Boundary integral methods for implicitly defined dynamic interfaces

Celebrating Prof. Yoshikazu Giga’s 60th Birthday

Richard Tsai

with
C. Chen, C. Kublik, Y. Wu

KTH Royal Institute of Technology, Sweden
and
The University of Texas at Austin, USA
Among the important things

- The existence of a second stomach for sweets
- The importance of “unagi” for survival of summer
- The completion of a meal by soba-yu
The L-solutions of Giga-Sato

Embed the solution graph as a level curve of a higher dimensional function, and look at convergence in Hausdorff distance of the subgraphs:

\[ u_t + H(t, x, u, u_x) = 0 \iff \phi_t - \phi_y H(t, x, y, -\frac{\phi_x}{\phi_y}) = 0. \]
Multivalued solutions

Non-overturning in level sets:

- $\phi(y)$ do not change sign for all time.

Then interpret \( \{ \phi(x, y, t) = 0 \} \) as the graph of the solution $u(x, t)$. 

Non-overturning conditions:

$\nabla \phi \cdot \nabla \phi + v(y) = 0$.

OR a more relaxed condition:

$V(y) = R_y v ds$ is concave!
Very singular diffusion of Giga

\[ \phi_t + \tilde{H}(t, x, y, \phi_x, \phi_y) = M |\nabla \phi| \frac{\partial}{\partial y} \left( \frac{\phi_y}{|\phi_y|} \right) \]

Numerical solution verifies the entropy condition. (Equal area rule).
Bunching and anisotropy
Multivalued solutions for high frequency waves

\[ \partial_t S + \frac{1}{2} |\nabla_x S|^2 + V(x) = 0 \]

\[ \nabla_x S_0(x) = (u_j^{(0)}(x)) \]

\[ \phi_j(x, p, 0) = p_j - u_j^{(0)}(x), j = 1, \cdots, d \]

\[ f(x, p, t) := \rho(x, p, t)|\nabla_p \phi| \]

\[ \tilde{\rho}(x, t) = \int \rho(x, p, t)\delta(\Phi)|\Phi_p|dp \]

\[ = \int f(x, p)\delta(\Phi)dp \]

[In the paper by Jin, Liu, and Osher-T:2005]
Geometric motion of an interface

Mullins-Sekerka dynamics: normal velocity of a moving interface

\[ v_n = \left[ \frac{\partial u}{\partial n} \right] \]

\[ \Delta u = 0 \quad x \notin \partial \Omega(t) \]

\[ u = \kappa, \quad x \in \partial \Omega(t) \]

+ far field condition
Solution of linear PDEs by boundary integrals

\[ \Delta u = 0, \quad x \in \Omega \]
\[ u(x) = f(x), \quad x \in \partial \Omega \]

- Solving for \( \gamma \):
  \[ f(x) = \lambda \gamma(y) + \int_{\partial \Omega} K(x, y)\gamma(y) dS_y \]

- Evaluating the solution:
  \[ u(x) = \int_{\partial \Omega} \tilde{K}(x, y)\gamma(y) dS_y \]
Breaking of convexity
Volume preservation
Merging
Nontrivial surface area-volume relation

Isothermal at infinity.
Closed simple curve $\Gamma: \gamma(s) = (X(s), Y(s))$, parametrized by arclength $s$. 

\[ l = \int f \circ \gamma(s) ds = \int_{\mathbb{R}^2} f(x,y) \delta(\Gamma, x, y) dx dy. \]

- **Explicit curves:** [Peskin, Lai]
  \[ \delta(\Gamma, x, y) = \int \delta(x - X(s))\delta(y - Y(s)) ds. \]

- **Implicit curves:**
  \[ \delta(\Gamma, x, y) = \delta(\phi(x, y))|\nabla \phi(x, y)|. \]

- **Surface integral:**
  \[ \int_{\mathbb{R}^d} f(x) \delta(\phi) |\nabla \phi| dx. \]
$$E_{\varepsilon,h} = \left| \sum_{i,j} \delta_{\varepsilon}(\Gamma, x_{i,j}) f(x_{i,j}) h^2 - \int_{\Gamma} f(\gamma(s)) ds \right|$$

Typically $E_{\varepsilon,h} \longrightarrow 0$ if $\varepsilon = h^\alpha$, $0 \leq \alpha < 1$. (wide support!)

Non-convergence if e.g. $\delta_{Ch}^{\cos}$
Relative error \( E = \frac{|S_h - S|}{S} \). Radii: \( r_x > r_\odot > r_\oplus \).
Theorem

\[ \sum_{i,j \in I_{p,q}} \delta_{\varepsilon}^L (d_{\Gamma} (x_{i,j})) h^2 \text{ exact}, \]

if \((p, q) = \nabla d_{\Gamma}\) are relative prime and \(\varepsilon = \varepsilon(p, q) = \frac{|p| + |q|}{\sqrt{p^2 + q^2}} h.\)
Defining a “fat” integral equation

\[ f(x) = \lambda \rho(y) + \int_{\partial \Omega} \Phi(x, y) \rho(y) dS_y \]

• Proper definition of the equation for computation using implicit surfaces: \( \partial \Omega = \{x : \phi(x) = 0\} \)

\[ f_{ext}(x) \approx \lambda \rho_{ext}(y) + \int_{\mathbb{R}^d} \Phi(x, y) \rho_{ext}(y) \, \delta_\epsilon(\phi) |\nabla \phi| dy \]

Need extension of \( f \) and the unknown \( \gamma \) from \( \partial \Omega \) to \( \{-\epsilon < \phi < \epsilon\} \)

• Parameterization of the interface using nearby level sets

• Averaging over different parameterizations
Advantages of implicit interface boundary integral method

- Inherit certain advantages of boundary integral methods: solution on exterior domain problem, easy for both Dirichlet, Neumann, Robin, transmission problems.

- Inherit certain benefits from implicit interface formulation: one grid for complicated, topologically changing dynamic interface

- Flexible on the grid geometry used to embed the interface. Multi-resolution grid possible.
Inverse problem and shape optimization
Section 2

This section describes, in 14 steps, how to use Sandia National Labs' CUBIT and Mathworks' MATLAB to convert micro-CT scans of trabacular bone structures into surface mesh element representations, which are then used in Boundary Element Method (BEM) computation.

Step 1:
We begin with a collection of cross-sectional micro-CT scans, converted into logical 2D arrays of zeros and ones. First, we stack these arrays into a single 3D image, using the matlab function \([X \ Y \ Z \ Data]\) = stack_new(stacknum, cut). The variable "stacknum" specifies the number of images you wish to include, while the variable "cut" specifies how far from each of the edges of the original images we wish to place the frame for our new image. In other words, "cut" is how much is cut off from each side. For simplicity, and to automatically produce a smoothing scheme for the picture, our actual call is to the function

\[> \text{test_stack_new.m}\]

Variables X, Y, Z, Data, and smoothL are globally defined by this function call.

Step 2:
We use the built-in matlab functions isosurface and isocaps to create two meshes representing the surface. We then consolidate these two meshes into a single mesh. This entire process is encapsulated in the matlab function facet_MKmerge(X, Y, Z, smoothL, k). The variables X, Y, Z, and smoothL come from step 1; the new value k determines how the isosurface and isocaps functions will build surfaces around the data. The value k determines exactly where contour lines are drawn on the surface of our 3D logical array to form a mesh surface. Typically, we use k = 0.5. For a more detailed explanation, see Mathworks' documentation on isosurface and isocaps.

Step 3:
We now import the facet file into Sandia National Labs' CUBIT with the following settings:

Compute effective material properties.

Segmented by an MBO type scheme [Esedoglu-T]
LIDARs and 3D scanners
New formulas for integration over implicit surfaces
Parametrizing by a nearby level set

\[
\int_{\partial \Omega} \rho(\gamma(s)) \Phi(x, \gamma(s)) \, ds = \int_{\partial \Omega_\eta} \rho(y^*(s_\eta)) \Phi(x, y^*(s_\eta)) J_\eta \, ds_\eta
\]
The Jacobian

\[
A_0 = (R_1 - \eta)\theta_1 \cdot (R_2 - \eta)\theta_2
\]

\[
A_\eta = ds \ d\tau
= R_1 \theta_1 \cdot R_2 \theta_2
\]

\[
\frac{A_\eta}{A_0} = \frac{R_1 R_2}{(R_1 - \eta)(R_2 - \eta)}
= 1 + (\kappa_1 + \kappa_2)\eta + O(\eta^2)
\]

In fact, we have:

\[
J_\eta = 1 + 2H_\eta \eta + G_\eta \eta^2
\]
Average the identities

\[ I_0 = \int_{\partial \Omega} \rho(\gamma(s))\Phi(x, \gamma(s))ds = \int_{\partial \Omega_\eta} \rho(y^*(s_\eta))\Phi(x, y^*(s_\eta))J_\eta ds_\eta = I_\eta \]

\[ I_0 = \int_{\partial \Omega} \rho(\gamma(s))\Phi(x, \gamma(s))ds = \int_{-\epsilon}^{\epsilon} \delta_\epsilon(\eta)I_\eta d\eta \]

\[ = \int_{\mathbb{R}^n} \rho(z^*)\Phi(x, z^*)\delta_\epsilon(d_\Omega(z))J(z; d_\Omega)dz \]

\[ z^* = z - d(z)\nabla d(z) \]
The singular values of $D\mathcal{P}_\Sigma$

$\mathcal{P}_\Sigma y = y^*$

$= y - \phi(y) \nabla \phi(y) \quad \phi(y) = \min_{x \in \Sigma} |y - x|$

**Proposition** The following identity holds for sufficiently small $\epsilon$:

$$\int_{\Sigma} g(x) dS_x = \int_{\mathbb{R}^3} g(\mathcal{P}_\Sigma(y)) \lambda_1 \lambda_2 K_\epsilon(\phi(y)) dy,$$

where $K_\epsilon$ is any symmetric kernel supported in $[-\epsilon, \epsilon]$ having unit mass.

$\lambda_1 \lambda_2 = 0$ at the boundary of $\Sigma$

[Kublik-T:2015]
The Jacobian and singular values of

**Proposition** \ Let \( \lambda_1(x) \) be the largest singular value of \( D\mathcal{P}_\Gamma(x) \). The following identity holds for \( \epsilon < \kappa_\infty \), where \( \kappa_\infty \) is the maximal unsigned curvature of \( \Gamma \).

\[
\int_{\Gamma} g ds = \frac{1}{2\pi} \int_{\mathbb{R}^3} g(\mathcal{P}_\Gamma(x)) \frac{\lambda_1(x) K_\epsilon(\phi)}{\phi} dx.
\]

\( \delta_\epsilon(x; \Gamma) \)

\( \lambda_1 = 0 \) near the end points of \( \Gamma \)
**High order approximation**

Integration over a torus:

<table>
<thead>
<tr>
<th>$n$</th>
<th>Relative Error</th>
<th>Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>32</td>
<td>$6.2030 \times 10^{-3}$</td>
<td>–</td>
</tr>
<tr>
<td>64</td>
<td>$1.8073 \times 10^{-4}$</td>
<td>5.10</td>
</tr>
<tr>
<td>128</td>
<td>$6.6838 \times 10^{-6}$</td>
<td>4.76</td>
</tr>
<tr>
<td>256</td>
<td>$4.1530 \times 10^{-7}$</td>
<td>4.01</td>
</tr>
<tr>
<td>512</td>
<td>$5.0379 \times 10^{-8}$</td>
<td>3.04</td>
</tr>
</tbody>
</table>
Surfaces with boundaries

3/4 sphere and third-order one-sided finite differencing:

<table>
<thead>
<tr>
<th>$n$</th>
<th>Relative Error</th>
<th>Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>32</td>
<td>$1.1726 \times 10^{-2}$</td>
<td>–</td>
</tr>
<tr>
<td>64</td>
<td>$1.1733 \times 10^{-3}$</td>
<td>3.32</td>
</tr>
<tr>
<td>128</td>
<td>$9.1325 \times 10^{-4}$</td>
<td>0.36</td>
</tr>
<tr>
<td>256</td>
<td>$3.8238 \times 10^{-4}$</td>
<td>1.26</td>
</tr>
<tr>
<td>512</td>
<td>$7.8308 \times 10^{-5}$</td>
<td>2.29</td>
</tr>
</tbody>
</table>
3.2 Integrating along curves in three dimensions

In codimension 2, we test our numerical integration on a coil wrapped around the helix defined parametrically as

\[ x(t) = (r \cos(t), r \sin(t), bt), \]

with \( r = 0.75 \) and \( b = 0.25 \). The coil is then wrapped around the helix at a distance of 0.2 from the helix. As our test case, we compute the length of the coil by integrating along the curve. The results are reported in Table 6.

### Table 6: Relative errors in the numerical approximation of a coil wrapped around a helix.

In this computation, we used a constant width for the tubular neighborhood \( \varepsilon = 0.1 \) and took the averaging kernels to be \( K_1, K_1 \varepsilon \) defined in (17).

<table>
<thead>
<tr>
<th>( n )</th>
<th>Relative Error</th>
<th>Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>60</td>
<td>( 5.5078 \times 10^{-3} )</td>
<td>–</td>
</tr>
<tr>
<td>120</td>
<td>( 1.1476 \times 10^{-3} )</td>
<td>2.63</td>
</tr>
<tr>
<td>240</td>
<td>( 2.3409 \times 10^{-4} )</td>
<td>2.29</td>
</tr>
<tr>
<td>480</td>
<td>( 3.7166 \times 10^{-5} )</td>
<td>2.66</td>
</tr>
</tbody>
</table>

3.3 One-sided discretization of the Jacobian matrix

Here for completeness, we describe the one-sided discretization used in computing results reported in Table 5.

For simplicity, we will describe the one-sided discretization by a uniform Cartesian grid. For \( P(x_{i,j}) = (U_{i,j}, V_{i,j}) \) with \( x_{i,j} = (ih, jh) \), \( i, j \in \mathbb{Z} \) and \( h > 0 \) being the step size. The Jacobian matrix will be approximated by simple finite differences defined below:

\[ \frac{\partial P}{\partial x}(x_{i,j}) \approx \sum_{\pm} \left( U_{\pm 1/2,j} \frac{1}{h} (U_{x})_{i,j} \pm V_{\pm 1/2,j} \frac{1}{h} (V_{x})_{i,j} \right). \]

The discretization of \( U \) and \( V \) have to be defined together because the two functions are not independent of each other. With \( (U_{\pm 1}, x_{i,j}) := \frac{1}{2} (U_{i,j} + U_{i \pm 1, j}) \pm \frac{1}{2} h (U_{ix})_{i,j} \), and the smoothness indicator \( S_{\pm i,j} = S_{\pm i,j}(U_{i,j}) := 4 + 4 |U_{i \pm 1,j}| \), we define \( (U_{x})_{i,j} := (U_{x})_{i,j} \) if \(|S_{\pm i,j}| \leq |S_{i,j}|\), otherwise.
Summation over point clouds

\[ \Gamma_N \subset \Sigma \]

\( \{d_{\Gamma_N} = \eta\} \) has improved regularity

• 30 \times 30 uniformly distributed point clouds sampling in spherical coordinate the quarter sphere patch.

• 50 \times 50 \times 50 uniform Cartesian grid discretizing \([-1, 1]^3\).

• Relative error using \( \epsilon = 0.05 = dx: -0.56 \).

• Relative error using \( \epsilon = 0.2 = 4dx: -0.061 \).

\[ P : x \in \{d_{\Gamma_N} = \eta\} \mapsto x - \eta \nabla d_{\Gamma_N}(x) \] “interpolates”
Implicit interface boundary integral equation

\[ x, y \in \{-\epsilon \leq d_\Omega \leq \epsilon\} \]

\[ f(x^*) = \lambda \rho(x^*) + \int_{\mathbb{R}^d} \Phi(x^*, y^*) \rho(y^*) \delta_\epsilon(y; d_\Omega) dy \]

\[ x^* = x - d_\Omega(x) \nabla d_\Omega(x) \]

\[ \delta_\epsilon(x; d_\Omega) := \delta_\epsilon(d_\Omega(x)) J(x; d_\Omega) \]

Discretized by simple Riemann sum over the grid nodes.

Solve the resulting linear system by an iterative solver.

[Kublik-Tanushev-T:2013]
Regularization

- Need regularization of \( \frac{\partial \Phi}{\partial n} \) near singularity

\[
\frac{\partial \Phi}{\partial n} (x, y) \approx A_\epsilon (x, y) \text{ for } ||x - y|| < \epsilon
\]

- Approximate surfaces by paraboloids

- Approximate \( \frac{\partial \Phi}{\partial n} \) weakly in a neighborhood of \( x \) on the paraboloid (the 3D case):

\[
\int_{U(x; \epsilon)} \frac{\partial \Phi}{\partial n_y} (x, y) \alpha(y) dS(y) \approx \int_{\tilde{U}(x; \epsilon)} A_\epsilon (x, y) \alpha(y) dS(y)
\]
Regularization

\[ \int_{U(x; h)} \frac{\partial \Phi}{\partial n_y} (x, y) dS(y) \simeq \frac{1}{8\pi h} (\kappa_x + \kappa_y) |U(x; h)| \]
# Condition numbers

$N = 50^3$

<table>
<thead>
<tr>
<th>$h$</th>
<th>Cond. number with the tangent regularization</th>
<th>Cond. number with the paraboloid regularization</th>
</tr>
</thead>
<tbody>
<tr>
<td>$4dx$</td>
<td>6.2113</td>
<td>7.8322</td>
</tr>
<tr>
<td>$2dx$</td>
<td>7.3343</td>
<td>6.9390</td>
</tr>
<tr>
<td>$dx$</td>
<td>7.0467</td>
<td>7.4197</td>
</tr>
<tr>
<td>$\frac{dx}{2}$</td>
<td>6.8948</td>
<td>6.7572</td>
</tr>
</tbody>
</table>

$N = 80^3$

<table>
<thead>
<tr>
<th>$h$</th>
<th>Cond. number with the tangent regularization</th>
<th>Cond. number with the paraboloid regularization</th>
</tr>
</thead>
<tbody>
<tr>
<td>$4dx$</td>
<td>8.0859</td>
<td>7.8024</td>
</tr>
<tr>
<td>$2dx$</td>
<td>8.2791</td>
<td>7.7919</td>
</tr>
<tr>
<td>$dx$</td>
<td>7.4127</td>
<td>8.1231</td>
</tr>
<tr>
<td>$\frac{dx}{2}$</td>
<td>7.8830</td>
<td>8.1192</td>
</tr>
</tbody>
</table>
Exterior Neumann Problem for the Helmholtz Equation

\[
\begin{aligned}
\Delta u(x) + k^2 u(x) &= 0 & x &\in \tilde{\Omega}^c \\
\frac{\partial u}{\partial n}(x) &= g(x) & x &\in \partial\Omega \\
\lim_{|x| \to \infty} |x|^{\frac{1}{2}} (\frac{\partial}{\partial |x|} - ik)u(x) &= 0
\end{aligned}
\]

<table>
<thead>
<tr>
<th>(dx)</th>
<th>(Re(u_e))</th>
<th>(Re(u_{KTT}))</th>
<th>(Err_{KTT})</th>
<th>(Re(u_{new}))</th>
<th>(Err_{new})</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\frac{1}{256})</td>
<td>-0.117847</td>
<td>-0.383978</td>
<td>2.26E+00</td>
<td>-0.097671</td>
<td>1.71E-01</td>
</tr>
<tr>
<td>(\frac{1}{512})</td>
<td>-0.365593</td>
<td>2.10E+00</td>
<td>-0.114460</td>
<td>2.87E-02</td>
<td></td>
</tr>
<tr>
<td>(dx)</td>
<td>(Im(u_e))</td>
<td>(Im(u_{KTT}))</td>
<td>(Err_{KTT})</td>
<td>(Im(u_{new}))</td>
<td>(Err_{new})</td>
</tr>
<tr>
<td>(1/256)</td>
<td>0.472381</td>
<td>-0.125402</td>
<td>1.27E+00</td>
<td>0.457065</td>
<td>3.24E-02</td>
</tr>
<tr>
<td>(1/512)</td>
<td>-0.120072</td>
<td>1.25E+00</td>
<td>0.471583</td>
<td>1.69E-03</td>
<td></td>
</tr>
</tbody>
</table>

Table 3: The solution and relative error using the KTT IBIM with only one potential \((6)\) and the new formulation with both potentials \((7)\). Wave number \(k = 2.4048255577\ldots\) is an eigenvalue. The tubular width for the KTT integral is \(\epsilon = 0.145\). The tubular width for the new method is \(\delta_0 = 0.005, \epsilon_0 = 0.15\).
Solutions to Helmholtz equation
Happy Birthday!!