $w_k : U_k \to \mathbf{R}$. Defining $\tilde{w}_k(s) = w_k(\xi_k^{-1}(s))$ for any point $s \in P_k$, and zero for other points of M, we require $\sum_k \tilde{w}_k(s) = 1$, for all points $s \in M$.

In the code, there are two realizations of this abstraction. Both require some way of identifying points on M.

In the specific case of manifold-based surfaces of (Ying and Zorin 2004), the domain M corresponds to an abstract (i.e., with no 3d embedding specified) quad mesh, i.e. a collection of unit squares (faces) with a coordinate system (c, d) specified on each square with an arbitrary vertex of the square chosen as the origin, with coordinates along edges. The edges of the squares are identified, so that each edge is shared by exactly two squares. Additionally, faces sharing a vertex form a loop such that each two squares in a sequence share an edge.

In this case, the points of M are identified by a pair (F, q), where $q \in [0, 1] \times [0, 1]$, and F is the index of the face. To maintain uniqueness, the edges and vertices are arbitrarily associated with one of the faces they belong to (this is never explicitly used int the code, as all surface samples are in the interior of faces).

A patch $P_k \subset M$, consists of all quads F_i sharing a vertex V_k. The map ξ_k and its inverse are analytically defined on each face of P_k, as $\xi_k(z) = z^{4/K}|z^{1-4/K}|$, where K is the vertex valence, and the coordinate system (c, d) on F_i is chosen with origin at V_k, with $z = c + id$. This is a conformal map rescaled in the radial direction to reduce distortion. $\xi_k(P_k)$ is a curved star-shaped domain in the plane; the coordinates in the plane are denoted (x, y) in the code.

The correspondence between the notation described here and the code is ξ_k is `face_point_to_xy`, ξ_k^{-1} is `xy_to_face_point`, g^k is `xy_to_patch_coords` (also can compute derivatives), w_k is `xy_to_patch_value` (the last two terms should be changed – these are misleading).

Notation

- d is the dimension of the density in the problem, 1 or 3 (vector density).
- δ is the target sampling step in 3d; per-patch parametric domain sampling step h_k is computed from δ;
- R is the refinement factor; for the points in Ω_1, refined sampling is used used with 3d spacing δ/R, $R > 1$.
- ρ is the radius, in the number of patch-depedent parametric domain step sizes, of the floating partition of unity η used in singular evaluation.
- L is the number of sample points used for near evaluation interpolation.
- A star-shaped patch U_k has zero-centered bounding square D_k of size $2b_k \times 2b_k$.
- *sample points* $s_{k,n}$ are associated with patches and are picked from a regular grid on D_k; $s_{k,n} = \xi_k^{-1}(-b_k + ih_k, -b_k + jh_k)$, for some integers (i, j). Points of $D_k \setminus U_k$ are skipped.

- *collocation points* $c_{k,m}$ are on-the-surface points at which the integrals are evaluated (may or may not be sample points: these are sample points in the solve, but can be any for solution evaluation); for a patch P_k, the list of collocation points includes both points sampled on D_k, as well as all points sampled on chart bounding boxes D_ℓ, for which $P_k \cap P_\ell \neq \emptyset$.
- Each patch U_k is associated with a connected component of the boundary, $m(k)$, which has an interior point z_m.
- $\mathcal{S}, \mathcal{D}, \mathcal{R}$ are single-layer, double-layer, and rotlet-type kernel for a given equation, the latter beyinf defined only for
- $r^\times$ denotes the linear operator of taking cross product with r.
- ϕ denotes density values;
- u denotes potentials in evaluation (in solve, the output is also a density).

Algorithms

Setup.

{**Sampling setup** }
{ `DN3dOv::setup()`}
for all patches P_k **do**
Estimated per-patch scale $\sigma_k := max(|g^k_x(0,0)|, |g^k_y(0,0)|)$.
Grid resolution $n_k := \lfloor 2b_k/(\delta/\sigma_k) \rfloor$.
Parametric domain step $h_k := 2b_k/n_k$;
Create the set of sample points $s_{k,i}$ on P_k, $i = 1 \dots N_k$, which are images of the subset of the $n_k \times n_k$ grid points on D_k inside U_k.
{Each point also has a global index n, and denoted s_n when a global index is used.}
for all $s_{k,i}$, $i = 1 \dots N_k$ **do**
Compute:
$p_{k,i} := g_k(s_{k,i})$, 3d position;
$n_{k,i}$, unit normal to the surface, pointing in the direction of the domain;
$w_{k,i}$, value of the POU function, supported on U_k;
$J_{k,i}$, determinant of the Jacobian of g_k;
$q_{k,i} := h_k^2$, integration weight of the point (periodic trapezoidal);
$W_{k,i} := w_{k,i} J_{k,i} q_{k_i}$, combined weight for summation.
end for
end for

{**Collocation setup**}
{`DN3dOv::distribute_collocation_points`}
Input: list of target surface samples $t_{k,i}$, at which the integrals are evaluated (not necessarily $s_{k,i}$)
Output: collocation points $c_{k,m}$
for all patches P_k **do**
for all patches P_ℓ overlapping P_k **do**
Add all sample points $s_{\ell,i}$ in $P_\ell \cap P_k$ to the list $c_{k,j}$.
$p^c_{k,j} := g^k(c_{k,j})$;
Define $S(k,j) := (\ell, i)$, the map from collocation point on patch k to its original sample point on patch ℓ.
end for
end for

{**Solver setup** }
{ `Bis3dOvGeneric::setup()`}
Call sampling setup for spacings δ and δ/R, to get samples $s_{k,i}$, and $s^r_{k,i}$. {The refined sampling is needed only for near evaluation, not described yet}
Call collocation setup for $t_{k,i}$ set to $s_{k,i}$ {This is called in`BE3dRon::setup()`}
For bounded-domain problems, M is the number of interior connected components of the boundary;
For unbounded-domain problems, M is the total number of boundary components.
if $d = 1$ **then**
Define $N \times M$ matrix B, $B_{nm} = \mathcal{S}(z_m, s_n)$, corresponding to the pole terms in the equation for mutiply-connected bounded domains and unbounded domains.
Define $N \times M$ constraint matrix C, with $C_{nm} = q_n$, the quadrature weight at n-th point.
else
Define $N \times 2M$ block matrix B, with 3×3 blocks $B_{mn} = \mathcal{S}(z_m, s_n)$, for $m \leq M$, and $B_{nm} = \mathcal{R}(z_m, s_n)$, $m > M$.
Define $N \times 2M$ constraint block matrix C with 3×3 blocks $C_{nm} = q_n I$, for $m \leq M$, and $C_{nm} = q_n(p_n - z_m)^\times$.
end if
Compute $P := (C^T B)^{-1}$ (Schur complement for preconditioning).

Next, we describe the matvec for the GMRES solve, assuming the setup above. The main component is evaluation of the singular integral $\int \mathcal{D}(x, y)\phi(y)dy$ over the surface.

Solve. The integral equation solved depends on the domain (see paper). The input is the righ-hand side b, defined at the sample points $s_{k,i}$, which is passed to the KSP solver. The complete vector we solve for is $x = [\phi, \gamma]$, where ϕ is the double-layer density defined at $s_{k,i}$, and $\gamma = \alpha$ for $d = 1$, and $\gamma = [\alpha, \beta]$ for $d = 3$. and α and β are vectors of pole charges defined at z_m, ("translational" and "rotational" in 3d), of size dM.

{**Singular integral evaluation**}
Input: ϕ at $s_{k,i}$, collocation points $c_{k,j}$. Output: potential u at target samples $t_{k,i}$.
{`BE3dOvRon::singular_evaluation`}
Set weighted density $\psi_n := W_n \phi_n$;
Call FMM to evaluate $u_n = \sum_m \mathcal{K}(p_m, p_n)\psi_m$ for all sample points s_n. {subtract the part of the integral FMM computed inaccurately}
for all patch P_k **do**
for all collocation points $c_{k,j}$ **do**
Find all sample points $s_{k,i}$, such that $|\xi_k(s_{k,i}) - \xi_k(c_{k,j})| < h_k\rho$
Here the collocation-to-samples map is used to update correct entries in density
$u_{S(k,j)} := u_{S(k,j)} - \sum_i \mathcal{K}(p^c_{k,j}, p_{k,i})W_{k,i}\phi_{k,i}\eta(s_{k,i})$.
end for
end for
{Add an accurate calculation of the same part using polar samples}
for all patch U_k **do**
for all collocation point $c_{k,m}$ **do**
Compute a radial sampling pattern $t_{k,m,i}$ in U_k, centered at $\xi_k(c_{k,m})$, radius $h_k\rho$, with radial number of samples $[\rho]$, and angular $2[\rho]$.
Interpolate weights, normals and densities to $t_{k,m,i}$ in U_k.
$u_{S(k,m)} := u_{S(k,m)} + \sum_i \mathcal{K}(p^c_{k,m}, g^k(t_{k,q}))W_{k,m,i}\phi_{k,m,i}\eta(t_{k,m,i})$.
end for
end for

For complete evaluation, the contribution from poles need to be added (using B), and constraint equation part of the matrix applied (using C). Additionally, for incompressible equations we add an extra summation with normal-derived kernel.

{**Full matvec for the Krylov solver iteration.**}
{`Bis3dOvGeneric::mmult()`}
Input: $x = [\phi, \gamma]$, Output: $y = [\phi^y, \gamma^y]$.
Compute $\phi^y := \frac{1}{2}\phi + \int \mathcal{D}(x,y)\phi(y)dy$ using singular evaluation above.
if incompressibility constraint is present **then**
  $\phi^y_m := \phi^y_m + W_m(n_m\dot{\phi}_m)n_m$
end if
if $M > 0$ **then**
  $\phi^y := \phi^y + B\gamma$; $\gamma^y := C^T\phi$.
end if

The final component of the solve is the preconditioner application, which is used only if $M > 0$.

{**Preconditioner application.**}
{`Bis3dOvGeneric::pcmult()`}
{`Supposed to agree with Greengard, Kropinski and Mayo`}
$\gamma^y := -\frac{1}{2}P(\gamma - 2C^T\phi)$
$\phi^y := 2\phi + B\gamma^y$

Evaluation. Given the extended density $[\phi, \gamma]$, evaluate the solution at arbitrary points in the domain, with contribution from γ added directly.

The density integral for far points (Ω_2 domain, greater than $C_1\delta$ from the boundary), FMM is used.

For on-the-surface samples $t_{k,i}$, compute collocation points $c_{k,m}$ and call singular integral evaluation above.

The remaining evaluation type is near evaluation

{**Near evaluation**}
Input: list of target 3d positions p^t_m closer than $C\delta$ to the surface, list of surface samples s^t_m with 3d positions $p^{t,c}_m$ closest to p^t_m, in-out flags for each target point; surface density ϕ_n at suface samples s_n.
Output: values u^t_m of the integral of ϕ at p^t_m.
{ `BE3dOv:setup()` and `eval()`}
Separate p^t into $p^{t,0}$, points closer than δ to the surface (Ω_0), and $p^{t,1}$ being the rest(Ω_1).
Generate a set of new target points $p^{t,0}_{m,q}$ for each $p^{t,0}_m$, $p^{t,0}_{m,q} := p^{t,c}_m + qn^{t,c}_m\delta$, $q = 1 \dots L-1$.
The union of $p^{t,1}$ and $p^{t,0}_{m,q}$ is $p^{t,int}$
Create collocation points $c^r_{k,n}$ for refined samples $s^r_{k,n}$
{these were initialized in the solver setup – should probably be moved here}
Create collocation points $c^t_{k,n}$ for $p^{t,c}_m$.
{Interpolate ϕ to finer samples; this is done first by patch to all collocation points in the patch, then for each sample point values obtained on different patches are averaged with blending function as weights}
Refine $\phi_{k,i}$ to obtain values $\phi^r_{k,i}$ at $s^r_{k,i}$.
for all P_k **do**
  for all $c^r_{k,j}$ **do**
    Interpolate $w_{k,i}\phi_{k,i}$ to samples $\xi_k(c^r_{k,j})$ to obtain values $\psi^{r,c}_{k,j}$
  end for
end for
for all $s^r_{k,j}$ **do**
  $\phi^r_{k,j} := \sum_{(\ell,m):S(\ell,m)=(k,j)} \psi^{r,c}_{\ell,m}$ {This is done through scatter}
end for
Set weighted density $\psi^r_n := W^r_n\phi^r_n$;
Call FMM to evaluate $u^{t,int}_\ell := \sum_m \mathcal{K}(p^r_m, p^{t,int}_\ell)\psi^r_m$ for all targets in Ω_1
{The above yields $u^{t,1}_m$.}
$u^c_m :=$ singular evaluation with ϕ as input at s^t_m.
$u^{jmp}_m :=$ jump evaluation with ϕ as input at s^t_m.
for all target points $p^{t,0}_m$ **do**
  $u^{surf}_m := \pm\frac{1}{2}u^{jmp}_m + u^c_m$ {the sign is set based on in-out flag for the point}
  interpolate u^{surf}_m and $u^{t,0}_{m,q}$, $q = 1 \dots L-1$ using 1d Lagrangian interpolation to $p^{t,0}_m$, to obtain $u^{t,0}_m$.
end for

TODO: jump evaluation, singularity cancellation in on-the-surface evaluation, parallel aspects.