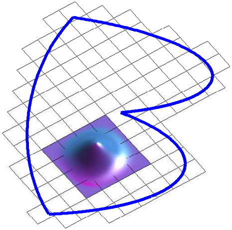
Finite Element Methods with WEB-Splines

Klaus Höllig



http://www.web-spline.de

Oslo, February 15, 2007

Cooperation partners: Prof. Dr. U. Reif, Dr. J. Wipper

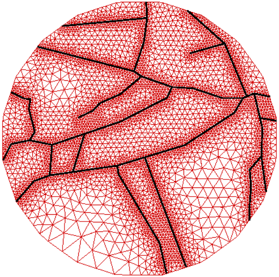
Contributions by:
C. Apprich, M. Boßle, Dr. A. Fuchs, J. Hörner, S. Kreitz, M. Pfeil, B. Russig

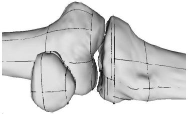
Supported by: TLB, HLRS

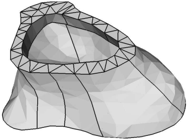
Table of Contents

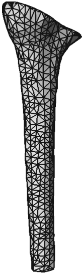
Standard Finite Elements

ART Triangulations (A. Fuchs)



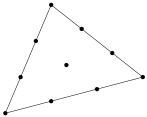






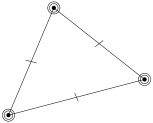
Element types for triangulations

Lagrange



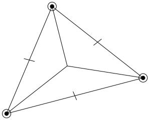
$(3, 0, 10)$

Argyris



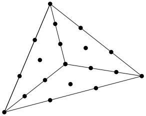
$(5, 1, 21)$

Clough-Tocher



$(3, 1, 12)$

Lagrange



$(3, 0, 20)$

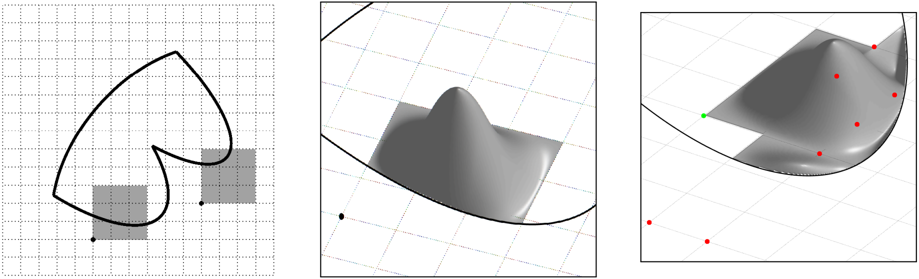
(degree, smoothness, dimension)

Splines: Some Historic Steps

1947	Curry and Schoenberg (B-Splines)
1959/66	de Casteljau, Bézier (Automated Design and Manufacturing)
1967	Ahlberg, Nilson, and Walsh (first book)
197*	de Boor, Schumaker (Numerical Analysis, Approximation)
1979	Dahmen, Michelli (Multivariate Splines)
1980	Oslo algorithm: Cohen, Lyche, and Riesenfeld (Computer Graphics and Image Processing)
1980	Böhm (Computer Aided Geometric Design)

BM-splines, D^m-splines, E-splines, $H^{m,p}$-splines, L-monosplines, L-splines, $Lg-$splines, M-splines, PDL_g splines, PL_{-g} splines, Q-splines, Λ splines, β-splines, ν-splines, ω-splines, τ-splines, $pLg-$splines, A-splines, ARMA splines, B-splines, Bernoulli splines, Box-splines Catmull-Rom splines, Dirichlet splines, Gibbs-Wilbraham splines, Helix splines, Hermite splines, Hermite-Birkhoff splines, Inf-convolution splines, K-splines, LB-splines, Lagrange splines, Legendre splines, NBV-splines, NURBS, ODR splines, Powell-Sabin splines, Schoenberg splines, Tschebyscheff splines, VP-splines, Wilson-Fowler splines, X-splines, box-splines, cardinal splines, discrete splines, elliptic splines, exponential Euler splines, exponential box splines, fundamental splines, g-splines, harmonic splines, histosplines, hyperbolic splines, metaharmonic splines, minimal-energy splines, monosplines, natural splines, perfect splines, polyharmonic splines, pseudo splines, simplex splines, smoothing splines, super splines, thin-plate splines, triangular splines, trigonometric splines, TURBS, v-splines, variation diminishing splines, vertex splines, web-splines, . . .

B-Splines as Finite Elements



homogeneous boundary condition via weight function

$$b_k \longrightarrow \quad w\, b_k$$

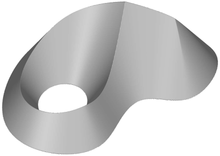
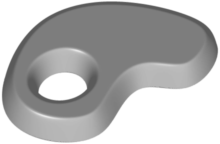
stability via extension

$$b_i \longrightarrow \quad b_i + \sum_{j \in J(i)} e_{i,j} b_j$$

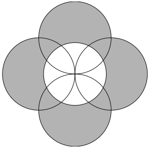
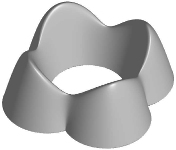
Weight Function

$$w(x) \asymp \operatorname{dist}(x, \partial D)\,, \qquad \text{smooth on } D$$

- Smoothed distance function: $w(x) = 1 - \max\left(0, 1 - d(x, \partial D)/\delta\right)^{\ell}$

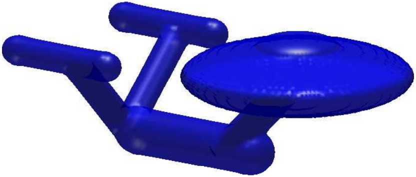
- R-functions: $r_c(w) = -w$, $r_{\cup/\cap}(w_1, w_2) = w_1 + w_2 \pm \sqrt{w_1^2 + w_2^2}$

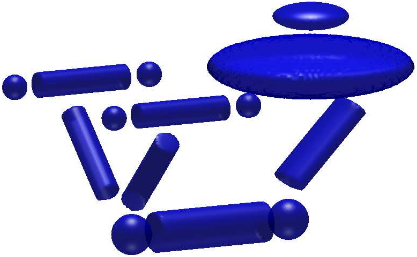




- Spline approximation
- Implicit representation of ∂D:

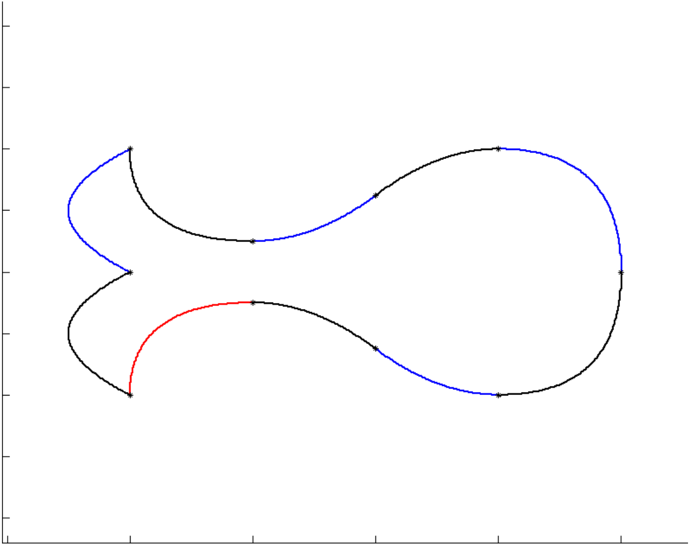
$$w(x) = 1 - x_1^2 - x_2^2 \quad \text{for} \quad D:\ |x| < 1$$

R-Functions in 3D

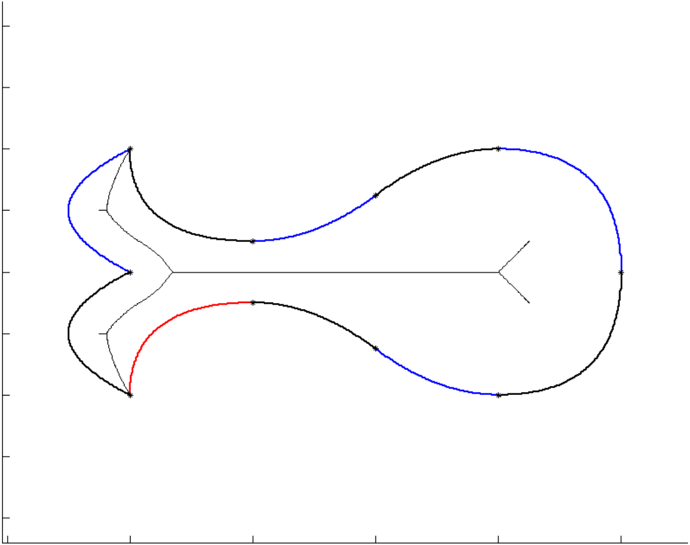




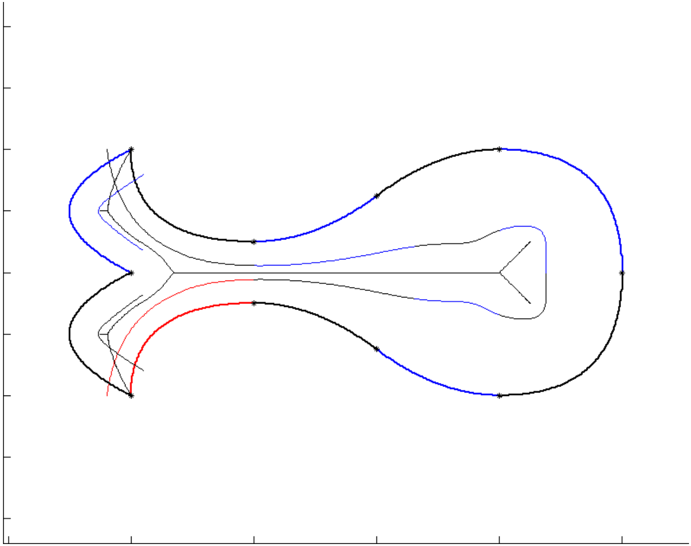
Weight Function with Variable Offset



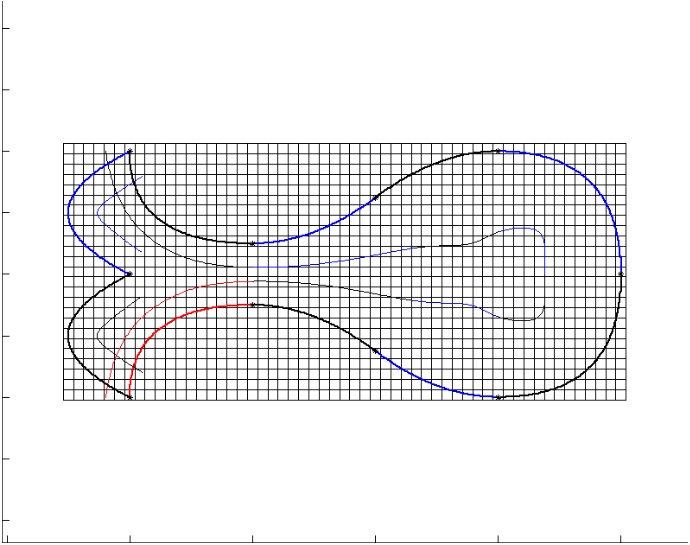
Weight Function with Variable Offset



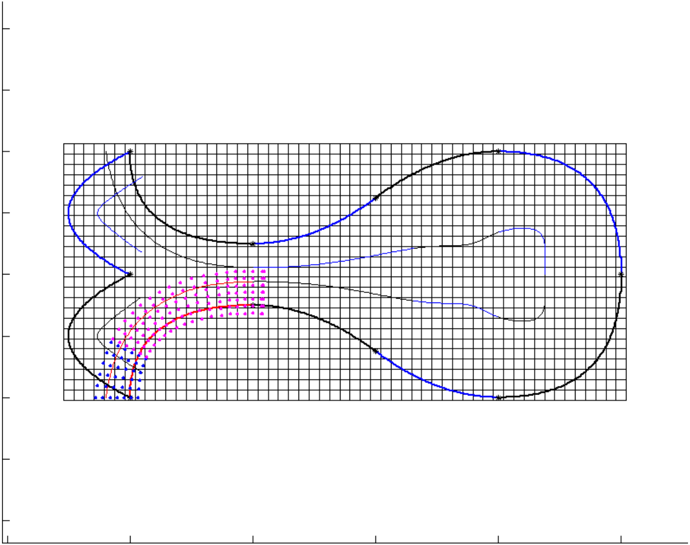
Weight Function with Variable Offset



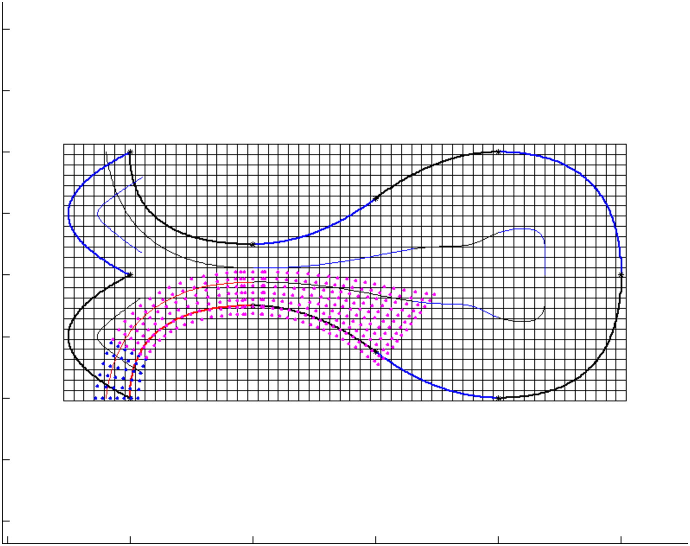
Weight Function with Variable Offset



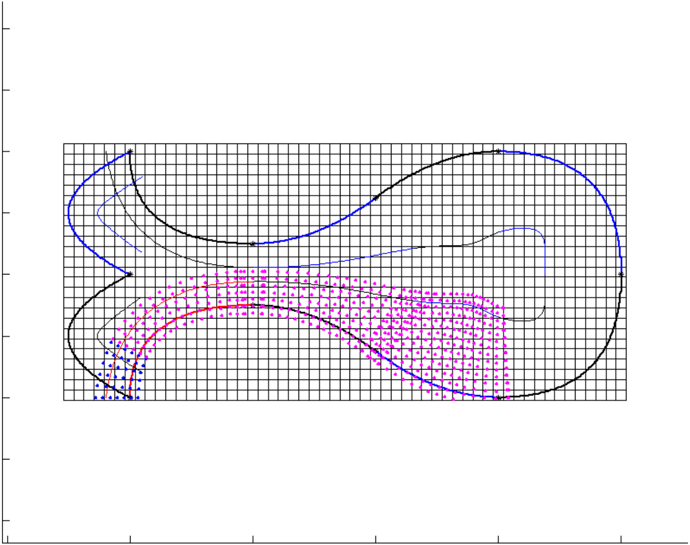
Weight Function with Variable Offset



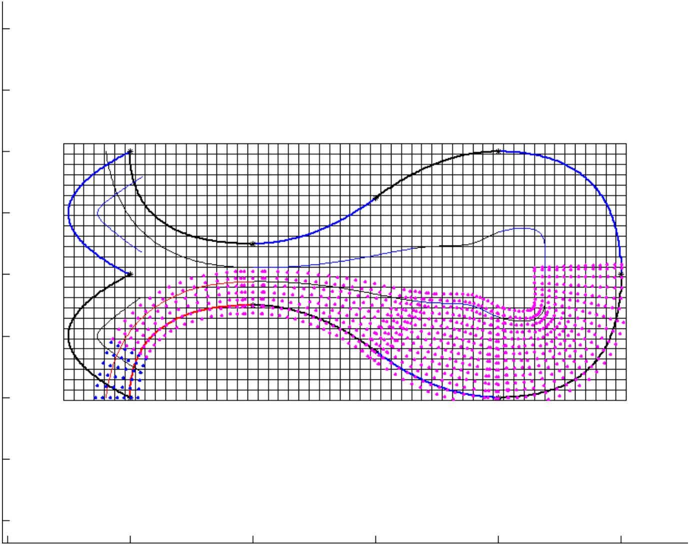
Weight Function with Variable Offset



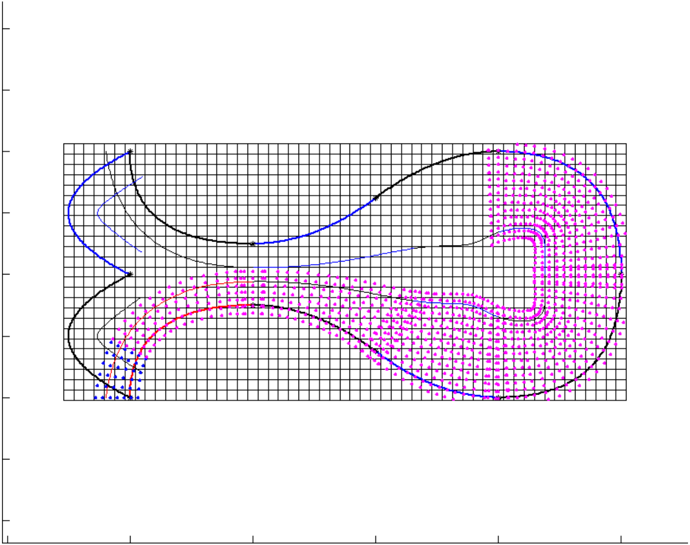
Weight Function with Variable Offset



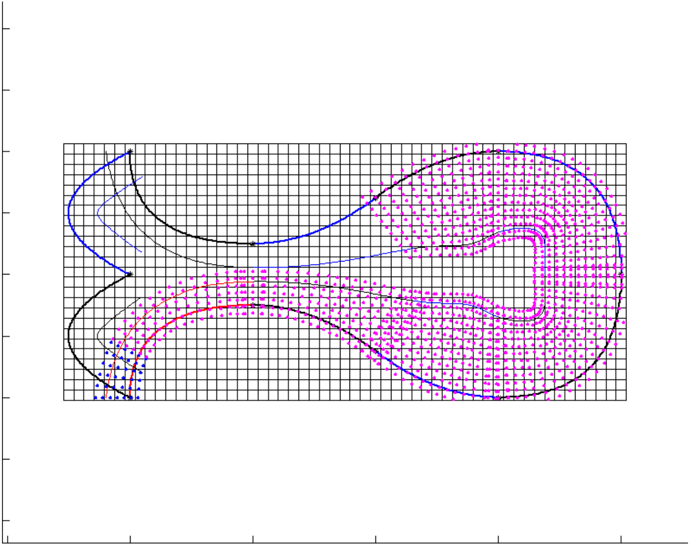
Weight Function with Variable Offset



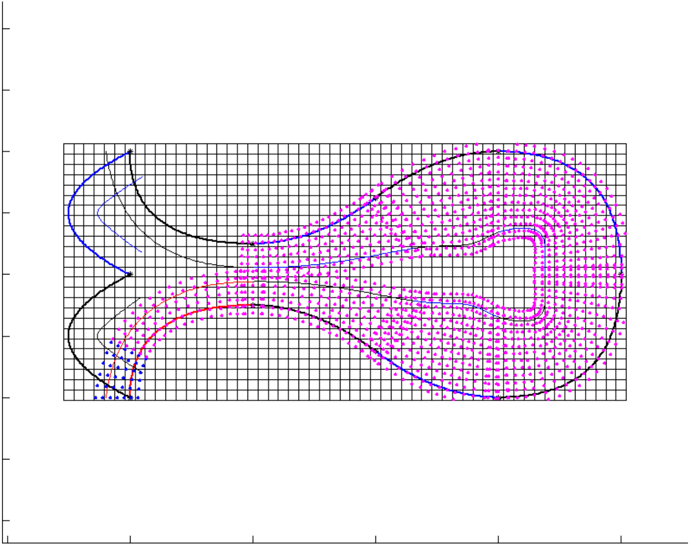
Weight Function with Variable Offset



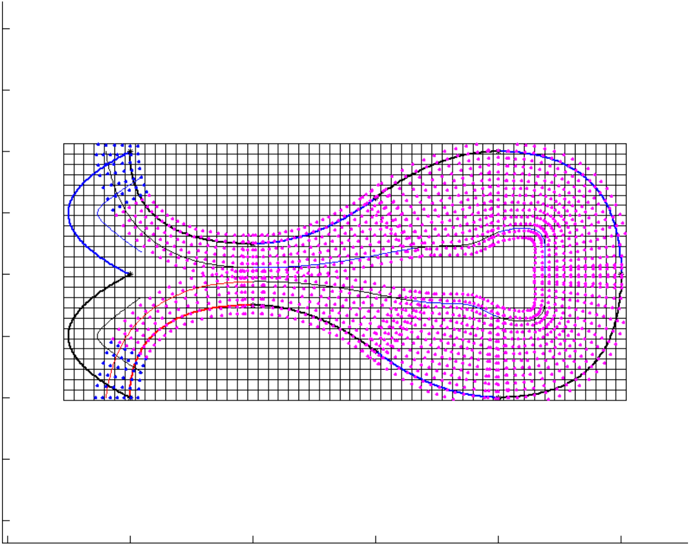
Weight Function with Variable Offset



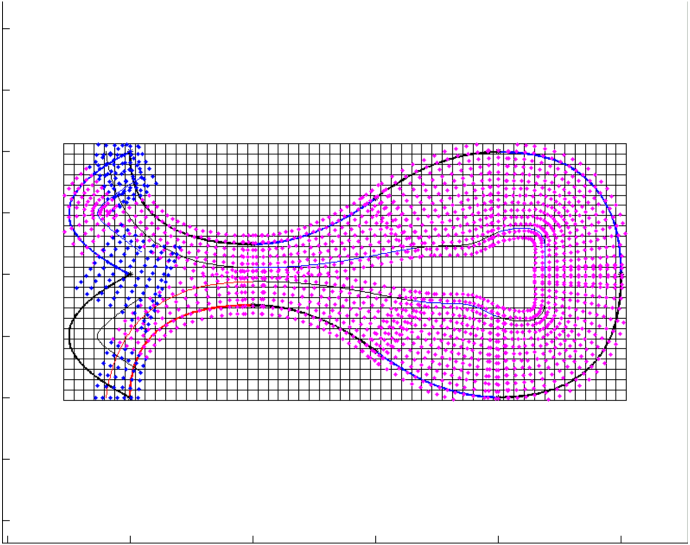
Weight Function with Variable Offset



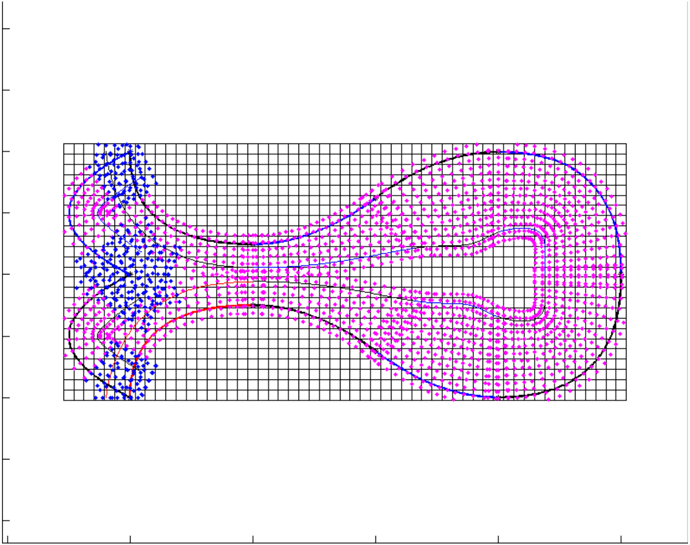
Weight Function with Variable Offset



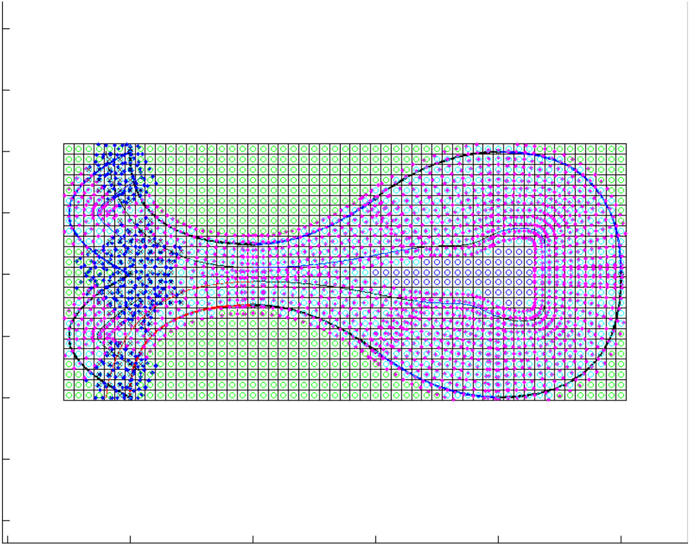
Weight Function with Variable Offset



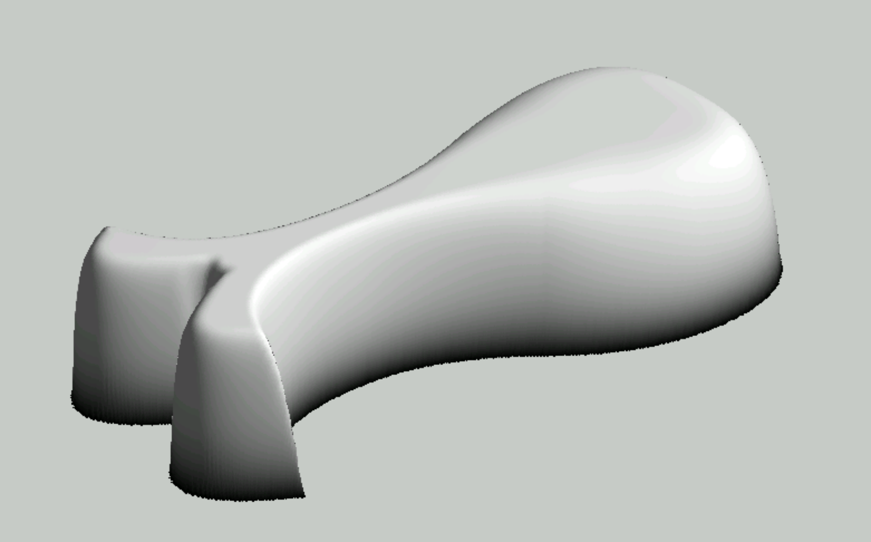
Weight Function with Variable Offset



Weight Function with Variable Offset



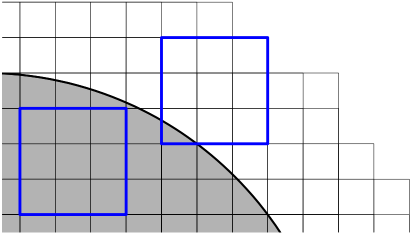
Weight Function with Variable Offset



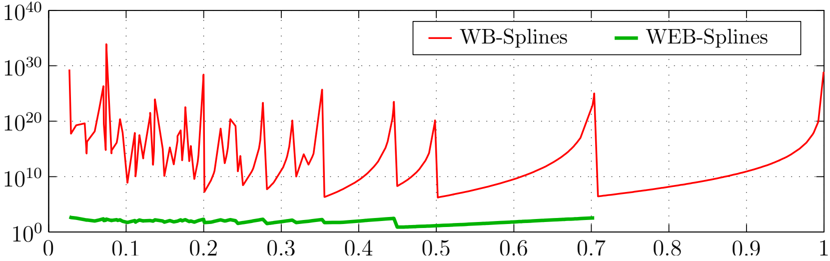
Instability of weighted B-Splines

Trial functions $wb_k \Longrightarrow$

- unstable basis
- slow convergence of solvers
- inaccurate results

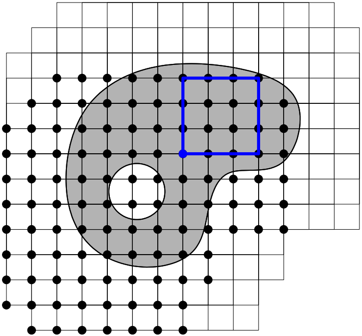


cond G_h for Ritz-Galerkin approximation of Poisson's problem on the unit disc:



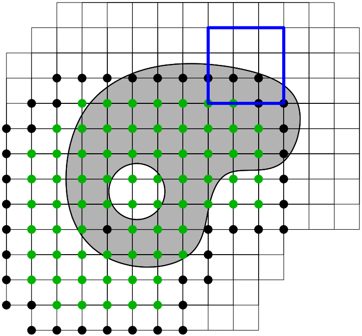
Classification of B-Splines

- Cell types: inner / boundary / outer



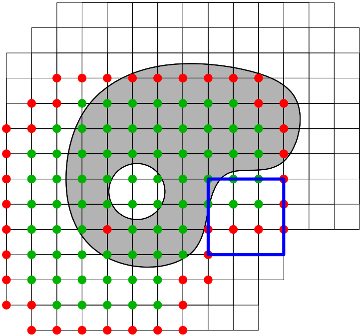
Classification of B-Splines

- Cell types: inner / boundary / outer
- Inner B-splines:
  supp b_k contains inner cell



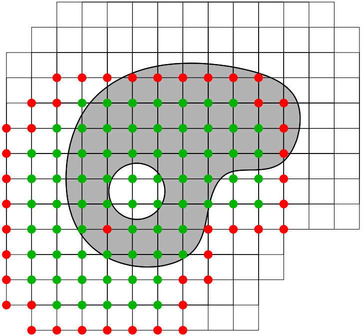
Classification of B-Splines

- Cell types: inner / boundary / outer
- Inner B-splines:
  supp b_k contains inner cell
- Outer B-splines:
  supp b_k contains boundary cell
  (and no inner cell)



Classification of B-Splines

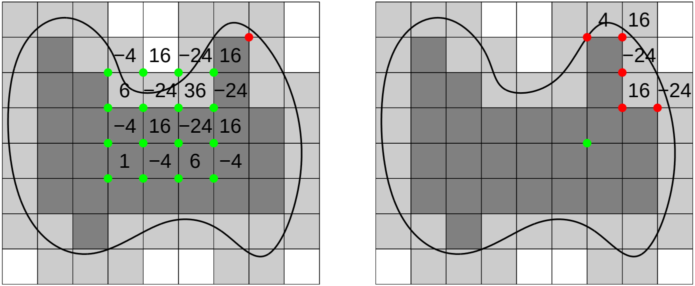
- Cell types: inner / boundary / outer
- Inner B-splines:
  supp b_k contains inner cell
- Outer B-splines:
  supp b_k contains boundary cell
  (and no inner cell)
- Relevant indices: $K = I \cup J$



Extension

Coupling outer with inner B-splines

$$B_i = b_i + \sum_{j \in J(i)} e_{i,j} b_j, \quad e_{i,j} = \prod_{\nu=1}^{m} \prod_{\substack{\mu=\ell_\nu \\ \mu \neq i_\nu}}^{\ell_\nu+n-1} \frac{j_\nu - \mu}{i_\nu - \mu}$$



$I(j)$: nearest $(n+1)^m$-array of inner indices

$J(i)$: dual sets $(i \in I(j))$

Extension Coefficients – Derivation

Marsden's Identity

$$p(x) = \sum_K q(k) b_k(x) = \sum_I q(i) b_i(x) + \sum_J q(j) b_j(x)$$

Lagrange interpolation

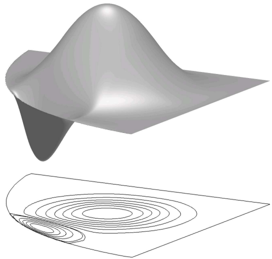
$$q(j) = \sum_{i \in I(j)} q(i) \underbrace{L_i^j(\xi)\Big|_{\xi=j}}_{e_{i,j}}$$

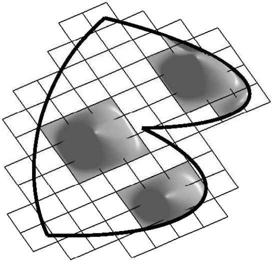
Complementary sets $\quad i \in I(j) \quad \Leftrightarrow \quad j \in J(i)$

$$\sum_I q(i) b_i(x) + \sum_J q(j) b_j(x) = \sum_I q(i) \left\{ b_i + \sum_{j \in J(i)} e_{i,j} b_j(x) \right\}$$

Weighted extended B-Splines

$$B_i = \frac{w}{w(x_i)}\left(b_i + \sum_{j\in J(i)} e_{i,j} b_j\right), \qquad i \in I$$





- piecewise polynomial, coordinate degree n
- local support, width $\asymp h$
- uniformly stable

Approximation with WEB-Splines

- Weighted dual functions

$$\langle \Lambda_\ell, B_i \rangle = \delta_{\ell,i}, \qquad \Lambda_\ell = \frac{w(x_\ell)}{w}\lambda_\ell$$

- Stability

$$\|C\| \asymp h^{-m/2} \Big\| \sum_{i \in I} c_i B_i \Big\|_0$$

- Optimal order

$$\|u - u_h\|_\ell \preceq h^{n+1-\ell} \|u\|_{n+1}, \quad u = 0 \text{ on } \partial D,$$

with

$$u_h = \sum_i \langle \Lambda_i, u \rangle B_i \,.$$

Boundary Regularity

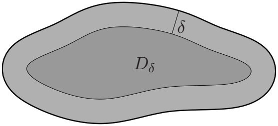
Factoring the error

$$u - u_h = w\Big(\underbrace{\frac{u}{w}}_{v} - \sum c_k b_k\Big)$$

Regularity of quotients

$$\|v\|_{2,D_\delta} \preceq \delta^{-1}\left(\|u\|_{2,D} + \|v\|_{2,D}\right)$$

$$\|v\|_{1,D} \preceq \|u\|_{2,D}$$



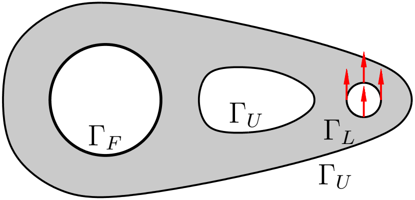
Test case

$$u(x) = x^{3/2+\varepsilon},\ w(x) = x,\ v(x) = x^{1/2+\varepsilon}, \quad x \in [0,1]$$

$\longrightarrow$ u just in H^2, v just in H^1

Linear Elasticity (Crank Arm)

$$\begin{aligned} -\operatorname{div}\sigma(u) &= f && \text{in } D, \\ u &= 0 && \text{on } \Gamma_F, \\ \sigma(u)n &= g && \text{on } \Gamma_L \end{aligned}$$

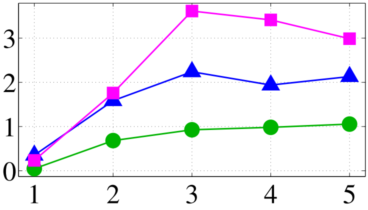


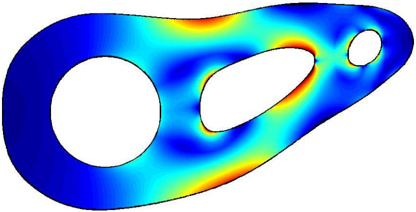
Weak formulation: $\forall v \in \left(H^1_{\Gamma_F}(D)\right)^3$

$$\int_D \sigma(u) : \varepsilon(v) = \int_D f \cdot v + \int_{\Gamma_L} g \cdot v$$

$h_i = 1.4 \cdot 2^{-i},\ i \in \{1, \dots, 6\}$
$n \in \{3, 4, 5\}$
$w(x) = x_1^2 + x_2^2 - 4$

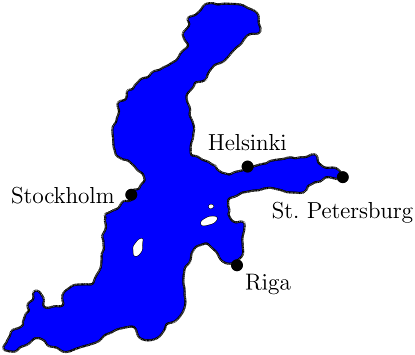
Rate of residual norm $\|\operatorname{div}\sigma(u) - f\|_0$





Water Waves

- connected domain $D \subset \mathbb{R}^2$
- water depth $H(x, y)$
- vertical, perfectly reflecting wall



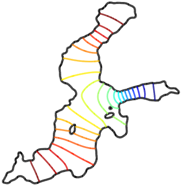
$$
\begin{aligned}
\operatorname{div}(H \operatorname{grad} u) &= \lambda u && \text{in } D \\
\partial_\perp u &= 0 && \text{on } \partial D
\end{aligned}
$$

Weak formulation:

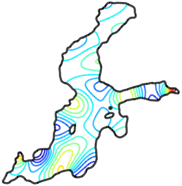
$$\int_D \operatorname{grad} v \, H \, \operatorname{grad} u = \lambda \int_D uv \qquad \forall v \in H^1(D)$$

Water Waves

first eigenfunction



second eigenfunction



20th eigenfunction

30th eigenfunction

Eigenvalue and eigenfunction errors

$$\left|\lambda_\ell^h - \lambda_\ell\right|,\ \left\|u_\ell^h - u_\ell\right\|_0 \preceq h^{2n}\lambda_\ell^2$$

Deformation of Thin Shells

Weak formulation of Koiters model

$$\forall v \in \left(H_0^1(D)\right)^2 \times H_0^2(D): \quad \int_D E^{ijkl}\left(\gamma_{ij}(u)\gamma_{kl}(v) + \frac{e^2}{12}\varrho_{ij}(u)\varrho_{kl}(v)\right) = \int_D f\, v$$

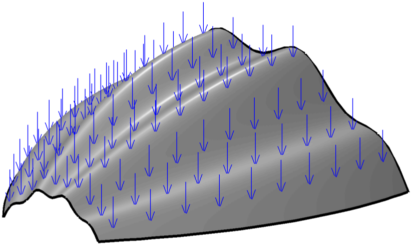
with

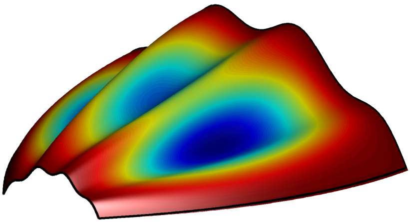
Strain tensor (corresponds to membrane forces)

$$\gamma_{ij}(u) = \tfrac{1}{2}\left(u_{i|j} + u_{j|i}\right) - b_{ij}u_3$$

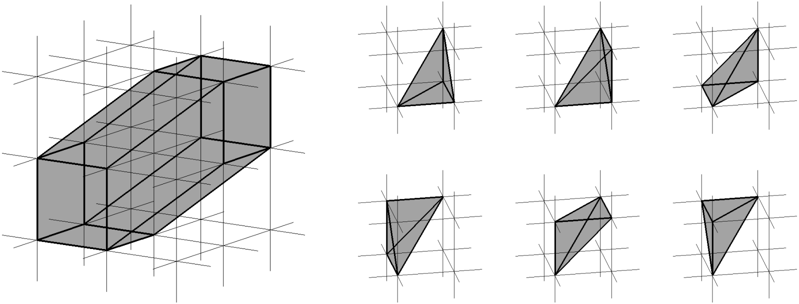
Change of curvature tensor (corresponds to bending forces)

$$\rho_{ij}(u) = u_{3|ij} - b_i^k b_{kj} u_3 + b_{i|j}^k u_k + b_i^k u_{k|j} + b_j^k u_{k|i}$$





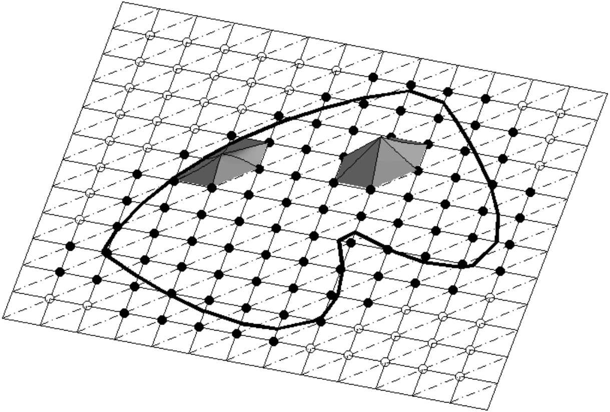
Linear Box-Splines



$$\Xi = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}, \quad B_\Xi(x) = \int_0^1 \chi_{[0,1]^3}\,(x - t(1,1,1))\,dt$$

Lagrange basis $\qquad b_k^h(x) = B_\Xi\,(x/h - k)\,, \quad k \in \mathbb{Z}^3$

Finite Element Basis



domain $\quad D \approx D^h : \quad w^h(x) > 0$

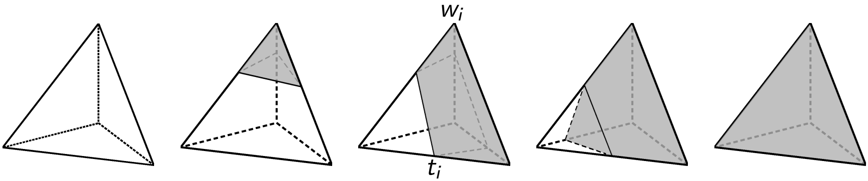
weighted B-splines $\quad w^h b_k^h , \quad k_\nu \in \mathbb{Z}$

Numerical Integration

Ritz-Galerkin integrals

$$\int_{S \cap D^h} a\left(w^h b_k^h, w^h b_{k'}^h\right) = I\left(w_S^h, k, k'\right)$$

intersection patterns



Poisson bilinear form

$$I \ : \text{rational function of } w_S^h \text{ or } t_S^h$$

(automatically generated, $6 * 16 * 10$ cases)

Expressions for Ritz-Galerkin Integrals

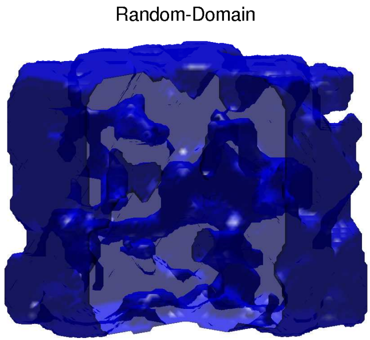
no intersection, full box (6 tetrahedra), common factor $1/120$:

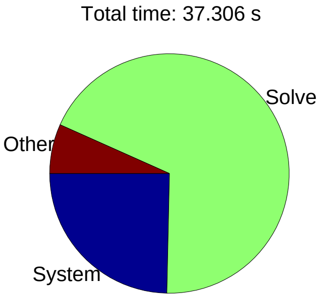
$$\begin{aligned}
&24\, w_8{}^2 - 8\, w_1\, w_3 + 48\, w_1{}^2 + 8\, w_5{}^2 + 24\, w_1\, w_8 + 8\, w_1\, w_6 \\
&-4\, w_5\, w_6 - 8\, w_1\, w_5 - 4\, w_8\, w_6 - 4\, w_3\, w_4 + 12\, w_6{}^2 - 4\, w_5\, w_7 \\
&+8\, w_1\, w_7 + 8\, w_3{}^2 - 4\, w_4\, w_8 + 12\, w_4{}^2 - 4\, w_6\, w_2 + 8\, w_1\, w_4 \\
&+12\, w_7{}^2 + 8\, w_2{}^2 - 8\, w_2\, w_1 - 4\, w_3\, w_7 - 4\, w_2\, w_4 - 4\, w_7\, w_8
\end{aligned}$$

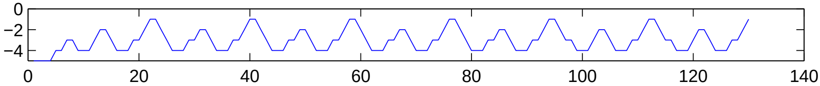
intersection pattern 1, one tetrahedron, common factor $w_1^2/60$:

$$\begin{aligned}
&2\,\frac{t_1\, t_3{}^3}{t_2} + 5\, t_3\, t_2 + 10\, t_3{}^2 + 20\,\frac{t_1\, t_3}{t_2} + 10\, t_1{}^2 - 2\, t_1{}^3 - 5\,\frac{t_1\, t_2{}^2}{t_3} - 5\,\frac{t_1{}^2 t_2}{t_3} - 20\, t_1 - t_3\, t_2{}^2 \\
&-2\, t_1{}^2 t_3 + 2\,\frac{t_1{}^2 t_3{}^2}{t_2} - 10\,\frac{t_3\, t_2{}^2}{t_1} - t_1\, t_2{}^2 + 20\,\frac{t_3\, t_2}{t_1} + \frac{t_1\, t_2{}^3}{t_3} + \frac{t_1{}^3 t_2}{t_3} + 5\, t_1\, t_2 + 10\,\frac{t_1\, t_2}{t_3} \\
&+\frac{t_1{}^2 t_2{}^2}{t_3} - t_1{}^2 t_2 + t_1\, t_3\, t_2 + 2\,\frac{t_3{}^2 t_2{}^2}{t_1} - 2\, t_3{}^3 + 2\,\frac{t_1{}^3 t_3}{t_2} + 10\, t_1\, t_3 - 10\,\frac{t_1\, t_3{}^2}{t_2} \\
&-10\,\frac{t_1{}^2 t_3}{t_2} - 20\, t_3 - 2\, t_1\, t_3{}^2 - t_3{}^2 t_2 + 2\,\frac{t_3{}^3 t_2}{t_1} + 2\,\frac{t_3\, t_2{}^3}{t_1} - 10\,\frac{t_3{}^2 t_2}{t_1}
\end{aligned}$$

Implementation with Linear Elements

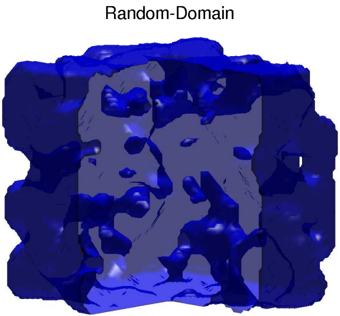


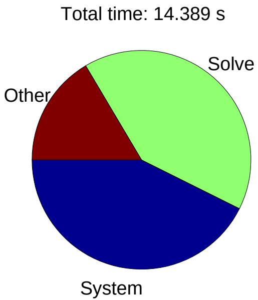


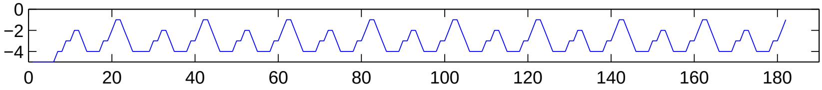


- Athlon 64 X2 4600+, 2.4 GHz (one CPU)
- 3,048,625 unknowns, relative residuum 3.1E-08
- 5 grids, 17.1 fine grid iterations, 3 SSOR smoothing steps, $\omega = 1.2$

Implementation with Linear Elements (vectorized)







- NEC SX 8, 2.0 GHz (one CPU, vector length 256)
- 3,048,625 unknowns, relative residuum 3.8E-08
- 5 grids, 22.343 fine grid iterations, 10 Jacobi smoothing steps, $\omega = 0.88$
- Ratio for Solving: 1:7, Ratio for System: 1:2

Program Demo

- Domains: unit disc with holes or random domain
- Programming language: FORTRAN 90
- MATLAB interface for visualization

disc, f constant	2 holes	4 holes
disc, f exponential function	2 holes	4 holes
random domain	f constant	f exponential function

Current Projects

- Fluid flow problems (U. Reif, B. Mößner)
- MIND: multiple integration over NURBS domains (J. Hörner)
- Parallelization (J. Hörner)
- Shells (M. Boßle)
- Singularities (C. Apprich)
- CSG–Models (s. Kreitz)
- Adaptive approximation (M. Mustahsan)

AND

www.mathematics-online.org

Advantages of the WEB-Method

- No grid generation
- Natural integration in CAD/CAM-systems based on tensor product B-splines
- Simple implementation and short computing times
- Approximations of arbitrary order of accuracy by appropriate choice of the degree of the basis functions
- Low dimensional approximation spaces
- Exact fulfillment of boundary conditions
- Well suited for multigrid methods and hierarchical refinement
- Natural parallelization of algorithms