Söllerhaus-Workshop 2009
October 16, 2009

HILBERT
Matlab Implementation of Adaptive 2D BEM

Dirk Praetorius

joint work with

M. Aurada, M. Ebner, S. Ferraz-Leite, P. Goldenits, M. Karkulik, M. Mayr

Vienna University of Technology
Institute for Analysis and Scientific Computing

What is HILBERT ?

Hilbert Is a Lovely Boundary Element Research Tool

- Matlab library for h-adaptive Galerkin BEM
- currently, lowest-order elements for 2D Laplacian
  - $\mathcal{P}^0$ for fluxes
  - $\mathcal{S}^1$ for traces
- research code for FWF project P21732
  - overview on current state of the art
  - starting point for further investigations
  - free for academic use

HILBERT?

Features of HILBERT

- C implementation of integral operators via MEX interface
  - $V(\mathcal{P}^0_\Gamma \times \mathcal{P}^0_\Gamma)$
  - $K(\mathcal{S}^1_\Gamma \times \mathcal{P}^0_\Gamma)$
  - $W(\mathcal{S}^1_\Gamma \times \mathcal{S}^1_\Gamma)$
  - $N(\mathcal{P}^1_\Omega \times \mathcal{P}^0_\Gamma)$
- remaining codes in Matlab fully vectorized
  - different error estimators
  - different marking strategies
  - local mesh-refinement
- demo files and adaptive algorithms for
  - weakly-singular integral equation
  - hypersingular integral equation
  - symmetric integral formulation of mixed BVP
  - with/without volume force

Current Restrictions of HILBERT

- $\Gamma = \partial\Omega$ piecewise affine boundary
- only lowest-order elements
- only canonical bases
  - characteristic functions for $\mathcal{P}^0$
  - hat functions for $\mathcal{S}^1$
- no matrix compression, i.e., dense matrices
- direct solution of linear systems by Matlab backslash operator

Weakly-Singular Integral Equation

Outline

Model Problem

Dirichlet Problem

$$\begin{aligned} -\Delta u &= 0 && \text{in } \Omega \\ u &= g && \text{on } \Gamma \end{aligned}$$

Weakly-Singular Integral Equation

$$V\phi = (K + 1/2)g$$

- $\mathcal{H} := H^{-1/2}(\Gamma)$ energy space
- $\|\cdot\|$ equivalent energy norm on $\mathcal{H}$ induced by V

Variational Formulation

Continuous Formulation

Seek $\phi \in \mathcal{H} := H^{-1/2}(\Gamma)$ s.t.

$$\langle V\phi\,,\psi\rangle = \langle (K+1/2)g\,,\psi\rangle \quad \text{for all } \psi \in \mathcal{H}$$

- $\mathcal{T}_\ell$ partition of Γ

Galerkin Formulation

Seek $\Phi_\ell^\star \in X_\ell := \mathcal{P}^0(\mathcal{T}_\ell)$ s.t.

$$\langle V\Phi_\ell^\star\,,\Psi_\ell\rangle = \langle (K+1/2)g\,,\Psi_\ell\rangle \quad \text{for all } \Psi_\ell \in X_\ell$$

- drawback: RHS hard to compute in general

Perturbed Galerkin Formulation 2/2

Matrix Formulation of Perturbed Galerkin Formulation

Compute coefficient vector $\mathbf{x}$ of Φ_ℓ by solving

$$\mathbf{V}\mathbf{x} = \Big(\mathbf{K} + \frac{1}{2}\mathbf{M}\Big)\,\mathbf{g}$$

where $\mathbf{g}$ is nodal vector of g

- mass-type matrix $\mathbf{M}$ analytically computable and sparse
- integral operator matrices $\mathbf{V}$, $\mathbf{K}$ analytically computable

Perturbed Galerkin Formulation 1/2

- assume additional regularity $g \in H^1(\Gamma)$
- $G_\ell \in \mathcal{S}^1(\mathcal{T}_\ell)$ nodal interpolant of g

Perturbed Galerkin Formulation

Seek $\Phi_\ell \in X_\ell$ s.t.

$$\langle V\Phi_\ell\,,\Psi_\ell\rangle = \langle (K+1/2)G_\ell\,,\Psi_\ell\rangle \quad \text{for all } \Psi_\ell \in X_\ell$$

Control Data Approximation

Reminder: Continuous Problem

$$V\phi = (K+1/2)g$$

Reminder: Perturbed Continuous Problem

$$V\phi_\ell = (K+1/2)G_\ell$$

Lemma (Aurada, Goldenits, P. '09)

$$\|\phi - \phi_\ell\| \lesssim \|g - G_\ell\|_{H^{1/2}(\Gamma)} \lesssim \|h_\ell^{1/2}(g-G_\ell)'\|_{L^2(\Gamma)} =: \mathrm{osc}_\ell$$

Control Galerkin Error

- Φ_ℓ Galerkin solution w.r.t. $\mathcal{T}_\ell$
- $\widehat{\Phi}_\ell$ Galerkin solution w.r.t. $\widehat{\mathcal{T}}_\ell$
- L^2-projection Π_ℓ onto $X_\ell = \mathcal{P}^0(\mathcal{T}_\ell)$

Lemma (Ferraz-Leite, P. '08)

There holds $\eta_\ell \le \widetilde{\eta}_\ell \lesssim \widetilde{\mu}_\ell \le \mu_\ell \lesssim \eta_\ell$ for error estimators

$$\eta_\ell = |||\widehat{\Phi}_\ell - \Phi_\ell||| \qquad \widetilde{\eta}_\ell = |||\widehat{\Phi}_\ell - \Pi_\ell\widehat{\Phi}_\ell|||$$
$$\mu_\ell = \|h_\ell^{1/2}(\widehat{\Phi}_\ell - \Phi_\ell)\|_{L^2(\Gamma)} \qquad \widetilde{\mu}_\ell = \|h_\ell^{1/2}(\widehat{\Phi}_\ell - \Pi_\ell\widehat{\Phi}_\ell)\|_{L^2(\Gamma)}$$

Local Refinement Indicators

$$\rho_\ell(T)^2 := \|h_\ell^{1/2}(\widehat{\Phi}_\ell - \Pi_\ell\widehat{\Phi}_\ell)\|_{L^2(T)}^2 + \|h_\ell^{1/2}(g - G_\ell)'\|_{L^2(T)}^2$$
$$|||\phi - \Phi_\ell|||^2 \lesssim (\eta_\ell + \mathrm{osc}_\ell)^2 \lesssim \widetilde{\mu}_\ell^2 + \mathrm{osc}_\ell^2 = \sum_{T\in\mathcal{T}_\ell} \rho_\ell(T)^2$$

- everything works also with μ_ℓ instead of $\widetilde{\mu}_\ell$

Control Galerkin and Data Approximation Error

Saturation Assumption for Non-Perturbed Problem

- $\Phi_\ell^\star$ Galerkin solution w.r.t. $\mathcal{T}_\ell$ and non-perturbed RHS
- $\widehat{\Phi}_\ell^\star$ Galerkin solution w.r.t. $\widehat{\mathcal{T}}_\ell$ and non-perturbed RHS
- $|||\phi - \widehat{\Phi}_\ell^\star||| \le q\,|||\phi - \Phi_\ell^\star|||$

Corollary (Aurada, Goldenits, P. '09)

- $\eta_\ell \lesssim |||\phi - \Phi_\ell||| + \mathrm{osc}_\ell$
- saturation assumption $\Longrightarrow$ $|||\phi - \Phi_\ell||| \lesssim \eta_\ell + \mathrm{osc}_\ell$

Adaptive Mesh-Refinement

- initial mesh $\mathcal{T}_0$
- adaptivity parameter $0 < \theta < 1$

Adaptive Algorithm

1. compute discrete solution $\widehat{\Phi}_\ell \in \widehat{X}_\ell = \mathcal{P}^0(\widehat{\mathcal{T}}_\ell)$
2. for all $T \in \mathcal{T}_\ell$, compute $\rho_\ell(T)$
3. find (minimal) set $\mathcal{M}_\ell \subseteq \mathcal{T}_\ell$ s.t.
$$\theta \sum_{T\in\mathcal{T}_\ell} \rho_\ell(T)^2 \le \sum_{T\in\mathcal{M}_\ell} \rho_\ell(T)^2$$
4. refine (at least) marked elements $T \in \mathcal{M}_\ell$ to obtain $\mathcal{T}_{\ell+1}$
   - ensure $\kappa(\mathcal{T}_{\ell+1}) \le 2\,\kappa(\mathcal{T}_0)$
5. increase counter $\ell \mapsto \ell + 1$ and iterate

Convergence of Adaptive Scheme

Theorem (Aurada, Goldenits, P. '09)

- Use refinement procedure of [AGP '09]
- Then, concept of estimator reduction applies, whence

$$\lim_{\ell\to\infty} (\widetilde{\mu}_\ell^2 + \mathrm{osc}_\ell^2) = 0$$

- Saturation assumption for non-perturbed problem implies

$$\lim_{\ell\to\infty} \|\phi - \widehat{\Phi}_\ell\| = 0 = \lim_{\ell\to\infty} \|\phi - \Phi_\ell\|$$

Data Structure

- $\mathcal{T}_\ell = \{T_1, \dots, T_N\}$ partition
- $\mathcal{N}_\ell = \{z_1, \dots, z_N\}$ nodes

coordinates

- $(N \times 2)$ double-array
- $z_j = (x, y) \in \mathbb{R}^2$ corresponds to coordinates$(j, :) = [\,x\,,\,y\,]$

elements

- $(N \times 2)$ int-array
- $T_i = \mathrm{conv}\{z_j, z_k\}$ corresponds to elements$(i, :) = [\,j\,,\,k\,]$

Introduction to HILBERT

Build Galerkin Data

buildV

- assembles $\mathbf{V} = V(\mathcal{P}^0 \times \mathcal{P}^0)$
- use of symmetry
- outer integration over element with $\mathrm{length}(T_j) \le \mathrm{length}(T_k)$
- outer integration by Gaussian quadrature for far-field entries

buildK

- assembles $\mathbf{K} = K(\mathcal{S}^1 \times \mathcal{P}^0)$

buildSymmRHS

- assembles $\mathbf{b}$ for perturbed RHS $(K + 1/2)G_\ell$

Local Mesh-Refinement

markElements

- returns index vector for marked elements $\mathcal{M}_\ell \subseteq \mathcal{T}_\ell$ from

$$\theta \sum_{T\in\mathcal{T}_\ell} \rho_\ell(T)^2 \le \sum_{T\in\mathcal{M}_\ell} \rho_\ell(T)^2$$

- even multiple triangulations, e.g., FEM-BEM, mixed BVP

refineBoundaryMesh

- algorithm from [Aurada, Goldenits, P. '09]
- complexity $\mathcal{O}(N\log N)$ for local mesh-refinement
- complexity $\mathcal{O}(N)$ for uniform mesh-refinement

$(h-h/2)$-Error Estimators 1/2

computeEstSlpEta

- returns $\eta_\ell^2 = \|\widehat{\Phi}_\ell - \Phi_\ell\|^2$.

computeEstSlpEtaTilde

- returns $\widetilde{\eta}_\ell^2 = \|\widehat{\Phi}_\ell - \Pi_\ell\widehat{\Phi}_\ell\|^2$.

Data Oscillations

computeOscDirichlet

- returns $\mathbf{v}\in\mathbb{R}^N$ with $\mathrm{osc}_\ell^2 \approx \widetilde{\mathrm{osc}}_\ell^2 = \sum_{j=1}^N \mathbf{v}_j$, where

$$\mathbf{v}_j \approx \mathrm{length}(T_j)\|(g-G_\ell)'\|_{L^2(T_j)}^2$$

- quadrature located at nodes and midpoints
- exact for $g\in\mathcal{S}^2$
- $\mathrm{osc}_\ell = \mathcal{O}(h^{3/2})$ and $|\mathrm{osc}_\ell - \widetilde{\mathrm{osc}}_\ell| = \mathcal{O}(h^{5/2})$ for smooth g

$(h-h/2)$-Error Estimators 2/2

computeEstSlpMu

- returns $\mathbf{v}\in\mathbb{R}^N$ with $\mu_\ell^2 = \sum_{j=1}^N \mathbf{v}_j$, where

$$\mathbf{v}_j = \mathrm{length}(T_j)\|\widehat{\Phi}_\ell - \Phi_\ell\|_{L^2(T_j)}^2$$

computeEstSlpMuTilde

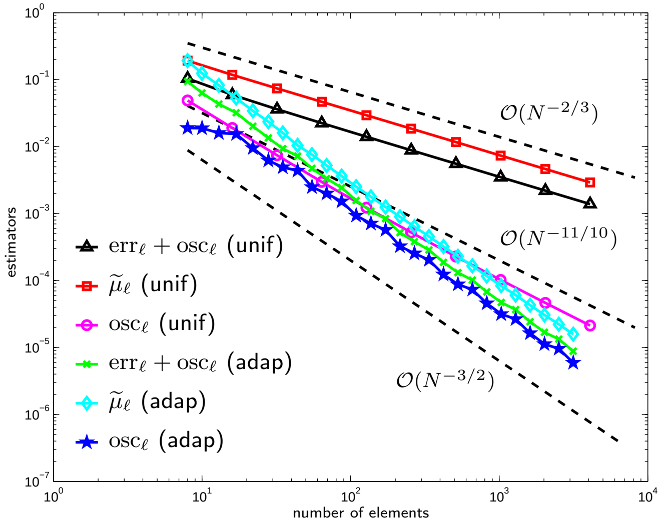
- returns $\widetilde{\mathbf{v}}\in\mathbb{R}^N$ with $\widetilde{\mu}_\ell^2 = \sum_{j=1}^N \widetilde{\mathbf{v}}_j$, where

$$\widetilde{\mathbf{v}}_j = \mathrm{length}(T_j)\|\widehat{\Phi}_\ell - \Pi_\ell\widehat{\Phi}_\ell\|_{L^2(T_j)}^2$$

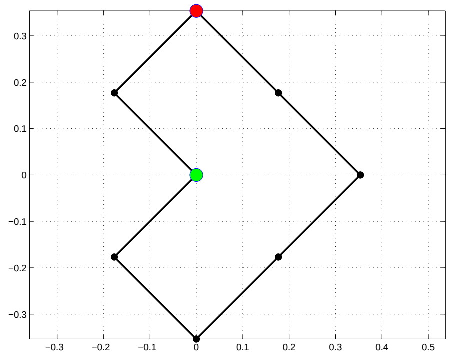
Adaptive Mesh-Refining Algorithm

```
while size(elements,1) < nEmax

    %*** build uniformly refined mesh
    [coordinates_fine,elements_fine,father2son] ...
        = refineBoundaryMesh(coordinates,elements);

    %*** compute fine-mesh solution
    V_fine = buildV(coordinates_fine,elements_fine);
    b_fine = buildSymmRHS(coordinates_fine,elements_fine,@g);
    x_fine = V_fine\b_fine;

    %*** compute (h-h/2)-error estimator tilde-mu
    mu_tilde = computeEstSlpMuTilde(coordinates,elements,father2son, ...
                                     x_fine);
    %*** compute data oscillations
    osc_fine = computeOscDirichlet(coordinates_fine,elements_fine,@g);
    osc = osc_fine(father2son(:,1)) + osc_fine(father2son(:,2));

    %*** mark elements for refinement
    marked = markElements(theta,mu_tilde + osc);

    %*** generate new mesh
    [coordinates,elements] = refineBoundaryMesh(coordinates,elements,marked);

end
```


Numerical Experiment 2/2



Numerical Experiment 1/2

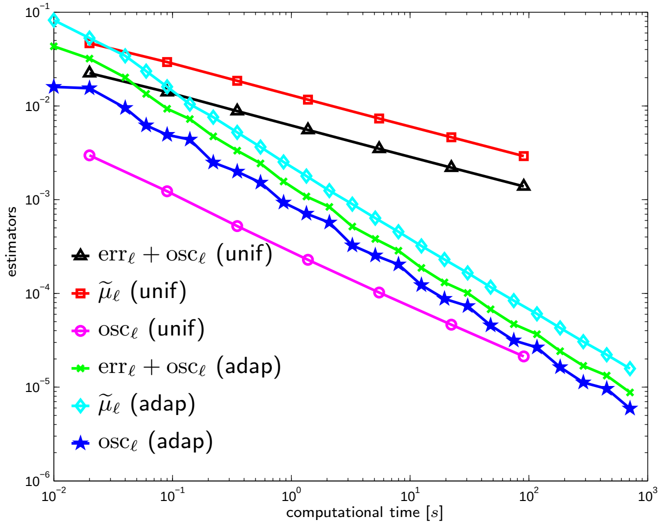


- known exact solution ϕ
- generic singularity of ϕ at reentrant corner **(green)**
- singularity of g at uppermost corner **(red)**

Adaptivity Pays 1/2

- adaptive pays w.r.t. number of elements
  - observe optimal convergence behaviour
- does adaptivity pay w.r.t. computational time?
  - $t_\ell^{(\mathrm{unif})} := \mathtt{unif}^{(\ell)}(\mathcal{T}_0) + \mathtt{assemble} + \mathtt{solve}$
  - $t_0^{(\mathrm{adap})} := \mathtt{assemble} + \mathtt{solve} + \mathtt{estimate} + \mathtt{mark} + \mathtt{refine}$
  - $t_{\ell+1}^{(\mathrm{adap})} := t_\ell^{(\mathrm{adap})} + \dots$

Adaptivity Pays 2/2



HILBERT — Release One

- release date: Friday 13. November 2009
- available at: `http://www.asc.tuwien.ac.at/abem`

- Release One contains lowest-order BEM for 2D Laplacian
  - Dirichlet problem
  - Neumann problem
  - mixed Dirichlet-Neumann problem
  - with/without volume forces
- Release One includes
  - $(h - h/2)$-error estimators for all problems
  - local mesh-refinement for boundary/volume meshes
  - adaptive algorithms & demo files
  - several visualization tools
  - full documentation of all Matlab codes (about 60 pages)

Conclusion & Outlook

Short-Term Extensions

- further error estimators, e.g.,
  - weighted-residual error estimators
  - two-level error estimators
  - Faermann error estimator
  - Steinbach-Schulz error estimator

- adaptive FEM-BEM coupling

- Lamé equation

Thank You for Listening

Dirk Praetorius

Vienna University of Technology
Institute for Analysis
and Scientific Computing

http://www.asc.tuwien.ac.at/abem