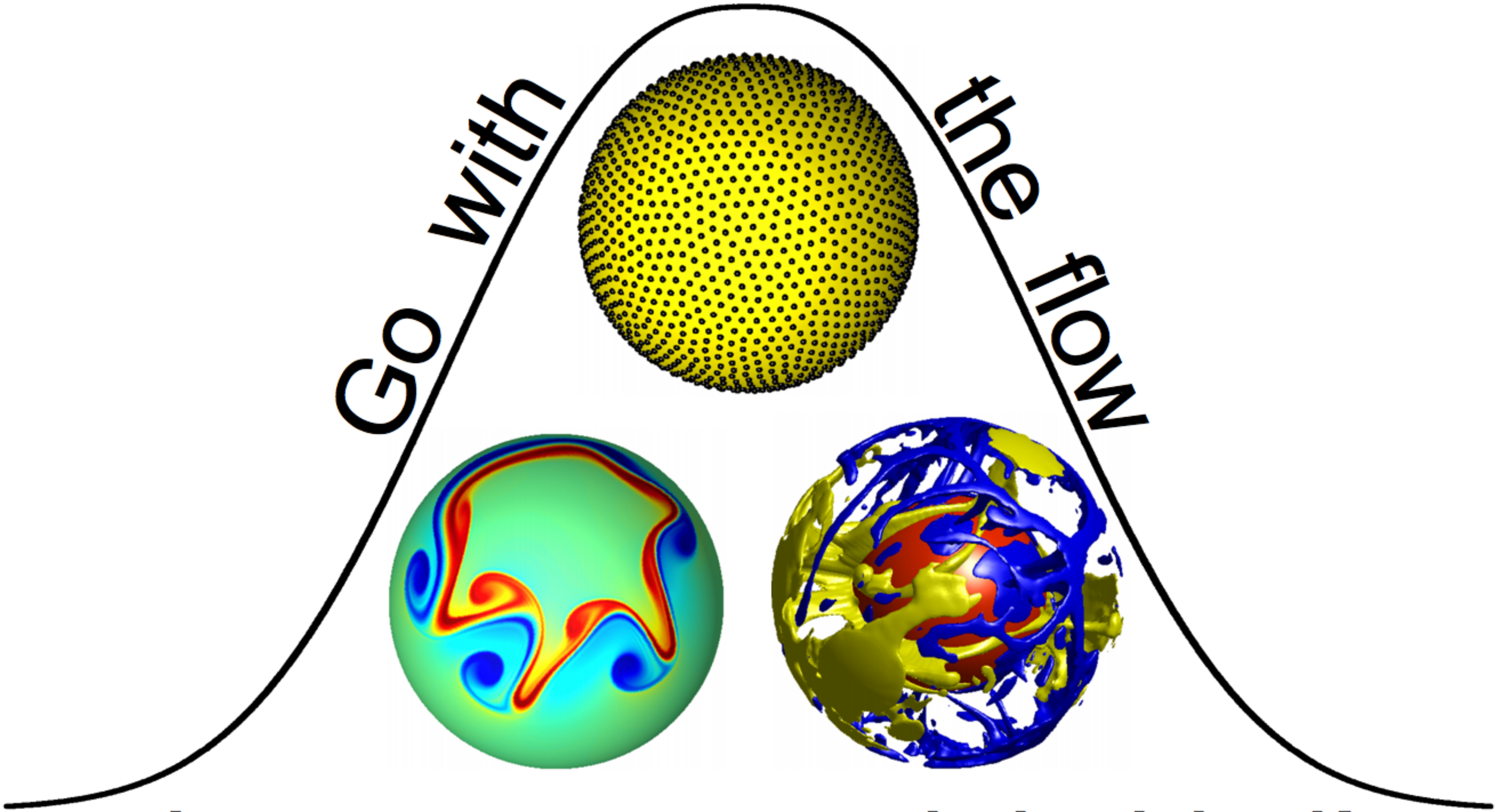


Leave your mesh behind!

Approximation on Surfaces with Kernels: Recent Developments and Applications

Grady B. Wright
Boise State University
gradywright@boisestate.edu

Acknowledgements

Collaborators:

- Ed Fuselier, High Point University
- Thomas Hangelbroek, University of Hawaii
- Fran Narcowich, Texas A&M
- Joe Ward, Texas A&M

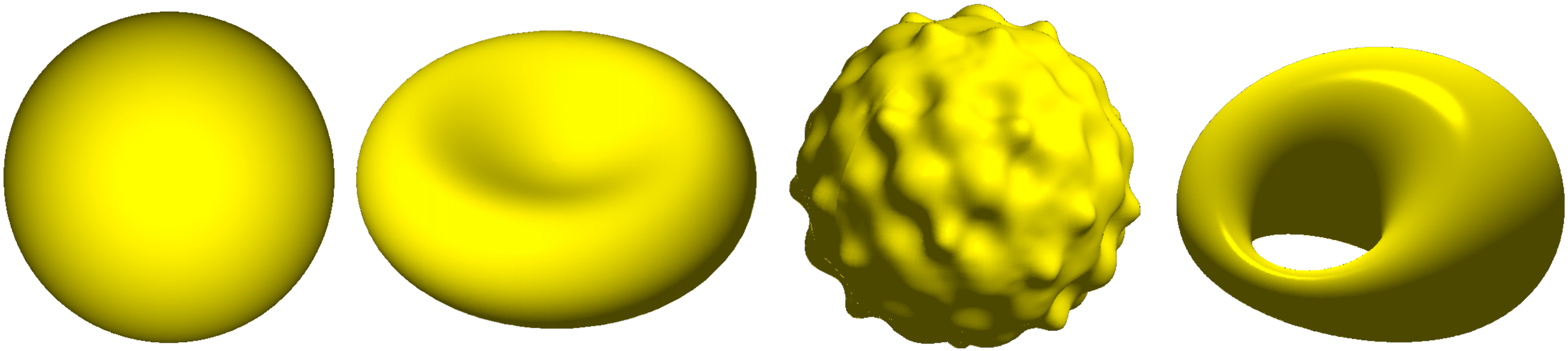
Special thanks:

- Natasha Flyer, National Center for Atmospheric Research, Boulder
- Doug Hardin and Ed Saff, Vanderbilt University

Extra special thanks: National Science Foundation

- ATM-0620090, DMS-0934581, and DMS 0540779

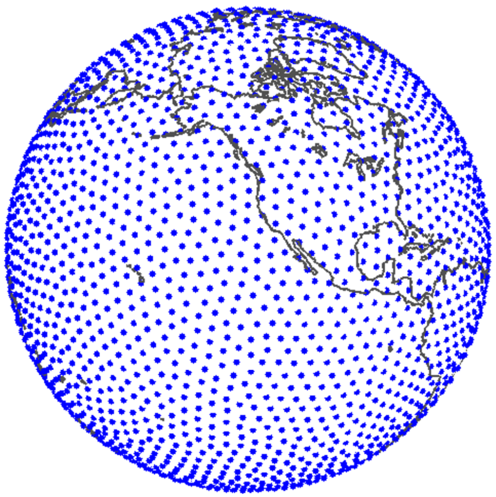
Overview

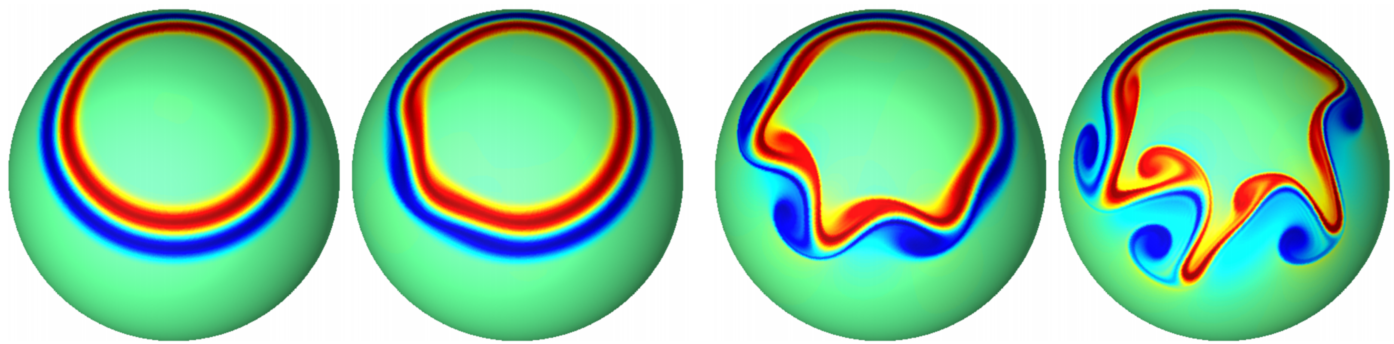


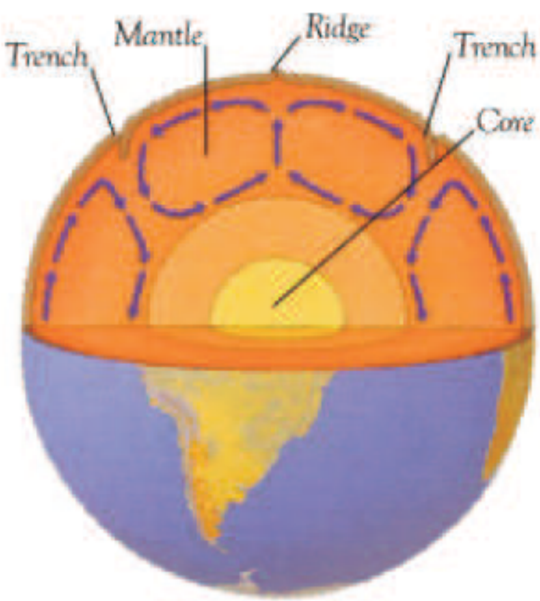
- Background
- Kernel approximation on surfaces
  - Applications to numerically solving PDEs on surfaces
- Local bases for kernels
  - Applications to interpolation and quadrature

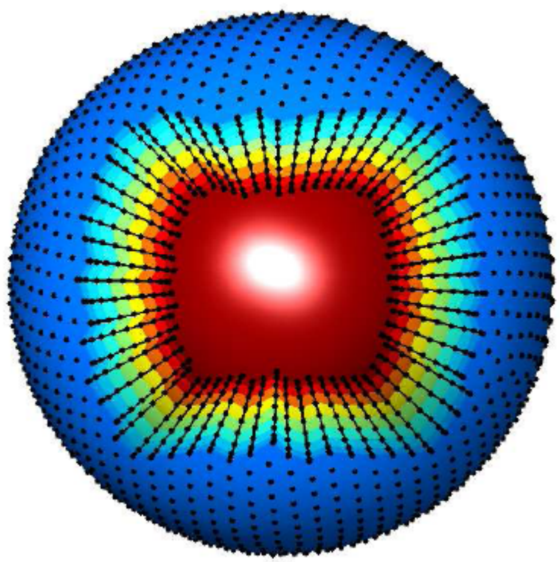
My Background

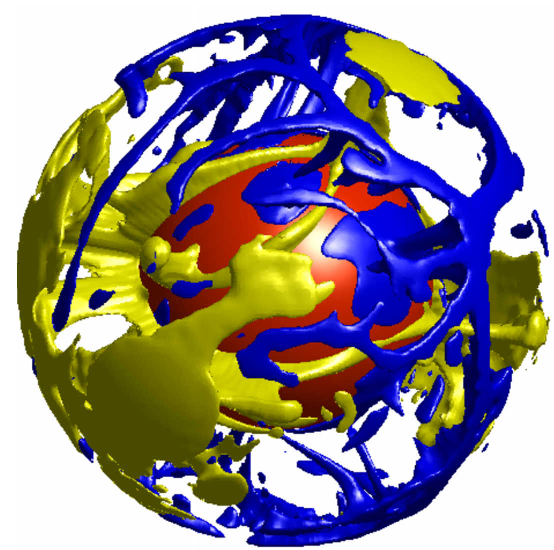
- Radial basis functions for geophysical applications











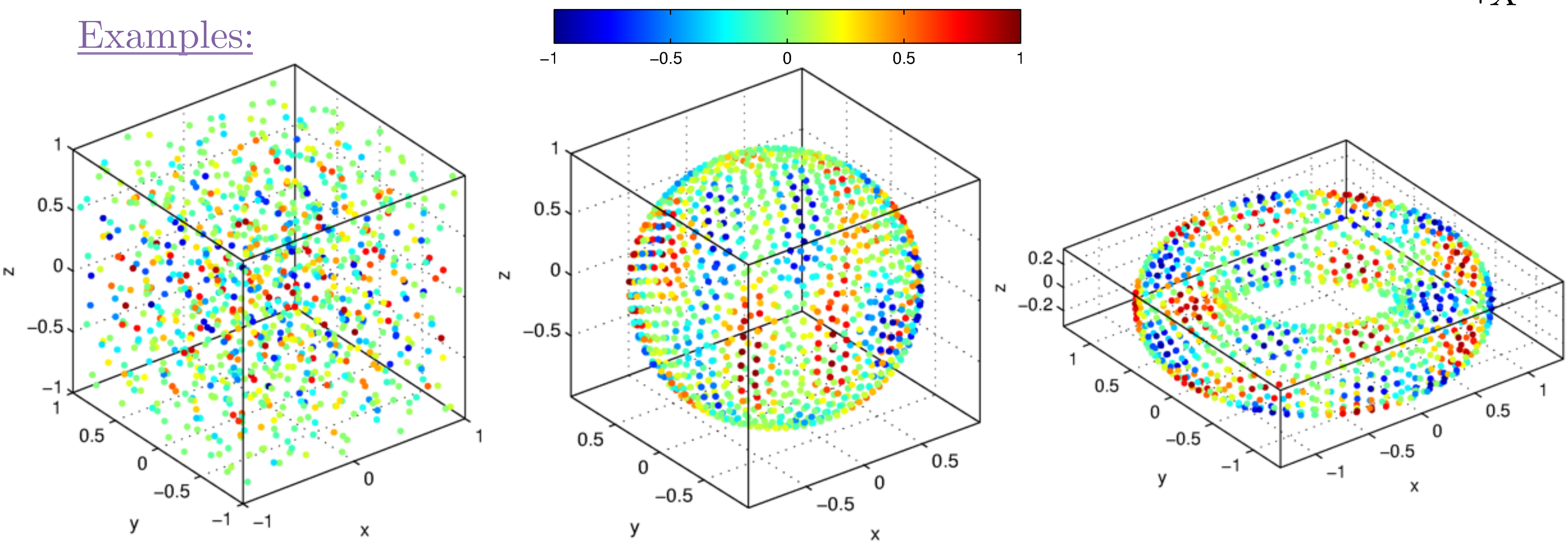
- Joint work with Natasha Flyer and Erik Lehto.

Interpolation with kernels

- Let $\Omega \subset \mathbb{R}^d$ and $X = \{\mathbf{x}_j\}_{j=1}^N$ a set of nodes on Ω.

- Consider a continuous target function $f : \Omega \to \mathbb{R}$ sampled at X: $f\big|_X$.

Examples:



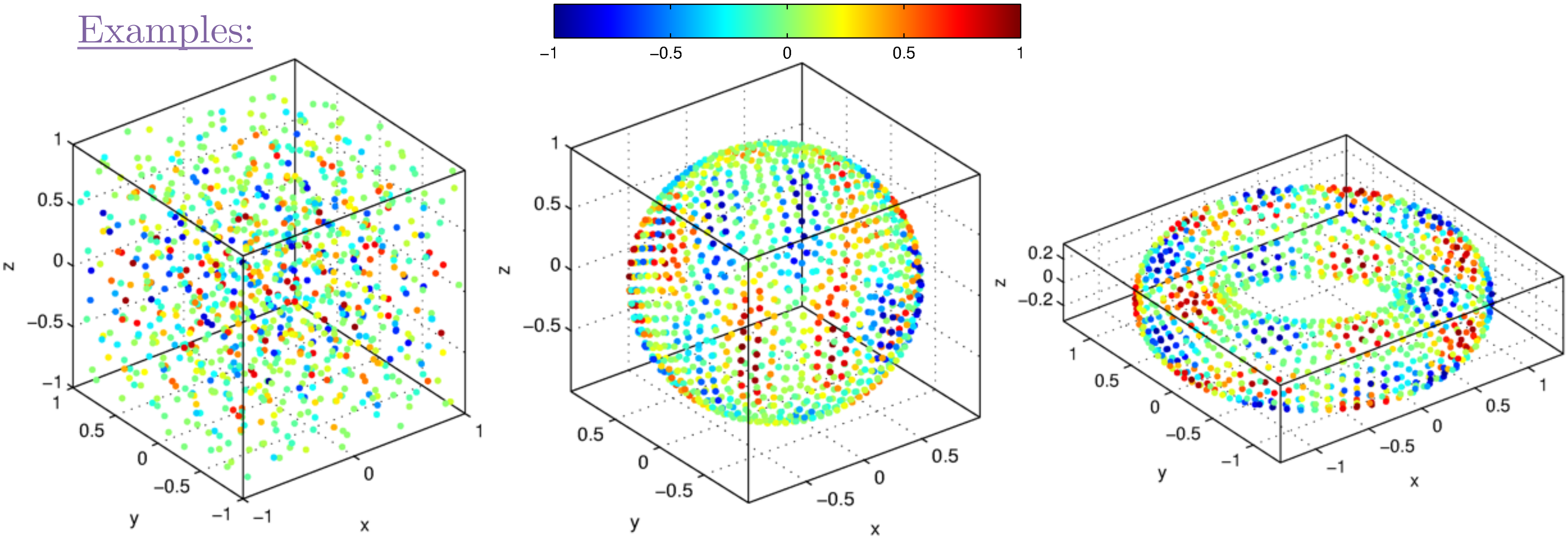
$\Omega = [-1, 1]$ $\qquad$ $\Omega = \mathbb{S}^2$ $\qquad$ $\Omega = \mathbb{T}^2$

- Kernel interpolant to $f\big|_X$: $\qquad I_X f = \sum_{j=1}^{N} c_j \phi(\cdot, \mathbf{x}_j)$

  where $\phi : \Omega \times \Omega \to \mathbb{R}$ and c_j come from requiring $I_X f\big|_X = f\big|_X$

Interpolation with kernels

Examples:



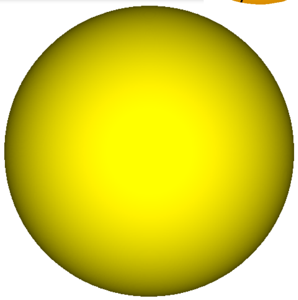
$\Omega = [-1, 1]$ $\qquad$ $\Omega = \mathbb{S}^2$ $\qquad$ $\Omega = \mathbb{T}^2$

- Kernel interpolant to $f\Big|_X$: $\qquad I_X f = \displaystyle\sum_{j=1}^{N} c_j \phi(\cdot, \mathbf{x}_j)$

- We call ϕ a positive definite kernel if $A = \{\phi(\mathbf{x}_i, \mathbf{x}_j)\}$ is positive definite for any $X = \{\mathbf{x}_j\}_{j=1}^N \subset \Omega$.

- In this case c_j are uniquely determined.

Kernel interpolation on surfaces



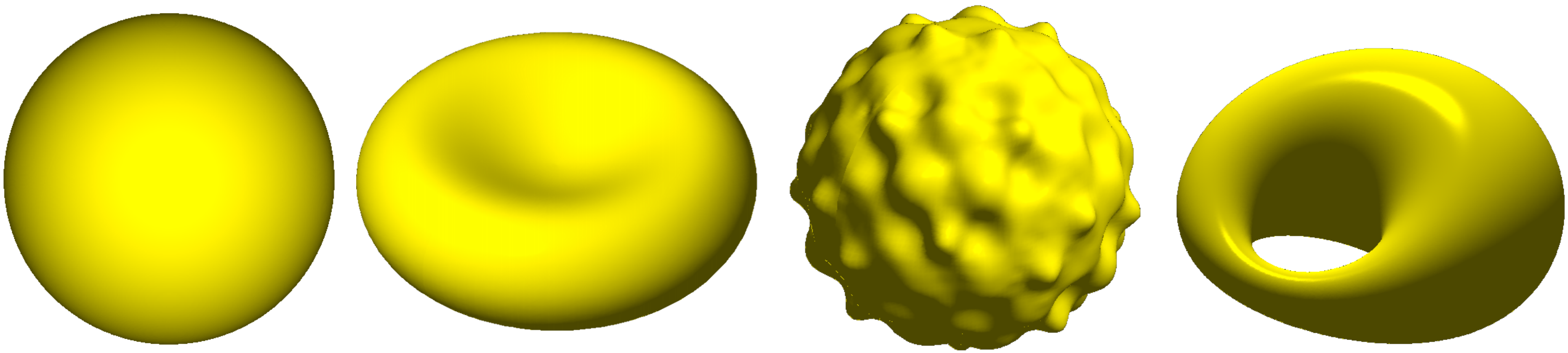
- Kernels on the sphere:
  - Schoenberg (1942)
  - Too many others to list.
- Kernels on specific manifolds ($SO(3)$, motion group, projective spaces):
  - Erb, Filber, Hangelbroek, Schmid, zu Castel,…
- Kernels on arbitrary Riemannian manifolds:
  - Narcowich (1995)
  - Dyn, Hangelbroek, Levesley, Ragozin, Schaback, Ward, Wendland.
- In these studies the kernels used are highly dependent on the manifold.
  - Inherent benefits to this.
  - However, for arbitrary manifolds it is difficult (or impossible) to compute these kernel.

Kernel interpolation on surfaces

- Types of surfaces: $\mathbb{M}$

  Compact, smooth embedded submanifolds of $\mathbb{R}^d$ without a boundary.

- Examples:



- Applications:
  - geophysics
  - atmospheric sciences
  - biology
  - chemistry
  - computer graphics

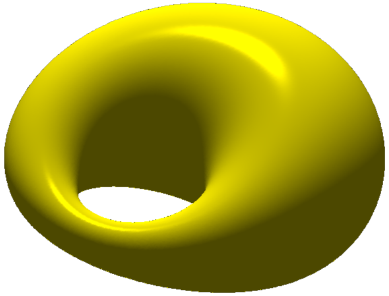
Kernel interpolation on surfaces

- One approach for kernels on general surfaces:

  Use a restricted positive definite kernel from $\mathbb{R}^d$

- Let ϕ be a positive definite kernel on $\mathbb{R}^d$, $\psi(\cdot,\cdot) = \phi(\cdot,\cdot)\Big|_{\mathbb{M},\mathbb{M}}$:

$$I_X f = \sum_{j=1}^{N} c_j \psi(\cdot, \mathbf{x}_j)$$



- Such ϕ are easy to come, e.g.

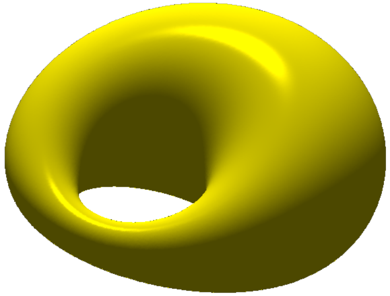
  – Let ϕ be a positive definite radial kernel (RBFs):
  $\phi(\mathbf{x},\mathbf{y}) = \phi(\|\mathbf{x}-\mathbf{y}\|_2) = \phi(r)$

Kernel interpolation on surfaces

- One approach for kernels on general surfaces:
  Use a restricted positive definite kernel from $\mathbb{R}^d$

- Let ϕ be a positive definite kernel on $\mathbb{R}^d$, $\psi(\cdot,\cdot) = \phi(\cdot,\cdot)\Big|_{\mathbb{M},\mathbb{M}}$:

$$I_X f = \sum_{j=1}^{N} c_j \psi(\cdot, \mathbf{x}_j)$$



- Such ϕ are easy to come, e.g.
  - Let ϕ be a positive definite radial kernel (RBFs): $\phi(\mathbf{x},\mathbf{y}) = \phi(\|\mathbf{x}-\mathbf{y}\|_2) = \phi(r)$

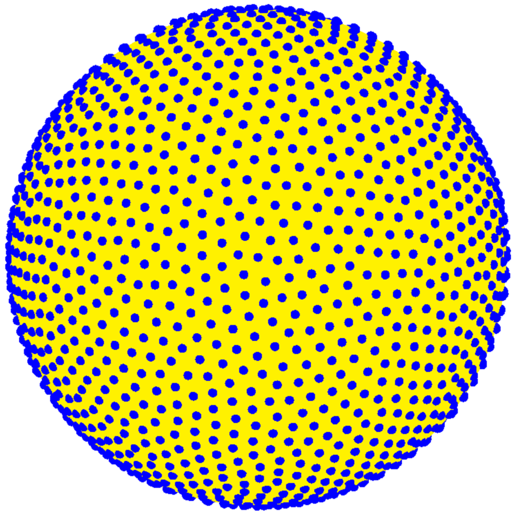
- For $\mathbb{M} = \mathbb{S}^2$, this approach has been thoroughly studied.

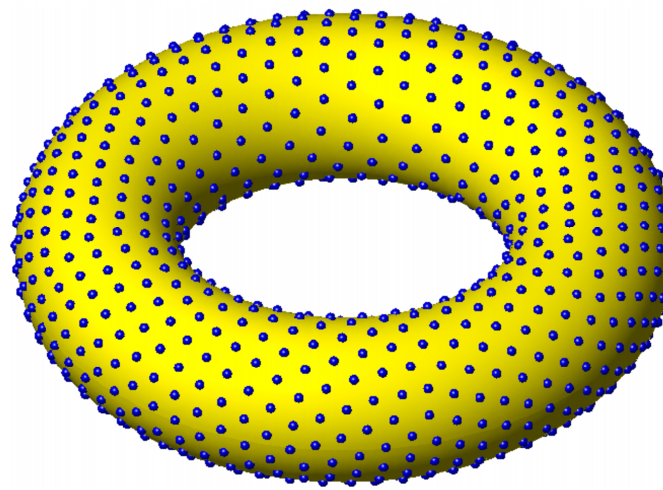
- Surprisingly, for general surfaces, virtually nothing had been done:
  - Powell (2001) DAMTP Technical Report.
  - Fasshauer (2007), p. 83

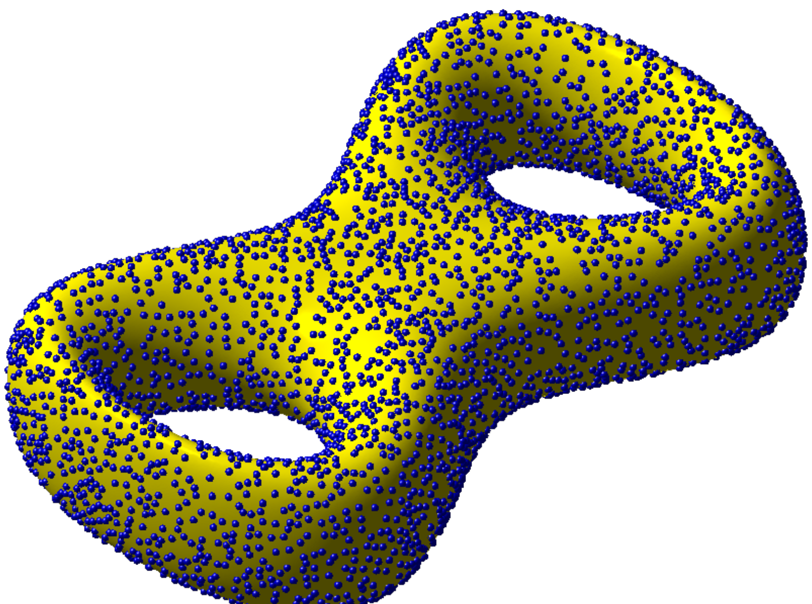
Nodes on surfaces

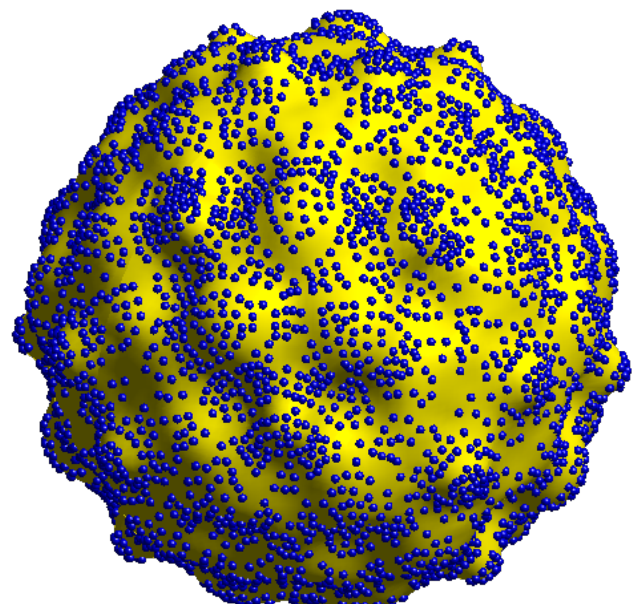
- Kernel methods do not require a mesh, just a set of nodes.

- Some ideas:









Some terminology

The following quantities arise often in kernel methods:

- Mesh norm

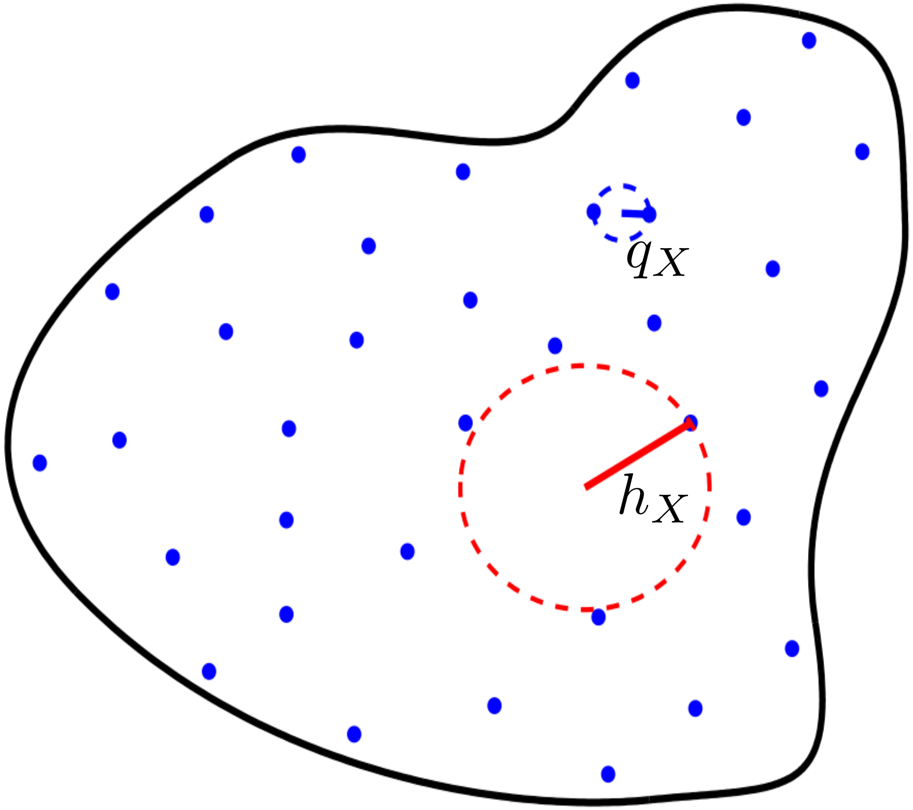
$$h_X = \sup_{\mathbf{x}\in\mathbb{M}} \operatorname{dist}_{\mathbb{M}}(\mathbf{x}, X)$$

- Separation radius

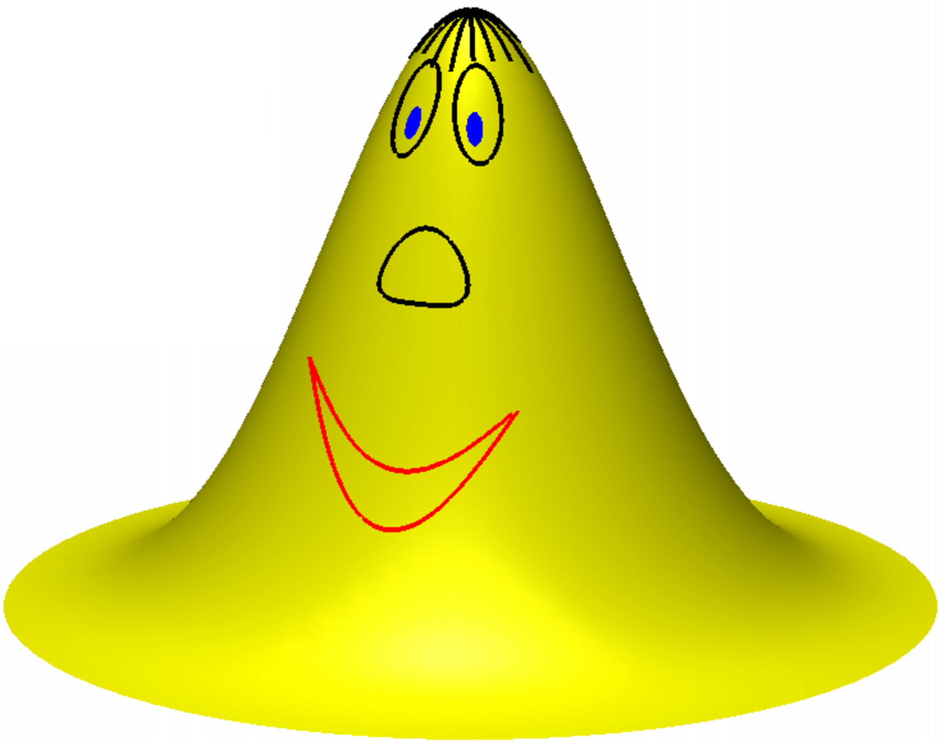
$$q_X = \frac{1}{2}\min_{i\neq j} \operatorname{dist}_{\mathbb{M}}(\mathbf{x}_i, \mathbf{x}_j)$$

- Mesh ratio

$$\rho_X = \frac{h_X}{q_X}$$



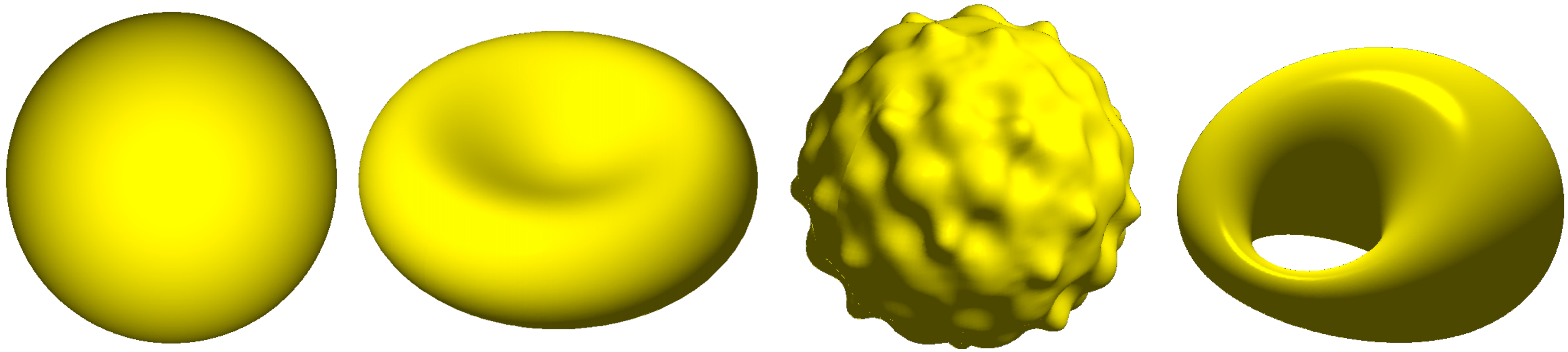
$$X = \{\mathbf{x}_j\}_{j=1}^{N} \subset \mathbb{M}$$



Approximation on surfaces with restricted kernels:

Applications to numerically solving PDEs

Reaction diffusion equations on surfaces



- Prototypical model: 2 interacting species

$$
\begin{aligned}
\frac{\partial u}{\partial t} &= \delta_u \Delta_{\mathbb{M}} u + f_u(t,u,v) \\
\frac{\partial v}{\partial t} &= \delta_v \Delta_{\mathbb{M}} v + f_v(t,u,v)
\end{aligned}
$$

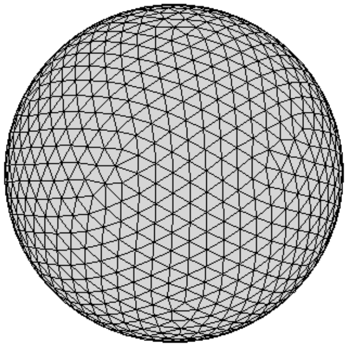
$\Delta_{\mathbb{M}}$ is the Laplace-Beltrami operator for the surface

- Applications
  - Biology: diffusive transport on a membrane, pattern formation on animal coats, and tumor growth.
  - Chemistry: waves in excitable media (cardiac arrhythmia, electrical signals in the brain).
  - Computer graphics: texture mapping and synthesis and image processing.

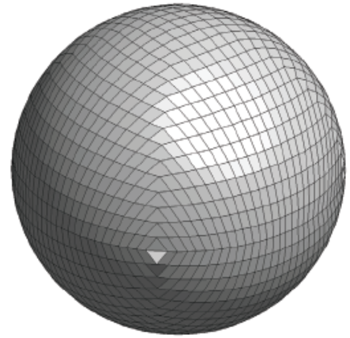
Current methods and kernel-based approach

- Current numerical method can be split into 2 categories:

1. Surface-based: approximate the PDE *on the surface* using *intrinsic* coordinates.

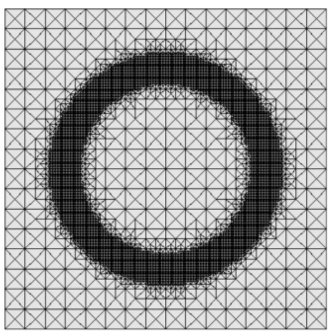


Triangulated Mesh
Dziuk (1988)
Stam (2003)
Xu (2004)
Dziuk & Elliot (2007)

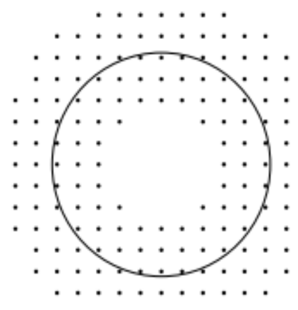


Logically rectangular grid
Calhoun and Helzel (2009)

2. Embedded: approximate the PDE in the *embedding space*, restrict solution to surface.

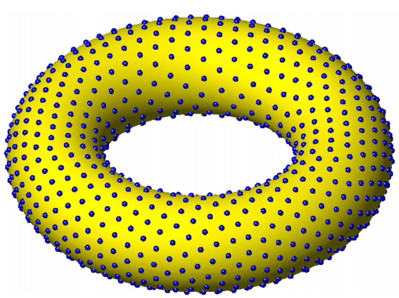


Level Set
Bertalmio *et al.* (2001)
Schwartz *et al.* (2005)
Greer (2006)
Sbalzarini *et al.* (2006)
Dziuk & Elliot (2010)



Closest point:
Ruuth & Merriman (2008)
MacDonald & Ruuth (2008)
MacDonald & Ruuth (2009)

- Kernel-based method: Fuselier & W (2013)



- Similarity to 1: approximate the PDE *on the surface*.
- Similarity to 2: use *extrinsic* coordinates.
- Differences: method is mesh-free;
  computations done in same dimension as the surface.

Interpolation with restricted PD kernels

- Let ϕ be a positive definite kernel on $\mathbb{R}^d$, $\psi(\cdot,\cdot) = \phi(\cdot,\cdot)\big|_{\mathbb{M},\mathbb{M}}$, and $k = \dim(\mathbb{M})$.
- Kernel interpolant: $I_X f = \displaystyle\sum_{j=1}^{N} c_j \psi(\cdot, \mathbf{x}_j)$, where $X = \{\mathbf{x}_j\}_{j=1}^N \subset \mathbb{M}$
- Approximation classes can be found from the native space of ψ: $\mathcal{N}_\psi$
  - $F_\psi = \left\{ f = \displaystyle\sum_j c_j \psi(\cdot, \mathbf{x}_j) \;\middle|\; c_j \in \mathbb{R},\ \mathbf{x}_j \in \mathbb{M} \right\}$
  - $\|f\|^2_{\mathcal{N}_\psi} = \displaystyle\sum_j \sum_k c_j c_k \psi(\mathbf{x}_j, \mathbf{x}_k)$, $f \in F_\psi$
  - $\mathcal{N}_\psi = \overline{F_\psi}$
- What is $\mathcal{N}_\psi$?

Interpolation with restricted PD kernels

- Let ϕ be a positive definite kernel on $\mathbb{R}^d$, $\psi(\cdot,\cdot) = \phi(\cdot,\cdot)\big|_{\mathbb{M},\mathbb{M}}$, and $k = \dim(\mathbb{M})$.
- Kernel interpolant: $I_X f = \sum_{j=1}^{N} c_j \psi(\cdot, \mathbf{x}_j)$, where $X = \{\mathbf{x}_j\}_{j=1}^{N} \subset \mathbb{M}$
- What is $\mathcal{N}_\psi$?
- Suppose the Fourier transform of ϕ on $\mathbb{R}^d$ satisfies $\hat{\phi}(\boldsymbol{\xi}) \sim (1+\|\boldsymbol{\xi}\|_2^2)^{-\tau}$ then $\mathcal{N}_\phi = H^\tau(\mathbb{R}^d)$
- Theorem (Fuselier,W 2012): If ϕ satisfies $\hat{\phi}(\boldsymbol{\xi}) \sim (1+\|\boldsymbol{\xi}\|_2^2)^{-\tau}$ with $\tau > d/2$, then $\mathcal{N}_\psi = H^{\tau-(d-k)/2}(\mathbb{M})$ with equivalent norms.

  Main idea: Trace theorem and restriction and extension operators on the native space from Schaback (1999).

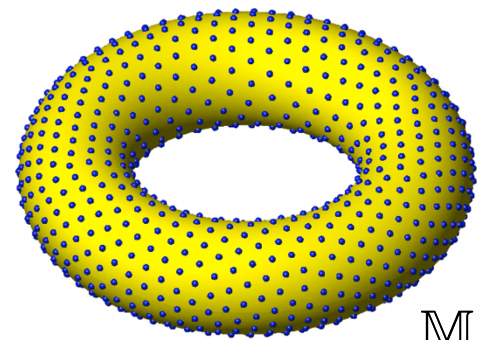
Interpolation error estimates

- Specific error estimate results from Fuselier & W (2012).
  - More general results are given in the paper.

<u>Notation:</u>

- $\mathbb{M} \subset \mathbb{R}^3$, $\dim(\mathbb{M}) = 2$.
- $\psi(\cdot,\cdot) = \phi(\cdot,\cdot)\big|_{\mathbb{M},\mathbb{M}}$
- $\hat{\phi}(\boldsymbol{\xi}) \sim (1 + \|\boldsymbol{\xi}\|_2^2)^{-\tau}$, $\tau > 3/2$
- $s = \tau - 1/2$

- $X = \{\mathbf{x}_j\}_{j=1}^N \subset \mathbb{M}$
- h_X = mesh-norm
- q_X = separation radius
- $\rho_X = h_X/q_X$, mesh ratio



$\mathbb{M}$

<u>Theorem</u>: target functions in the native space

$$\text{If } f \in H^s(\mathbb{M}) \text{ then } \|f - I_X f\|_{L_2(\mathbb{M})} = \mathcal{O}(h_X^s)$$

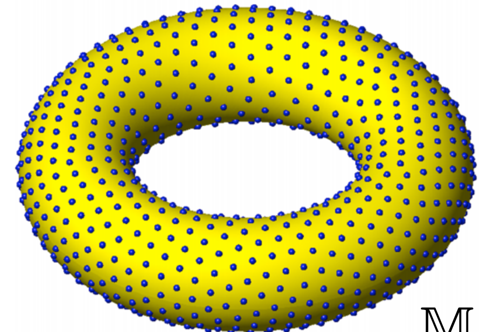
Interpolation error estimates

- Specific error estimate results from Fuselier & W (2012).
    - More general results are given in the paper.

Notation:

- $\mathbb{M} \subset \mathbb{R}^3$, $\dim(\mathbb{M}) = 2$.
- $\psi(\cdot,\cdot) = \phi(\cdot,\cdot)\big|_{\mathbb{M},\mathbb{M}}$
- $\hat{\phi}(\boldsymbol{\xi}) \sim (1 + \|\boldsymbol{\xi}\|_2^2)^{-\tau}$, $\tau > 3/2$
- $s = \tau - 1/2$

- $X = \{\mathbf{x}_j\}_{j=1}^N \subset \mathbb{M}$
- h_X = mesh-norm
- q_X = separation radius
- $\rho_X = h_X / q_X$, mesh ratio



$\mathbb{M}$

Corollary: target functions approx. twice as smooth as the native space

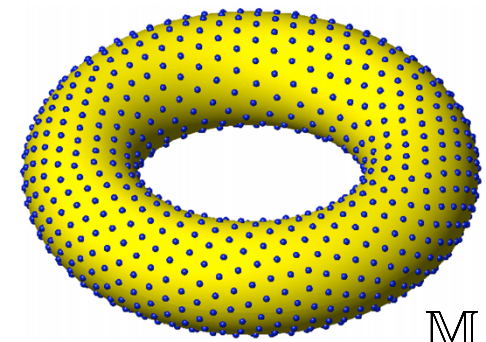
$$\text{If } f \in H^s(\mathbb{M}) \text{ and } T^{-1} f \in L_2(\mathbb{M}) \text{ then } \|f - I_X f\|_{L_2(\mathbb{M})} = \mathcal{O}(h_X^{2s})$$

Interpolation error estimates

- Specific error estimate results from Fuselier & W (2012).
  - More general results are given in the paper.

Notation:

- $\mathbb{M} \subset \mathbb{R}^3$, $\dim(\mathbb{M}) = 2$.
- $\psi(\cdot,\cdot) = \phi(\cdot,\cdot)\big|_{\mathbb{M},\mathbb{M}}$
- $\hat{\phi}(\boldsymbol{\xi}) \sim (1 + \|\boldsymbol{\xi}\|_2^2)^{-\tau}$, $\tau > 3/2$
- $s = \tau - 1/2$
- $X = \{\mathbf{x}_j\}_{j=1}^N \subset \mathbb{M}$
- h_X = mesh-norm
- q_X = separation radius
- $\rho_X = h_X/q_X$, mesh ratio



$\mathbb{M}$

Theorem: target functions rougher than the native space

$$\text{If } f \in H^{\beta}(\mathbb{M}) \text{ with } s > \beta > 1 \text{ then } \|f - I_X f\|_{L_2(\mathbb{M})} = \mathcal{O}(h_X^{\beta}\rho_X^{s-\beta})$$

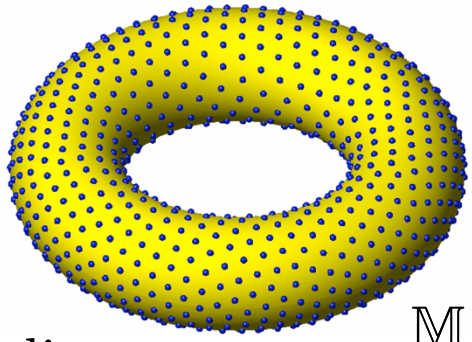
Proof required results Narcowich, Ward, & Wendland (2005; 2006) on $\mathbb{R}^d$

Interpolation error estimates

- Specific error estimate results from Fuselier & W (2012).
  - More general results are given in the paper.

Notation:

- $\mathbb{M} \subset \mathbb{R}^3$, $\dim(\mathbb{M}) = 2$.
- $\psi(\cdot,\cdot) = \phi(\cdot,\cdot)\big|_{\mathbb{M},\mathbb{M}}$
- $\hat{\phi}(\boldsymbol{\xi}) \sim (1 + \|\boldsymbol{\xi}\|_2^2)^{-\tau}$, $\tau > 3/2$
- $s = \tau - 1/2$
- $X = \{\mathbf{x}_j\}_{j=1}^N \subset \mathbb{M}$
- h_X = mesh-norm
- q_X = separation radius
- $\rho_X = h_X/q_X$, mesh ratio

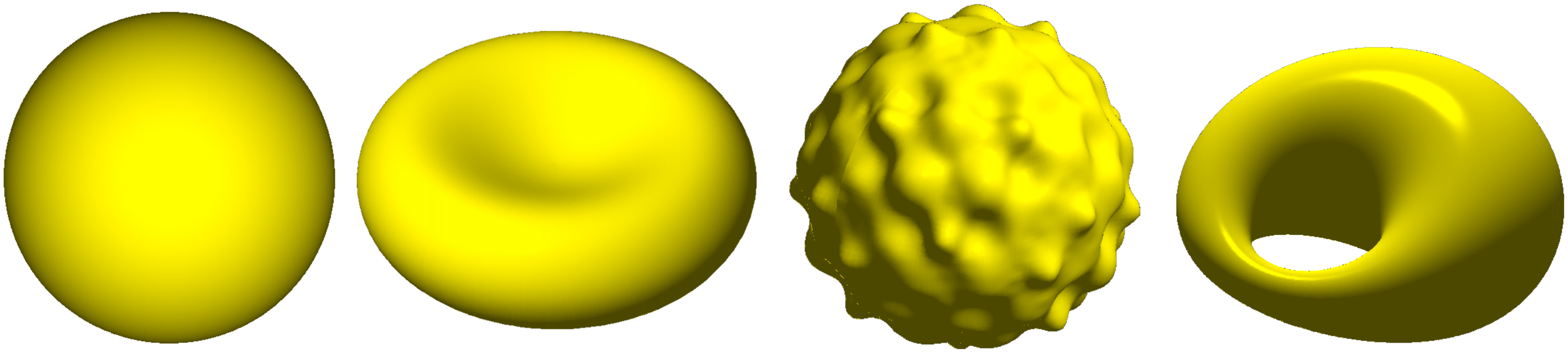


$\mathbb{M}$

Main point: can use simple RBFs for interpolation on surfaces:

$$I_X f = \sum_{j=1}^{N} c_j \psi(\mathbf{x}, \mathbf{x}_j) = \sum_{j=1}^{N} c_j \phi(\|\mathbf{x} - \mathbf{x}_j\|_2)$$

Return: Reaction diffusion equations on surfaces



- Prototypical model: 2 interacting species

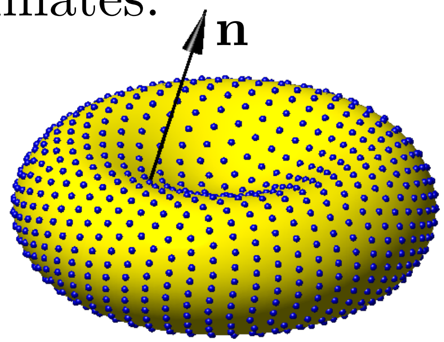
$$\frac{\partial u}{\partial t} = \delta_u \Delta_{\mathbb{M}} u + f_u(t,u,v)$$
$$\frac{\partial v}{\partial t} = \delta_v \Delta_{\mathbb{M}} v + f_v(t,u,v)$$

$\Delta_{\mathbb{M}}$ is the Laplace-Beltrami operator for the surface

- Applications
  - Biology: diffusive transport on a membrane, pattern formation on animal coats, and tumor growth.
  - Chemistry: waves in excitable media (cardiac arrhythmia, electrical signals in the brain).
  - Computer graphics: texture mapping and synthesis and image processing.

Differential operators on surfaces

- Surface gradient on $\mathbb{M}$ in *extrinsic* (or Cartesian) coordinates:



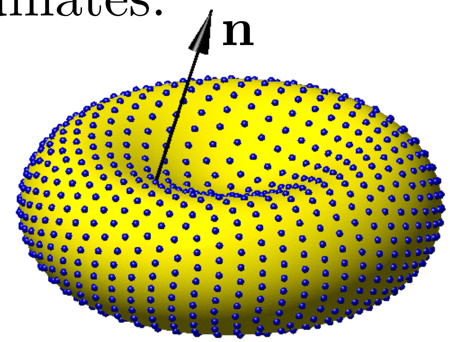
$$\nabla_{\mathbb{M}} := \mathbf{P}\nabla = (\mathbf{I} - \mathbf{n} \otimes \mathbf{n})\,\nabla$$

- After some manipulations

$$\nabla_{\mathbb{M}} := \begin{bmatrix} (\mathbf{e}_x \cdot \mathbf{P})\,\nabla \\ (\mathbf{e}_y \cdot \mathbf{P})\,\nabla \\ (\mathbf{e}_z \cdot \mathbf{P})\,\nabla \end{bmatrix} = \begin{bmatrix} (\mathbf{e}_x - n_x \mathbf{n}) \cdot \nabla \\ (\mathbf{e}_y - n_y \mathbf{n}) \cdot \nabla \\ (\mathbf{e}_z - n_z \mathbf{n}) \cdot \nabla \end{bmatrix} = \begin{bmatrix} \mathbf{p}_x \cdot \nabla \\ \mathbf{p}_y \cdot \nabla \\ \mathbf{p}_z \cdot \nabla \end{bmatrix} = \begin{bmatrix} \mathcal{G}^x \\ \mathcal{G}^y \\ \mathcal{G}^z \end{bmatrix}$$

Differential operators on surfaces

- Surface gradient on $\mathbb{M}$ in *extrinsic* (or Cartesian) coordinates:



$$\nabla_{\mathbb{M}} := \mathbf{P}\nabla = (\mathbf{I} - \mathbf{n} \otimes \mathbf{n})\,\nabla$$

- After some manipulations

$$\nabla_{\mathbb{M}} := \begin{bmatrix} (\mathbf{e}_x \cdot \mathbf{P})\,\nabla \\ (\mathbf{e}_y \cdot \mathbf{P})\,\nabla \\ (\mathbf{e}_z \cdot \mathbf{P})\,\nabla \end{bmatrix} = \begin{bmatrix} (\mathbf{e}_x - n_x\mathbf{n}) \cdot \nabla \\ (\mathbf{e}_y - n_y\mathbf{n}) \cdot \nabla \\ (\mathbf{e}_z - n_z\mathbf{n}) \cdot \nabla \end{bmatrix} = \begin{bmatrix} \mathbf{p}_x \cdot \nabla \\ \mathbf{p}_y \cdot \nabla \\ \mathbf{p}_z \cdot \nabla \end{bmatrix} = \begin{bmatrix} \mathcal{G}^x \\ \mathcal{G}^y \\ \mathcal{G}^z \end{bmatrix}$$

- Surface divergence of smooth vector field $\mathbf{f} : \mathbb{R}^3 \to \mathbb{R}^3$ ($\mathbf{f} = (f_x, f_y, f_z)$):

$$\nabla_{\mathbb{M}} \cdot \mathbf{f} := (\mathbf{P}\nabla) \cdot \mathbf{f} = \mathcal{G}^x f_x + \mathcal{G}^y f_y + \mathcal{G}^z f_z$$

- Laplace-Beltrami operator on $\mathbb{M}$ in *extrinsic coordinates*:

$$\Delta_{\mathbb{M}} := (\mathbf{P}\nabla)\cdot(\mathbf{P}\nabla) = \mathcal{G}^x\mathcal{G}^x + \mathcal{G}^y\mathcal{G}^y + \mathcal{G}^z\mathcal{G}^z = \mathcal{D}_{xx} + \mathcal{D}_{yy} + \mathcal{D}_{zz}$$

$\Delta_{\mathbb{M}}$ is the Laplace-Beltrami operator for the surface.

Kernel approximation of surface derivative operators

Idea from Fuselier & W (2013):

- Let $X = \{\mathbf{x}_j\}_{j=1}^N \subset \mathbb{M}$ and some smooth target $f : \mathbb{M} \to \mathbb{R}$.
- Interpolate $\underline{f} := f\big|_X$, using restricted (RBF) kernel interpolant:

$$I_X f = \sum_{j=1}^{N} c_j \psi(\mathbf{x}, \mathbf{x}_j) = \sum_{j=1}^{N} c_j \phi(\|\mathbf{x} - \mathbf{x}_j\|)$$

Kernel approximation of surface derivative operators

Idea from Fuselier & W (2013):

- Let $X = \{\mathbf{x}_j\}_{j=1}^{N} \subset \mathbb{M}$ and some smooth target $f : \mathbb{M} \to \mathbb{R}$.
- Interpolate $\underline{f} := f\big|_X$, using restricted (RBF) kernel interpolant:

$$I_X f = \sum_{j=1}^{N} c_j \psi(\mathbf{x}, \mathbf{x}_j) = \sum_{j=1}^{N} c_j \phi(\|\mathbf{x} - \mathbf{x}_j\|)$$

- Apply $\mathcal{G}^x$, $\mathcal{G}^y$, $\mathcal{G}^z$ to $I_X f$ and evaluate at X:

$$(\mathcal{G}^x[I_X f])\big|_X = G_x \underline{f}, \quad (\mathcal{G}^y[I_X f])\big|_X = G_y \underline{f}, \quad (\mathcal{G}^z[I_X f])\big|_X = G_z \underline{f}$$

Kernel approximation of surface derivative operators

Idea from Fuselier & W (2013):

- Let $X = \{\mathbf{x}_j\}_{j=1}^N \subset \mathbb{M}$ and some smooth target $f : \mathbb{M} \to \mathbb{R}$.
- Interpolate $\underline{f} := f\big|_X$, using restricted (RBF) kernel interpolant:

$$I_X f = \sum_{j=1}^{N} c_j \psi(\mathbf{x}, \mathbf{x}_j) = \sum_{j=1}^{N} c_j \phi(\|\mathbf{x} - \mathbf{x}_j\|)$$

- Apply $\mathcal{G}^x$, $\mathcal{G}^y$, $\mathcal{G}^z$ to $I_X f$ and evaluate at X:

$$(\mathcal{G}^x[I_X f])\big|_X = G_x \underline{f}, \quad (\mathcal{G}^y[I_X f])\big|_X = G_y \underline{f}, \quad (\mathcal{G}^z[I_X f])\big|_X = G_z \underline{f}$$

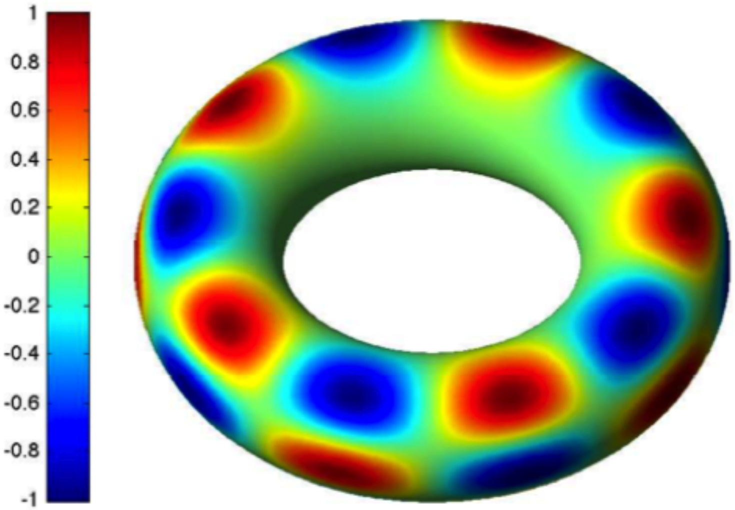
- Approximate $(\Delta_{\mathbb{M}} f)\big|_X$ using G_x, G_y, G_z:

$$(\Delta_{\mathbb{M}} f)\big|_X = ([\mathcal{G}^x\mathcal{G}^x + \mathcal{G}^y\mathcal{G}^y + \mathcal{G}^z\mathcal{G}^z] f)\big|_X \approx \underbrace{(G_x G_x + G_y G_y + G_z G_z)}_{L_{\mathbb{M}}} \underline{f}$$

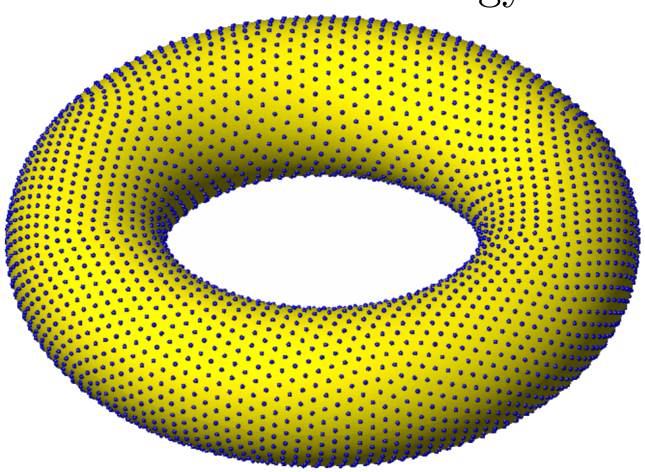
- $L_{\mathbb{M}}$ is an $N \times N$ differentiation matrix

Example: convergence of discrete surface Laplacian

Smooth target f



N near-minimal Riesz energy nodes

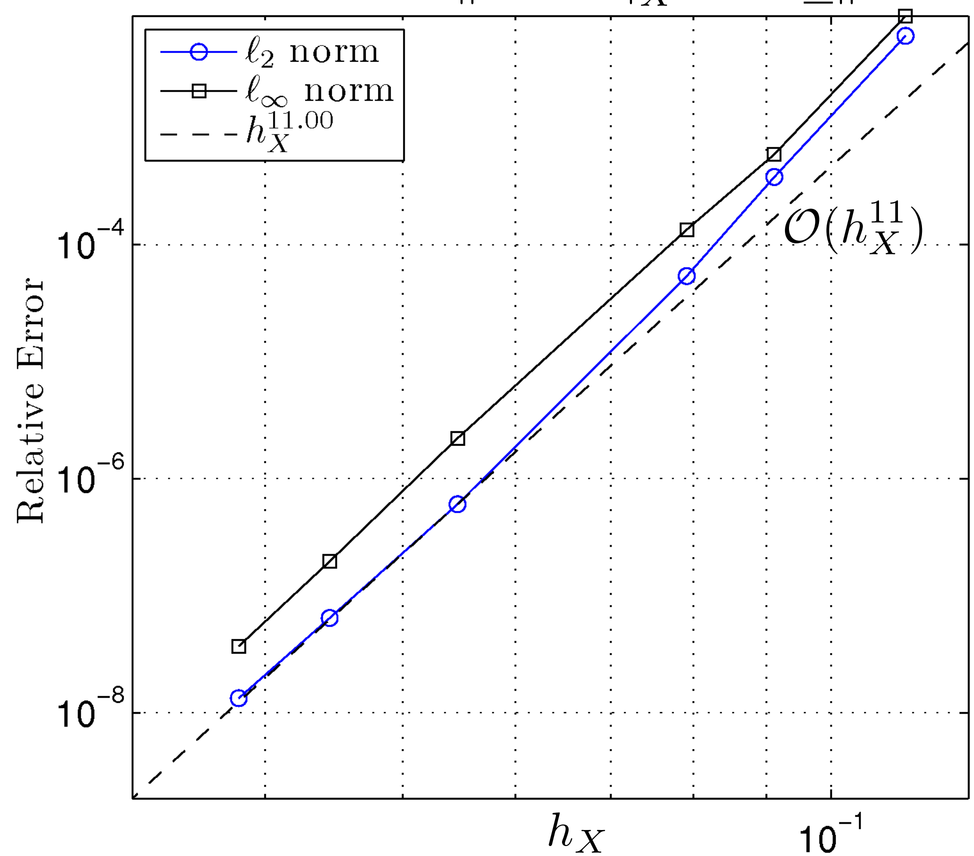


Mesh-norm: $h_X \sim 1/\sqrt{N}$

Matérn kernel: $\psi \Rightarrow \phi(r) = (\varepsilon r)^{9/2} K_{9/2}(\varepsilon r)$

$\mathcal{N}_\psi = H^{11/2}(\mathbb{M})$

Error: $\left\| (\Delta_{\mathbb{M}} f)\big|_X - L_{\mathbb{M}} \underline{f} \right\|$



- Error estimates given in Fuselier & W (2013)
- Observed convergence rate is 2 orders higher than theory predicts. (More on this later...)

Applications: Turing patterns

- Pattern formation via non-linear reaction-diffusion systems; Turing (1952)

Possible mechanism for animal coat formation (and other morphogenesis phenomena)



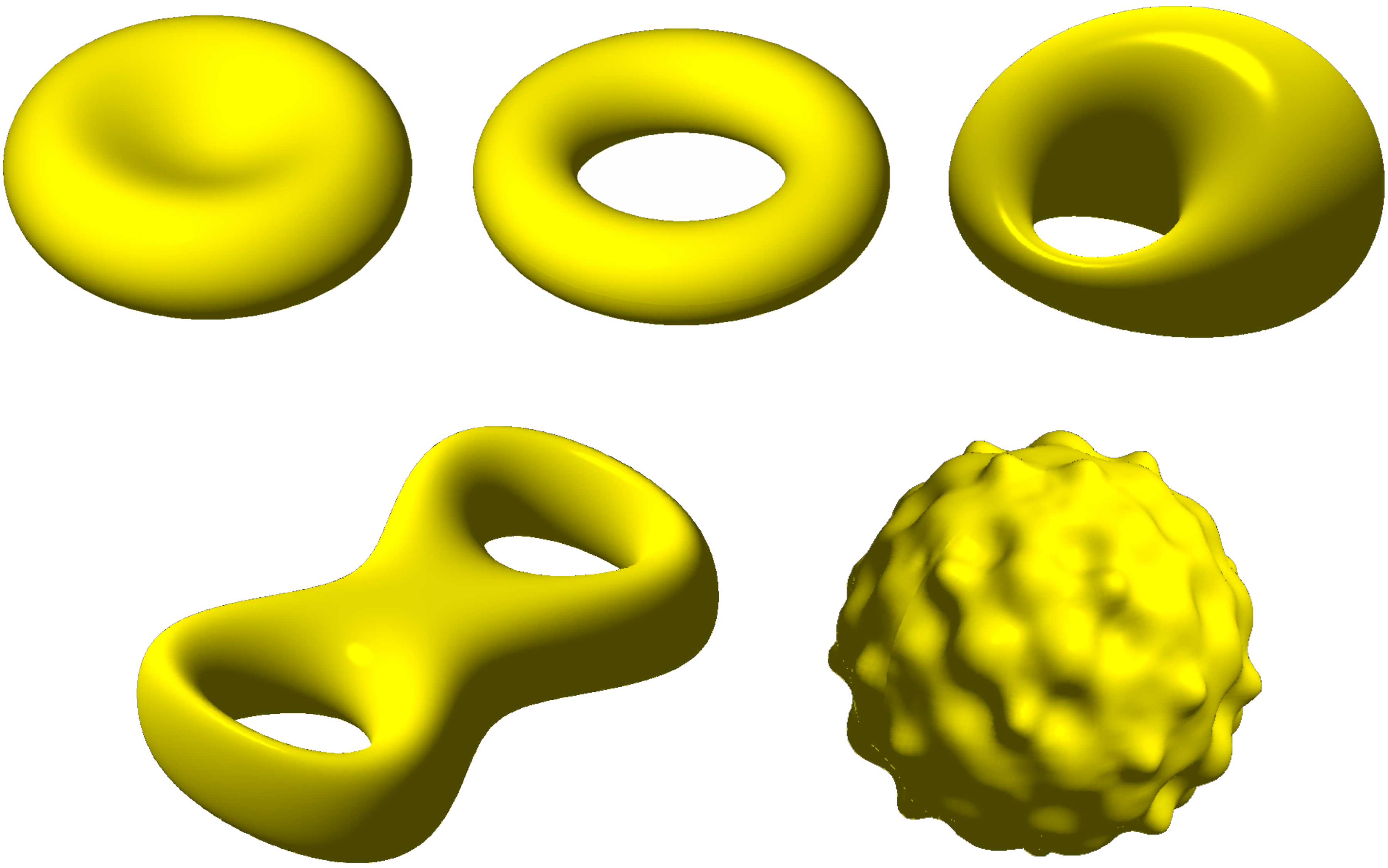
- Example system: Barrio *et al.* (1999)

$$
\begin{aligned}
\frac{\partial u}{\partial t} &= \delta_u \Delta_{\mathbb{M}} u + \alpha u(1 - \tau_1 v^2) + v(1 - \tau_2 u) \\
\frac{\partial v}{\partial t} &= \delta_v \Delta_{\mathbb{M}} v + \beta v \left(1 + \frac{\alpha \tau_1}{\beta} uv\right) + u(\gamma + \tau_2 v)
\end{aligned}
$$

- These types of systems have been studied extensively in planar domains.
- Recent studies have focused on the sphere.
- Growing interest in studying these on more general surfaces.
- Numerical method: collocation and method-of-lines

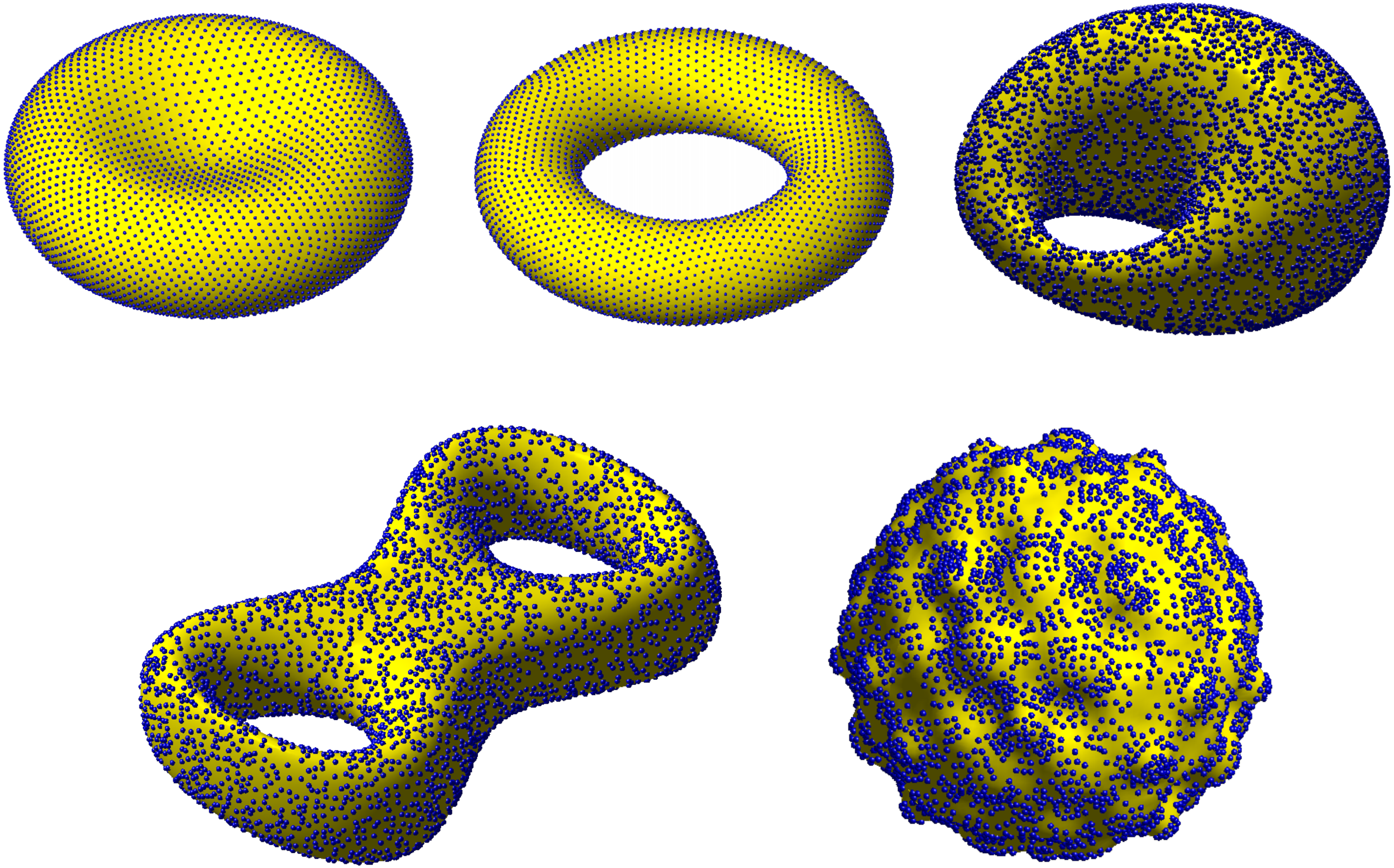
Application: Turing patterns

- Surfaces used in the numerical experiments:



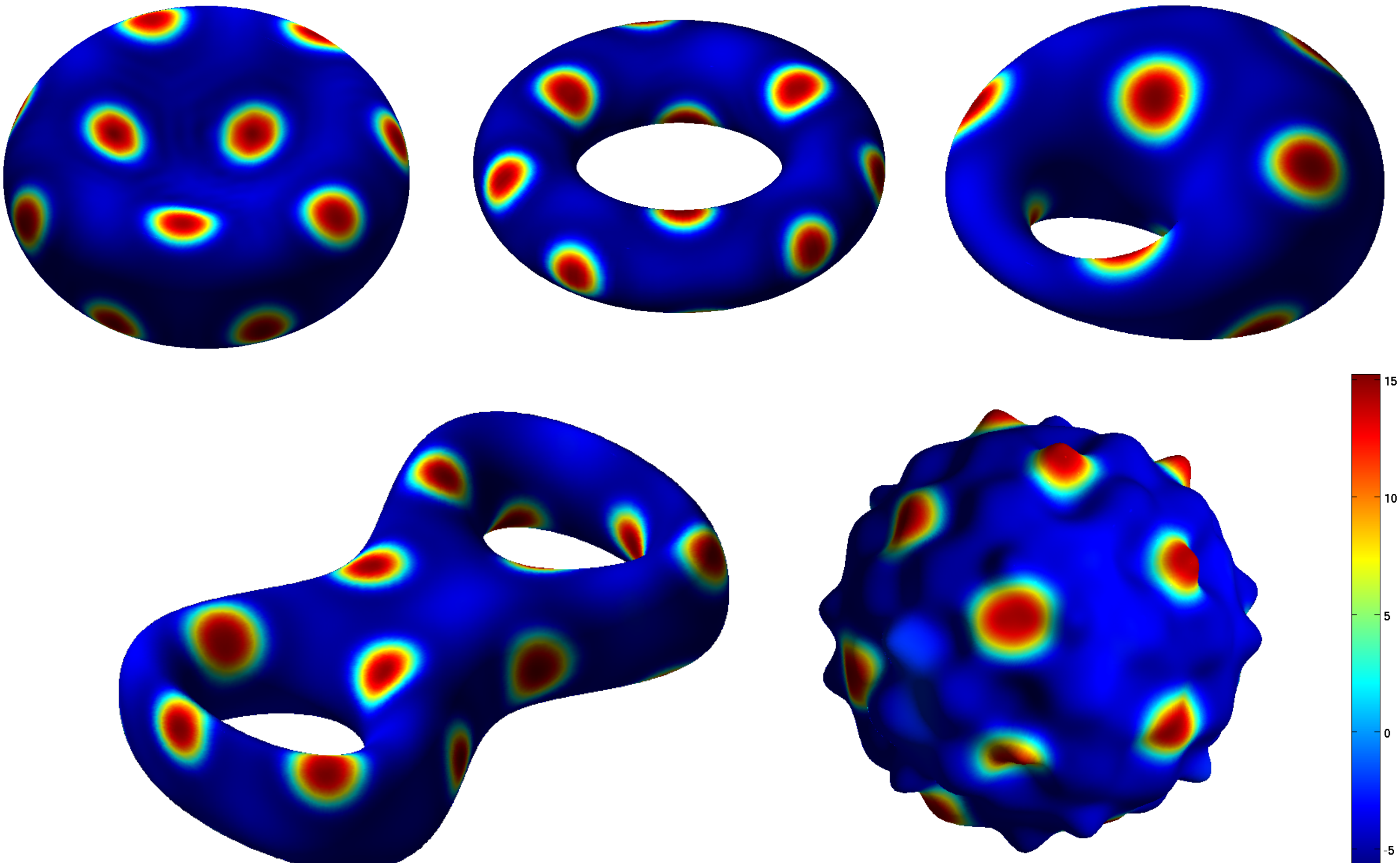
Application: Turing patterns

- Node sets X used in the numerical experiments:



Application: Turing patterns

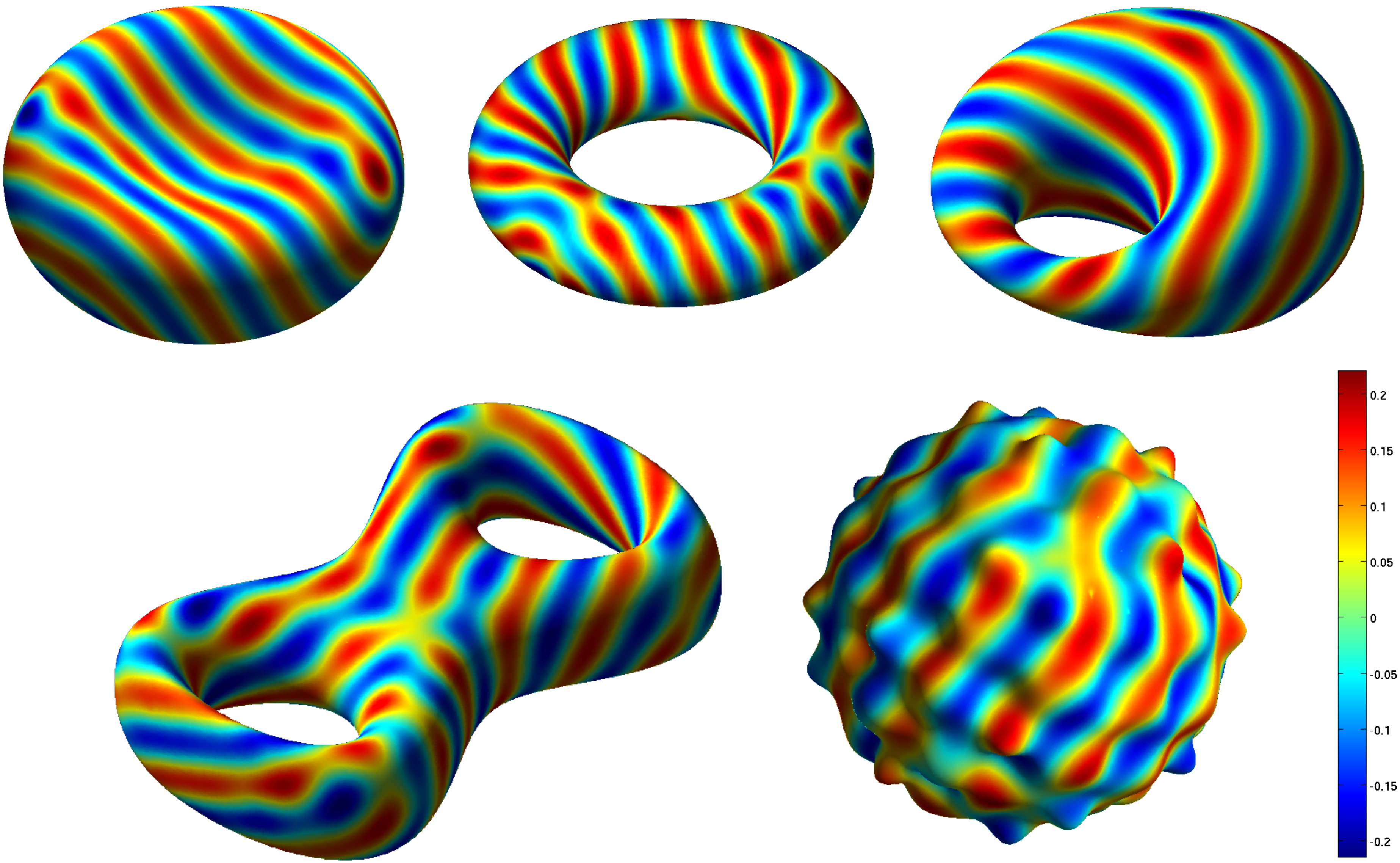
- Numerical solutions: *steady* spot patterns (visualization of u component)



Initial condition: u and v set to random values between +/- 0.5

Application: Turing patterns

- Numerical solutions: *steady* stripe patterns (visualization of u component)



Initial condition: u and v set to random values between +/- 0.5

Application: spiral waves in excitable media

- Example system: Barkley (1991)

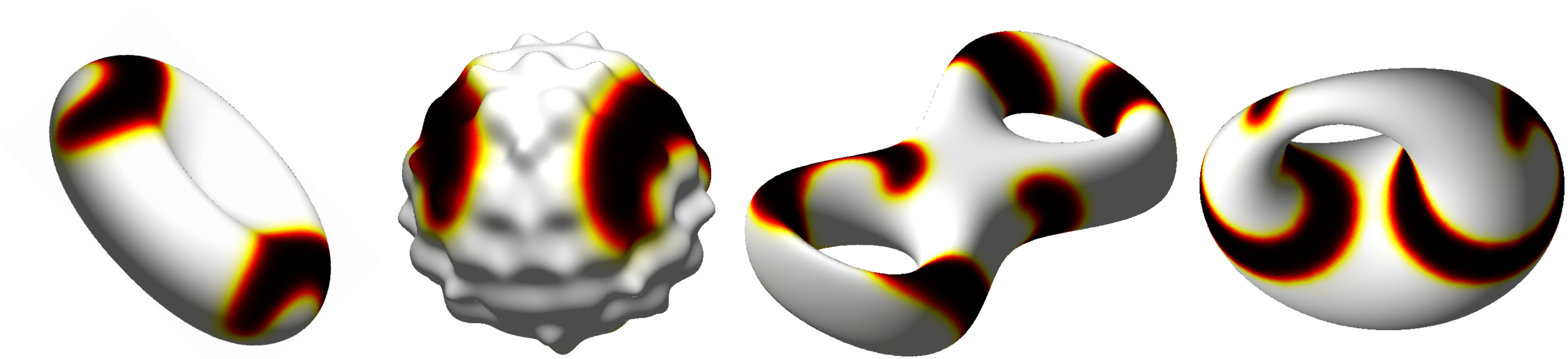
$$\frac{\partial u}{\partial t} = \delta_u \Delta_{\mathbb{M}} u + \frac{1}{\epsilon} u (1-u) \left(u - \frac{v+b}{a} \right)$$

$$\frac{\partial v}{\partial t} = \delta_v \Delta_{\mathbb{M}} v + u - v$$

u=activator species
v=inhibitor species

Simplification of FitzHugh-Nagumo model for a spiking neuron.

- Studied extensively on planar regions and somewhat on the sphere.
- Growing interest more physically relevant domains like surfaces.
- Snapshots from different numerical simulations with our method:



visualization of the u (activator) component

Summary and future

- Restricted kernels offer a relatively simple method for interpolation on rather general surfaces.
  - Interpolation error estimates are similar to what you expect from $\mathbb{R}^d$.
- Method can be used to approximate surface derivatives in a relatively straightforward manner.
  - These approximation can provide high rates of approximation.
  - Can be used to also solve PDEs to high accuracy.
- Future: Biological Applications
  - PDEs on moving surfaces.
  - PDEs that feed back on the shape of the object.
- Future: Improve computational cost
  - Radial basis finite difference formulas (RBF-FD)
  - Partition of unity methods
  - Localized bases

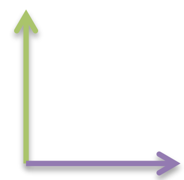
Part II

Good bases for kernel spaces

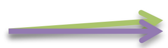
- Let $\Omega \subseteq \mathbb{R}^d$, $\phi : \Omega \times \Omega \to \mathbb{R}$ be PD, and $X = \{\mathbf{x}_j\}_{j=1}^N \subset \Omega$
- Observation: The space spanned by linear combinations of $\phi(\cdot, \mathbf{x}_j)$ is good for approximation.

  BUT, the standard basis $\{\phi(\cdot, \mathbf{x}_1), \ldots, \phi(\cdot, \mathbf{x}_N)\}$ can be problematic.

Analogy:
(Fornberg)



Good basis for $\mathbb{R}^2$



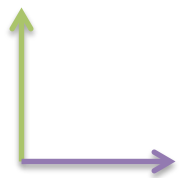
Bad basis for $\mathbb{R}^2$

Good bases for kernel spaces

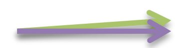
- Let $\Omega \subseteq \mathbb{R}^d$, $\phi : \Omega \times \Omega \to \mathbb{R}$ be PD, and $X = \{\mathbf{x}_j\}_{j=1}^N \subset \Omega$
- Observation: The space spanned by linear combinations of $\phi(\cdot, \mathbf{x}_j)$ is good for approximation.

  BUT, the standard basis $\{\phi(\cdot, \mathbf{x}_1), \ldots, \phi(\cdot, \mathbf{x}_N)\}$ can be problematic.

Analogy:
(Fornberg)



Good basis for $\mathbb{R}^2$



Bad basis for $\mathbb{R}^2$

- Can a better basis be constructed?
  - Difference functionals: Dyn, Levin & Rippa; Sibson & Stone; Beatson, Levesley, & Mouat.
  - Approximate cardinal functions: Beatson & Powell; Faul & Powell; Beatson, Cherrie, & Mouat.
  - Orthonormal: Schaback & Müller; Schaback & Pazouki; De Marchi & Santin
  - RBF-QR: Fornberg & Piret; Fornberg, Larsson, & Flyer; Fasshauer & McCourt

Lagrange functions on the sphere

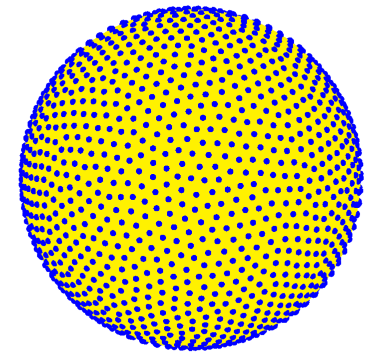
Restrict our attention to $\Omega = \mathbb{S}^2$ and $\psi_\ell(t) = (1-t)^{\ell-1}\log(1-t)$

Standard zonal interpolant:

$$s_X(\mathbf{x}) = \sum_{j=1}^{N} c_j \psi_\ell(\mathbf{x}\cdot\mathbf{x}_j) + \sum_{k=1}^{\ell^2} b_k p_k(\mathbf{x}) \quad \sum_{j=1}^{N} c_j p_k(\mathbf{x}_j) = 0,\ 1 \le k \le \ell^2$$

Lagrange form:

$$s_X(\mathbf{x}) = \sum_{j=1}^{N} L_j(\mathbf{x}) f_j, \qquad L_i(\mathbf{x}_j) = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

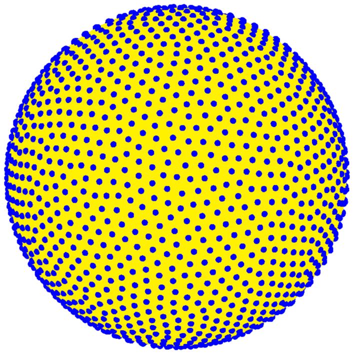


Lagrange functions on the sphere

Restrict our attention to $\Omega = \mathbb{S}^2$ and $\psi_\ell(t) = (1-t)^{\ell-1}\log(1-t)$

Standard zonal interpolant:
$$s_X(\mathbf{x}) = \sum_{j=1}^{N} c_j \psi_\ell(\mathbf{x}\cdot\mathbf{x}_j) + \sum_{k=1}^{\ell^2} b_k p_k(\mathbf{x}) \quad \sum_{j=1}^{N} c_j p_k(\mathbf{x}_j) = 0,\ 1 \le k \le \ell^2$$

Lagrange form:
$$s_X(\mathbf{x}) = \sum_{j=1}^{N} L_j(\mathbf{x}) f_j, \quad L_i(\mathbf{x}_j) = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

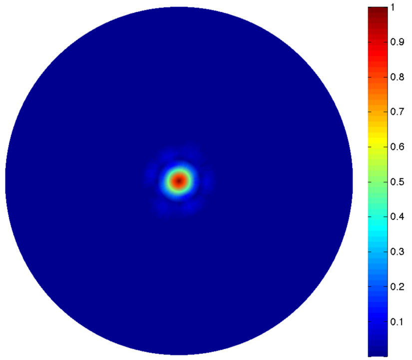


Results on the Lagrange functions for quasi-uniform X:
(Hangelbroek, Narcowich, Sun, Ward)

1. Lagrange basis is local (HNW, 2010):
$$|L_j(\mathbf{x})| \le C \exp\left[-\nu \frac{\mathrm{dist}(\mathbf{x}_j,\mathbf{x})}{h_X}\right]$$

2. Lebesgue constant is bounded (HNW, 2010):
$$\mathcal{L}_X := \max_{\mathbf{x}\in\mathbb{S}^2} \sum_{j=1}^{N} |L_j(\mathbf{x})| \le C$$

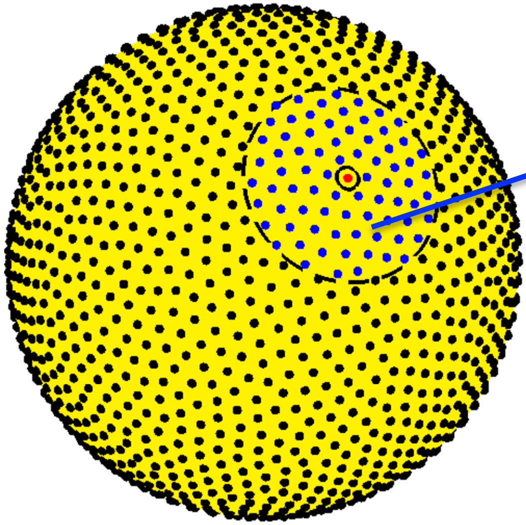
3. Lagrange basis is stable (HNSW, 2011)



$|L_j(\mathbf{x})|$

Local Lagrange functions on the sphere

$X = \{\mathbf{x}_j\}_{j=1}^{N} \subset \mathbb{S}^2$



Algorithm: For $i = 1, \ldots, N$

1. Choose $n << N$ nearest neighbors to $\mathbf{x}_i$:

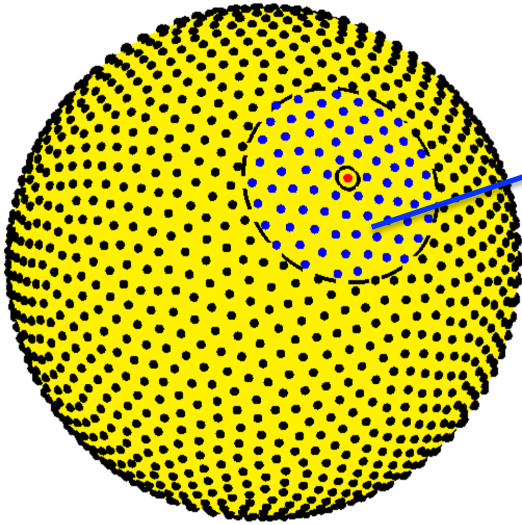
   $X_i = \{\mathbf{x}_j^i\}_{j=1}^{n} \subset X$

2. Construct the local Lagrange function on X:

$$\widetilde{L}_i(\mathbf{x}) = \sum_{j=1}^{n} c_j^i \psi_\ell(\mathbf{x}, \mathbf{x}_j^i) + \sum_{k=1}^{\ell^2} b_k p_k(\mathbf{x})$$

Local Lagrange functions on the sphere

$X = \{\mathbf{x}_j\}_{j=1}^{N} \subset \mathbb{S}^2$



Algorithm: For $i = 1, \ldots, N$

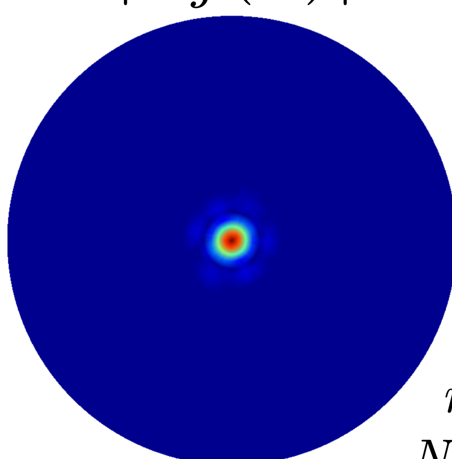
1. Choose $n << N$ nearest neighbors to $\mathbf{x}_i$:

   $X_i = \{\mathbf{x}_j^i\}_{j=1}^{n} \subset X$

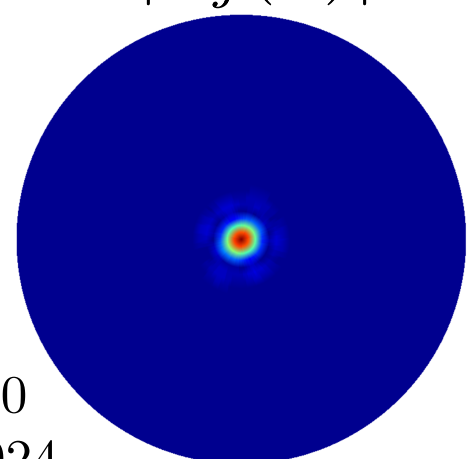
2. Construct the local Lagrange function on X:

$$\widetilde{L}_i(\mathbf{x}) = \sum_{j=1}^{n} c_j^i \psi_\ell(\mathbf{x}, \mathbf{x}_j^i) + \sum_{k=1}^{\ell^2} b_k p_k(\mathbf{x})$$

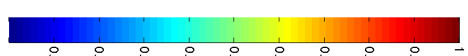
$|L_j(\mathbf{x})|$



$|\widetilde{L}_j(\mathbf{x})|$

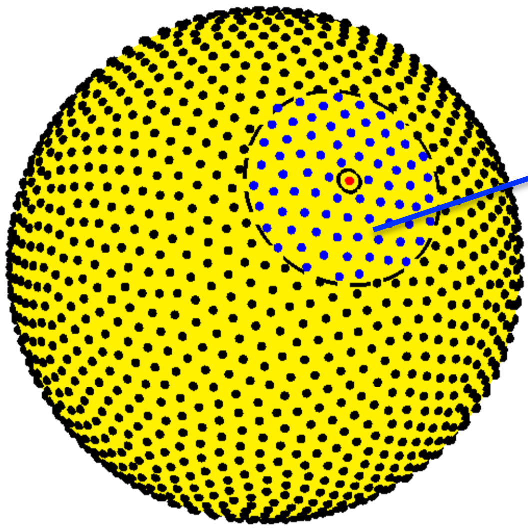


n=70
N=1024



Local Lagrange functions on the sphere

$X = \{\mathbf{x}_j\}_{j=1}^{N} \subset \mathbb{S}^2$



Algorithm: For $i = 1, \ldots, N$

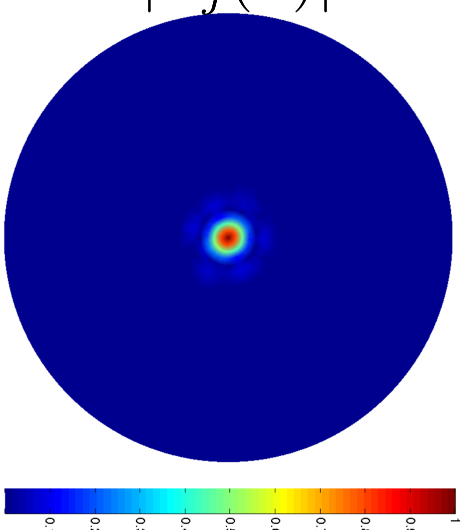
1. Choose $n << N$ nearest neighbors to $\mathbf{x}_i$:

   $X_i = \{\mathbf{x}_j^i\}_{j=1}^{n} \subset X$

2. Construct the local Lagrange function on X:

$$\widetilde{L}_i(\mathbf{x}) = \sum_{j=1}^{n} c_j^i \psi_\ell(\mathbf{x}, \mathbf{x}_j^i) + \sum_{k=1}^{\ell^2} b_k p_k(\mathbf{x})$$

$|\widetilde{L}_j(\mathbf{x})|$



Estimates: (FHNWW, 2013)

If each $\widetilde{L}_j(\mathbf{x})$ is constructed from $n = M(\log N)^2$ neighbors then

$$\|\widetilde{L}_j - L_j\|_\infty \le C\, h_X^J$$

$$|L_j(\mathbf{x})| \le C\big(1 + \mathrm{dist}(\mathbf{x}, \mathbf{x}_j)/h_X\big)^{-J}$$

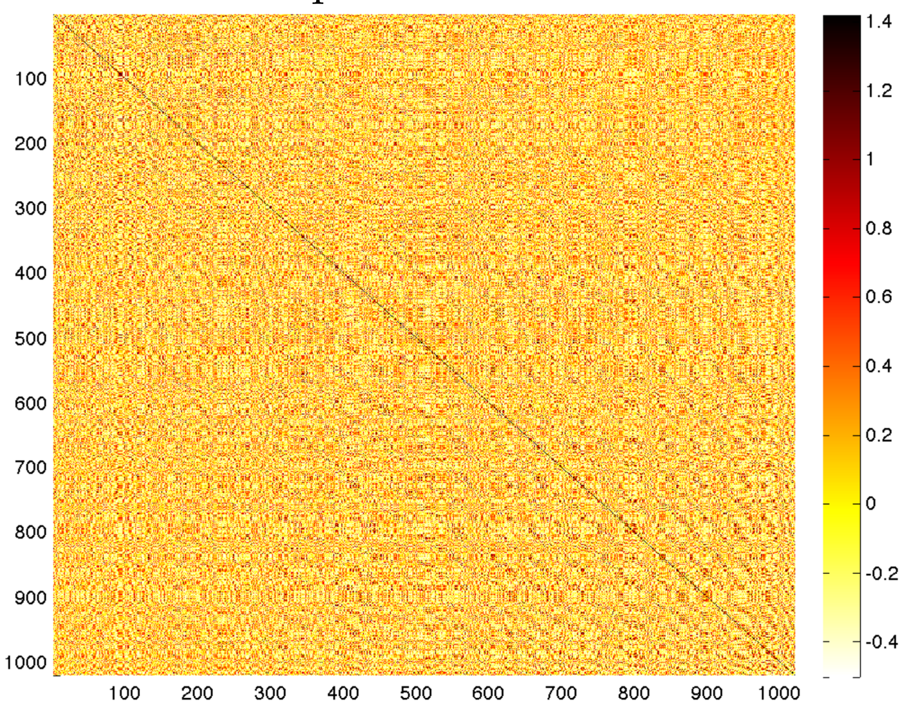
Interpolation matrices

- Example: N=1024, n=70

Standard basis:

$$s(\mathbf{x}) = \sum_{j=1}^{N} c_j \psi_\ell(\mathbf{x},\mathbf{x}_j) + \sum_{k=1}^{\ell^2} b_k p_k(\mathbf{x})$$

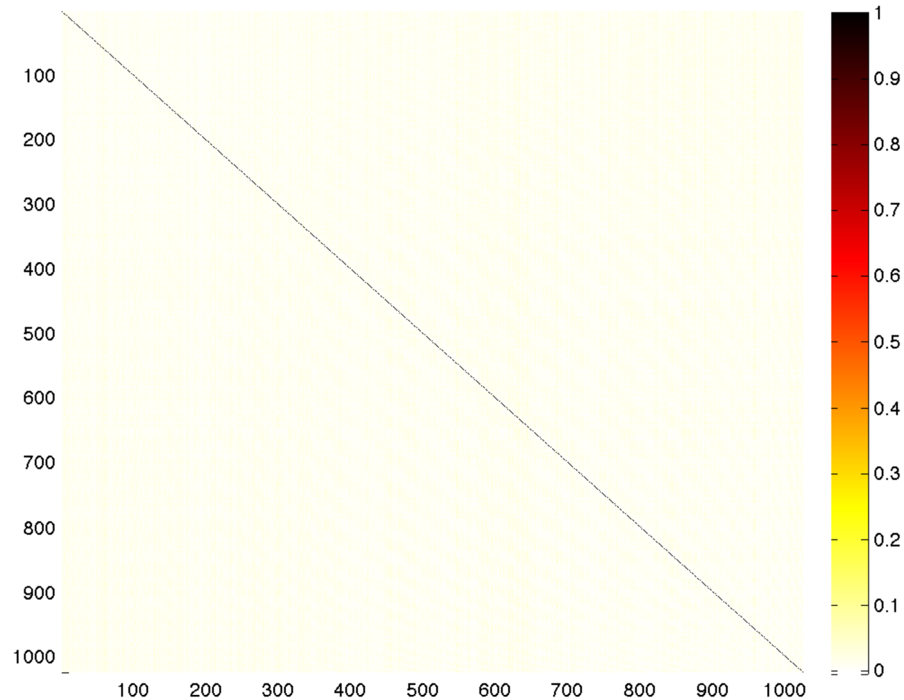
Interpolation matrix



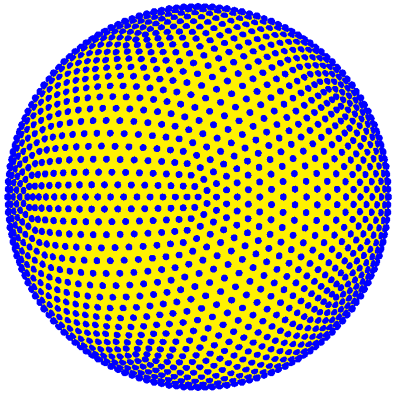
Approximate Lagrange basis:

$$s(\mathbf{x}) = \sum_{j=1}^{N} a_j \widetilde{L}_j(\mathbf{x})$$

Interpolation matrix

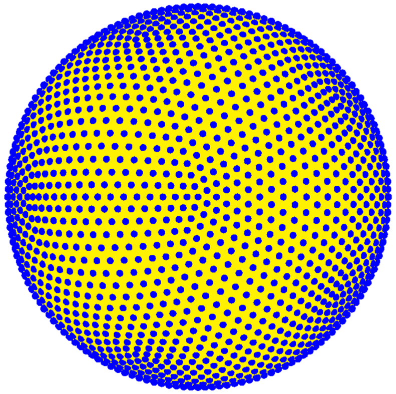


Solving "preconditioned" systems



- Numerical experiment: $\quad s(\mathbf{x}) = \sum_{j=1}^{N} a_j \widetilde{L}_j(\mathbf{x})$
  - Target f: random values distributed between $[-1, 1]$.
  - $\widetilde{L}_j$ constructed from $n = 7\lceil (\log_{10} N)^2 \rceil$ neighbors
  - Systems solved using GMRES iterative method (Saad & Schultz, 1986)

Solving "preconditioned" systems



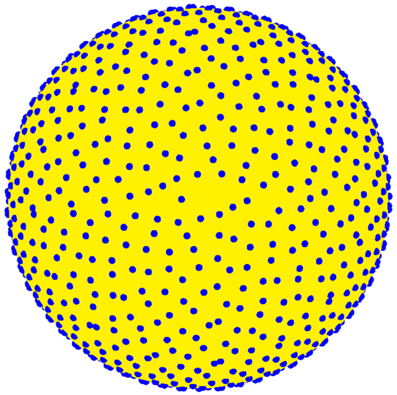
- Numerical experiment: $s(\mathbf{x}) = \sum_{j=1}^{N} a_j \widetilde{L}_j(\mathbf{x})$
  - Target f: random values distributed between $[-1, 1]$.
  - $\widetilde{L}_j$ constructed from $n = 7\lceil(\log_{10} N)^2\rceil$ neighbors
  - Systems solved using GMRES iterative method (Saad & Schultz, 1986)

			Number GMRES iterations			
			$tol = 10^{-6}$	$tol = 10^{-8}$	$tol = 10^{-10}$	$tol = 10^{-12}$
N	n	ρ_X	Icosahedral nodes			
2562	84	1.650	7	8	9	10
10242	119	1.679	5	7	8	9
23042	140	1.688	6	7	8	9
40962	154	1.693	5	7	7	8
92162	175	1.688	6	8	9	10
163842	196	1.701	5	7	7	8

Note: Each iteration takes $\mathcal{O}(N^2)$ operations, but may be reduced to $\mathcal{O}(N \log N)$ using NFFT (Keiner, Kunis, & Potts, 2006).

Solving “preconditioned” systems

- Numerical experiment: $s(\mathbf{x}) = \displaystyle\sum_{j=1}^{N} a_j \widetilde{L}_j(\mathbf{x})$
  - Target f: random values distributed between $[-1, 1]$.
  - $\widetilde{L}_j$ constructed from $n = 7\lceil(\log_{10} N)^2\rceil$ neighbors
  - Systems solved using GMRES iterative method (Saad & Schultz, 1986)

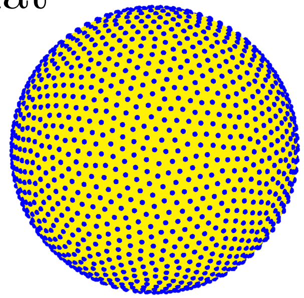


			Number GMRES iterations			
			$tol = 10^{-6}$	$tol = 10^{-8}$	$tol = 10^{-10}$	$tol = 10^{-12}$
N	n	ρ_X	Hammersley nodes			
4000	91	24.56	8	10	11	12
8000	112	34.74	8	9	11	12
16000	126	49.13	7	9	10	11
32000	147	69.48	7	8	10	11
64000	168	98.26	7	9	10	12

Note: Also appears to work well for less uniform nodes, but no theory (yet!).

Application: Quadrature on the sphere

- Problem: Given $X = \{\mathbf{x}\}_{j=1}^{N} \subset \mathbb{S}^2$, find weights $\{w_j\}_{j=1}^{N}$ such that



$$\int_{\mathbb{S}^2} f(\mathbf{x}) d\mu(\mathbf{x}) \approx \sum_{j=1}^{N} w_j f(\mathbf{x}_j) =: Q(f), \quad f \in C(\mathbb{S}^2)$$

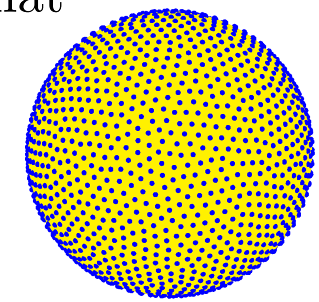
- Our solution: Find the weights from the kernel interpolant of f on X:

$$\int_{\mathbb{S}^2} f(\mathbf{x}) d\mu(\mathbf{x}) \approx \int_{\mathbb{S}^2} s_X(\mathbf{x}) d\mu(\mathbf{x})$$

- So what are the weights?

Application: Quadrature on the sphere

- Problem: Given $X = \{\mathbf{x}\}_{j=1}^{N} \subset \mathbb{S}^2$, find weights $\{w_j\}_{j=1}^{N}$ such that



$$\int_{\mathbb{S}^2} f(\mathbf{x}) d\mu(\mathbf{x}) \approx \sum_{j=1}^{N} w_j f(\mathbf{x}_j) =: Q(f), \quad f \in C(\mathbb{S}^2)$$

- Our solution: Find the weights from the kernel interpolant of f on X:

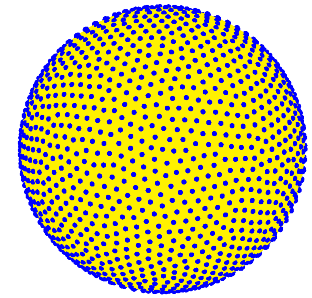
$$\int_{\mathbb{S}^2} f(\mathbf{x}) d\mu(\mathbf{x}) \approx \int_{\mathbb{S}^2} s_X(\mathbf{x}) d\mu(\mathbf{x})$$

- So what are the weights?

$$\text{Lagrange form}: \int_{\mathbb{S}^2} s_X(\mathbf{x}) d\mu(\mathbf{x}) = \sum_{j=1}^{N} \underbrace{\left(\int_{\mathbb{S}^2} L_j(\mathbf{x}) d\mu(\mathbf{x}) \right)}_{w_j} f_j$$

- How can this be made computationally tractable for large N?

Quadrature on the sphere



- For simplicity suppose ψ is a positive definite zonal kernel, i.e.

$$\implies \psi(\mathbf{x},\mathbf{y}) = \psi(\mathbf{x}\cdot\mathbf{y})$$

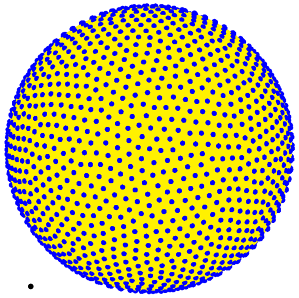
- Thus

$$\int_{\mathbb{S}^2} s_X(\mathbf{x})d\mu(\mathbf{x}) = \sum_{j=1}^{N}\left(\int_{\mathbb{S}^2}\psi(\mathbf{x}\cdot\mathbf{x}_j)d\mu(\mathbf{x})\right)c_j$$

$$= \left(\int_{\mathbb{S}^2}\psi(\mathbf{x}\cdot\mathbf{x}_1)d\mu(\mathbf{x})\right)\sum_{j=1}^{N}c_j = J\begin{bmatrix}1 & 1 & \cdots & 1\end{bmatrix}\begin{bmatrix}c_1\\ c_2\\ \vdots\\ c_N\end{bmatrix}$$

$$= \underbrace{J\begin{bmatrix}1 & 1 & \cdots & 1\end{bmatrix}\begin{bmatrix} & & \\ & \psi(\mathbf{x}_i\cdot\mathbf{x}_j) & \\ & & \end{bmatrix}^{-1}}_{\begin{bmatrix}w_1 & w_2 & \cdots & w_N\end{bmatrix}}\begin{bmatrix}f_1\\ f_2\\ \vdots\\ f_N\end{bmatrix}$$

Quadrature on the sphere



- Problem: Given $X = \{\mathbf{x}\}_{j=1}^{N} \subset \mathbb{S}^2$, find weights $\{w_j\}_{j=1}^{N}$ such that

$$\int_{\mathbb{S}^2} f(\mathbf{x})d\mu(\mathbf{x}) \approx \sum_{j=1}^{N} w_j f(\mathbf{x}_j) =: Q(f), \quad f \in C(\mathbb{S}^2)$$

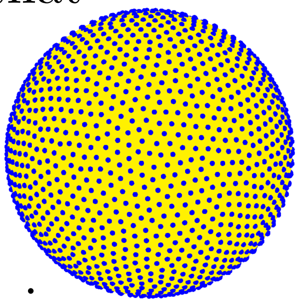
- Our solution: Find the weights from the kernel interpolant of $f\Big|_X$:

$$\begin{bmatrix} & & \\ & \psi(\mathbf{x}_i \cdot \mathbf{x}_j) & \\ & & \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_N \end{bmatrix} = \begin{bmatrix} J \\ \vdots \\ J \end{bmatrix}$$

Quadrature on the sphere

- Problem: Given $X = \{\mathbf{x}\}_{j=1}^{N} \subset \mathbb{S}^2$, find weights $\{w_j\}_{j=1}^{N}$ such that

$$\int_{\mathbb{S}^2} f(\mathbf{x})d\mu(\mathbf{x}) \approx \sum_{j=1}^{N} w_j f(\mathbf{x}_j) =: Q(f), \quad f \in C(\mathbb{S}^2)$$



- Our solution: Find the weights from the kernel interpolant of $f\Big|_X$:

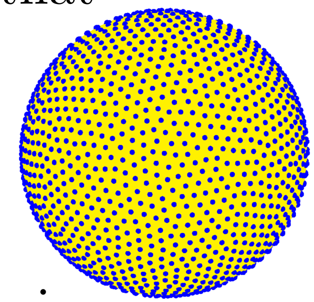
$$\begin{bmatrix} & & \\ & \psi(\mathbf{x}_i \cdot \mathbf{x}_j) & \\ & & \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_N \end{bmatrix} = \begin{bmatrix} J \\ \vdots \\ J \end{bmatrix}$$

- Note that this idea can be extended to CPD kernels as well: (FHNWW-2, 2013)

$$s_X(\mathbf{x}) = \sum_{j=1}^{N} c_j \psi_\ell(\mathbf{x} \cdot \mathbf{x}_j) + \sum_{k=1}^{\ell^2} b_k p_k(\mathbf{x}), \quad \psi_\ell(t) = (1-t)^{\ell-1}\log(1-t)$$

Error: $$\left| \int_{\mathbb{S}^2} f(\mathbf{x})d\mu(\mathbf{x}) - \sum_{j=1}^{N} w_j f_j \right| \leq \begin{cases} h_X^r \|f\|_{C^r(\mathbb{S}^2)} & 0 < r \leq 2\ell \\ h_X^r \|f\|_{H^r(\mathbb{S}^2)} & 1 < r \leq \ell \end{cases}$$

Quadrature on the sphere



- Problem: Given $X = \{\mathbf{x}\}_{j=1}^{N} \subset \mathbb{S}^2$, find weights $\{w_j\}_{j=1}^{N}$ such that

$$\int_{\mathbb{S}^2} f(\mathbf{x})d\mu(\mathbf{x}) \approx \sum_{j=1}^{N} w_j f(\mathbf{x}_j) =: Q(f), \quad f \in C(\mathbb{S}^2)$$

- Our solution: Find the weights from the kernel interpolant of $f\big|_X$:

$$\begin{bmatrix} & & \\ & \psi(\mathbf{x}_i \cdot \mathbf{x}_j) & \\ & & \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_N \end{bmatrix} = \begin{bmatrix} J \\ \vdots \\ J \end{bmatrix}$$

- Note that this idea can be extended to CPD kernels as well: (FHNWW-2, 2013)

$$s_X(\mathbf{x}) = \sum_{j=1}^{N} c_j \psi_\ell(\mathbf{x} \cdot \mathbf{x}_j) + \sum_{k=1}^{\ell^2} b_k p_k(\mathbf{x}), \quad \psi_\ell(t) = (1-t)^{\ell-1}\log(1-t)$$

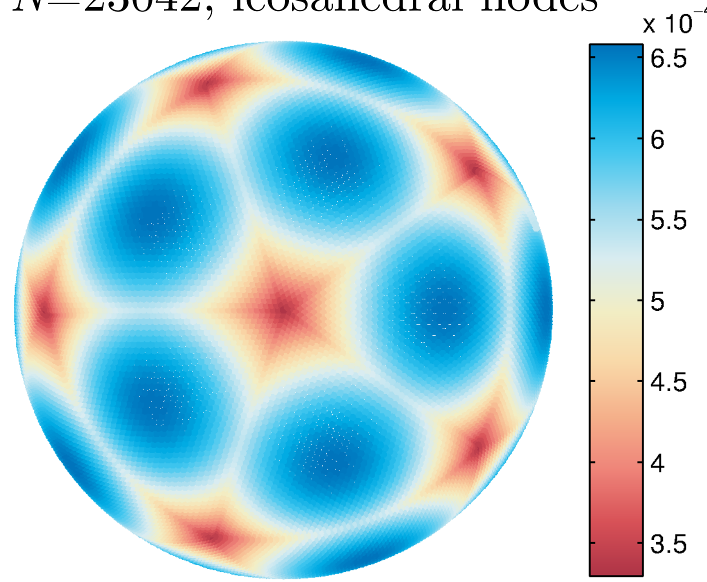
- How ELSE can this be made computationally tractable for large N?

Local Lagrange basis!

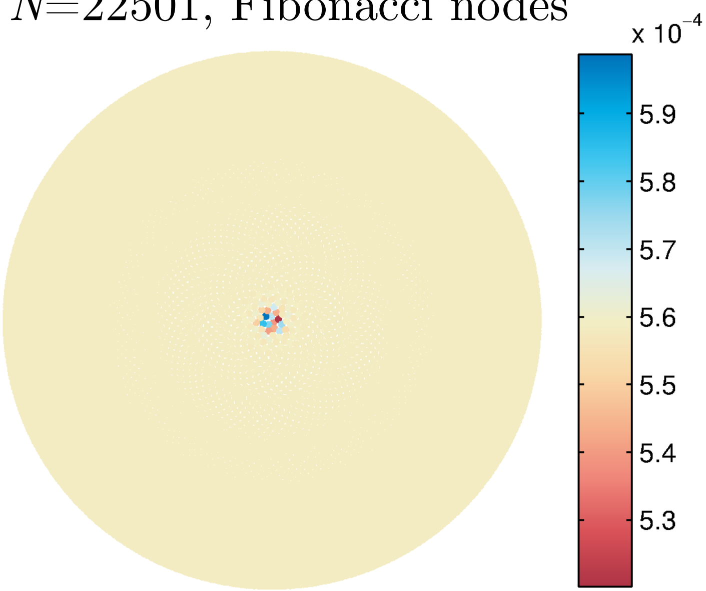
Example of quadrature weights

- Quadrature weights computed using
$$\psi_2(t) = (1-t)\log(1-t)$$
- Local Lagrange preconditioner

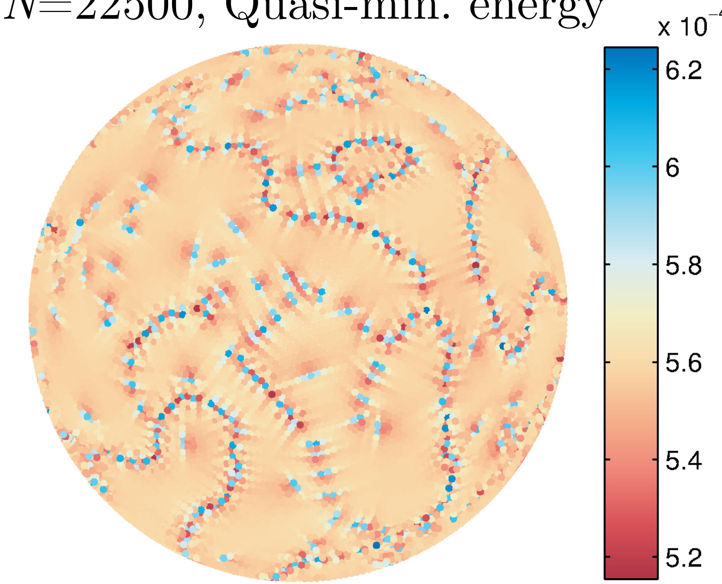
N=23042, icosahedral nodes



N=22501, Fibonacci nodes

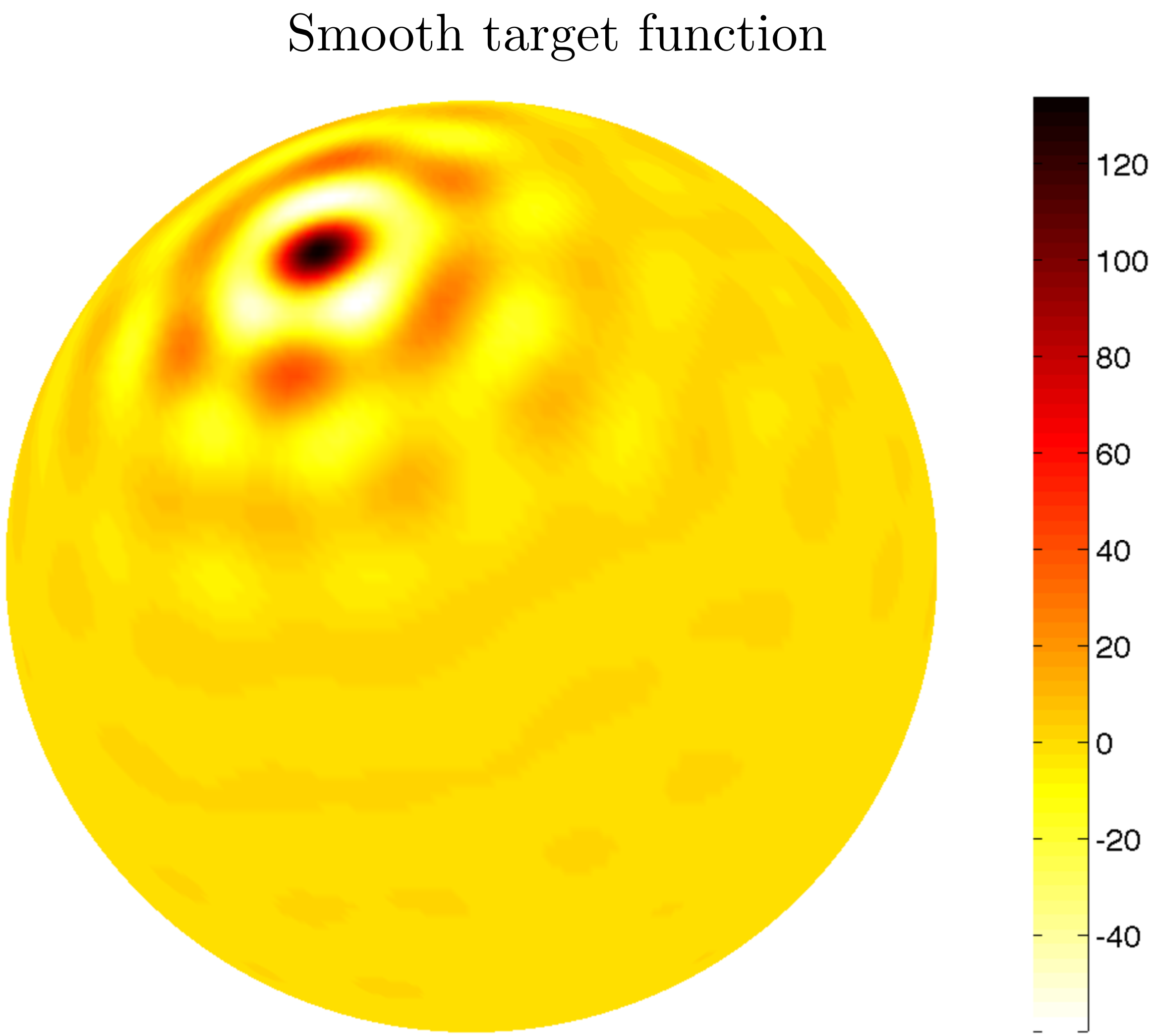


N=22500, Quasi-min. energy



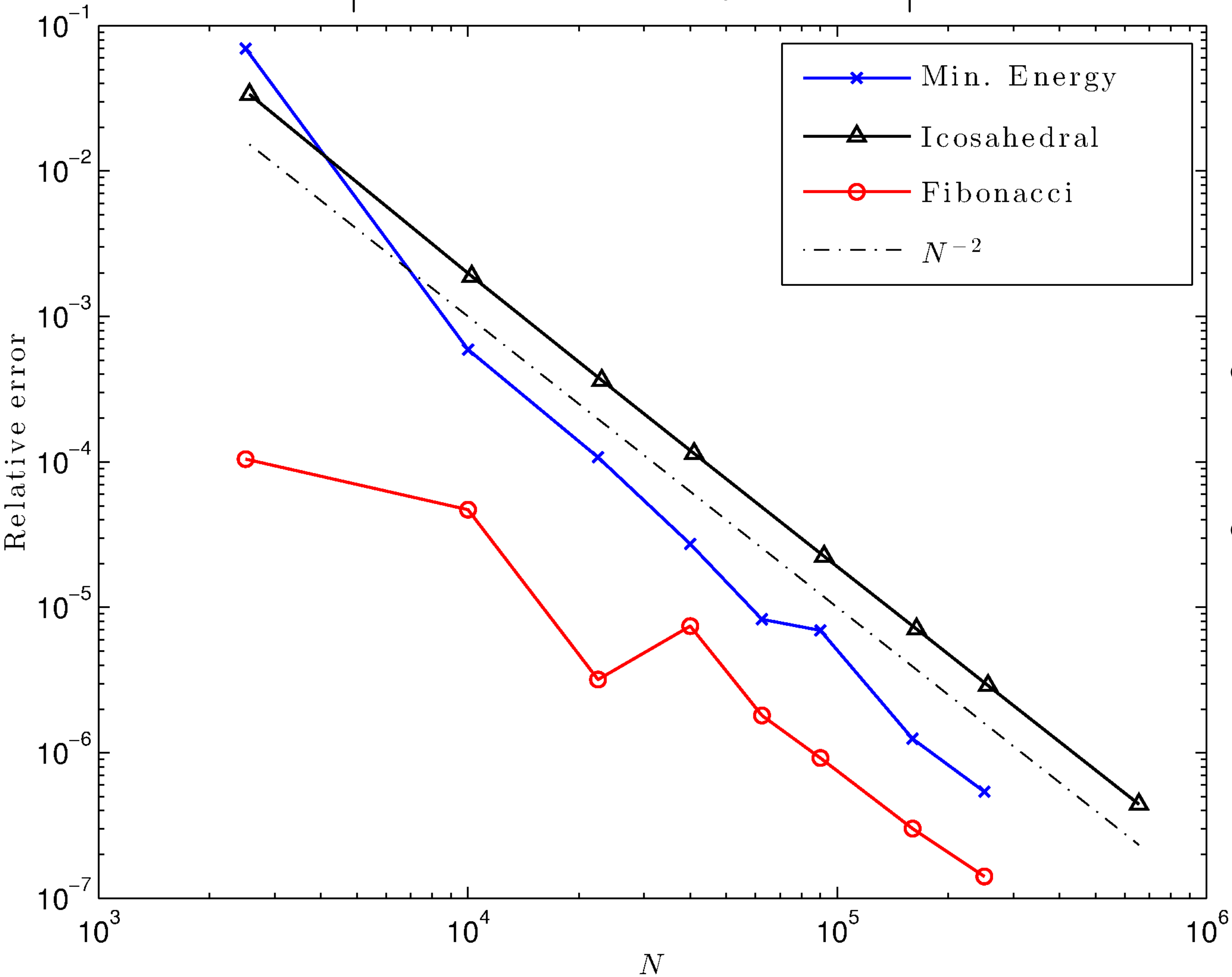
Borodachov, Hardin, Saff (2013)

Numerical example



Numerical example

$$\left| \int_{\mathbb{S}^2} f(\mathbf{x}) d\mu(\mathbf{x}) - \sum_{j=1}^{N} w_j f_j \right|$$



- Weights computed up to $N = 650000$
- Convergence is $\mathcal{O}(h_X^4)$.

Summary and future

- Local Lagrange basis appears to provide a good bases for certain kernel spaces on $\mathbb{S}^2$.
  - Can be computed using $\mathcal{O}(N(\log N)^2)$ nearest neighbors.
  - Computation is embarrassingly parallel.
  - Works very well as a preconditioner for global interpolation problem.
  - Future: implement fast *evaluation* algorithm to reduce the global cost.
- Future: exploit local Lagrange basis as a quasi-interpolant
- Future: develop theory and numerics to handle non-uniform nodes.
- Local lagrange basis can be extended to other manifolds:
  - Future: computable kernels for general manifolds.
- Future: Can more be said about smooth kernels on manifolds: error estimates and good bases (or stable algorithms).

Thank you!