Exponential fast convergence on 'complicated' geometries with BEM

M. Maischak

School of Information Systems, Computing & Mathematics
Brunel University

JSA-Rennes, 30.8.2013

Outline

1. hp-version
   - Convergence
   - Quadrature–exact
   - Quadrature–numerical
2. Numerical Quadrature
   - Curve (2d)
   - Surface (3d)
3. Numerical experiments
   - Curve (2d)
   - Surface (3d)

Weakly singular integral equation

Given $f \in H^{1/2}(\Gamma)$, find $\psi \in \tilde{H}^{-1/2}(\Gamma)$, s.t.

$$V\psi = f \text{ on } \Gamma$$

Single layer potential

$$Vw(x) = 2 \int_\Gamma G(x,y) w(y)\, ds_y,$$

with $G(x,y) = \begin{cases} -\frac{1}{2\pi} \log|x-y|, & d = 2 \\ \frac{1}{4\pi} |x-y|^{-1}, & d = 3 \end{cases}.$

Galerkin's method:
Find $\psi_N \in X_N \subset \tilde{H}^{-1/2}(\Gamma)$ s.t.

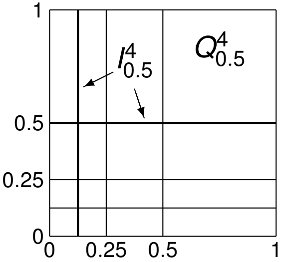
$$\langle V\psi_N, w \rangle = \langle f, w \rangle \;\; \forall w \in X_N$$

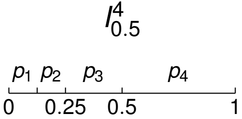
Given $Q = [0,1]^2$, $n \in \mathbb{N}$, $0 < \sigma < 1$, deg. vec. $p = (p_1, \ldots, p_n)$
Partition I_σ^n of I into n subintervals (x_{j-1}, x_j), $j = 1, \ldots, n$, where

$$x_0 = 0, \qquad x_j = \sigma^{n-j}, \quad j = 1, \ldots, n.$$

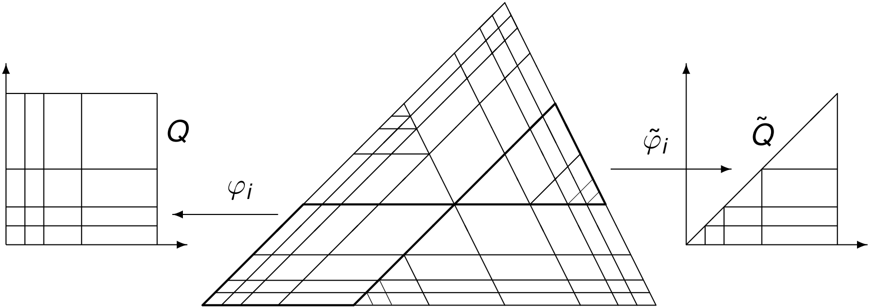
Partition Q_σ^n of Q into n^2 rectangles $R_{kl} = [x_{k-1}, x_k] \times [x_{l-1}, x_l]$.
Define

$$S^{p,r}(Q_\sigma^n) = \{ w \in C^{r-1}(Q) \;:\; w|_{[x_{k-1},x_k]\times[x_{l-1},x_l]} \in P_{p_k,p_l}(R_{kl}) \}$$





Geometric mesh on the square Q ($\sigma = 0.5$, $n = 4$).



Geometric mesh with $\sigma = 0.5$ on a polyhedral face

Theorem (Heuer, Stephan 1992)

Let f be piecewise analytic, and Γ polygonal or piecewise analytic. Then there holds $\psi_N \in X_N = S^{p,0}(\Gamma_\sigma^n)$

$$\|\psi - \psi_N\|_{\tilde{H}^{-1/2}(\Gamma)} \leq C\, e^{-bn} \lesssim C\, e^{-b\sqrt{\dim X_N}}$$

Theorem (Heuer, Maischak, Stephan 1995)

Let f be piecewise analytic, and Γ polyhedral or piecewise analytic. Then there holds for $\psi_N \in X_N = S^{p,0}(\Gamma_\sigma^n)$

$$\|\psi - \psi_N\|_{\tilde{H}^{-1/2}(\Gamma)} \leq C\, e^{-bn} \lesssim C\, e^{-b\sqrt[4]{\dim X_N}}$$

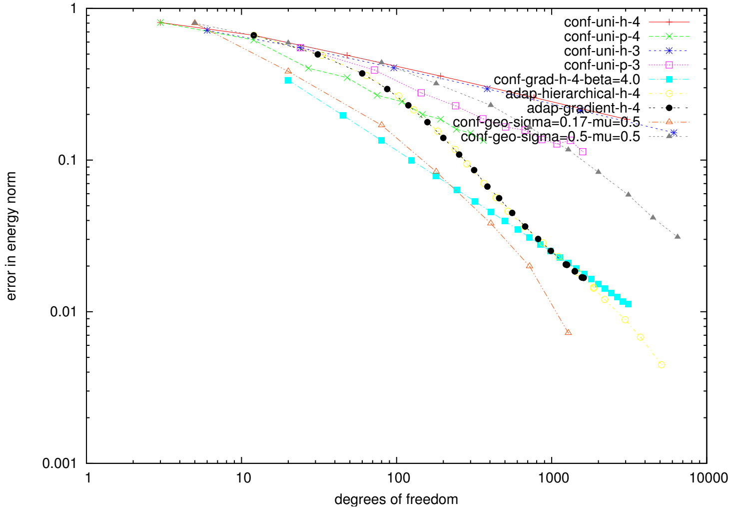


Figure : $V\psi = 1$ on L-Shape (3d)

Polyhedral surface decomposed in parallelograms and triangles

$$\Gamma = \cup_{i=1}^{N} \Gamma_i, \quad \Gamma_i = \{a_i t_1 + b_i t_2 + x_i \,:\, (t_1, t_2) \in \{\Delta, \Box\}\}, \quad a_i, b_i, x_i \in \mathbb{R}^3$$

$\Box = [-1, 1]^2$, $\Delta = \{(t_1, t_2) \,:\, 0 \le t_1 \le 1 - t_2 \le 1\}$.

Single layer potential

$$\begin{aligned} V\varphi_{kl}^{i}(x) &:= \int_\Gamma \frac{1}{|x-y|} \varphi_{kl}^{i}(y)\, ds_y \\ &= |a_i \times b_i| \int_{\{\Delta,\Box\}} |a_i t_1 + b_i t_2 + x_i - x|^{-1} t_1^k t_2^l \, dt_2 \, dt_1 \end{aligned}$$

Galerkin element

$$F_{klmn}^{ij,p} := \int\limits_{\{\Delta,\Box\}} t_1^k t_2^l \int\limits_{\{\Delta,\Box\}} s_1^m s_2^n |a_i t_1 + b_i t_2 + a_j s_1 + b_j s_2 + x_i - x_j|^{2p} \, ds_2 \, ds_1 \, dt_2 \, dt_1$$

Orientation	same plane	not parallel	parallel
Analytical integrations	4	3	2
Parallel edges needed	0	1	2

Brunel UNIVERSITY LONDON

Recurrence relations

Advantages

- In 2d all Galerkin elements can be computed analytical
- In 3d some numerical integration is necessary, but singular integrals can always computed analytical
- No need for numerical integration of singular integrals
- Very fast for near-field computation

Disadvantages

- Recurrence relations use monomials for basis functions, i.e. high complexity for transformation to other basis functions, $n^2 p^4$ in 2d.
- Naive implementation leeds to numerical instability.
- Numerical stability for higher p requires very carefull implementation
- Only polygonal or polyhedral boundaries.
- Restricted to logarithmic or integer power of $|x - y|$ kernels.

Numerical quadrature on curved surfaces

h-Version

- H-Matrices, ACA
- Multipole
- Panel-clustering
- Wavelets

p-Version

- Large Nearfield, no H-Matrices, no Multipole
- Far field: evaluation of surface representation expensive
- Near field: Danger of loss of precision due to cancellation
- For optimality: Anisotropic elements in 3D

- Compute matrix
- Compute rhs
- Solve system

$$Ax = b$$

or

$$(A + \delta A)(x + \delta x) = b + \delta b$$

with error

$$\frac{\|\delta x\|}{\|x\|} \leq \frac{\mathrm{cond}(A)}{1 - \|A^{-1}\delta A\|} \left(\frac{\|\delta A\|}{\|A\|} + \frac{\|\delta b\|}{\|b\|} \right)$$

- Postprocessing
  - Exact solution $\|u - u_N\|_{H^1(\Omega)}$
  - Energy norm $\|u - u_N\|_E = \sqrt{\|u\|_E^2 - \|u_N\|_E^2} = \sqrt{\|u\|_E - \|u_N\|_E}\sqrt{\|u\|_E + \|u_N\|_E}$

Brunel UNIVERSITY LONDON

p-BEM	graded quadrature of singular kernel (cancellation) high-order polynomials (complexity)
hp-BEM	small elements (cancellation) graded quadrature of singular kernel (cancellation) high-order polynomials (complexity)

Remedies

problem	remedy
complexity	pre-computation/re-use of function values
	sum-factorization
cancellation	change data model, store distances
	for transformations $F(x) - F(y) = G(x - y, x, y)$
	multiple precision
- Using multiple precision software increases the complexity again and increases the storage requirements!
- Also, we don't want to manage more than one code version

Sum factorization

```
for i=1:N
 for j=1:N
  for k=1:N
   for l=1:N
    d(i,j)+=a(i,k)*b(k,l)*c(l,j)
   end
  end
 end
end
```

Complexity: $\mathcal{O}(N^4)$ $(D = A \cdot B \cdot C)$

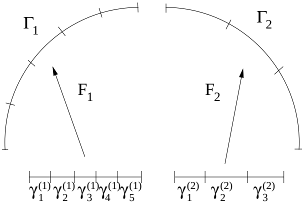
```
for i=1:N
 for k=1:N
  for l=1:N
    e(i,l)+=a(i,k)*b(k,l)
  end
 end
end
for i=1:N
 for j=1:N
  for l=1:N
   d(i,j)+=e(i,l)*c(l,j)
  end
 end
end
```

Complexity: $\mathcal{O}(N^3)$ $(E = A \cdot B, D = E \cdot C)$

Parametrization

Curve decomposed into M patches

$$\Gamma = \bigcup_{m=1}^{M} \Gamma_m$$



every patch represented by local polynomial of order n, $F_m : [0,1] \to \Gamma_m$.

$$\vec{x}(q) = \sum_{k=0}^{n} a_k^{(m)} q^k, \quad q \in [0,1]$$

Every local partition $[0,1] = \cup_{i=1}^{N_m} \gamma_i^{(m)}$ defines partition of Γ_m,i.e.

$$\Gamma = \bigcup_{m=1}^{M} \bigcup_{i=1}^{N_m} F_m(\gamma_i^{(m)}) = \bigcup_{m=1}^{M} \bigcup_{i=1}^{N_m} \Gamma_{m,i}$$

We have to compute

$$a_{m,i,k;n,j,l} = \int_{\Gamma_{m,i}} \int_{\Gamma_{n,j}} K(\vec{x},\vec{y}) L_k(\vec{x}) L_l(\vec{y})\, ds_y\, ds_x$$
$$= \int_{\gamma_i^{(m)}} \int_{\gamma_j^{(n)}} K(F_m(t), F_n(s)) L_k(t) L_l(s) \left|\frac{dF_m}{dt}\right| \left|\frac{dF_n}{ds}\right| ds\, dt$$

for $0 \le k \le p_{m,i}$, $0 \le l \le p_{n,j}$.

There are three cases

- Identical element, i.e. $\Gamma_{m,i} = \Gamma_{n,j}$.
- Vertex-Vertex, i.e. $\Gamma_{m,i} \cap \Gamma_{n,j} \neq \emptyset$.
- Farfield, integrals are regular

Far field

$$a_{m,i,k;n,j,l} = \int_{\gamma_i^{(m)}} \int_{\gamma_j^{(n)}} K(F_m(t), F_n(s)) L_k(t) L_l(s) \left| \frac{dF_m}{dt} \right| \left| \frac{dF_n}{ds} \right| ds\, dt$$

$$\approx \sum_{r=1}^{q_{m,i}} \sum_{o=1}^{q_{n,j}} K(F_m(t_r), F_n(s_o)) L_k(t_r) L_l(s_o) \left| \frac{dF_m(t_r)}{dt} \right| \left| \frac{dF_n(s_o)}{ds} \right| w_r w_o$$

for $0 \le k \le p_{m,i}$, $0 \le l \le p_{n,j}$.
For every quadrature node on every element we have to evaluate

$$F_m(t_r), \quad \left| \frac{dF_m(t_r)}{dt} \right|, \quad 1 \le r \le q_{m,i}$$

Avoiding repeated evaluation, by precomputing and saving the data.

Naive algorithm

```
For m=1:M; For i=1:N(m)
 For n=1:M; For j=1:N(n)
  If (.not.Farfield(m,n,i,j)) cycle
  For k=1:p(m,i)
   For l=1:p(n,j)
    For r=1:gp
     For o=1:gp
      a(m,i,k;n,j,l)+=K(F(m,t(r))-F(n,s(o)))
                              *L(k,t(r))*L(l,s(o))*w(r)*w(o)
                              *det(F,m,t(r))*det(F,n,s(o))
     End
    End
   End
  End
 End; End
End; End
```

Optimizations (p-Version)

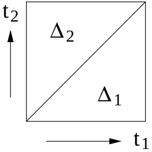
```
For m=1:M; For i=1:N(m)
 For n=1:M; For j=1:N(n)
  If (.not.Farfield(m,n,i,j)) cycle
  For r=1:gp
   For o=1:gp
    Compute Lm(k):=L(k,t(r)) for k=0:p(m,i)
    Compute Ln(l):=L(l,s(o)) for l=0:p(n,j)
    a(m,i,0:p;n,j,0:p)+=K(F(m,t(r))-F(n,s(o)))
                    *Lm(0:p)*Ln(0:p)*w(r)*w(o)
                    *det(F,m,t(r))*det(F,n,s(o))
   End
  End
 End; End
End; End
```

Optimizations (Farfield)

```
Compute for m=1:M,i=1:N(m):r=1:gp
 x(m,i,r):=F(m,t(r)) and det(m,i,r):=det(F,m,t(r))
For m=1:M; For i=1:N(m)
 For n=1:M; For j=1:N(n)
  If (.not.Farfield(m,n,i,j)) cycle
  For r=1:gp
   For o=1:gp
    Compute Lm(k):=L(k,t(r)) for k=0:p(m,i)
    Compute Ln(l):=L(l,s(o)) for l=0:p(n,j)
    a(m,i,0:p;n,j,0:p)+=K(x(m,i,r)-x(n,j,s))
                 *Lm(0:p)*Ln(0:p)*w(r)*w(o)
                 *det(m,i,r)*det(n,j,s)
   End
  End
 End; End
End; End
```

Vertex-Vertex, Duffy-transformation

$$\int_0^1 \int_0^1 f(t_1, t_2)\, dt_2\, dt_1$$
$$= \int_{\Delta_1} f(t_1, t_2)\, dt_2\, dt_1 + \int_{\Delta_2} f(t_1, t_2)\, dt_2\, dt_1$$
$$= \int_{\Box} \underbrace{f(q_1, q_2 q_1) q_1}_{=:g_1(q_1,q_2)} + \int_{\Box} \underbrace{f(q_1 q_2, q_2) q_2}_{=:g_2(q_1,q_2)}$$

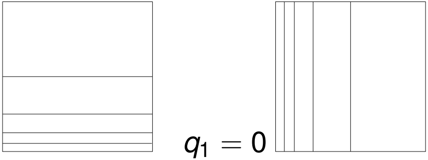


Singularity in (0,0)

$$\Delta_1 = \{(t_1, t_2) \,|\, t_2 \leq t_1\}$$
$$\Delta_2 = \{(t_1, t_2) \,|\, t_1 \leq t_2\}$$

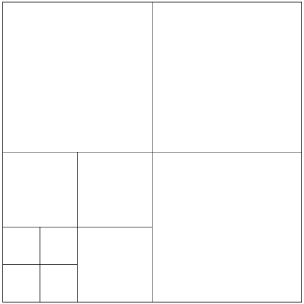
Graded quadrature towards edges

$q_2 = 0$ $q_1 = 0$



Vertex-Vertex, geometric grading

Point singularity in (0,0) can also treated by graded quadrature towards the vertex.[1]

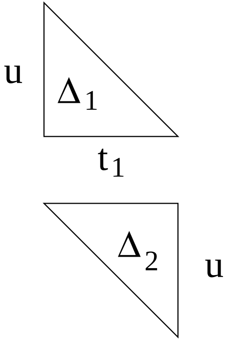


Duffy regularisation reduces order of singularity before edge-grading.

[1] C. Schwab: Variable Order Composite Quadrature of Singular and Nearly Singular Integrals Computing 53 (1994), 173-194

Identical-elements

Singularity for $t_1 = t_2$



$$\int_0^1 \int_0^1 f(t_1, t_2)\, dt_2\, dt_1$$

$$= \int_0^1 \int_{-t_1}^{1-t_1} f(t_1, t_1 + u)\, du\, dt_1$$

$$= \int_{\Delta_1 \cup \Delta_2} f(t_1, t_1 + u)\, du\, dt_1$$

$$= \int_{\square} f(q_1(1 - q_2), q_1(1 - q_2) + q_2) \cdot (1 - q_2)$$

$$+ \int_{\square} f(1 - q_1(1 - q_2), 1 - q_1(1 - q_2) - q_2) \cdot (1 - q_2)$$

Singularity remains for $q_2 = 0$, edge-grading

Surface patch parametrization/ local meshes

Surface decomposed into M patches

$$\Gamma = \bigcup_{m=1}^{M} \Gamma_m$$

every patch represented by local polynomial of order n, $F_m : [0,1]^2 \to \Gamma_m$.

$$\vec{x}(q_1, q_2) = \sum_{k=0}^{n} \sum_{l=0}^{n} a_{kl}^{(m)} q_1^k q_2^l, \quad (q_1, q_2) \in [0,1]^2$$

Every local partition $[0,1]^2 = \cup_{i=1}^{N_m} \gamma_i^{(m)}$ defines partition of Γ_m,i.e.

$$\Gamma = \bigcup_{m=1}^{M} \bigcup_{i=1}^{N_m} F_m(\gamma_i^{(m)})$$

Local partition elements $\gamma_i^{(m)}$ described by vertices $\vec{z}_1, \vec{z}_2, \vec{z}_3, \vec{z}_4$.

$$\begin{aligned}\vec{q} = \ & \vec{z}_1(1 - t_1 - t_2 + t_1 t_2) + \vec{z}_2(1 + t_1 - t_2 - t_1 t_2) \\ & + \vec{z}_3(1 + t_1 + t_2 + t_1 t_2) + \vec{z}_4(1 - t_1 + t_2 - t_1 t_2)\end{aligned}$$

$$\begin{aligned}\frac{\partial \vec{q}}{\partial t_1} &= z_1(-1 + t_2) + z_2(1 - t_2) + z_3(1 + t_2) + z_4(-1 - t_2) \\ \frac{\partial \vec{q}}{\partial t_2} &= z_1(-1 + t_1) + z_2(-1 - t_1) + z_3(1 + t_1) + z_4(1 - t_1)\end{aligned}$$

$$\begin{aligned}\frac{\partial \vec{x}}{\partial t_1} &= \frac{\partial \vec{x}}{\partial q_1}\frac{\partial q_1}{\partial t_1} + \frac{\partial \vec{x}}{\partial q_1}\frac{\partial q_2}{\partial t_1} \\ \frac{\partial \vec{x}}{\partial t_2} &= \frac{\partial \vec{x}}{\partial q_1}\frac{\partial q_1}{\partial t_2} + \frac{\partial \vec{x}}{\partial q_1}\frac{\partial q_2}{\partial t_2}\end{aligned}$$

We have to compute

$$a_{m,i,\vec{k};n,j,\vec{l}} = \int_{\Gamma_{m,i}} \int_{\Gamma_{n,j}} K(\vec{x},\vec{y}) L_{k_1}(\vec{x}) L_{k_2} L_{l_1}(\vec{y}) L_{l_2}(\vec{y})\, ds_y\, ds_x$$
$$= \int_{\gamma_i^{(m)}} \int_{\gamma_j^{(n)}} K(F_m(t), F_n(s)) L_{k_1}(t_1) L_{k_2}(t_2) L_{l_1}(s_1) L_{l_2}(s_2) \left|\frac{dF_m}{dt}\right| \left|\frac{dF_n}{ds}\right| ds\, dt$$

for $0 \leq k_1, k_2 \leq p_{m,i}$, $0 \leq l_1, l_2 \leq p_{n,j}$.

There are four cases

- Identical element, i.e. $\Gamma_{m,i} = \Gamma_{n,j}$.
- Edge-Edge, i.e. $\Gamma_{m,i} \cap \Gamma_{n,j}$ is a line.
- Vertex-Vertex, i.e. $\Gamma_{m,i} \cap \Gamma_{n,j}$ is a point.
- Farfield, integrals are regular

Typical integral transformations

- Identical element: 8 integrals
- Edge-Edge: 6 integrals
- Vertex-Vertex: 4 integrals

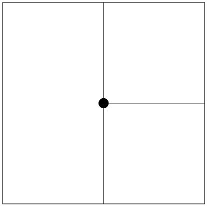
$$\int_{[0,1]^4} K(\vec{x},\vec{y}) L_{k_1}(s_1(1-u_1)) L_{k_2}(s_2(1-u_1u_2)) \\ L_{l_1}(s_1(1-u_1)+u_1) L_{l_2}(s_2(1-u_1u_2)+u_1u_2) \\ u_1(1-u_1)(1-u_1u_2)\, ds_1\, ds_2\, du_1\, du_2$$

Polynomial degree in

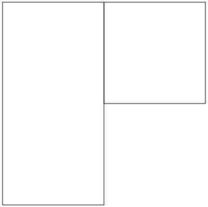
$$\begin{aligned} s_1 &: k_1 + l_1 \\ s_2 &: k_2 + l_2 \\ u_1 &: k_1 + k_2 + l_1 + l_2 + 3 \\ u_2 &: k_2 + l_2 + 1 \end{aligned}$$

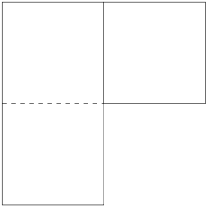
Non-conforming mesh, hanging nodes

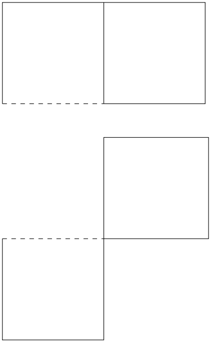
1-irregular mesh

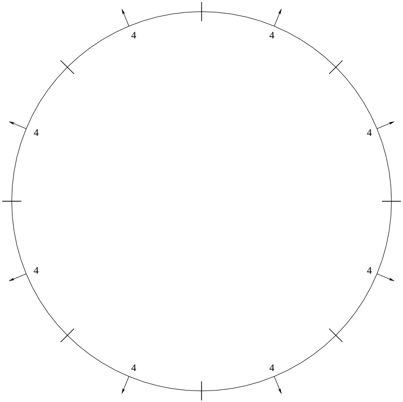


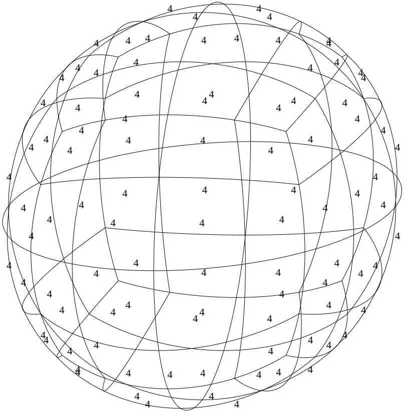
Partitioning for quadrature











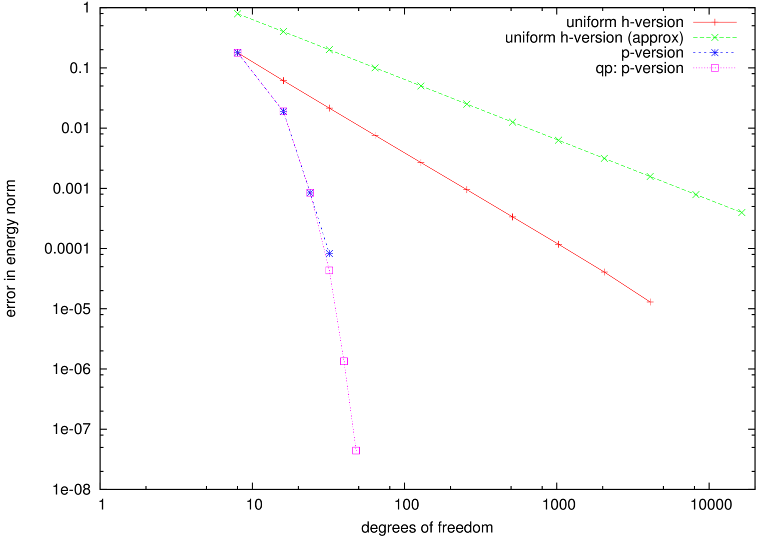
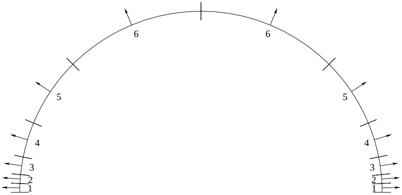
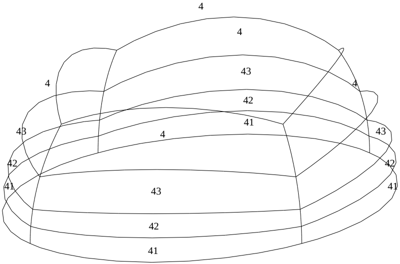


Figure : Convergence on the Circle, $V\psi = x + y$, $\|\psi\|_V = \sqrt{2\pi}$.


6
6
5
5
4
4
3
3
2
2
1
1


4
4
43
4
4
42
41
43
43
4
42
42
41
41
43
42
41

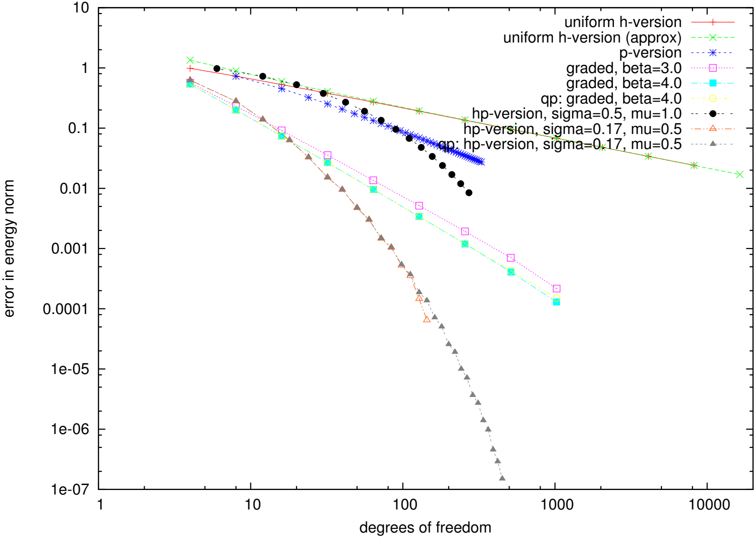


Figure : Convergence on the Half-Circle, $V\psi = 1$.

We compare computations with different optimizations

- Polynomial of degree 20 or 10 for boundary parametrization (n20 or n10)
- Far field optimizations (fopt)
- Vertex-Vertex elements with Duffy or graded mesh only

For the numerical quadrature we use

- a general baseline of 10 quadrature points, i.e. $gp = 10 + p$ for a polynomial of degree p.
- $\sigma = 0.17, \mu = 0.5$ for the graded quadrature with $n = 10$ levels.

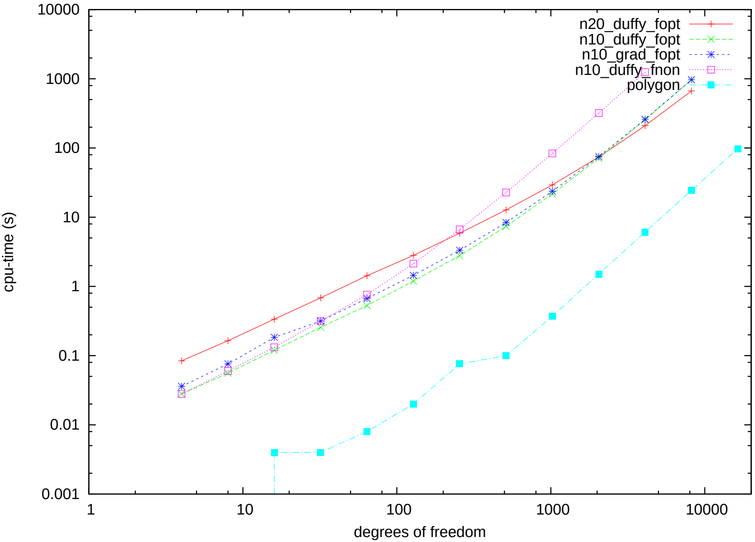


Figure : h-version, time versus dof

Brunel UNIVERSITY LONDON

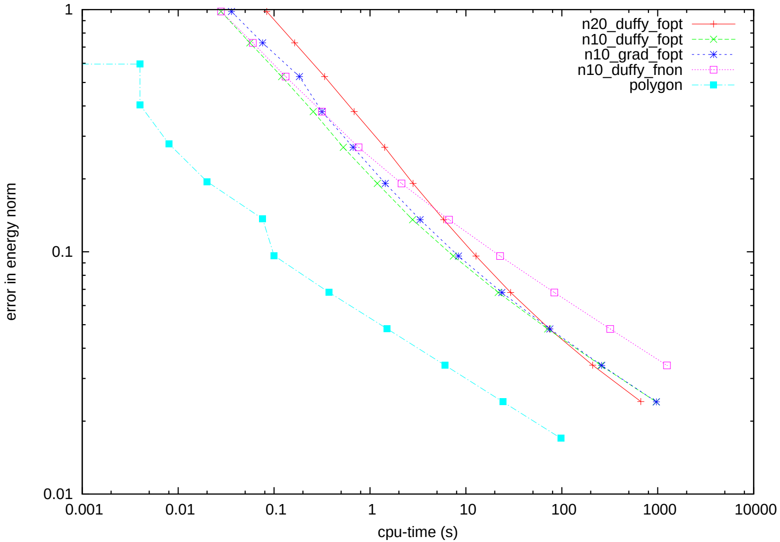


Figure : h-version, error versus time

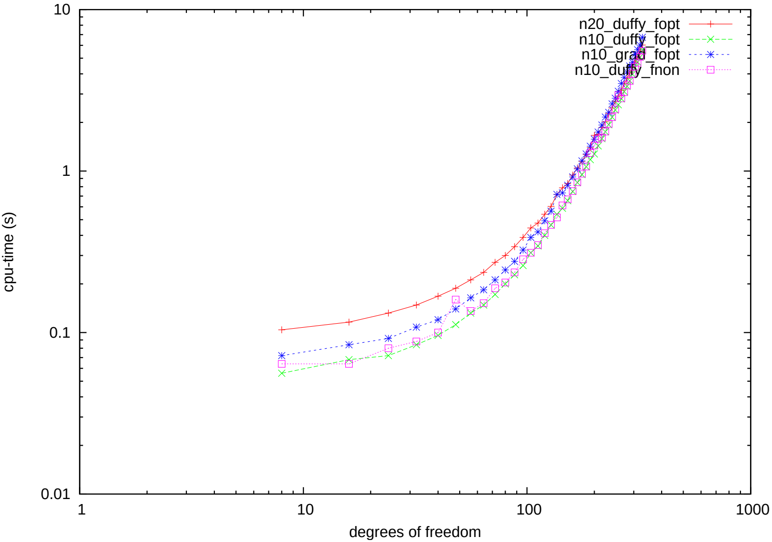


Figure : p-version, time versus dof

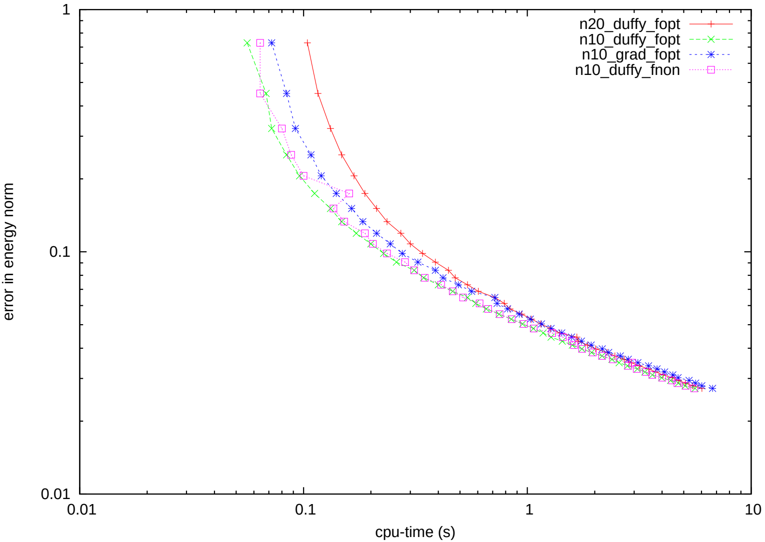


Figure : p-version, error versus time

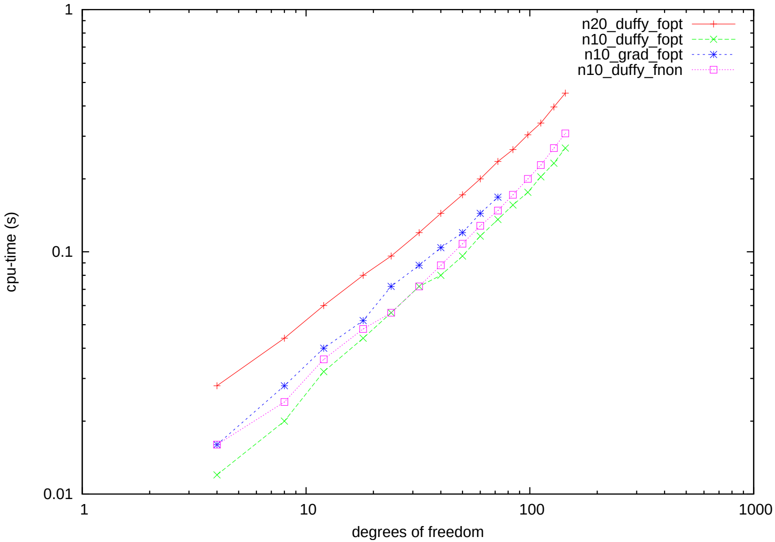


Figure : hp-version (geom mesh, $\sigma = 0.17, \mu = 0.5$), time versus dof

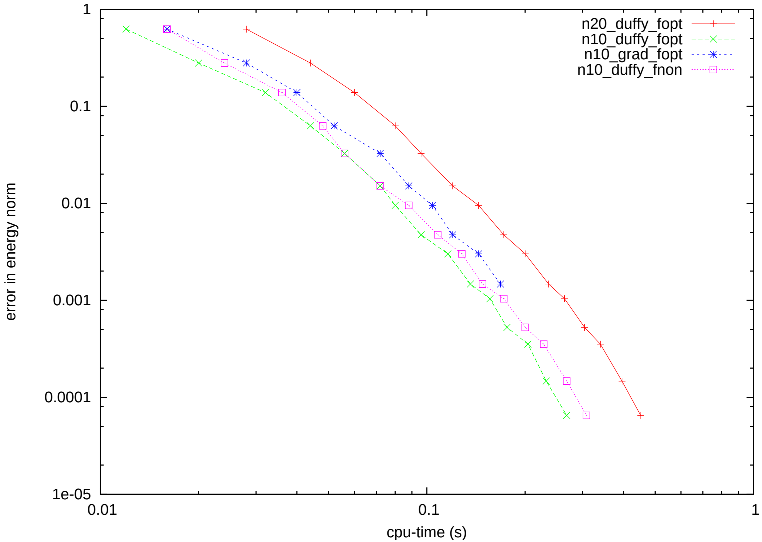


Figure : hp-version (geom mesh, $\sigma = 0.17, \mu = 0.5$), error versus time

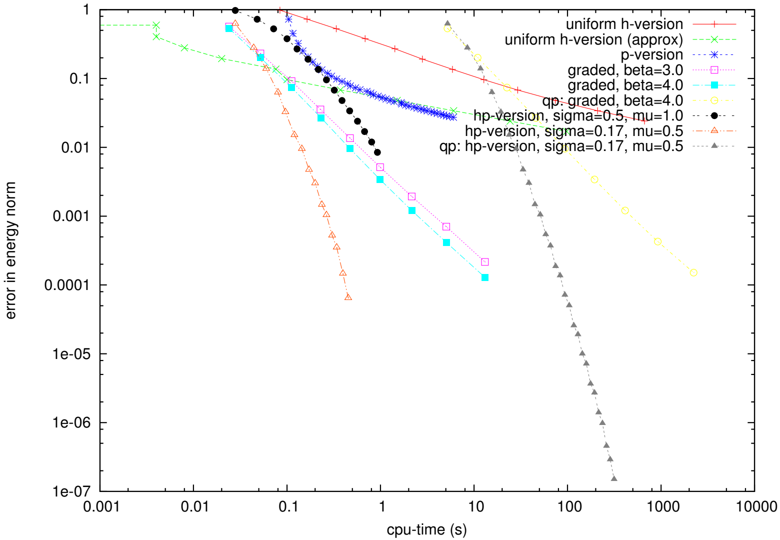


Figure : Complexity on the Half-Circle, $V\psi = 1$, n20_duff_fopt.

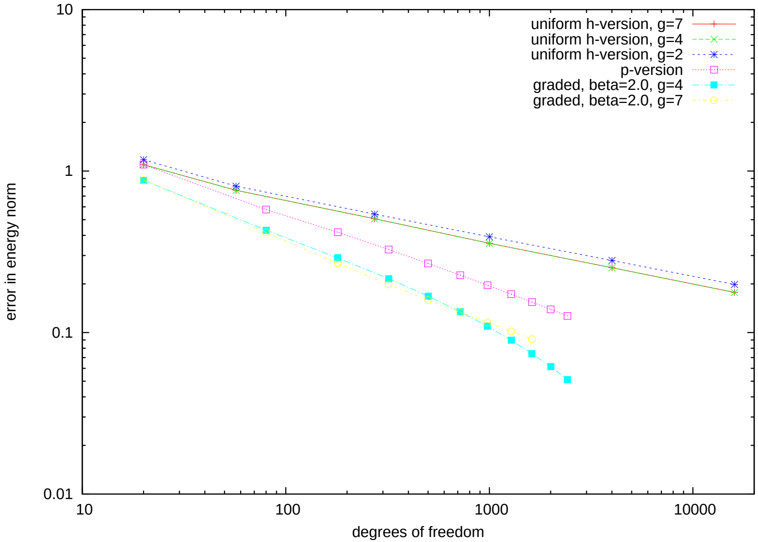


Figure : Convergence on the Half-Sphere, $V\psi = x + y + z$.

We compare computations with different optimizations

- Polynomial of degree 10 for boundary parametrization (n10)
- Far field optimizations (fopt)
- Sum factorization (sopt)

For the numerical quadrature we use

- a general baseline of 4 or 7 quadrature points, i.e. $gp = 4 + p$ or $gp = 7 + p$ for a polynomial of degree p.

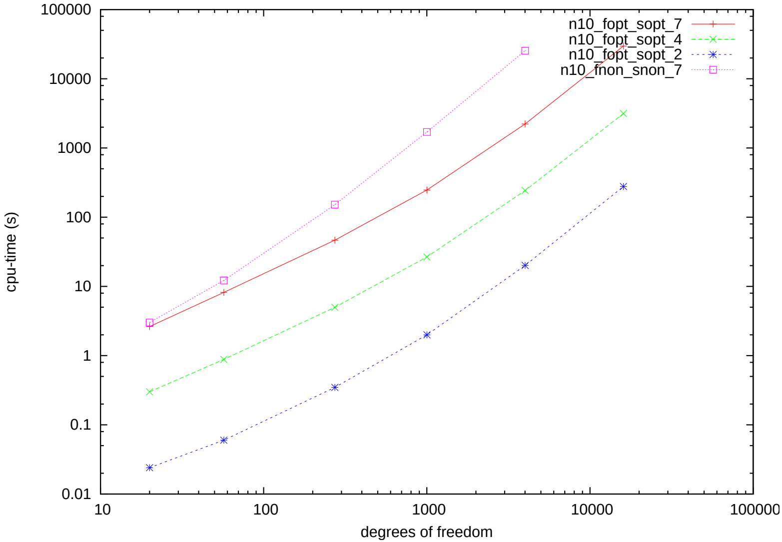


Figure : h-version, time versus dof

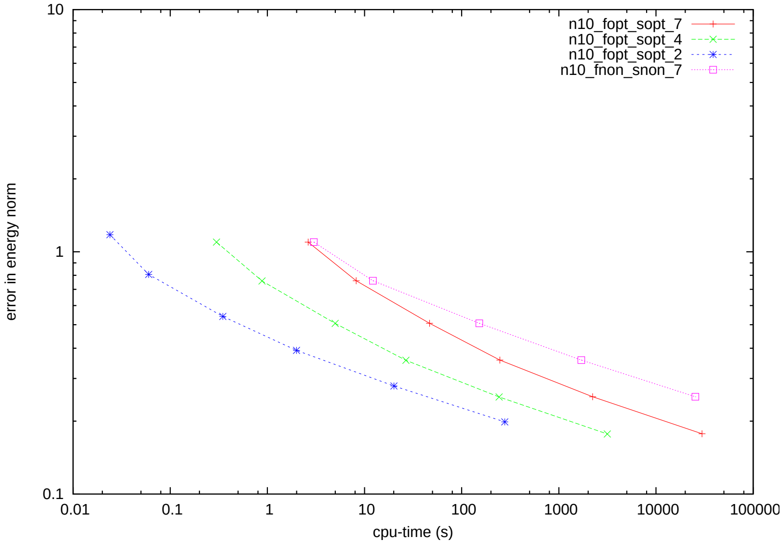


Figure : h-version, error versus time

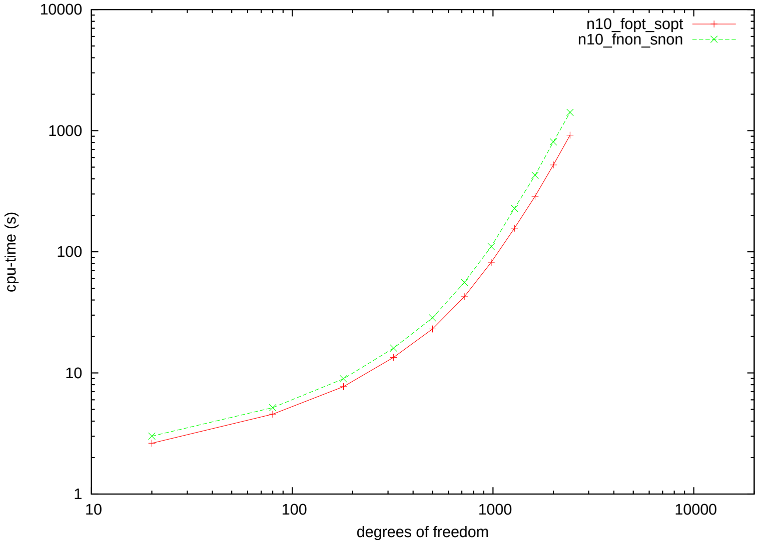


Figure : p-version, time versus dof

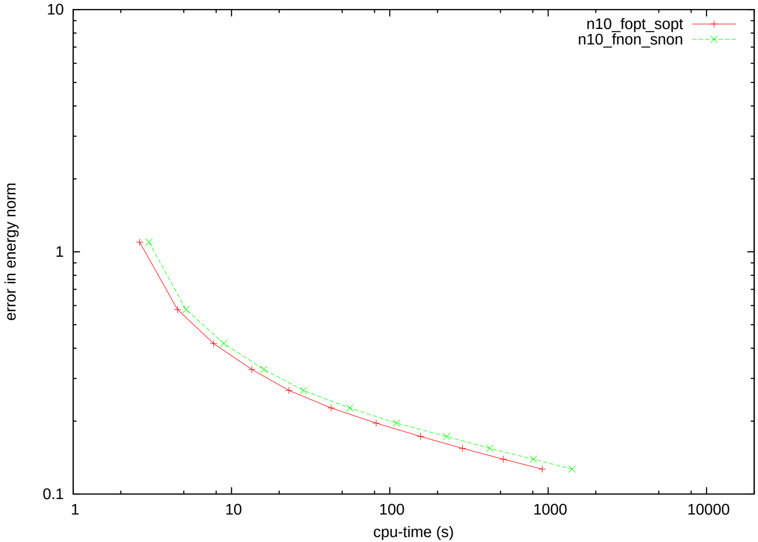


Figure : p-version, error versus time

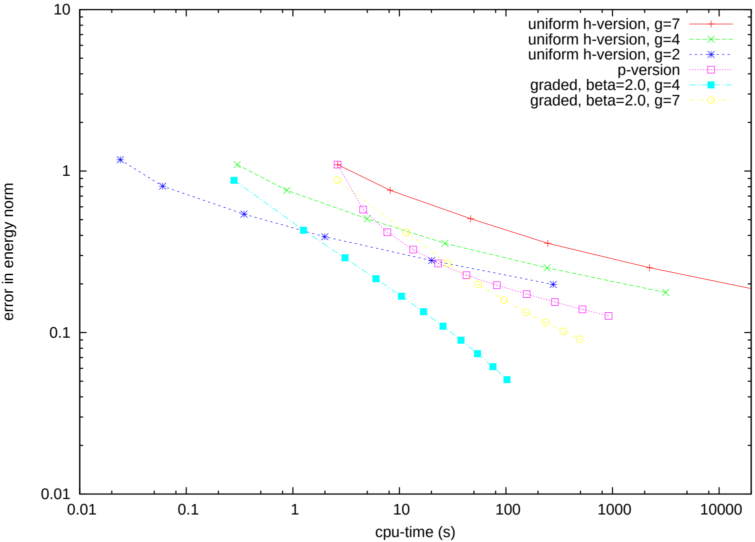
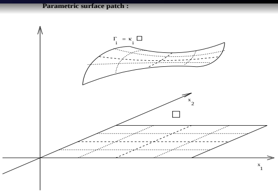


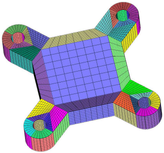
Figure : Complexity on the Half-Sphere, $V\psi = x + y + z$.

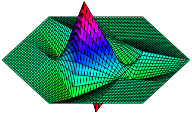
From CAD till wavelet BEM (H. Harbrecht)

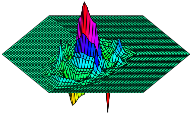
- $\partial\overline{D} = \cup_{i=1}^{M}\Gamma_i, \quad \Gamma_i = \gamma_i(\square), \quad i = 1, \ldots, M$
- $\Gamma_i \cap \Gamma_j$, $i \neq j$, is either empty or a common edge or vertex
- wavelets on manifolds by lifting wavelets from $\square$ to $\partial\overline{D}$ via the parametrization



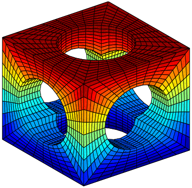
IGES Entity	ID number	IGES-code
Line	110	LINE
Circular arc	100	ARC
Polynomial/rational B-spline curve	126	B_SPLINE
Composite curve	102	CCURVE
Surface of revolution	120	SREV
Tabulated cylinder	122	TCYL
Polynomial/rational B-spline surface	128	SPLSURF
Trimmed parametric surface	144	TRM_SRF
Transformation matrix	124	XFORM

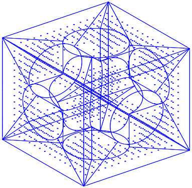






Numerical results for model problem





Indirect formulation with the single layer potential

$$Vu = f \text{ on } \Gamma$$

$$U = Vu \text{ in } \Omega$$

Here: $f = x^2 + 2y^2 - 3z^2$
Computing

$$\|U - u\|_{\infty,\Omega}$$

in discrete points.
Nehalem 5570, 2.93GHz, 48Gbyte, 8 cores.

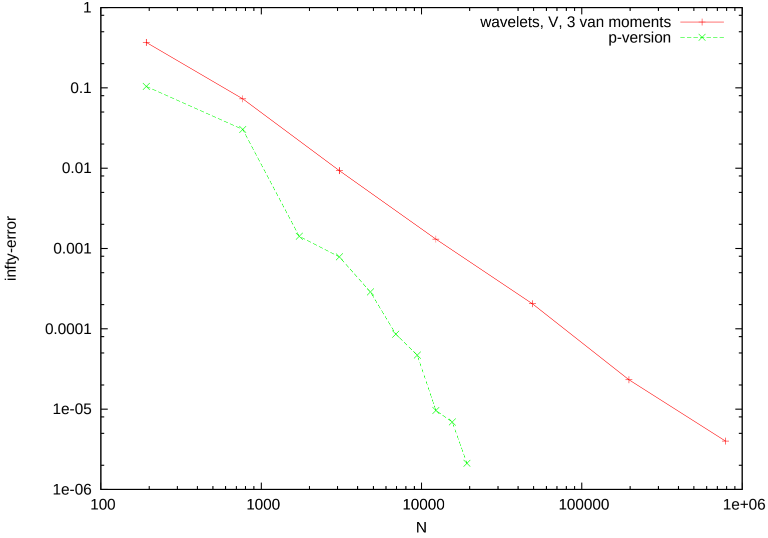


Figure : Convergence on the Toy-Cube.

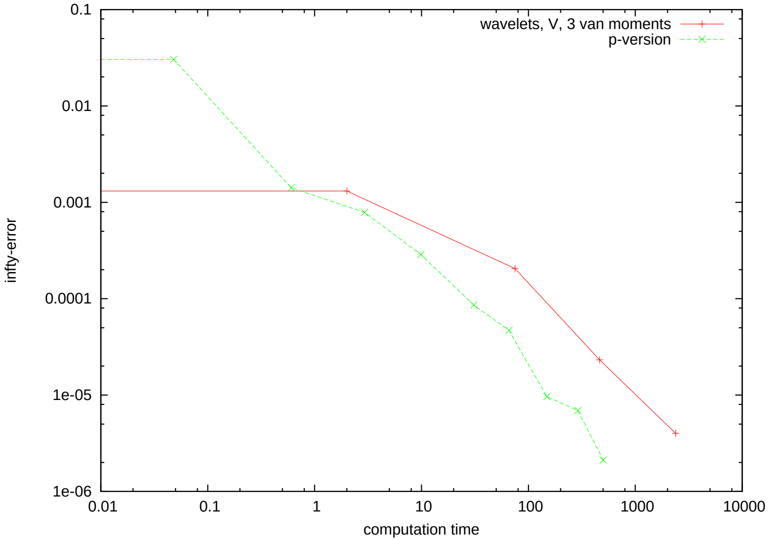


Figure : Complexity on the Toy-Cube.

Conclusion

- hp-method on geometrical refined meshes gives exponential fast convergence on polygonal/polyhedral domains [5, 6].
- Numerical quadrature retains exponential fast convergence on curved surfaces [2, 7], if the surface parametrisation is relatively cheap.
- Here a general surface parametrisation was used, cf. [4].
- Emphasize is given to balancing the computational costs for evaluation of the surface parametrisation
- Examples have been presented, showing exponential fast convergence on complicated geometries.
- Using multiple precision software and/or quadruple precision it is possible to achieve errors of 10^{-7}.

W. Dahmen, H. Harbrecht, and R. Schneider. Compression techniques for integral equations. Asymptotically optimal complexity estimates. SIAM J. Numer. Anal., 43(6):2251-2271, 2006.

S. Erichsen and S. A. Sauter, *Efficient automatic quadrature in 3-d Galerkin BEM*, Computer Methods in Applied Mechanics and Engineering, 157 (1998), pp. 215–224.

H. Harbrecht and M. Randrianarivony. From Computer Aided Design to Wavelet BEM. Bericht 07-18, Kiel, Germany, 2007. (to appear in Comput. Vis. Sci.).

H. Harbrecht and R. Schneider, *Wavelet galerkin schemes for boundary integral equations*, SIAM J. Sci. Comput., 27 (2006), pp. 1347–1370.

N. Heuer, M. Maischak, and E. P. Stephan, *Exponential convergence of the hp-version for the boundary element method on open surfaces*, Numer. Math., 83 (1999), pp. 641–666.

H. Holm, M. Maischak, and E. P. Stephan, *Exponential convergence of the $h-p$ version bem for mixed bvp's on polyhedrons*, Math. Meth. Appl. Sci., 31 (2008), pp. 2069–2093.