Brief summary of Finite Difference methods

Presented at NSF-CBMS Regional Research Conference
University of Massachusetts, Dartmouth, by

Bengt Fornberg and Natasha Flyer

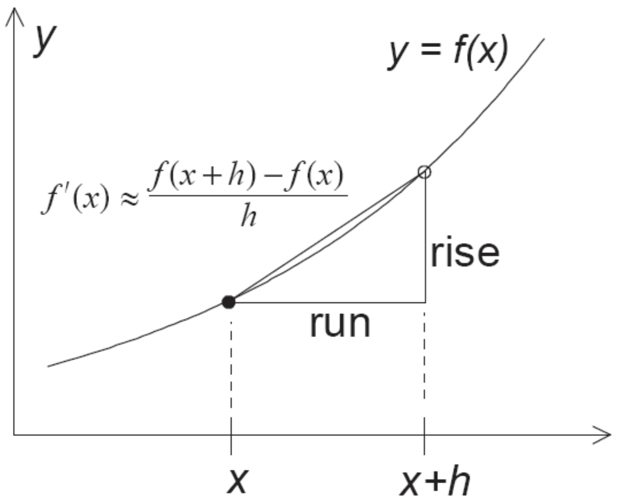
Introduction to FD formulas:

Definition of first derivative; simplest FD formula:

$$\frac{df}{dx} = \lim_{h\to 0} \frac{f(x+h)-f(x)}{h}$$

Corresponds to 1st order FD approximation; Taylor expand:

$$f'(x) = \frac{f(x+h)-f(x)}{h} + O(h^1)$$



General approach to calculate FD formulas:

With nodes $\{x_1, x_2, \ldots, x_n\}$, the FD *weights* $\{w_1, w_2, \ldots, w_n\}$ for approximating the operator *L* at the location $x = x_c$ can be obtained by solving the linear system

$$\begin{bmatrix} 1 & 1 & \cdots & 1 \\ x_1 & x_2 & \cdots & x_n \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{n-1} & x_2^{n-1} & \cdots & x_n^{n-1} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} = \begin{bmatrix} L\,1|_{x=x_c} \\ L\,x|_{x=x_c} \\ \vdots \\ L\,x^{n-1}|_{x=x_c} \end{bmatrix}.$$

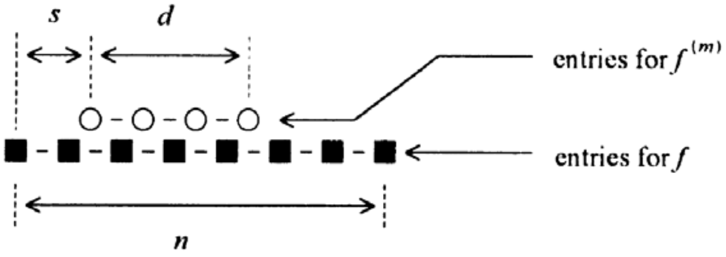
The successive lines enforce that the weights lead to the correct results for the functions $1, x, x^2, \ldots, x^{n-1}$, and thus, by linearity, for all polynomials up through degree $n-1$.

This approach generalized to RBF-FD stencils.

Numerous more effective approaches are available for creating FD formulas:

Padé - based algorithm:

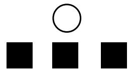
Fornberg (1998)



In Mathematica:

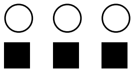
t = PadeApproximant[x^{s}(Log[x]/h)^{m},{x,1,{n,d}}];
CeofficientList[{Denominator[t],Numerator[t]},x]

Examples:



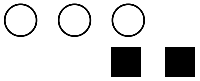
s=1,d=0,n=2,m=2 $\{\{h^2\},\{1,-2,1\}\}$

$$f''(x)=\{f(x-h)-2f(x)+f(x+h)\}/h^2$$



s=1,d=2,n=2,m=2 $\left\{\left\{\frac{h^2}{12},\frac{5h^2}{6},\frac{h^2}{12}\right\},\{1,-2,1\}\right\}$

$$\tfrac{1}{12}f''(x-h)+\tfrac{5}{6}f''(x)+\tfrac{1}{12}f''(x+h)=\{f(x-h)-2f(x)+f(x+h)\}/h^2$$



s=-2,d=2,n=1,m=1 $\left\{\left\{\frac{5h}{12},-\frac{4h}{3},\frac{23h}{12}\right\},\{-1,1\}\right\}$

$$f(x+h)=f(x)+\frac{h}{12}(23f'(x)-16f'(x-h)+5f'(x-2h))$$ AB3

Adams-Bashforth	(ABp)	s = 1-p,	d = p-1,	n = 1,	m = 1,
Adams-Moulton	(AMp)	s = 2-p,	d = p-1,	n = 1,	m = 1,
Backward Differentiation	(BDp)	s = p,	d = 0,	n = p,	m = 1.

Example of algorithm for arbitrarily spaced nodes:

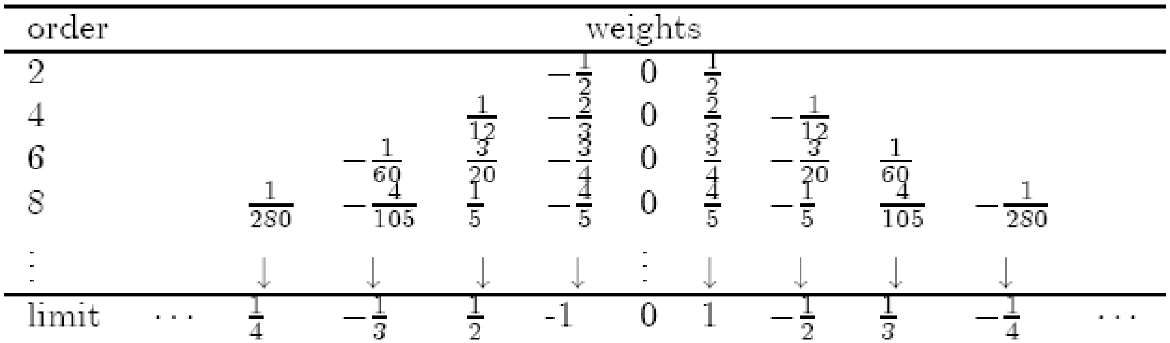
```
function c=weights(z,x,m)
% Calculates FD weights. The parameters are:
  %  z   location where approximations are to be accurate,
 %  x   vector with x-coordinates for grid points,
 %  m   highest derivative that we want to find weights for
 %  c   array size m+1,lentgh(x) containing (as output) in
 %      successive rows the weights for derivatives 0,1,...,m.
 n=length(x); c=zeros(m+1,n); c1=1; c4=x(1)-z; c(1,1)=1;
 for i=2:n
    mn=min(i,m+1); c2=1; c5=c4; c4=x(i)-z;
     for j=1:i-1
        c3=x(i)-x(j);  c2=c2*c3;
        if j==i-1
           c(2:mn,i)=c1*((1:mn-1)'.*c(1:mn-1,i-1)-c5*c(2:mn,i-1))/c2;
           c(1,i)=-c1*c5*c(1,i-1)/c2;
        end
        c(2:mn,j)=(c4*c(2:mn,j)-(1:mn-1)'.*c(1:mn-1,j))/c3;
        c(1,j)=c4*c(1,j)/c3;
     end
     c1=c2;
  end
```

For example, the statement **weights(0,-2:2,6)** returns the output

```
 0         0         1.0000    0         0
 0.0833   -0.6667    0         0.6667   -0.0833
-0.0833    1.3333   -2.5000    1.3333   -0.0833
-0.5000    1.0000    0        -1.0000    0.5000
 1.0000   -4.0000    6.0000   -4.0000    1.0000
 0         0         0         0         0
 0         0         0         0         0
```

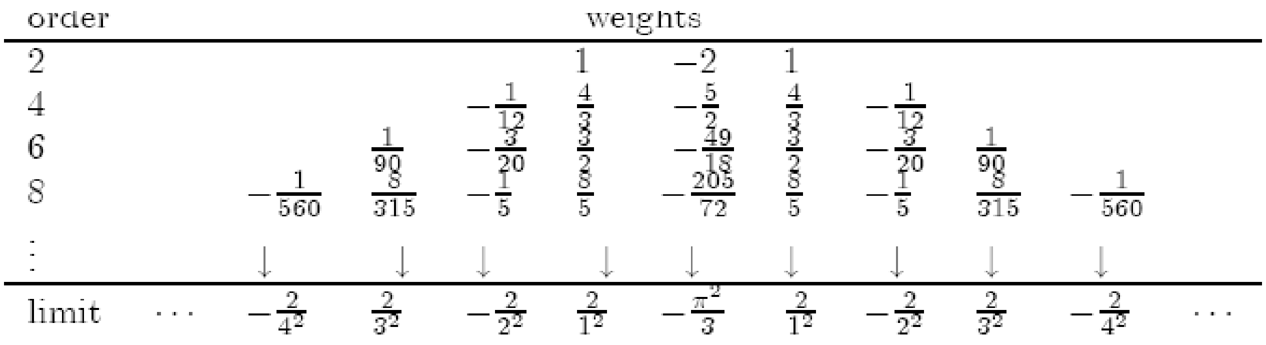
Examples of tables for centered FD approximations:

First derivative ⇒



order	weights										
2					$-\frac{1}{2}$	0	$\frac{1}{2}$				
4				$\frac{1}{12}$	$-\frac{2}{3}$	0	$\frac{2}{3}$	$-\frac{1}{12}$			
6			$-\frac{1}{60}$	$\frac{3}{20}$	$-\frac{3}{4}$	0	$\frac{3}{4}$	$-\frac{3}{20}$	$\frac{1}{60}$		
8		$\frac{1}{280}$	$-\frac{4}{105}$	$\frac{1}{5}$	$-\frac{4}{5}$	0	$\frac{4}{5}$	$-\frac{1}{5}$	$\frac{4}{105}$	$-\frac{1}{280}$	
⋮		↓	↓	↓	↓	⋮	↓	↓	↓	↓	
limit	⋯	$\frac{1}{4}$	$-\frac{1}{3}$	$\frac{1}{2}$	-1	0	1	$-\frac{1}{2}$	$\frac{1}{3}$	$-\frac{1}{4}$	⋯

Second derivative ⇒



order	weights										
2					1	-2	1				
4				$-\frac{1}{12}$	$\frac{4}{3}$	$-\frac{5}{2}$	$\frac{4}{3}$	$-\frac{1}{12}$			
6			$\frac{1}{90}$	$-\frac{3}{20}$	$\frac{3}{2}$	$-\frac{49}{18}$	$\frac{3}{2}$	$-\frac{3}{20}$	$\frac{1}{90}$		
8		$-\frac{1}{560}$	$\frac{8}{315}$	$-\frac{1}{5}$	$\frac{8}{5}$	$-\frac{205}{72}$	$\frac{8}{5}$	$-\frac{1}{5}$	$\frac{8}{315}$	$-\frac{1}{560}$	
⋮		↓	↓	↓	↓	↓	↓	↓	↓	↓	
limit	⋯	$-\frac{2}{4^2}$	$\frac{2}{3^2}$	$-\frac{2}{2^2}$	$\frac{2}{1^2}$	$-\frac{\pi^2}{3}$	$\frac{2}{1^2}$	$-\frac{2}{2^2}$	$\frac{2}{3^2}$	$-\frac{2}{4^2}$	⋯

Note that the limits of increasing orders exist and take very simple forms.

Examples of FD approximations for the Laplacian in 2-D and 3-D:

2-D

$$\begin{bmatrix} & 1 & \\ 1 & -4 & 1 \\ & 1 & \end{bmatrix} / h^2 \qquad O(h^2)\ ; \qquad \begin{bmatrix} & & -\frac{1}{12} & & \\ & & \frac{4}{3} & & \\ -\frac{1}{12} & \frac{4}{3} & -5 & \frac{4}{3} & -\frac{1}{12} \\ & & \frac{4}{3} & & \\ & & -\frac{1}{12} & & \end{bmatrix} / h^2 \qquad O(h^4)$$

$$\begin{bmatrix} 1 & 4 & 1 \\ 4 & -20 & 4 \\ 1 & 4 & 1 \end{bmatrix} / (6h^2) \quad \begin{cases} O(h^2) & \text{when approximating } \Delta u \\ O(h^4) & \text{when approximating a solution to } \Delta u = f \text{ by means of} \\ O(h^6) & \text{when approximating a solution to } \Delta u = 0. \end{cases} \quad \begin{bmatrix} 1 & 4 & 1 \\ 4 & -20 & 4 \\ 1 & 4 & 1 \end{bmatrix} u/(6h^2) = \begin{bmatrix} & 1 & \\ 1 & 8 & 1 \\ & 1 & \end{bmatrix} f/12.$$

3-D

$$\begin{bmatrix} & & [0\ \ 1\ \ 0] \\ & [1\ \ 2\ \ 1] & \\ [0\ \ 1\ \ 0] & & \\ -\ -\ - & & -\ - \\ & & [1\ \ 2\ \ 1] \\ & [2\ \ -24\ \ 2] & \\ [1\ \ 2\ \ 1] & & \\ -\ -\ - & & -\ - \\ & & [0\ \ 1\ \ 0] \\ & [1\ \ 2\ \ 1] & \\ [0\ \ 1\ \ 0] & & \end{bmatrix} / (6h^2) \begin{cases} O(h^2) & \text{when approximating } \Delta u \\ O(h^4) & \text{when approximating a solution to } \Delta u = f \text{ by means of} \\ O(h^4) & \text{when approximating a solution to } \Delta u = 0. \end{cases}$$

$$\begin{bmatrix} & & [0\ \ 1\ \ 0] \\ & [1\ \ 2\ \ 1] & \\ [0\ \ 1\ \ 0] & & \\ -\ -\ - & & -\ - \\ & & [1\ \ 2\ \ 1] \\ & [2\ \ -24\ \ 2] & \\ [1\ \ 2\ \ 1] & & \\ -\ -\ - & & -\ - \\ & & [0\ \ 1\ \ 0] \\ & [1\ \ 2\ \ 1] & \\ [0\ \ 1\ \ 0] & & \end{bmatrix} u/(6h^2) = \begin{bmatrix} & & [0\ \ 0\ \ 0] \\ & [0\ \ 1\ \ 0] & \\ [0\ \ 0\ \ 0] & & \\ -\ -\ - & & -\ - \\ & & [0\ \ 1\ \ 0] \\ & [1\ \ 6\ \ 1] & \\ [0\ \ 1\ \ 0] & & \\ -\ -\ - & & -\ - \\ & & [0\ \ 0\ \ 0] \\ & [0\ \ 1\ \ 0] & \\ [0\ \ 0\ \ 0] & & \end{bmatrix} f/12$$

- Implicit (compact) approximations can increase order without increased computational cost.
- Diagonal dominance is possible also for approximations above second order accuracy.
- RBF-FD formulas will be designed to generalize approximations such as the ones above to meshfree settings (and be 'immune' to singularities, no matter how nodes are scattered).

Method of Lines (MOL) time stepping

Concept: Discretize in space and then time step by a standard ODE procedure

In case of periodic problems on equispaced lattices, von Neumann stability analysis is available.

Else, *including for RBF-FD methods*, find the eigenvalues of the spatial operator, and then choose the ODE solver and the time step *k* so the eigenvalues fall within the solver's *stability domain*.

ODE stability domain:

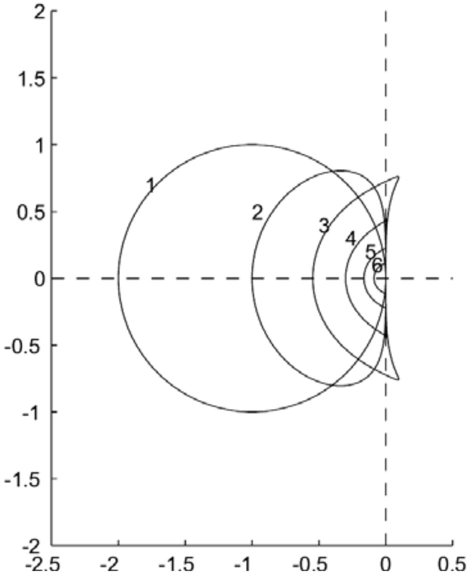
Apply the ODE solver to the constant coefficient ODE $y' = \lambda y$ and find the region in the complex $\xi = \lambda k$ for which there are no growing solutions.

Example: Find the stability domain for Forward Euler: $y(t+k) = y(t) + k\,y'(t)$.

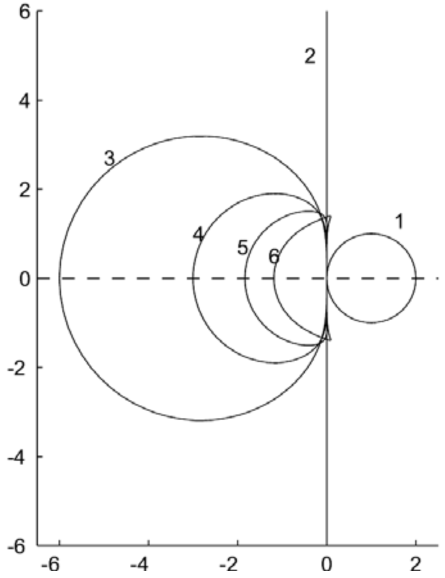
Applied to $y' = \lambda y$, the scheme becomes $y(t+k) = (1+\lambda k)y(t)$. With $\xi = \lambda k$, the condition for no-growth becomes $|1+\xi| \le 1$, i.e. a circle of radius 1 centered at $\xi = -1$.

Illustrations of the stability domains for some standard ODE solvers

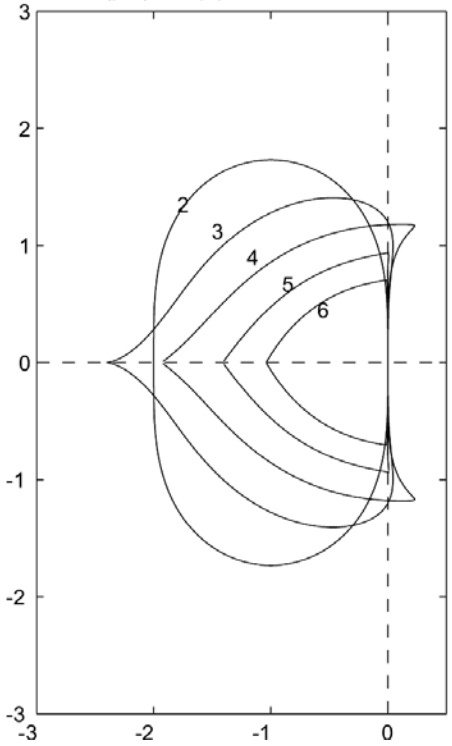
Adams-Bashforth



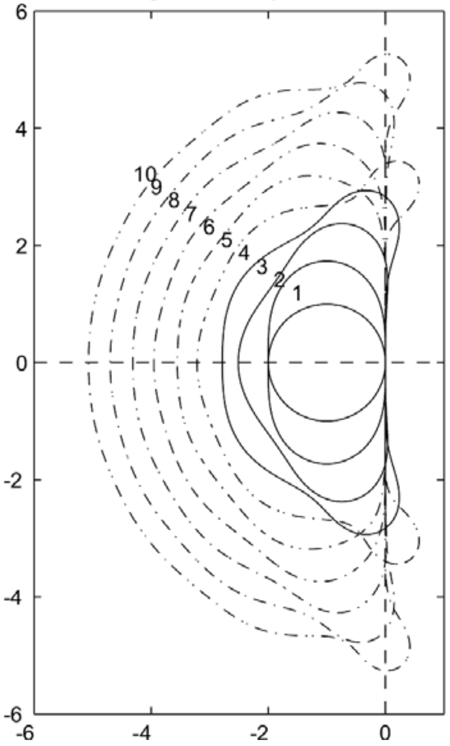
Adams-Moulton



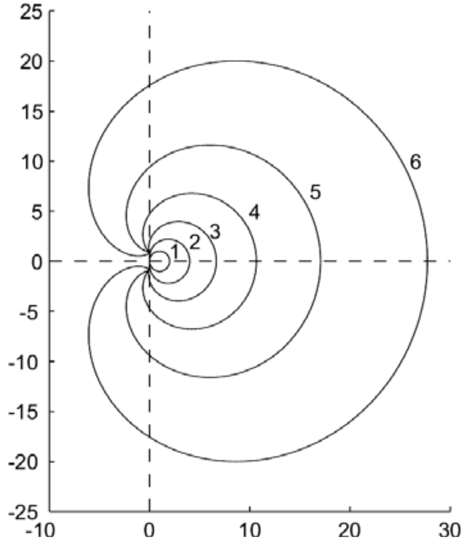
AB(p-1) / AMp predictor / corrector



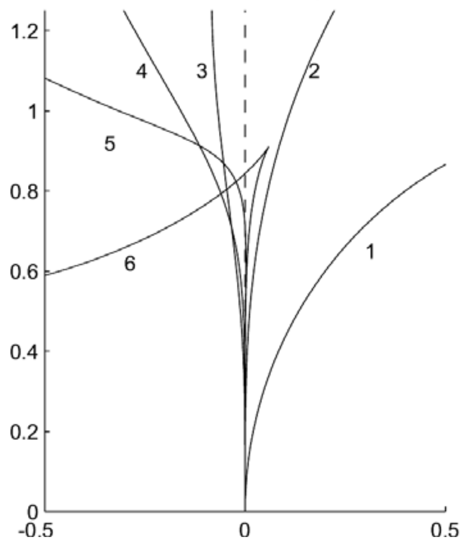
Runge-Kutta / Taylor series



BD method



Detail near the origin



Brief summary of Pseudospectral Methods

Presented at NSF-CBMS Regional Research Conference
University of Massachusetts, Dartmouth, by

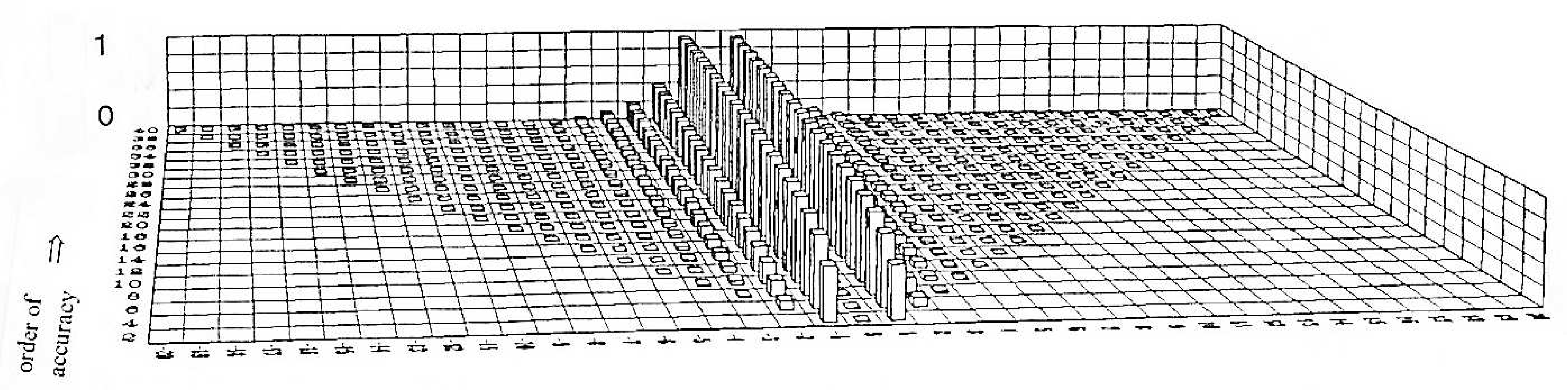
Bengt Fornberg and Natasha Flyer

Recall from the Introduction to FD Methods

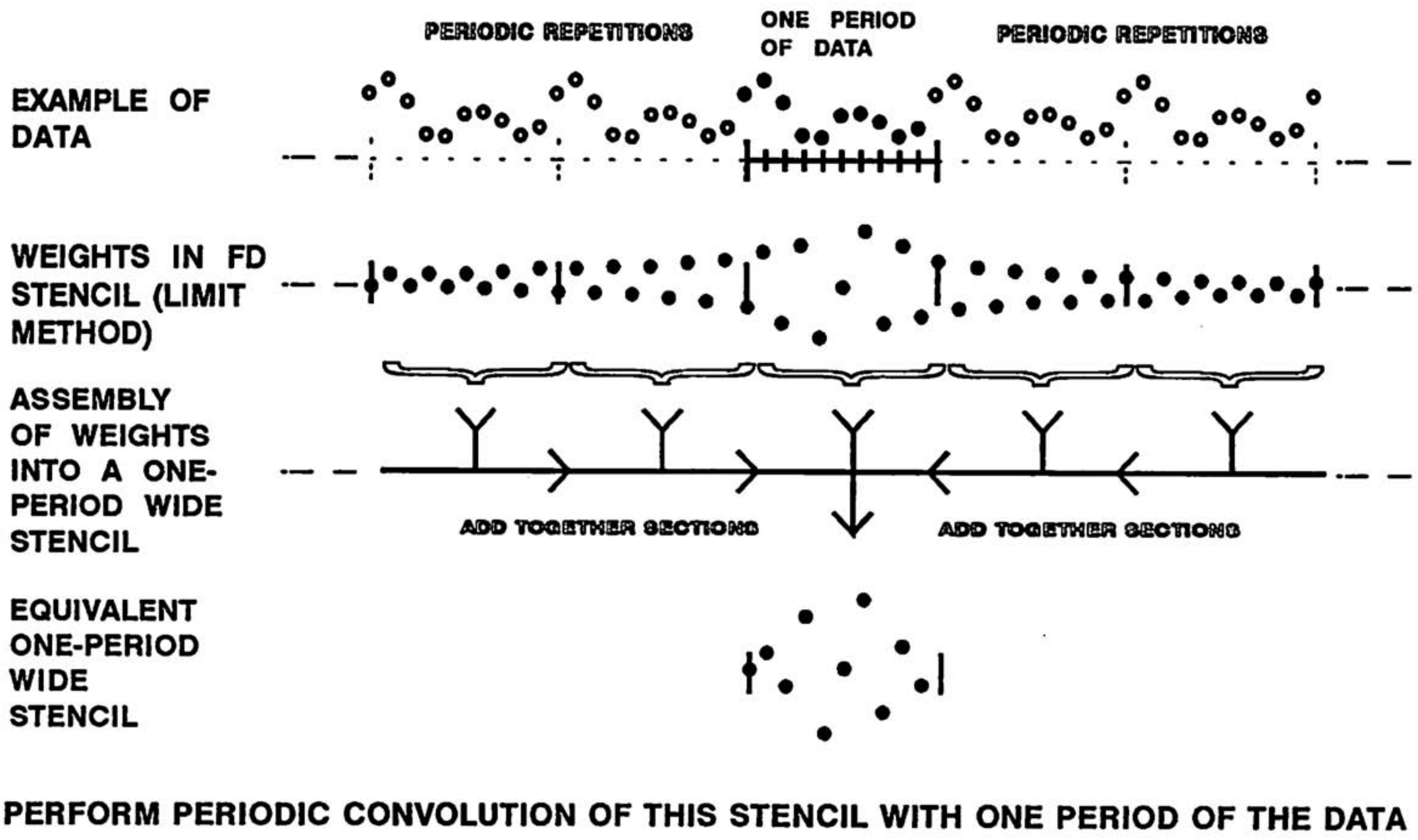
First derivative ⇒

order	weights										
2					$-\frac{1}{2}$	0	$\frac{1}{2}$				
4				$\frac{1}{12}$	$-\frac{2}{3}$	0	$\frac{2}{3}$	$-\frac{1}{12}$			
6			$-\frac{1}{60}$	$\frac{3}{20}$	$-\frac{3}{4}$	0	$\frac{3}{4}$	$-\frac{3}{20}$	$\frac{1}{60}$		
8		$\frac{1}{280}$	$-\frac{4}{105}$	$\frac{1}{5}$	$-\frac{4}{5}$	0	$\frac{4}{5}$	$-\frac{1}{5}$	$\frac{4}{105}$	$-\frac{1}{280}$	
$\vdots$		$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\vdots$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	
limit	$\cdots$	$\frac{1}{4}$	$-\frac{1}{3}$	$\frac{1}{2}$	-1	0	1	$-\frac{1}{2}$	$\frac{1}{3}$	$-\frac{1}{4}$	$\cdots$

Magnitude of weights for increasing orders of accuracy



Limiting FD method in the case of periodic data


PERIODIC REPETITIONS
ONE PERIOD OF DATA
PERIODIC REPETITIONS
EXAMPLE OF DATA
WEIGHTS IN FD STENCIL (LIMIT METHOD)
ASSEMBLY OF WEIGHTS INTO A ONE-PERIOD WIDE STENCIL
ADD TOGETHER SECTIONS
ADD TOGETHER SECTIONS
EQUIVALENT ONE-PERIOD WIDE STENCIL
PERFORM PERIODIC CONVOLUTION OF THIS STENCIL WITH ONE PERIOD OF THE DATA

The calculation can be carried out numerically as a convolution between the data and the equivalent one-period wide stencil

The result becomes equivalent to the periodic Fourier-PS method, which typically is implemented by:

i. Take FFT of data
ii. Take analytical derivative of the obtained (interpolating) trigonometric polynomial
iii. Return (by a second FFT) to physical space

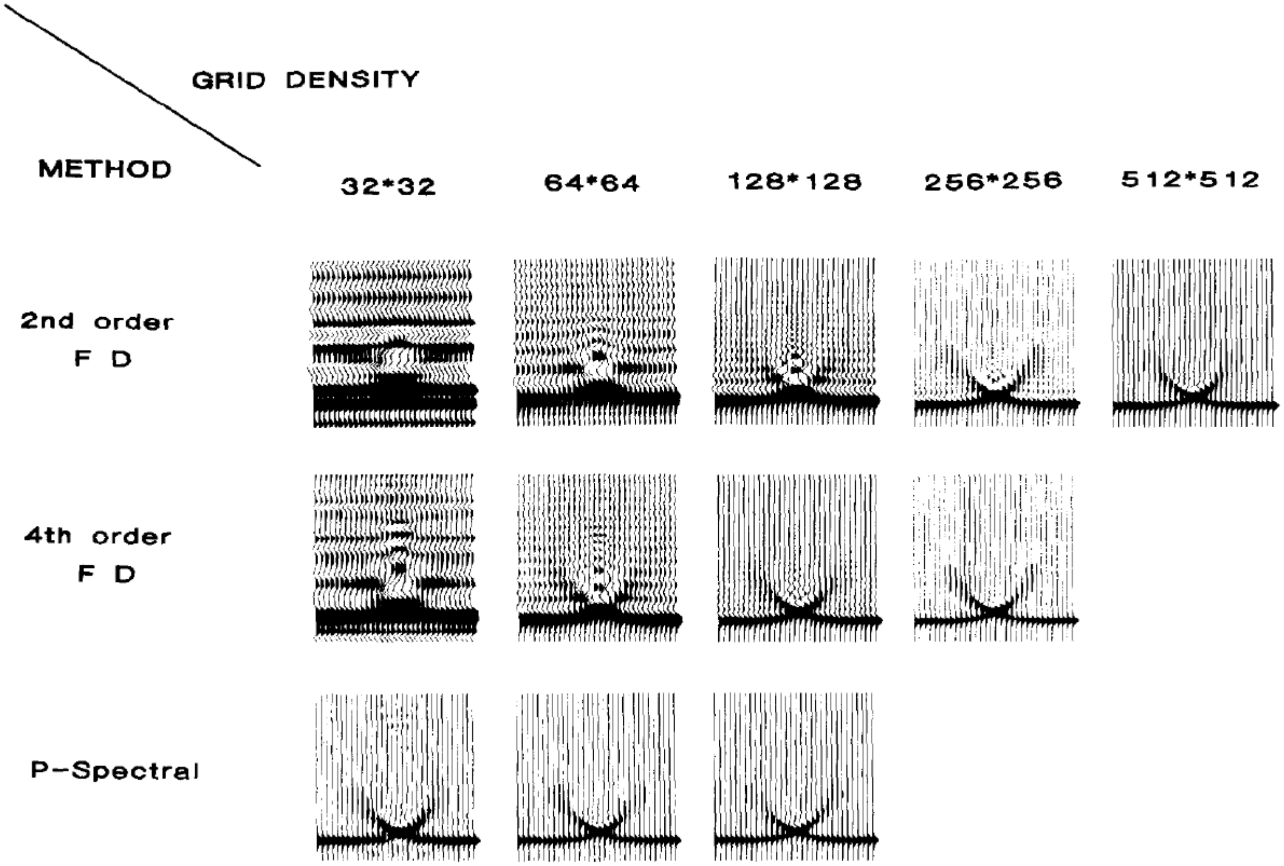
How well does this work? A first periodic PS example:

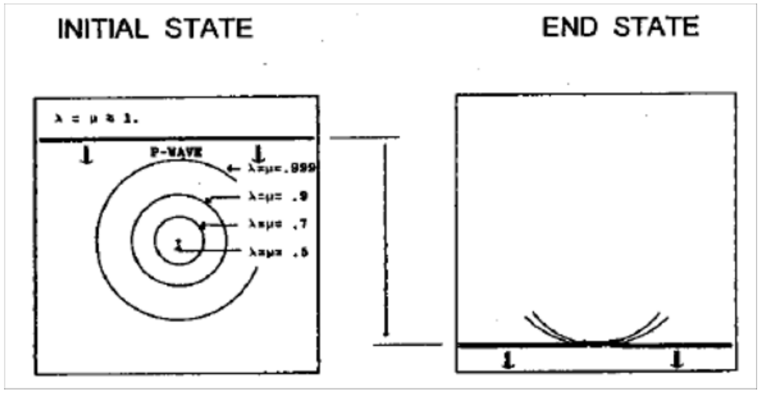
2-D Elastic Wave equation:

Fornberg (1987)

P-wave transition through a variable density medium

$$\begin{cases} \rho u_t = f_x + g_y \\ \rho v_t = g_x + h_y \\ \quad f_t = (\lambda + 2\mu) u_x + \lambda v_y \\ \quad g_t = \mu v_x + \mu u_y \\ \quad h_t = \lambda u_x + (\lambda + 2\mu) v_y \end{cases}$$





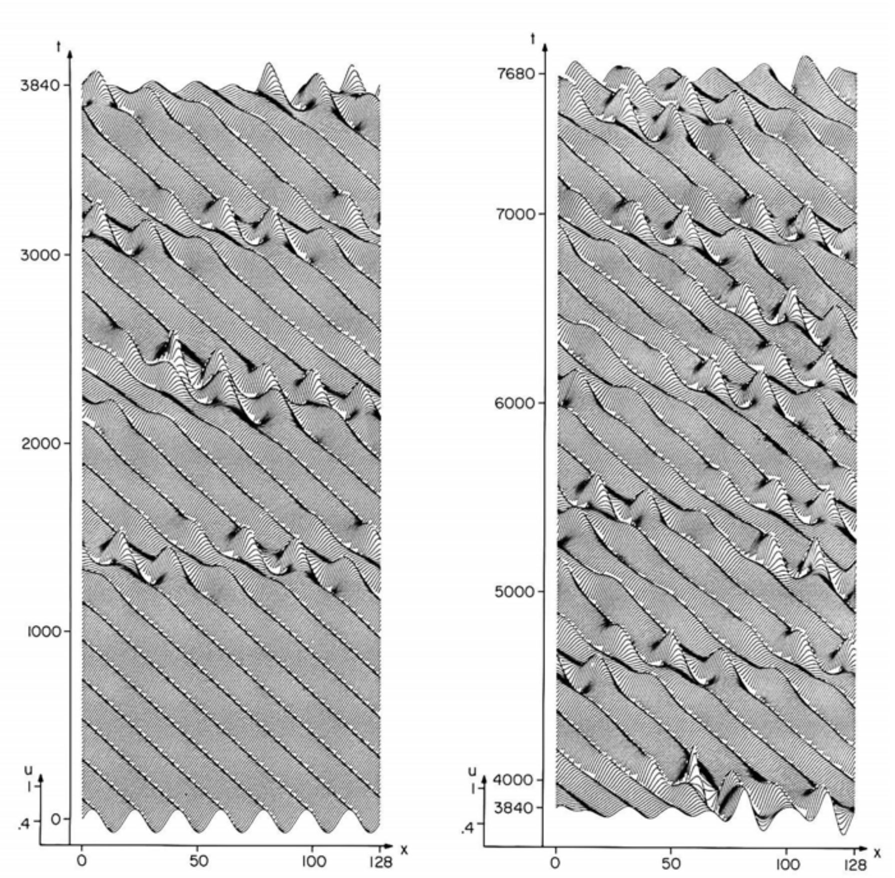
A second periodic PS example:

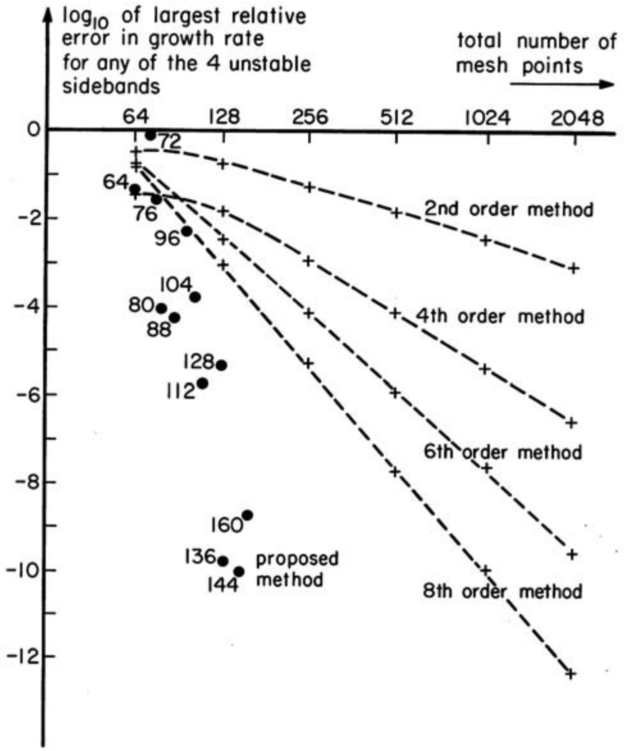
Nonlinear instabilities: Modified KdV equation:

Fornberg and Whitham (1978)

Growth rates for different side bands in the mKdV equation:

Dominant sideband modes	Growth rate α
6, 8	0.0101916188
5, 9	0.0184643383
4, 10	0.0220767943
3, 11	0.0130214545





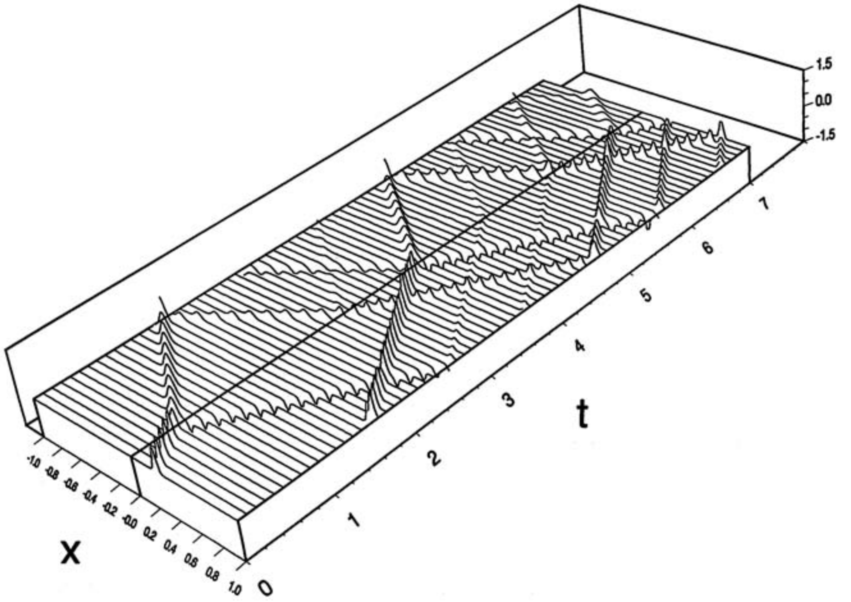
Pioneering calculation by Zabusky and Kruskal (1965): Discovery of **solitons**.

$$\frac{u(x,t+k)-u(x,t-k)}{2k}+\frac{1}{3}(u(x+h,t)+u(x,t)+u(x-h,t))\cdot\frac{u(x+h,t)-u(x-h,t)}{2h}+\frac{u(x+2h,t)-2u(x+h,t)-2u(x-h,t)+u(x-h,t)}{2h^3}=0$$

Second order: For 10^{-12} accuracy, need about 10^{8} nodes in space; stability cond k/h^{3} < const => k about 10^{-18}. If 1 µs/node/time step, need 10^{23} seconds; cf. age of universe about 10^{17} seconds.

A third periodic PS example:

1-D pulse propagation in a discontinuous medium:



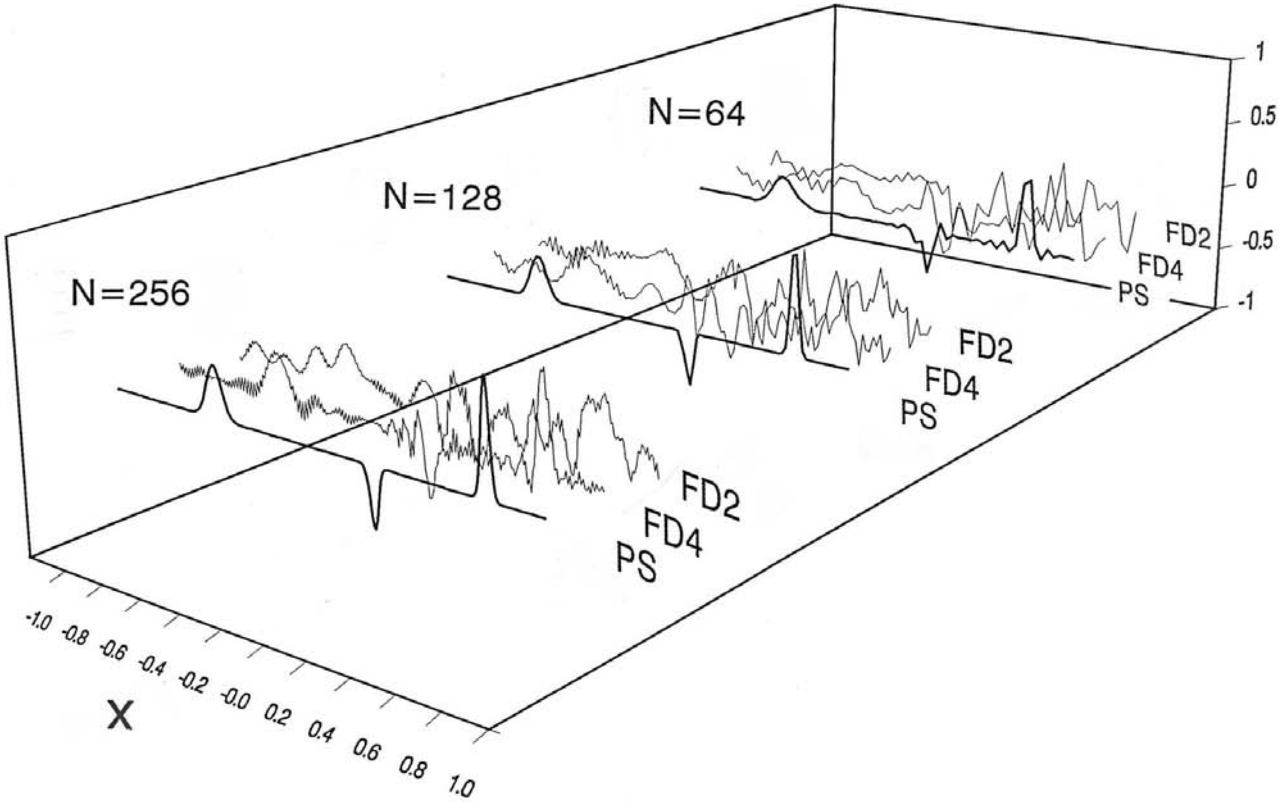
1-D acoustic wave equation:

$$\begin{cases} u_t &= v_x \\ v_t &= c^2(x)\, u_x \end{cases}$$

where

$$c(x) = \begin{cases} 1 & \text{if } -1 < x < 0 \\ \frac{1}{2} & \text{if } \quad 0 < x < 1 \end{cases}$$

and IC: $u(x,0) = 2v(x,0) = e^{-1600(x-\frac{1}{4})^2}$.



A fourth periodic PS example:

Double Fourier Method for convective flow over a sphere:

Merilees (1973), Fornberg and Merrill (1997)

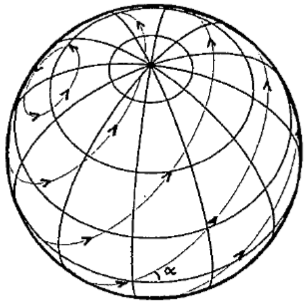
Governing equation

$$\frac{\partial h}{\partial t} + \frac{u}{a\,\cos\theta} + \frac{v}{a}\frac{\partial h}{\partial \theta} = 0$$

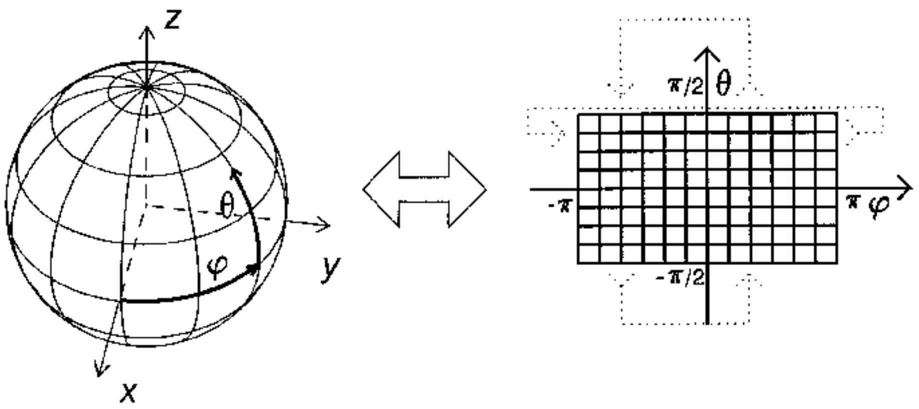
where

$$\begin{cases} u = u_0(\cos\theta\cos\alpha - \sin\theta\sin\alpha) \\ v = -u_0\cos\varphi\sin\alpha \end{cases}$$

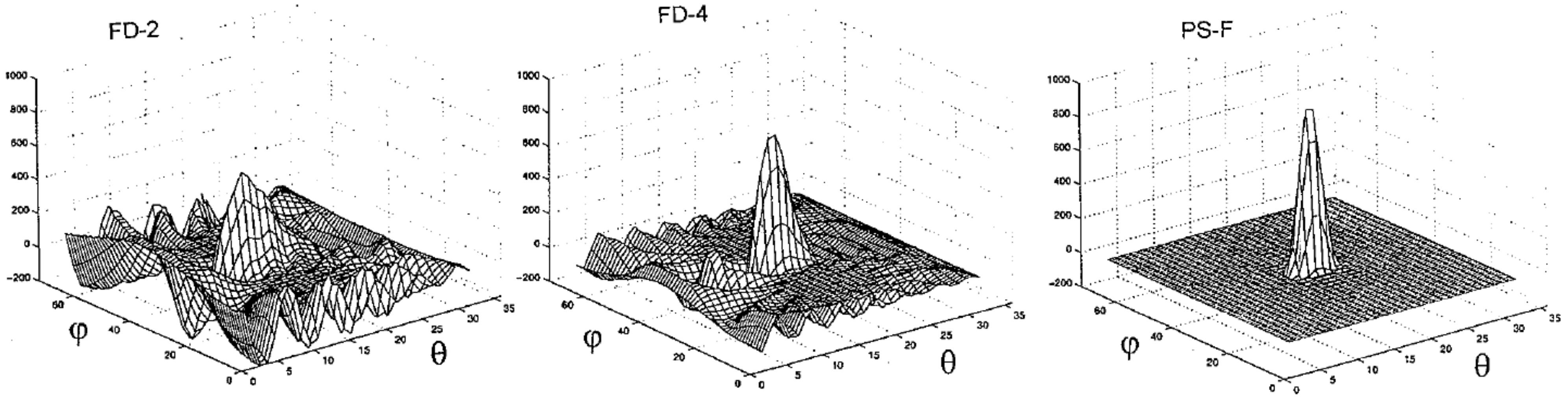
Convection: $\alpha = \frac{\pi}{2} - 0.05$



Grid display



Numerical solution after one revolution, 64×32 grid:


FD-2

FD-4

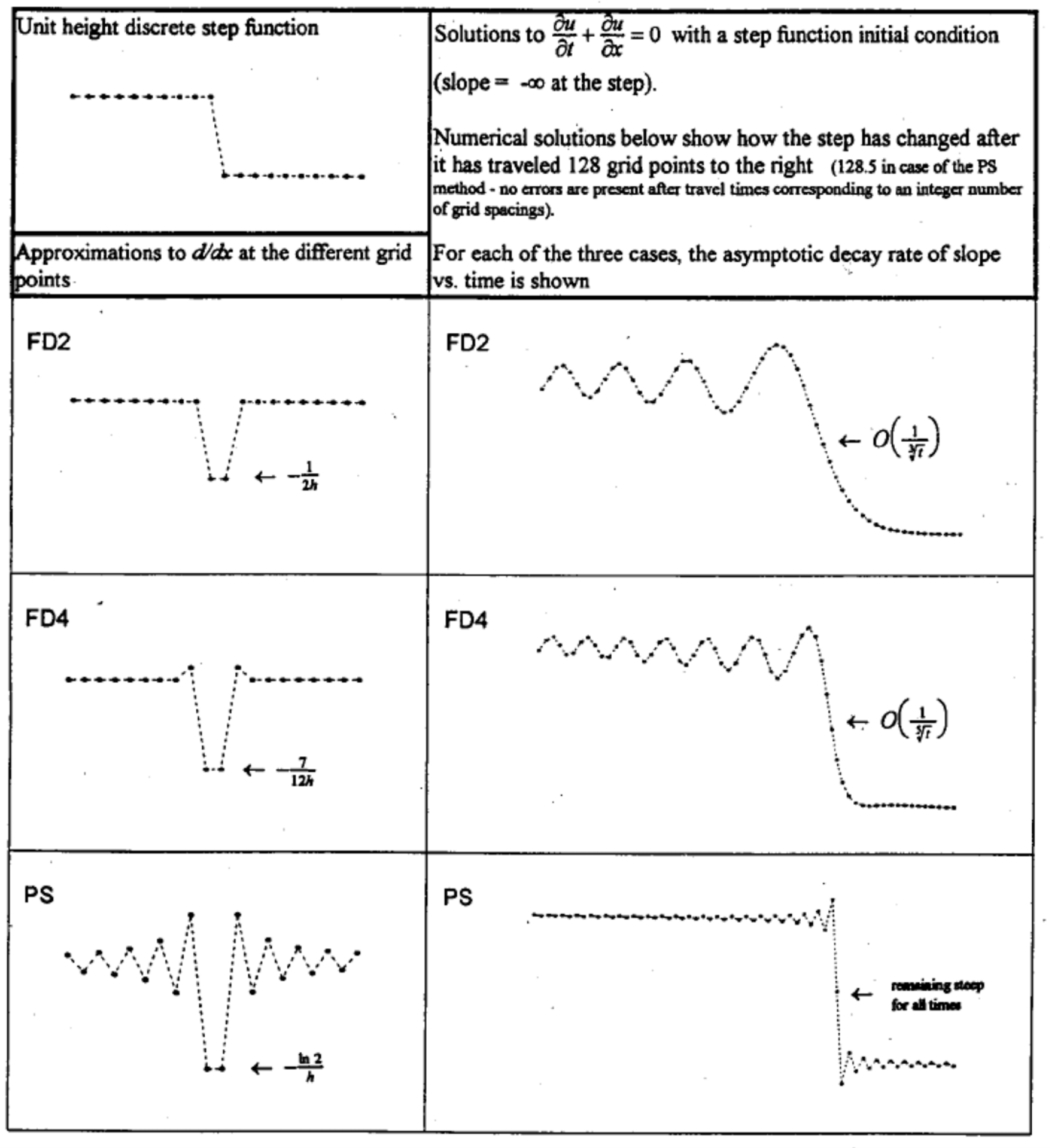
PS-F

Errors when approximating a step function

With FDp in space, the slope 'flattens' in time as $O\left(\frac{1}{\sqrt[p+1]{t}}\right)$.

For *convective PDEs*, it is thus advantageous to use high order approximations *even if the transported solution does not have the matching smoothness*.

Same will hold in RBF contexts.



Unit height discrete step function

Solutions to $\frac{\partial u}{\partial t}+\frac{\partial u}{\partial x}=0$ with a step function initial condition (slope = $-\infty$ at the step).

Numerical solutions below show how the step has changed after it has traveled 128 grid points to the right (128.5 in case of the PS method - no errors are present after travel times corresponding to an integer number of grid spacings).

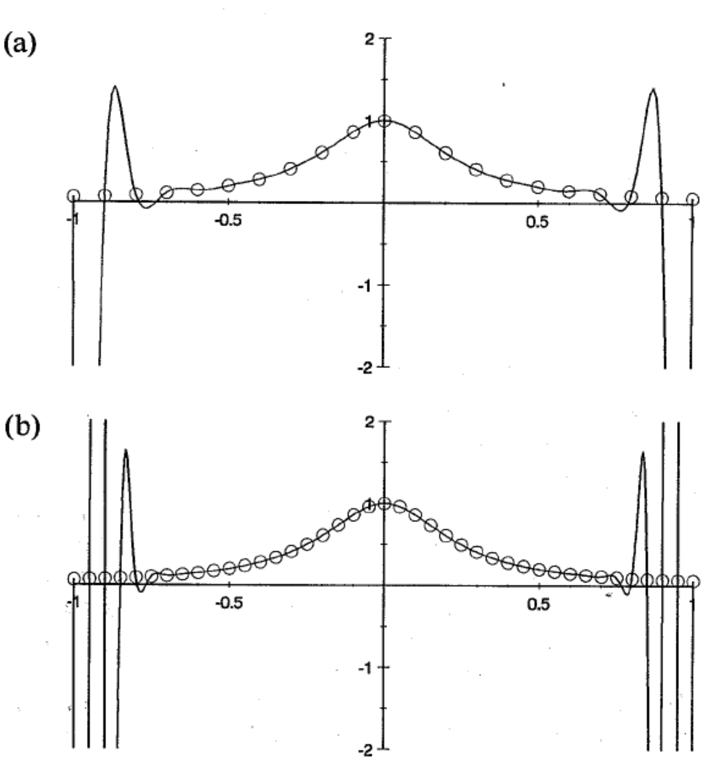
Approximations to d/dx at the different grid points

For each of the three cases, the asymptotic decay rate of slope vs. time is shown

FD2
$\leftarrow -\frac{1}{2h}$
FD2
$\leftarrow O\left(\frac{1}{\sqrt[3]{t}}\right)$
FD4
$\leftarrow -\frac{7}{12h}$
FD4
$\leftarrow O\left(\frac{1}{\sqrt[5]{t}}\right)$
PS
$\leftarrow -\frac{\ln 2}{h}$
PS
$\leftarrow$ remaining steep for all times

Non-periodic PS methods: Background

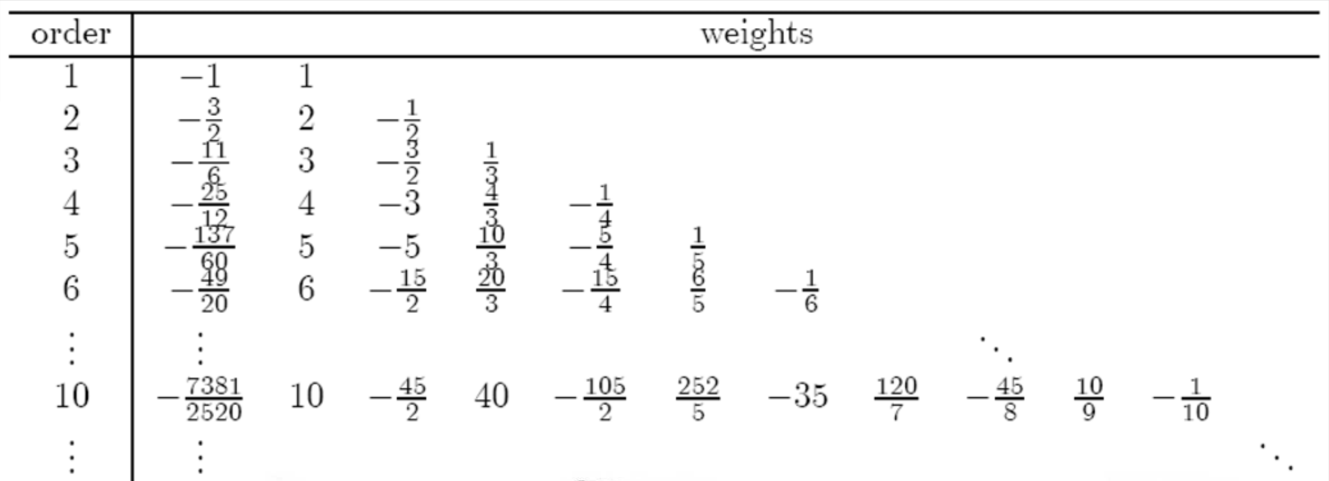
Runge Phenomenon:



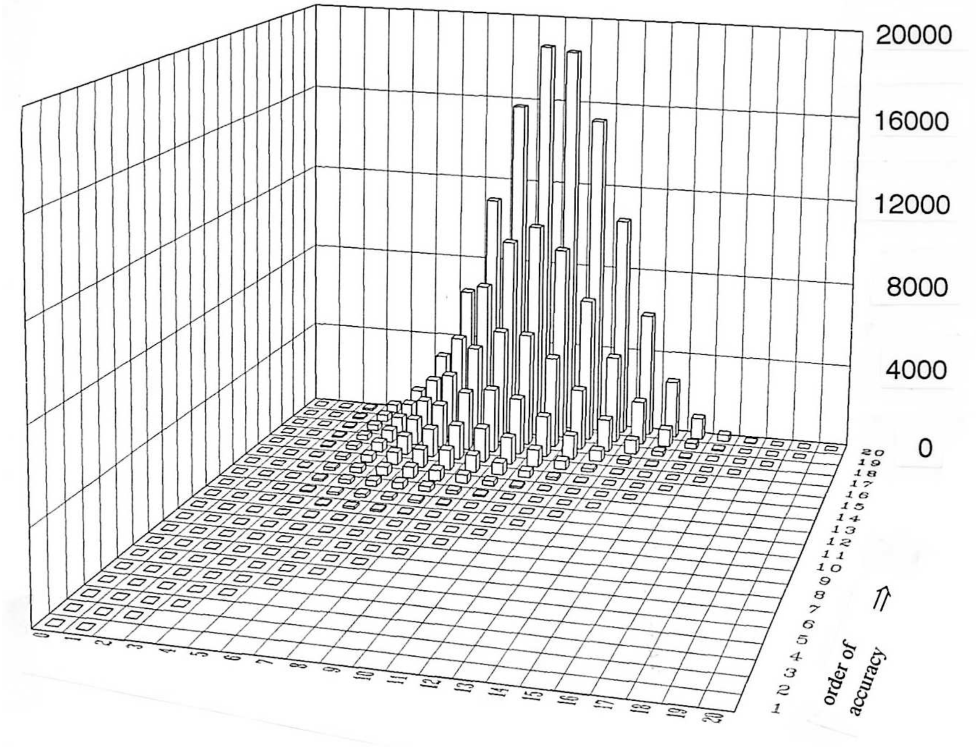
n = 21 and n = 41 equispaced interpolations of $f(x) = \dfrac{1}{1+16x^2}$ over [-1,1]

Exponential (spectral) convergence in middle; exponential divergence at the edges.

Chebyshev-type node clustering equals out errors across the interval.



order	weights										
1	-1	1									
2	$-\frac{3}{2}$	2	$-\frac{1}{2}$								
3	$-\frac{11}{6}$	3	$-\frac{3}{2}$	$\frac{1}{3}$							
4	$-\frac{25}{12}$	4	-3	$\frac{4}{3}$	$-\frac{1}{4}$						
5	$-\frac{137}{60}$	5	-5	$\frac{10}{3}$	$-\frac{5}{4}$	$\frac{1}{5}$					
6	$-\frac{49}{20}$	6	$-\frac{15}{2}$	$\frac{20}{3}$	$-\frac{15}{4}$	$\frac{6}{5}$	$-\frac{1}{6}$				
$\vdots$	$\vdots$								$\ddots$		
10	$-\frac{7381}{2520}$	10	$-\frac{45}{2}$	40	$-\frac{105}{2}$	$\frac{252}{5}$	-35	$\frac{120}{7}$	$-\frac{45}{8}$	$\frac{10}{9}$	$-\frac{1}{10}$
$\vdots$	$\vdots$										$\ddots$



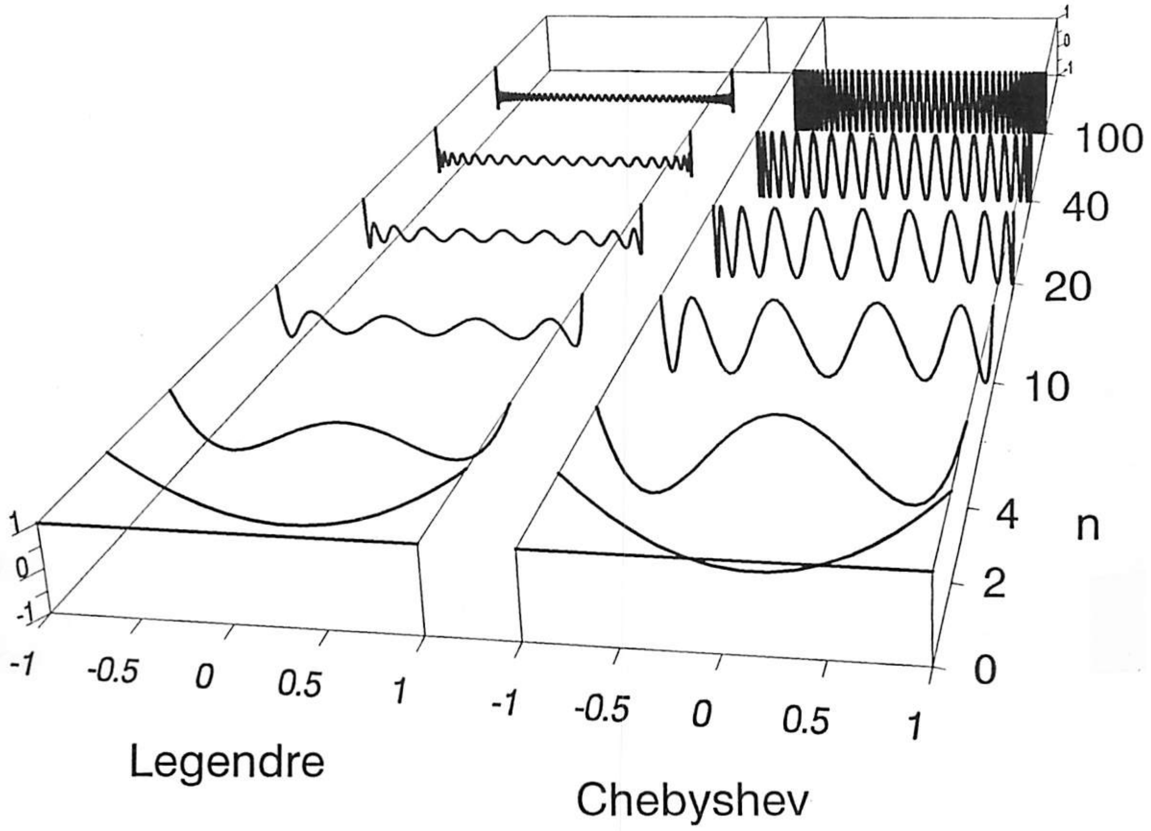
Magnitudes of one-sided FD weights for d/dx.

Requirements for PS expansion functions

Want to interpolate discrete data $u(x_i),\ i=0,1,\ldots,n$ as $u(x)=\sum_{k=0}^{n} a_k\phi_k(x)$. Requirements on choice of $\phi_k(x)$:

1. The expansion $\Sigma_{k=0}^{n} a_k\phi_k(x)$ must converge rapidly for smooth functions
2. Given coefficients a_k, it should be easy/fast to find b_k such that $\frac{d}{dx}\left(\sum_{k=0}^{n} a_k\phi_k(x)\right)=\sum_{k=0}^{n} b_k\phi_k(x)$
3. It should be fast to convert between coefficients $a_k, ki=0,1,\ldots,n$ and node values $u(x_i),\ i=0,1,\ldots,n$.

Trig functions are the 'obvious' choice in the periodic case. Jacobi-type orthogonal polynomials are 'natural' in non-periodic case, e.g.



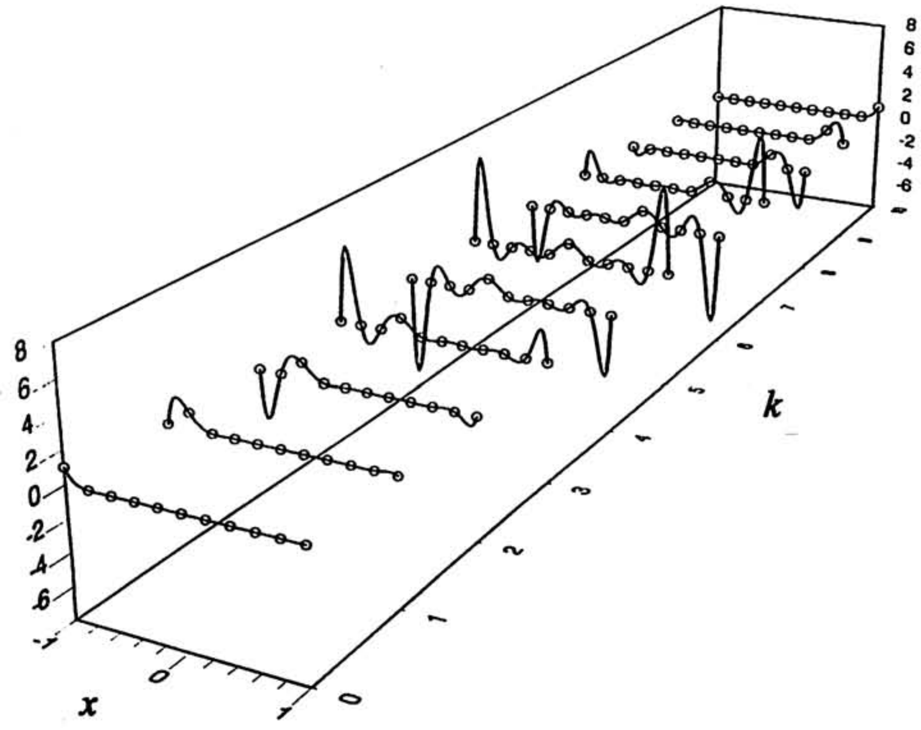
Associated node clustering at edges indeed controls the RP

Illustration be means of Lagrange's interpolation formula:

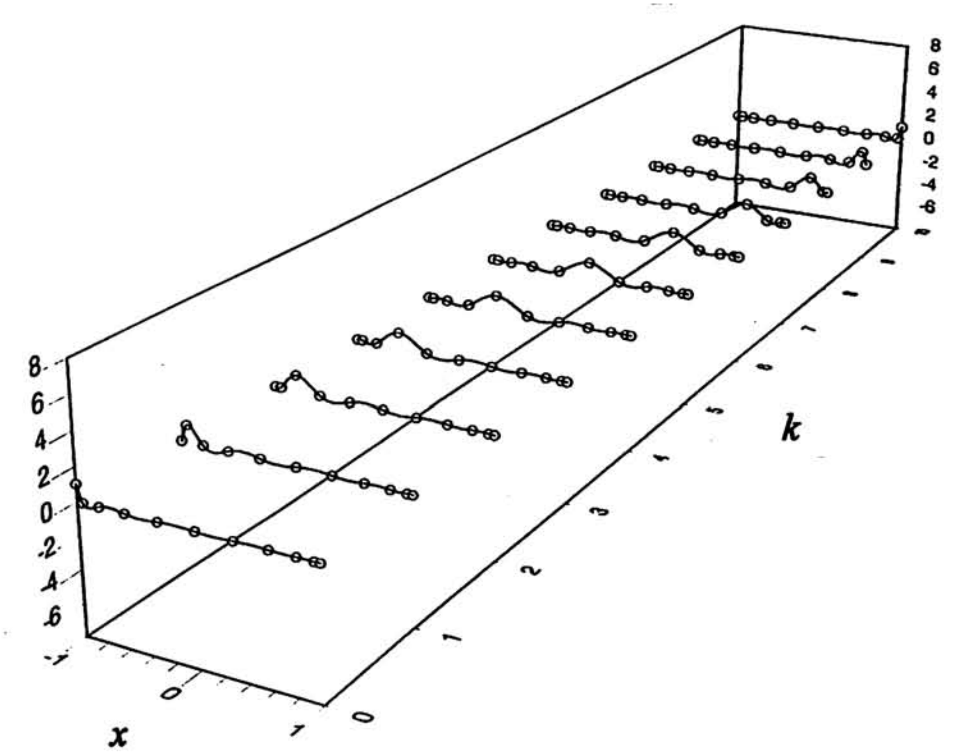
$$p_k(x) = \frac{(x - x_0)\cdot\ldots\cdot(x - x_{k-1})(x - x_{k+1})\cdot\ldots\cdot(x - x_n)}{(x_k - x_0)\cdot\ldots\cdot(x_k - x_{k-1})(x_k - x_{k+1})\cdot\ldots\cdot(x_k - x_n)}$$

N = 10

Equi-spaced

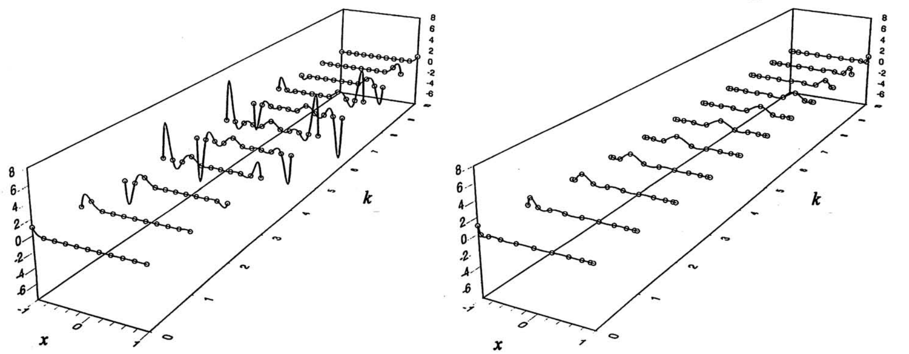


Chebyshev



Issue will arise also for RBF methods near boundaries.

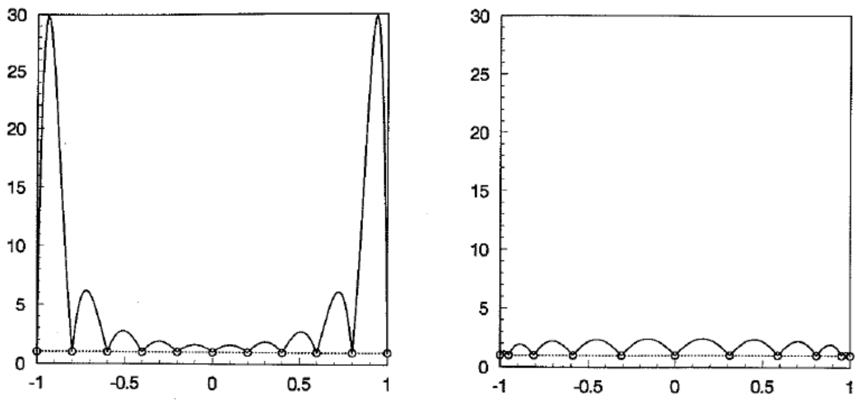
Lebesgue constants



Equispaced

Chebyshev

Add the magnitudes of all the curves above:



The peak heights of these resulting curves are known as the interpolation method's Lebesgue constant.

These values offer a precise measure of the interpolation procedures' quality (sensitivity to perturbations)

The concept is used also for RBF interpolation.

Key formula: Relates interpolation error to the optimal polynomial

$$\|f - P_N^{interp}\| \le (1 + \Lambda_N)\|f - P_N^{optimal}\|$$

(max norn, *NO* restriction on *f*(*x*))

Examples of polynomial Lebesgue constants:

$$\Lambda_N^{eq} = O\left(\frac{2^N}{N \log N}\right)$$

$$\Lambda_N^{Leg} = O(\sqrt{N})$$

$$\Lambda_N^{Cheb} = \frac{2}{\pi}(\log N + \gamma + \log \frac{8}{\pi}) + o(1)$$

$$\Lambda_N^{Opt} = \frac{2}{\pi}(\log N + \gamma + \log \frac{4}{\pi}) + o(1)$$

MOL discretization

Model problem: $\frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} = 0,\ u(-1) = 0;$ MOL: $\frac{\partial}{\partial t}\begin{bmatrix} u \end{bmatrix} = \begin{bmatrix} \quad D \quad \end{bmatrix}\begin{bmatrix} u \end{bmatrix}$

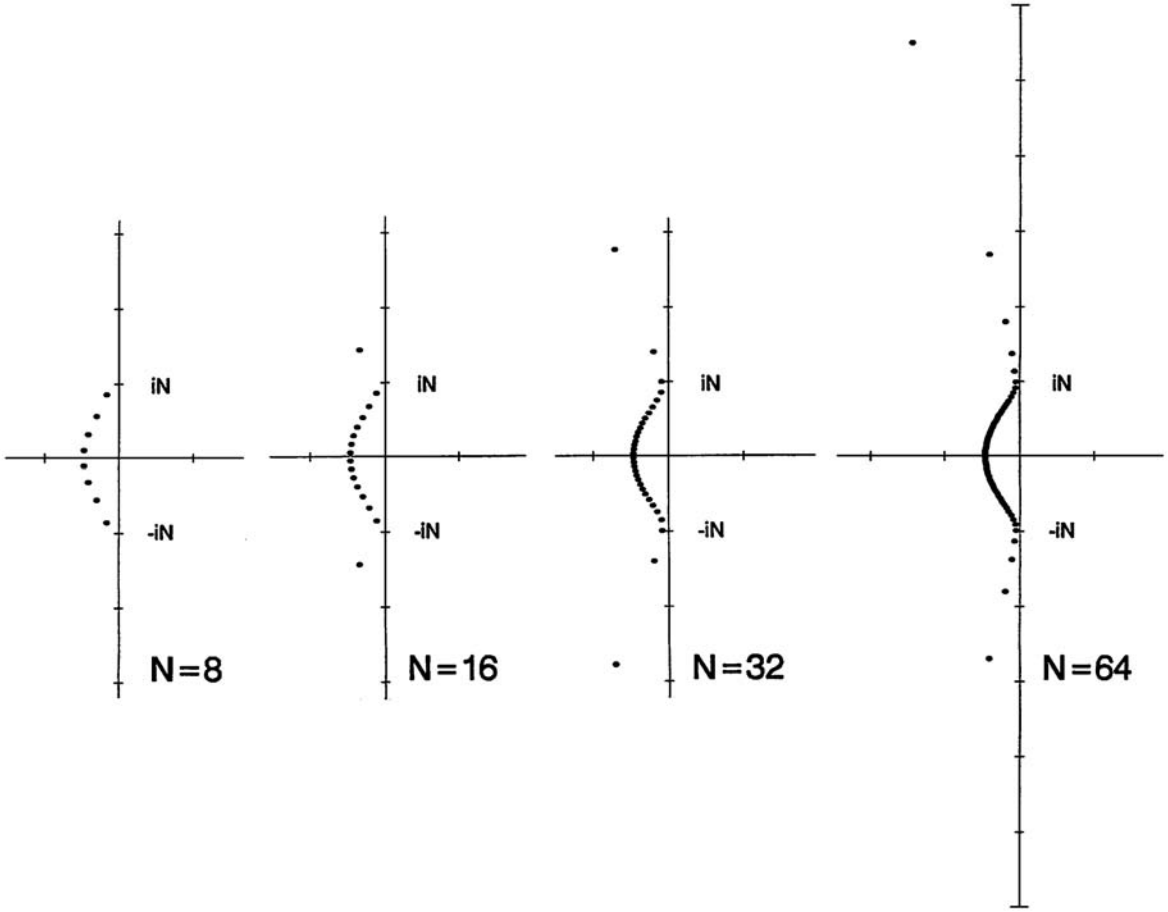
Product *Du* can be evaluated via FFTs or direct matrix×vector multiplication.

For choice of time stepping scheme and time step, inspect the eigenvalues of the DM. In the present case

:

High divergence rate spurious eigenvalues an immediate consequence of the quadratic node clustering.

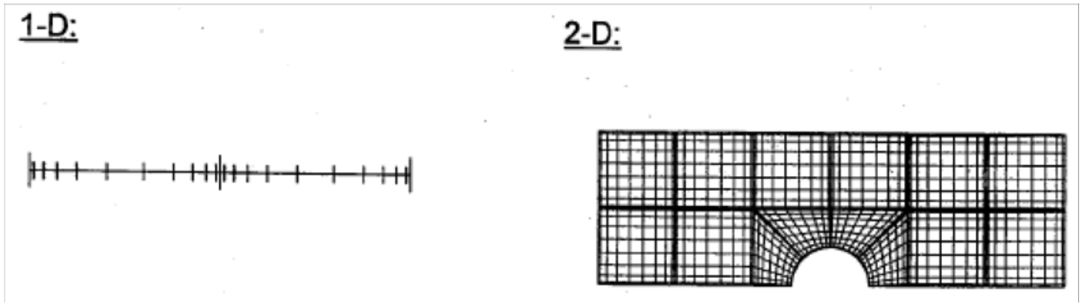
When later coming to RBFs, key issue will be to handle boundaries with much less ODE stiffness.



Non-periodic PS methods offer very limited geometric flexibility

Spectral Elements:

Concerns include errors/stiffness from artificial internal boundaries and complexity in cases of local refinement.



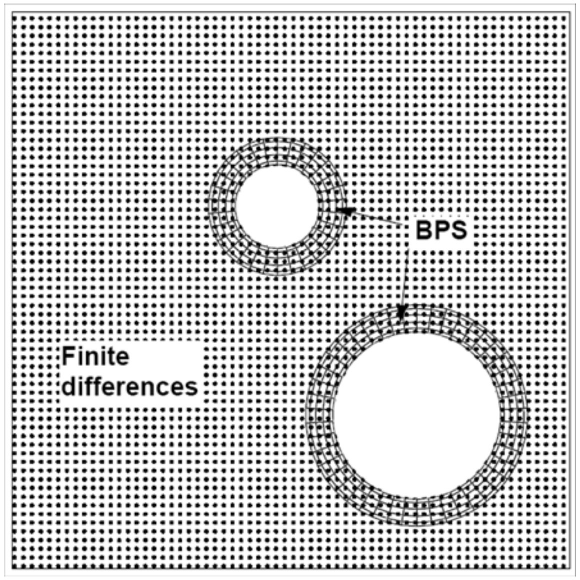
Hybrid FD-PS combinations:

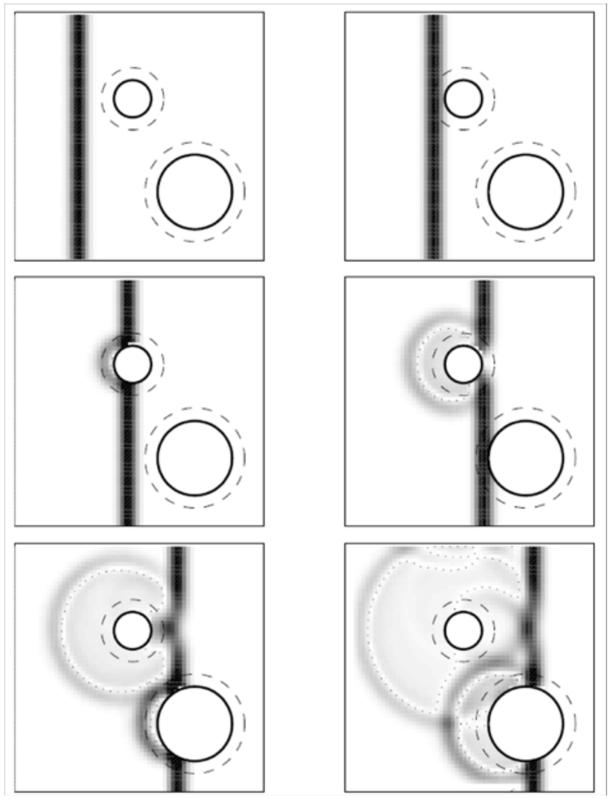
(Driscoll and Fornberg, 1999)

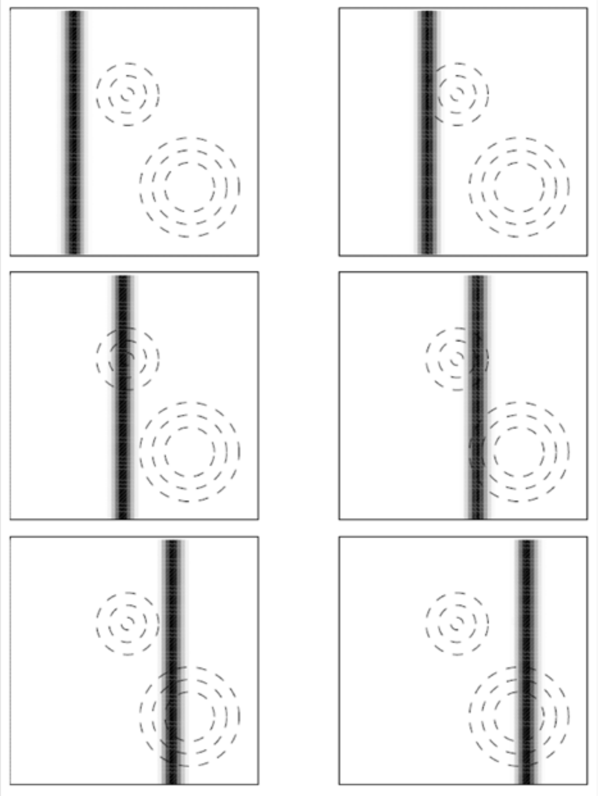
Example: 2-D TE (transverse electric) Maxwell's equations.

$$\left\{\begin{aligned} \varepsilon(x,y)\frac{\partial E_x}{\partial t} &= \frac{\partial H_z}{\partial y} - \sigma(x,y)E_x \\ \varepsilon(x,y)\frac{\partial E_y}{\partial t} &= -\frac{\partial H_z}{\partial x} - \sigma(x,y)E_y \\ \mu(x,y)\frac{\partial H_z}{\partial t} &= \frac{\partial E_x}{\partial y} - \frac{\partial E_y}{\partial x} - \sigma^*(x,y)H_z \end{aligned}\right.$$

E_x, E_y	Electric field
H_z	Magnetic field
ε, μ	Permetivity, permeability
σ, σ^*	Electric and magnetic resistivity





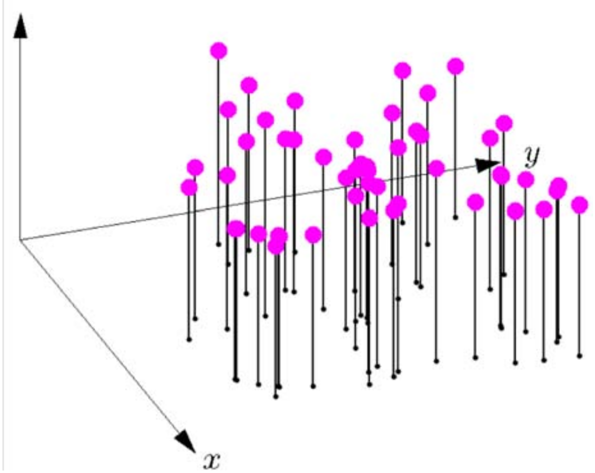


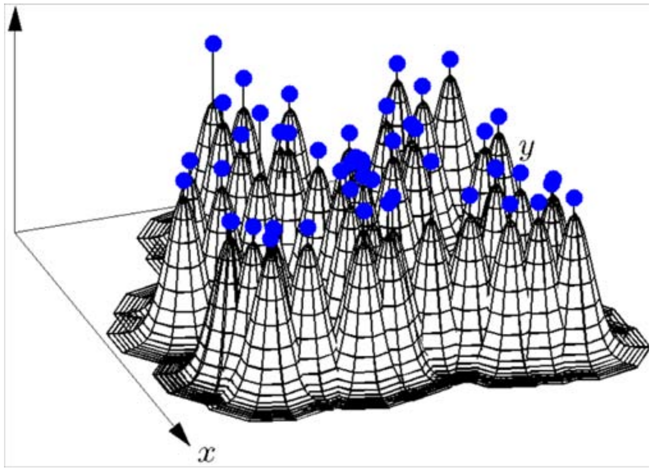
Introduction to Radial Basis Functions

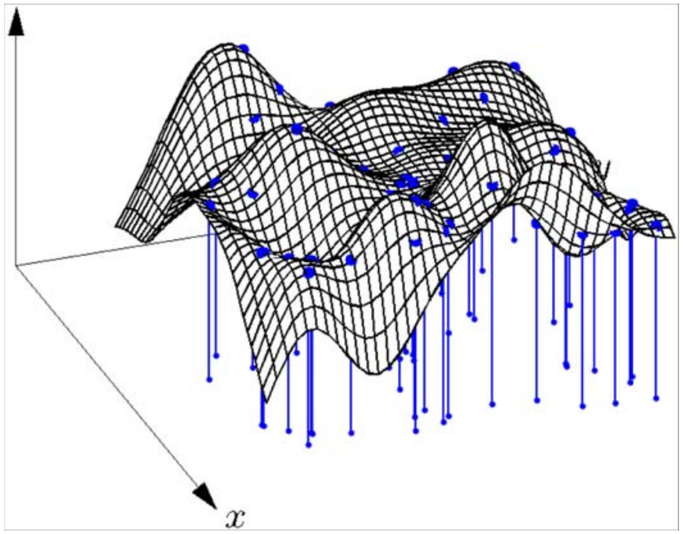
Presented at NSF-CBMS Regional Research Conference
University of Massachusetts, Dartmouth, by

Bengt Fornberg and Natasha Flyer

RBF idea, In pictures:







RBF idea, In formulas:

Given scattered data ($\underline{x}_k$, f_k), k = 1, 2, ... , N, in d-D, the RBF interpolant is

$$s(\underline{x}) = \sum_{k=1}^{N} \lambda_k \, \phi(\| \underline{x} - \underline{x}_k \|)$$

The coefficients λ_k can be found by collocation: $s(\underline{x}_k) = f_k$, k = 1, 2, ... , N *:*

$$\begin{bmatrix} \phi(\|\underline{x}_1 - \underline{x}_1\|) & \phi(\|\underline{x}_1 - \underline{x}_2\|) & \cdots & \phi(\|\underline{x}_1 - \underline{x}_N\|) \\ \phi(\|\underline{x}_2 - \underline{x}_1\|) & \phi(\|\underline{x}_2 - \underline{x}_2\|) & \cdots & \phi(\|\underline{x}_2 - \underline{x}_N\|) \\ \vdots & \vdots & \ddots & \vdots \\ \phi(\|\underline{x}_N - \underline{x}_1\|) & \phi(\|\underline{x}_N - \underline{x}_2\|) & \cdots & \phi(\|\underline{x}_N - \underline{x}_N\|) \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \vdots \\ \vdots \\ \lambda_N \end{bmatrix} = \begin{bmatrix} f_1 \\ \vdots \\ \vdots \\ f_N \end{bmatrix}$$

Inclusion of constant and polynomial terms:

Examples in 2-D:

$$s(\underline{x}) = \sum_{k=1}^{N} \lambda_k \, \phi(\| \underline{x} - \underline{x}_k \|) + \gamma_1 \qquad \text{with constraint} \quad \sum_{k=1}^{N} \lambda_k = 0$$

$$s(\underline{x}) = \sum_{k=1}^{N} \lambda_k \, \phi(\| \underline{x} - \underline{x}_k \|) + \gamma_1 + (\gamma_2 x + \gamma_3 y) \qquad \text{with constraints} \quad \sum_{k=1}^{N} \lambda_k = \sum_{k=1}^{N} \lambda_k x_k = \sum_{k=1}^{N} \lambda_k y_k = 0$$

Four main purposes:

1. The *A*-matrices for the modified systems may become positive (or negative) definite in cases where the original system lacked these features

   Examples: Already including constant will create negative definite system matrices for $\phi(r) = r$ and $\phi(r) = \sqrt{1+(\varepsilon r)^2}$.
   Including also linear terms creates positive definite matrices for $\phi(r) = r^2 \log r$ and $\phi(r) = r^3$.

2. The interpolant $s(\underline{x})$ becomes less prone to oscillations - routinely utilized in RBF-FD contexts.

3. *Saturation errors* may get reduced (or eliminated).

4. The inclusions may improve RBF accuracy at boundaries

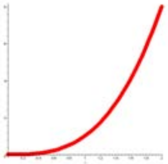
   Example: With $\phi(r) = r^3$, $s(\underline{x})$ typically diverges as $O(\|\underline{x}\|^3)$ for $\|\underline{x}\|$ increasing. The two versions above reduce this to $O(\|\underline{x}\|^2)$ and to $O(\|\underline{x}\|^1)$, respectively.

Many types of RBFs are available:

Piecewise smooth $\phi(r)$

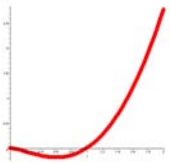
Cubics

r^3



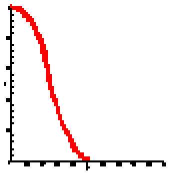
TP splines

$r^2 \log r$



Compact

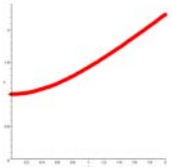
support



Infinitely smooth $\phi(r)$

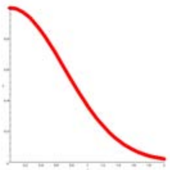
Multiquadric

$\sqrt{1+(\varepsilon r)^2}$



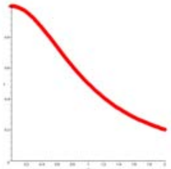
Gaussian

$e^{-(\varepsilon r)^2}$



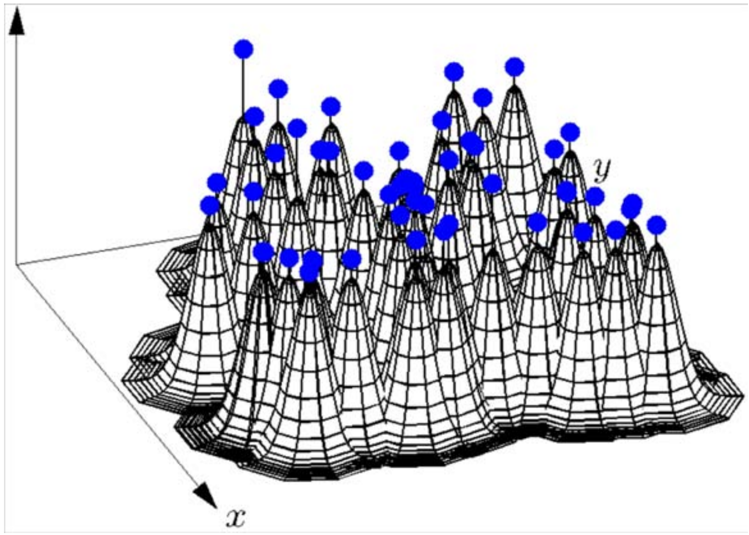
Inverse quadratic

$\dfrac{1}{1+(\varepsilon r)^2}$



Interpolation guaranteed nonsingular for all commonly used radial functions.

- Piecewise smooth RBFs:
  - algebraic accuracy (cf. cubic splines)
  - often only mild Gibbs oscillations
- Infinitely smooth RBFs:
  - spectral accuracy (if no Runge phenomenon)



While compactly supported RBFs (e.g. Wendland functions) can be highly effective in certain applications, we will not focus on them in the present survey because (i) only algebraic accuracy, and (ii) compactness is lost when forming derivative approximations (Differentiation Matrices; DMs).

Non-singularity of the *A*-matrix

Bochner (1933), Schoenberg (1938)

For Gaussian (GA) RBF:

1. Prove that *A* is positive semidefinite

Fourier transform of GA in *d* dimensions:

$$e^{-\|\underline{\omega}\|^2} = \frac{1}{(4\pi)^{d/2}} \int_{R^d} e^{-\|\underline{x}\|^2/4}\, e^{-i\underline{x}\cdot\underline{\omega}}\, d\underline{x}$$

Let $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_n)^T$ be an arbitrary column vector. Then:

$$\begin{aligned}
\underline{\alpha}^T A\, \underline{\alpha} &= \Sigma_{j=1}^n \Sigma_{k=1}^n \alpha_j \alpha_k\; e^{-\|\underline{x}_j-\underline{x}_k\|^2} \\
&= \Sigma_{j=1}^n \Sigma_{k=1}^n \alpha_j \alpha_k\, \frac{1}{(4\pi)^{d/2}} \int_{R^d} e^{-\|\underline{x}\|^2/4}\, e^{-i\underline{x}\cdot(\underline{x}_j-\underline{x}_k)}\, d\underline{x} \\
&= \frac{1}{(4\pi)^{d/2}} \int_{R^d} e^{-\|\underline{x}\|^2/4} \underbrace{\left(\Sigma_{j=1}^n \Sigma_{k=1}^n \alpha_j \alpha_k\;\; e^{-i\underline{x}\cdot(\underline{x}_j-\underline{x}_k)}\right)}_{(\Sigma_{j=1}^n\; \alpha_j e^{-i\underline{x}\cdot\underline{x}_j})(\Sigma_{k=1}^n\; \alpha_k e^{i\underline{x}\cdot\underline{x}_k}) = |\Sigma_{m=1}^n\; \alpha_m e^{-i\underline{x}\cdot\underline{x}_m}|^2 \geq 0} d\underline{x}
\end{aligned}$$

2. Refine argument to show that *A* in fact is positive definite

Idea: If x_m are distinct, show that a function $f(\underline{x}) = \Sigma_{m=1}^n\; \alpha_m e^{-i\underline{x}\cdot\underline{x}_m}$ cannot be identically zero unless all the coefficients α_m are zero.

Additional RBF classes that give nonsingular *A*-matrices:

- Any $\phi(r)$ such that $\hat{\phi}(\underline{\omega}) > 0$ and $\int_{R^d} \phi(\|\underline{x}\|)^2 d\underline{x} < \infty$.
- Any $\phi(r)$ such that $\phi(\sqrt{r}\,)$ is the Laplace transform of a non-negative function

Comletely Monotone functions

Definition 1: A $C^\infty(0,\infty)$ function $\psi(r)$ which has a bounded first derivative at the origin, is said to be completely monotone if and only if $(-1)^k \frac{d^k}{dr^k}\psi(r) \geq 0$ for $r > 0$ and $k = 0, 1, \ldots$

Definition 2: A function $\psi(r)$, $r \geq 0,$ is said to be completely monotone if and only its inverse Laplace transform $\gamma(s)$ is nonnegative (i.e. $\gamma(s) \geq 0$ when $\psi(r) = \int_0^\infty \gamma(s)\, e^{-sr} ds$).

Theorem: If $\psi(r)$ is completely constant (but not constant), then the RBF matrix *A* based on the radial function $\phi(r) = \psi(r^2)$ will be positive definite.

Example: Show that $\phi(r) = \frac{1}{(1+r^2)^\beta}$ is positive definite when $\beta > 0$.

Since $\phi(r) = \psi(r^2)$, we need to consider $\psi(r) = \frac{1}{(1+r)^\beta}$. Then:

Use Definition 1: $(-1)^k \frac{d^k}{dr^k}\psi(r) = (\prod_{i=0}^{k-1}(\beta+i))/(1+r)^{\beta+k} \geq 0$ if $\beta > 0,\ k = 0, 1, \ldots$

Use Definition 2: $\gamma(s) = e^{-s} s^{\beta-1}/\Gamma(\beta) > 0$ if $\beta > 0.$

Comletely Monotone functions:

The proof that Definition 2 leads to the stated theorem follows along the lines of the following examples:

Taking the inverse Laplace transform of $\phi(\sqrt{r}\,)$ for different radial functions $\phi(r)$ gives formulas such as

$$\text{IQ:} \qquad \frac{1}{1+(\varepsilon r)^2} \qquad = \int_0^\infty e^{-s}\, e^{-s(\varepsilon r)^2}\, ds$$

$$\text{IMQ:} \qquad \frac{1}{\sqrt{1+(\varepsilon r)^2}} \qquad = \int_0^\infty \frac{e^{-s}}{\sqrt{\pi s}}\, e^{-s(\varepsilon r)^2}\, ds$$

Direct demonstration that IQ is positive definite:

Whenever the factor in front of $e^{-s(\varepsilon r)^2}$ is positive, we observe (using here IQ as an illustration)

$$\underline{\alpha}^T A\, \underline{\alpha} = \sum_{j=1}^n \sum_{k=1}^n \alpha_j \alpha_k \frac{1}{1+\varepsilon^2 \|\underline{x}_j - \underline{x}_k\|^2} = \int_0^\infty e^{-s} \left(\sum_{j=1}^n \sum_{k=1}^n \alpha_j \alpha_k\, e^{-s\varepsilon^2 \|\underline{x}_j - \underline{x}_k\|^2} \right) ds.$$

From the nonsingularity proof for GA RBFs, we know that the double sum is positive whenever the vector $\underline{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_n]^T$ is not identically zero. Therefore, the integral and, with that, the quantity $\underline{\alpha}^T A\, \underline{\alpha}$ will also be positive, i.e. *A* is a positive definite matrix.

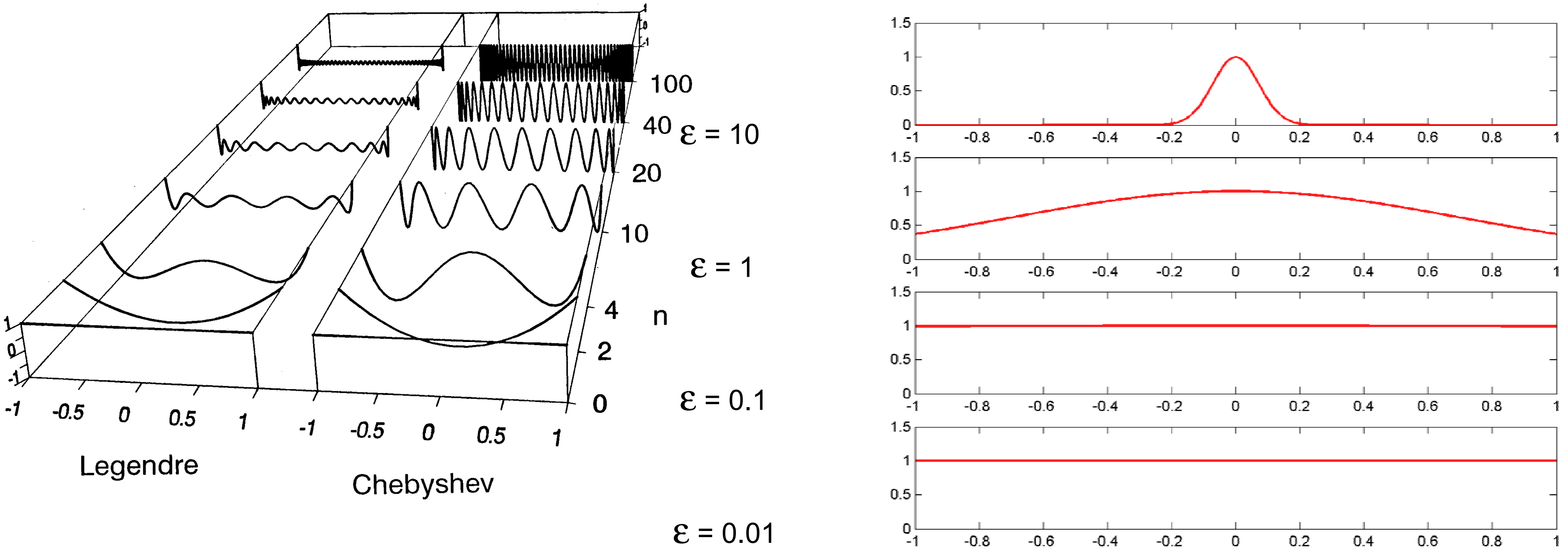
A variation of the Completely Monotone results above shows that MQ can never be singular

Micchelli (1984), particularly simple proof by Powell (2005).

Numerical conditioning, and the flat RBF limit (ε → 0)

Classical basis functions are usually highly oscillatory

RBFs are translates of one single function - here $\phi(r) = e^{-(\varepsilon r)^2}$



In case of 41 scattered nodes in 1-D: cond(A) = $O(\varepsilon^{-80})$, det(A) = $O(\varepsilon^{1640})$.

2-D: cond(A) = $O(\varepsilon^{-16})$, det(A) = $O(\varepsilon^{416})$.

Exact formulas available for any number of nodes in any number of dimensions

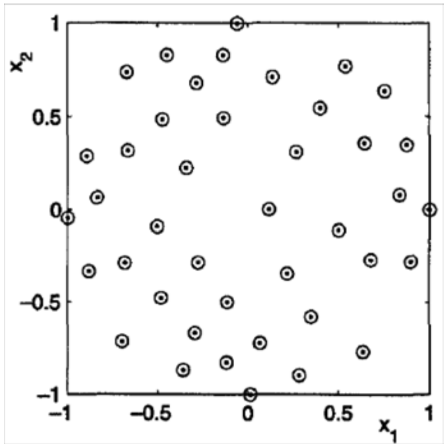
Fornberg and Zuev (2007)

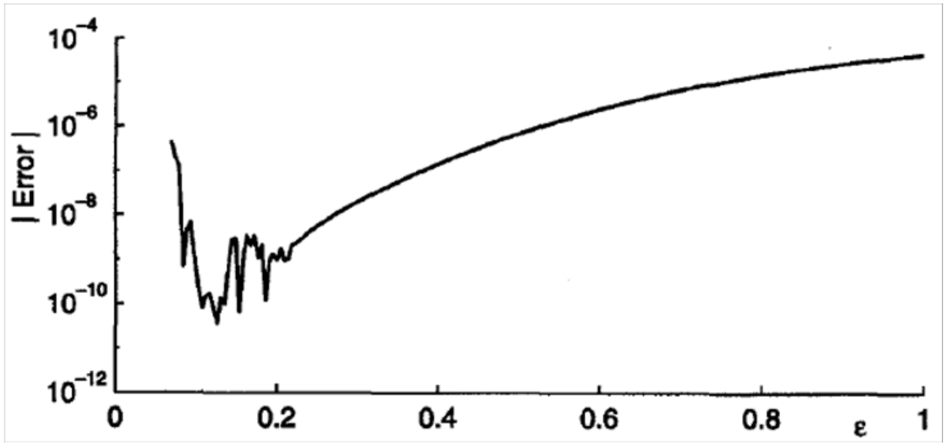
Extreme ill-conditioning typical as ε → 0.

Why are flat (or near-flat) RBFs interesting ?

- Intriguing error trends as $\varepsilon \to 0$

  'Toy-problem' example: 41 node MQ interpolation of $f(x_1,x_2)=\dfrac{59}{67+(x_1+\frac{1}{7})^2+(x_2-\frac{1}{11})}$





- RBF interpolant in 1-D reduces to Lagrange's interpolation polynomial

  Driscoll and Fornberg (2002)

- The $\varepsilon \to 0$ limit reduces to 'classical' PS methods if used on tensor type grids.

- The RBF approach generalize PS methods in many ways:
  - Guaranteed nonsingular also for scattered nodes on irregular geometries (suggests RBF-FD mehtods)
  - Allows spectral accuracy to be combined with local mesh refinement
  - Best accuracy may be obtained for non-zero ε.

Solving $A\underline{\lambda}=\underline{f}$ followed by evaluating $s(\underline{x},\varepsilon)=\sum_{k=1}^{N} \lambda_k\, \phi(||\underline{x}-\underline{x}_k||)$
is merely an unstable algorithm for a stable problem

Rough estimate of the A-matrix ill-conditioning

RBFs based on *odd* powers of *r* (such as $\phi(r) = r$ and $\phi(r) = r^3$) feature a discontinuous derivative at the origin, and therefore (low order) algebraic accuracy.

RBFs based on *even* powers of *r* (such as $\phi(r) = r^2$ and $\phi(r) = r^4$) lead generally to singular *A*-matrices:

Example: Using $\phi(r) = r^2$ in 1-D will lead to an interpolant of the form $s(x) = \sum_{k=1}^{n} \lambda_k (x - x_k)^2$, which is a parabola, and is therefore unable to interpolate more than $n = 3$ points.

One finds similarly that

$$\phi(r) = c_0 + c_1(\varepsilon r)^2 + c_2(\varepsilon r)^4 + \ldots + c_m(\varepsilon r)^{2m} \qquad (1)$$

in *d*-D can interpolate at most $n = \frac{2m+d}{m+d}\binom{m+d}{d}$ points (recovering the value $n = 3$ for $m = d = 1$).

Example: With $n = 300$ and $d = 2$, $\phi(r)$ will give a singular *A*-matrix up through $m = 16$. Nonsingularity for an *A*-matrix will therefore depend on the further terms, omitted in (1)

$$c_{17}(\varepsilon r)^{34} + c_{18}(\varepsilon r)^{36} + \ldots ,$$

i.e. an $O(\varepsilon^{34})$-sized perturbation of the entries of *A* will suffice to make it singular.

The estimate above turns out to be sharp in case we allow spatially varialbe ε; else it turns out that $O(\varepsilon^{46})$-sized perturbations suffice to make *A* singular.

Note that the estimate did not require any specific knowledge about the radial function $\phi(r)$.

RBF Conditioning

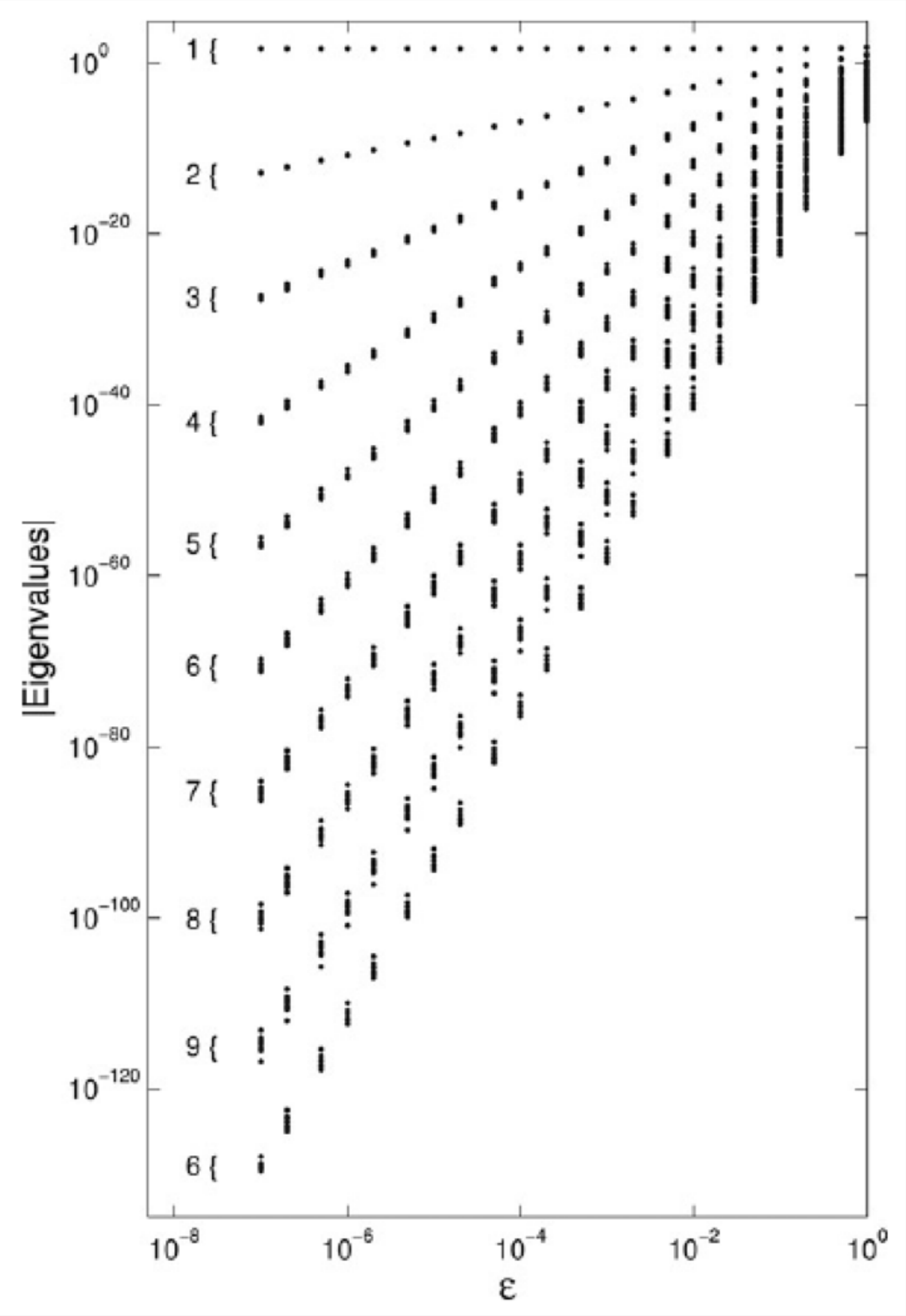
Fornberg and Zuev (2007)

Eigenvalue patterns for the RBF *A*-Matrix:

Example: Scatter n = 51 nodes in 2-D; use MQ
Calculate eigenvalues as functions of ε.

1 *e* - value $O(1)$
2 *e* - values $O(\varepsilon^2)$
3 *e* - values $O(\varepsilon^4)$
4 *e* - values $O(\varepsilon^6)$

etc. until last *e* - value is reached.



Pattern for groups independent of:

- Number of nodes
- How nodes are scattered
- RBF type (GA, IQ, IMQ, MQ, ...)
  (certain types, such as BE, can be exceptional)

Different eigenvalue patterns for different geometries

Number of eigenvalues of different sizes:

Geometry	Power of ε							
	0	2	4	6	8	10	12	...
1-D non-periodic	1	1	1	1	1	1	1	...
1-D on circle periphery	1	2	2	2	2	2	2	...
2-D non-periodic	1	2	3	4	5	6	7	...
On spherical surface	1	3	5	7	9	11	13	...
3-D non-periodic	1	3	6	10	15	21	28	...

$\text{Cond}\,(A) = 1/\varepsilon^{\alpha(n)}$ with $\alpha(n)$ shown below:

Geometry	Number of nodes n						
	1	10	100	1,000	10,000	100,000	...
1-D non-periodic	0	18	198	1,998	19,998	199,998	...
1-D on circle periphery	0	10	100	1,000	10,000	100,000	...
2-D non-periodic	0	6	26	88	280	892	...
On spherical surface	0	6	18	62	198	632	...
3-D non-periodic	0	4	14	34	76	166	...

Closed formulas available:

Example: 2-D non-periodic: $\text{Cond}\,(A) = O(1/\varepsilon^{2[(\sqrt{8n-7}\,-1)/2]})$

where [...] denotes integer part (Matlab 'floor')

Some variations in the eigenvalue pattern formulas

Random 'noise' on the ε-values:

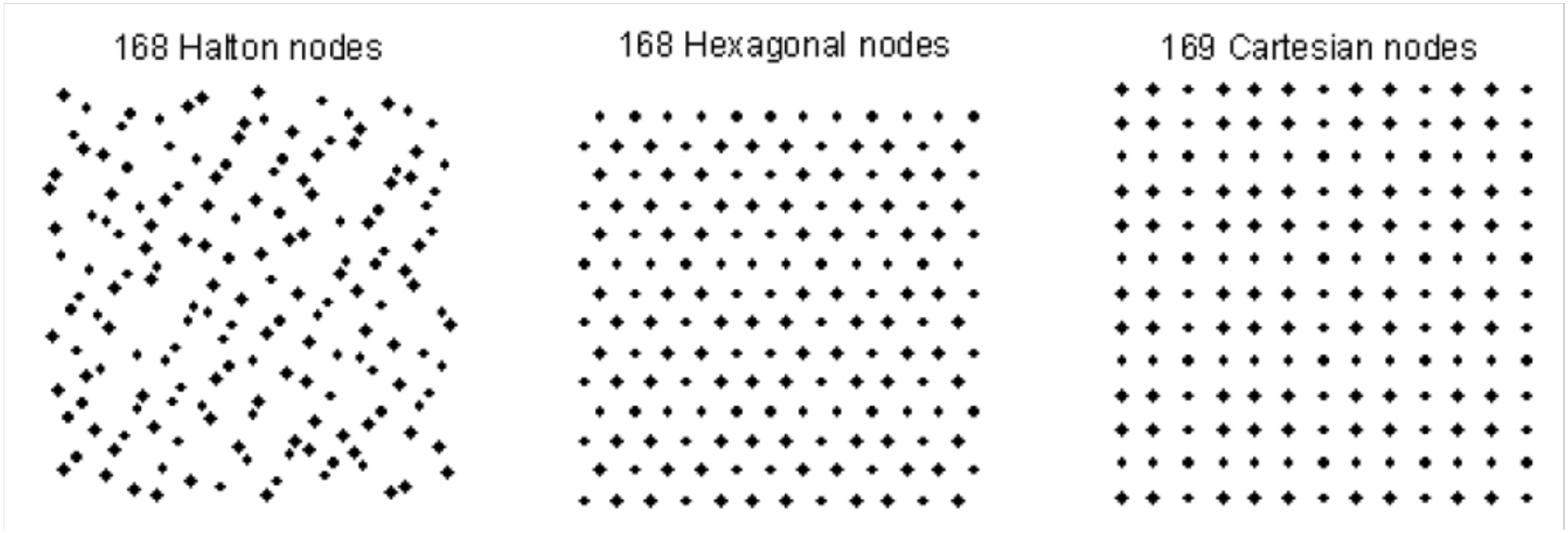
Choose for the different nodes $\varepsilon_k = \varepsilon \cdot$ {random numbers on [0,1]}, and then let $\varepsilon \to 0$.

Geometry	Shape	Power of ε							
	parameter	0	2	4	6	8	10	12	...
1-D non-periodic	const	1	1	1	1	1	1	1	...
1-D non-periodic	variable	1	2	2	2	2	2	2	...
1-D on circle periphery	const	1	2	2	2	2	2	2	...
1-D on circle periphery	variable	1	2	2	2	2	2	2	...
2-D non-periodic	const	1	2	3	4	5	6	7	...
2-D non-periodic	variable	1	3	5	7	9	11	13	...
On spherical surface	const	1	3	5	7	9	11	13	...
On spherical surface	variable	1	3	5	7	9	11	13	...
3-D non-periodic	const	1	3	6	10	15	21	28	...
3-D non-periodic	variable	1	4	9	16	25	36	49	...

- Conditioning improved in non-periodic cases (suggests that boundary effects play a role)
- Suggestive that *A*-matrix singularity, while possible, might be unlikely
- Practical utility unclear (e.g. effect on accuracy)

Some variations in the eigenvalue pattern formulas

Degradation in case of lattice-based node layouts (2-D non-periodic):



Number of eigenvalues of sizes $\varepsilon^0, \varepsilon^2, \varepsilon^4, \ldots$

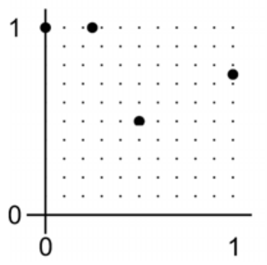
Halton	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 | 15
Hexagonal	1 2 3 4 5 6 7 8 9 10 11 12 13 14 |14 14 14 14 | 7
Cartesian	1 2 3 4 5 6 7 8 9 10 11 12 13 |12 11 10 9 8 7 6 5 4 3 2 1

resulting in condition numbers $O(\varepsilon^{-34}), O(\varepsilon^{-36}), O(\varepsilon^{-48})$, respectively.

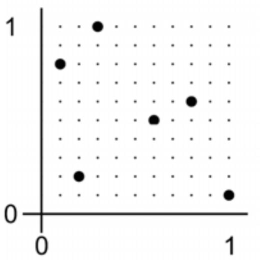
There is a severe degradation in the Cartesian case in terms of conditioning (AND in accuracy).

Some examples of different limits as $\varepsilon \rightarrow 0$

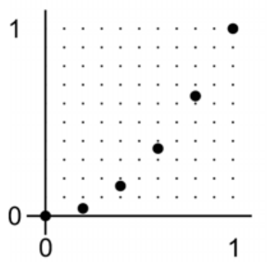
Result depends both on node distribution and on type of basis function



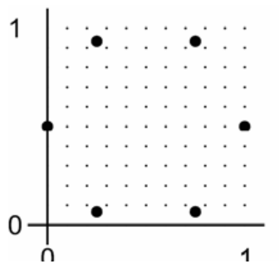
MQ, IM, IQ,GA,J0 all different limits; all polynomials degree 2.



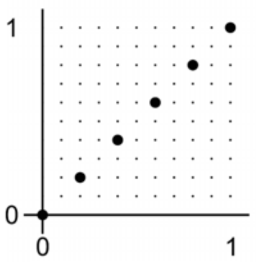
MQ, IM, IQ, GA same limit, pol. degree 2; J0 different limit, pol. degree 3.



All limits different; all polynomials degree 3.



All converge to the to the same limit.



MQ, IM, IQ diverge; GA, J0 converge to different limits

What is the widest class of radial functions possible such that divergence will never occur, no matter how the nodes are distributed?

Result true for GA: Conjectured 2004 (Fornberg, Wright, Larsson), Proved 2005 (Schaback)
BE widest class possible: Conjectured 2006 (Fornberg, Larsson, Wright)

.

A closed form expression for RBF interpolants

For cardinal data $y_k = \begin{cases} 1 & k = 1 \\ 0 & \text{otherwise} \end{cases}$, the RBF interpolant $s(\underline{x}) = \sum_{k=1}^{N} \lambda_k \, \phi(\| \underline{x} - \underline{x}_k \|)$ becomes

$$s(\underline{x}) = \frac{\det \begin{bmatrix} \phi(\|\underline{x} - \underline{x}_1\|) & \phi(\|\underline{x} - \underline{x}_2\|) & \cdots & \phi(\|\underline{x} - \underline{x}_n\|) \\ \phi(\|\underline{x}_2 - \underline{x}_1\|) & \phi(\|\underline{x}_2 - \underline{x}_2\|) & \cdots & \phi(\|\underline{x}_2 - \underline{x}_n\|) \\ \vdots & \vdots & \ddots & \vdots \\ \phi(\|\underline{x}_n - \underline{x}_1\|) & \phi(\|\underline{x}_n - \underline{x}_2\|) & \cdots & \phi(\|\underline{x}_n - \underline{x}_n\|) \end{bmatrix}}{\det \begin{bmatrix} \phi(\|\underline{x}_1 - \underline{x}_1\|) & \phi(\|\underline{x}_1 - \underline{x}_2\|) & \cdots & \phi(\|\underline{x_1} - \underline{x}_n\|) \\ \phi(\|\underline{x}_2 - \underline{x}_1\|) & \phi(\|\underline{x}_2 - \underline{x}_2\|) & \cdots & \phi(\|\underline{x}_2 - \underline{x}_n\|) \\ \vdots & \vdots & \ddots & \vdots \\ \phi(\|\underline{x}_n - \underline{x}_1\|) & \phi(\|\underline{x}_n - \underline{x}_2\|) & \cdots & \phi(\|\underline{x}_n - \underline{x}_n\|) \end{bmatrix}}$$

With $\phi(r) = a_0 + a_1(\varepsilon r)^2 + a_2(\varepsilon r)^4 + a_3(\varepsilon r)^6 + \ldots$ we get

$$s(\underline{x}) = \boxed{\underbrace{\frac{1}{\varepsilon^{2p}} s_{-2p}(\underline{x}) + \frac{1}{\varepsilon^{2p-2}} s_{-2p+2}(\underline{x}) + \ldots + \frac{1}{\varepsilon^{2}} s_{-2}(\underline{x})}} + \underbrace{s_0(\underline{x})} + \underbrace{\varepsilon^2 s_2(\underline{x}) + \varepsilon^4 s_4(\underline{x}) + \ldots}$$

Divergent part; rarely present ↑

Flat basis function limit (generalized pseudospectral method)

Corrections to flat basis function limit

Some issues:

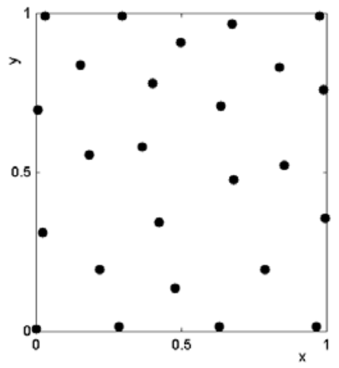
- When is the divergent parp absent?
- Can we evaluate the individual *s*($\underline{x}$)-functions numerically?

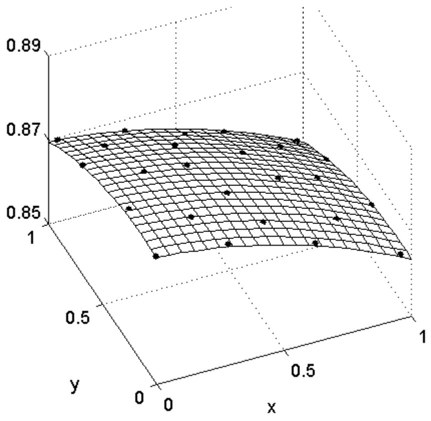
  (Will be achieved by the Contour-Padé algorithm)

Example

$$f\,(x,\,y) = \frac{59}{67+(x+\frac{1}{7})^2+(y-\frac{1}{11})^2}$$

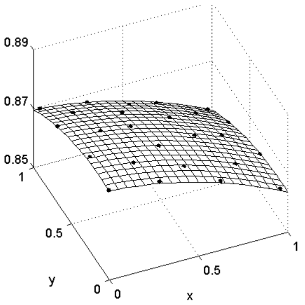
sampled at scattered
points in [0, 1]×[0, 1]



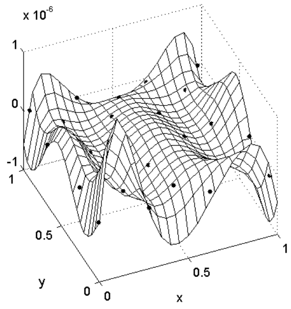


Multiquadric interpolant: $s(\underline{x},\varepsilon) = s_0(\underline{x}) + \varepsilon^2\, s_2(\mathrm{x}) + \varepsilon^4\, s_4(\underline{x}) + \varepsilon^6\, s_6(\underline{x}) + \ldots$ where

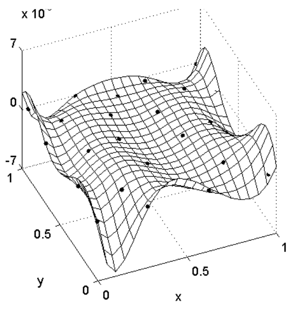
$s_0(\underline{x})$



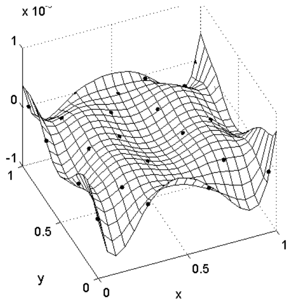
$s_2(\mathrm{x})$



$s_4(\underline{x})$



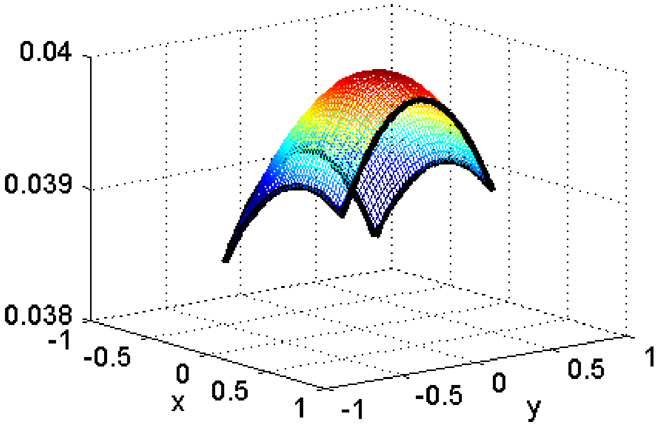
$s_6(\underline{x})$



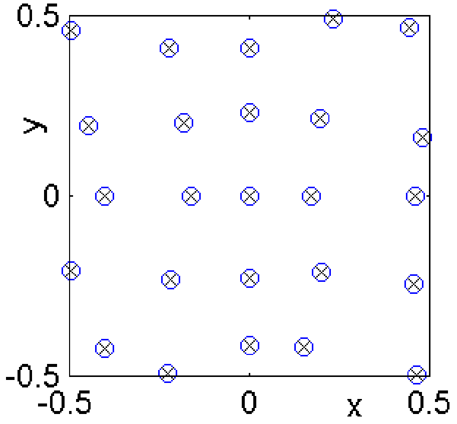
| max error | = $9.4 \cdot 10^{-9}$

Is ε = 0 best? Typically NOT

Function $f(x,y) = \dfrac{1}{25 + (x - \frac{1}{5})^2 + 2y^2}$

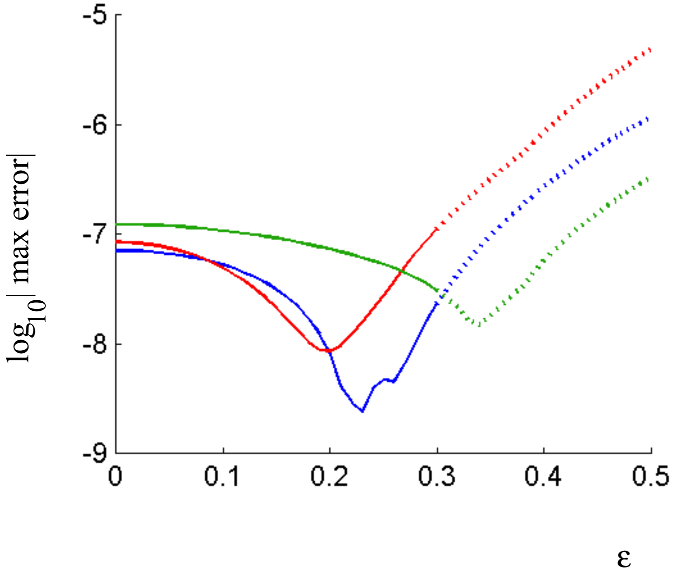


Sampling (25 scattered points)



Max interpolation errors over full square when using

Gaussians	(GA)
Inverse quadratics	(IQ)
Multiquadrics	(MQ)

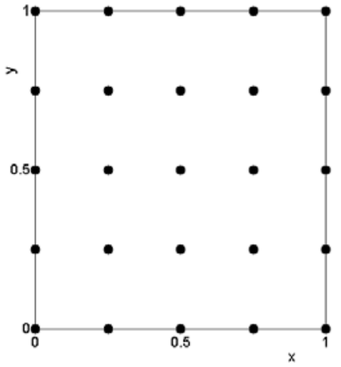


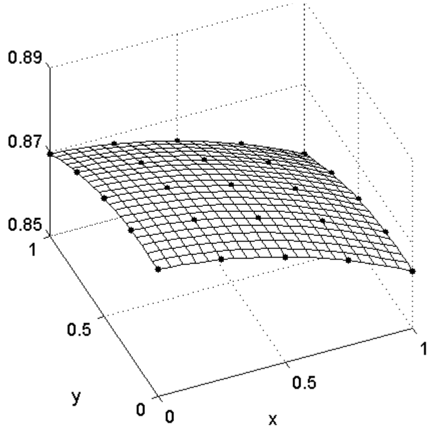
- Solid parts (below ε = 0.3) can not be computed directly with RBF-Direct - requires a stable algorithm.

Example

$$f(x, y) = \frac{59}{67+(x+\frac{1}{7})^2+(y-\frac{1}{11})^2}$$

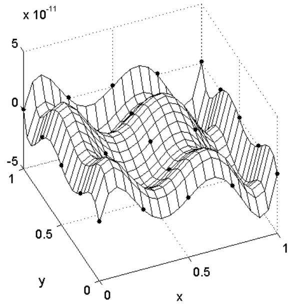
(same as in last example)
sampled at Cartesian points in [0, 1]×[0, 1]



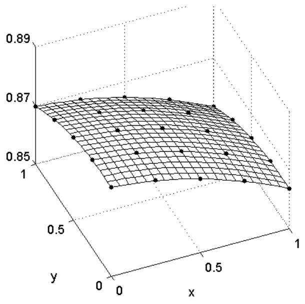


Multiquadric interpolant: $s(\underline{x},\varepsilon) = \frac{1}{\varepsilon^2} s_{-2}(\underline{x}) + s_0(\underline{x}) + \varepsilon^2 s_2(x) + \varepsilon^4 s_4(\underline{x}) + \varepsilon^6 s_6(\underline{x}) + \dots$ where

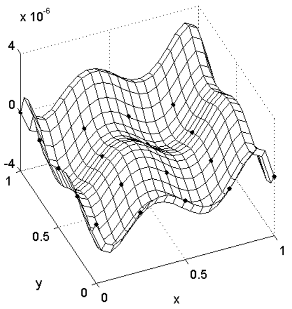
$s_{-2}(\underline{x})$



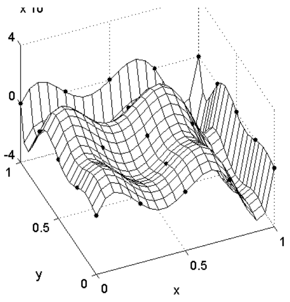
$s_0(\underline{x})$



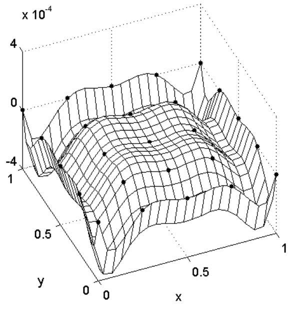
$s_2(\underline{x})$



$s_4(\underline{x})$



$s_6(\underline{x})$



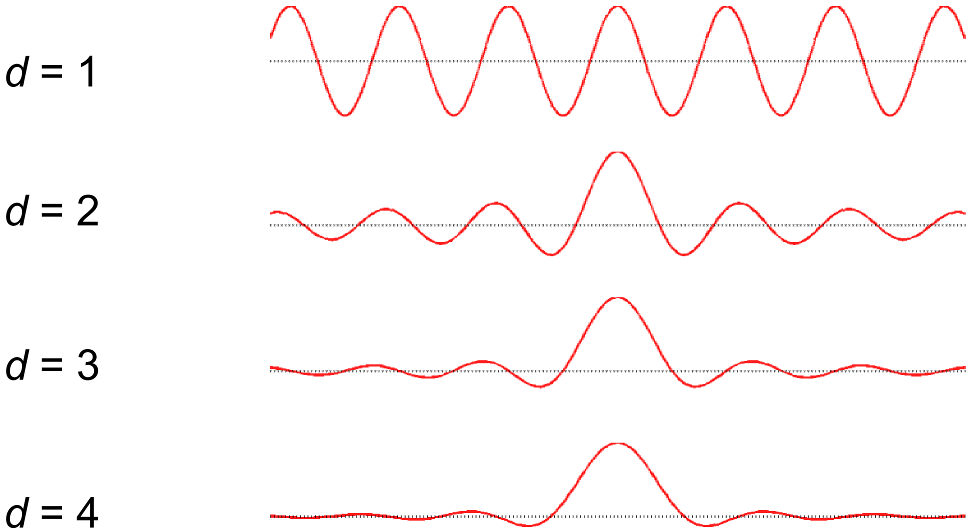
| max error | = $3.2 \cdot 10^{-8}$

The Bessel class of oscillatory radial functions

Schoenberg (1938), Richards (1985), Fornberg, Larsson, Wright (2006), Flyer (2006)

$$\phi_d(r) = \frac{J_{\frac{d}{2}-1}(\varepsilon r)}{(\varepsilon r)^{\frac{d}{2}-1}}$$

Converges to GA in limit of $d \to \infty$.



- The Fourier transform of $\phi_d(r)$ has compact support:

$$\hat{\phi}(\omega) = \frac{1}{2\pi}\int_{-\infty}^{\infty} \phi_d(r)\, e^{-i\omega r} dr = \begin{cases} (1-(\frac{\omega}{\varepsilon})^2)^{\frac{d}{2}-1} & -\frac{1}{\varepsilon} \le \omega \le \frac{1}{\varepsilon} \\ 0 & \text{otherwise} \end{cases}$$

- RBF interpolants based on $\phi_d(r)$ are non-singular for arbitrary point distributions in up to d dimensions. Proof similar to the one for GA RBFs, but based on $\phi_d(\|\underline{x}\|) = \frac{1}{(2\pi)^{d/2}} \int_{\|\underline{\omega}\|=1} e^{i\underline{x}\cdot\underline{\omega}}\, d\underline{\omega}$.

- Exact polynomial reproduction of arbitrary order on infinite lattice in d-D.

- When data is located along a straight line, RBF interpolants based on $\phi_d(r)$ will never diverge off the line.

Is the BE class the most general one possible which will never diverge in the ε → 0 limit?

With 4 points x_1, x_2, x_3, x_4 along a line (x-axis) and the interpolant evaluated at (x,y) off the line, it will never diverge. For more points: divergence unless the coefficients in $\phi(r) = a_0 + a_1(\varepsilon r)^2 + a_2(\varepsilon r)^4 + \ldots$ satisfy

5 points $\quad a_1 a_2^2 - 2 \cdot \frac{3}{2} a_1^2 a_3 + \frac{3}{1} a_0 a_2 a_3 = 0$

6 points $\quad a_2 a_3^2 - 2 \cdot \frac{4}{3} a_2^2 a_4 + \frac{4}{2} a_1 a_3 a_4 = 0$

7 points $\quad a_3 a_4^2 - 2 \cdot \frac{5}{4} a_3^2 a_5 + \frac{5}{3} a_2 a_4 a_5 = 0$

8 points $\quad a_4 a_5^2 - 2 \cdot \frac{6}{5} a_4^2 a_6 + \frac{6}{4} a_3 a_5 a_6 = 0$

If this pattern continues indefinitely, the following theorem will hold:

Theorem: The **ONLY** type of RBFs which will never diverge off a straight line is the BE type

$$\phi_d(r) = \frac{J_{\frac{d}{2}-1}(\varepsilon r)}{(\varepsilon r)^{\frac{d}{2}-1}} \ .$$

Conjecture: RBFs based on $\phi_d(r)$ are unique in never diverging (in < d dimensions), no matter how the (distinct) data points are located.

Noted above: GA RBFs form a special limiting case of the BE class, due to the relation

$$\lim_{\delta \to 0} 2^\delta \delta! \frac{J_\delta(2\sqrt{\delta}\ r)}{(2\sqrt{\delta}\ r)} = e^{-r^2} \ .$$

Also: Formulas for the RBF-QR algorithm in general 2-D domain (obeying the necessary 'counting rule') has so far only been found for GA and BE RBFs.

Numerical computations for small values of ε (near-flat RBFs)

It is possible to create algorithms that completely bypass ill-conditioning all the way into $\varepsilon \to 0$ limit, while using only standard precision arithmetic:

Concept: Find a computational path from $\underline{f}$ to $s(\underline{x},\varepsilon)$ that does **not** go via the ill-conditioned expansion coefficients $\underline{\lambda}$.

Contour-Padé algorithm

First algorithm of its kind; established that the concept is possible

Based on contour integration in a complex ε-plane.

Limited to relatively small N-values

First version	(Fornberg and Wright, 2004).
FFT-based rational approximation	(Gonnet, Pachón, Trefethen, 2011).
RBF-RA	(Wright and Fornberg, in progress).

RBF-QR method

Initially developed for nodes scattered over the surface of a sphere

No limit on N; cost about six times that of RBF-Direct

Original version (for nodes on sphere)	(Fornberg and Piret, 2007).
Versions for 1-D, 2-D, and 3-D	(Fornberg, Larsson, and Flyer, 2010).
Codes for generating RBF-FD stencils	(Larsson, Lehto, Heryudono, Fornberg, in progress).
Version without any infinite expansions	(Fornberg, Powell, Lehto, in progress).

Probably many more completely stable algorithms to come

The Contour-Padé algorithm:

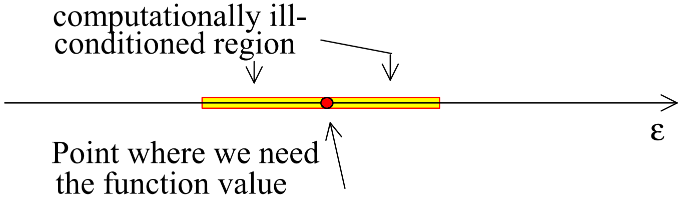
Numerically stable computations for ε near or equal to zero

Fornberg and Wright (2004)

Concept of algorithm:

Example: Evaluate $f(\varepsilon) = \underbrace{(1-\cos\varepsilon)\cdot\frac{1}{\varepsilon^2}}_{\text{'black box'}}$ for $\varepsilon = 0$ (or ε very small)

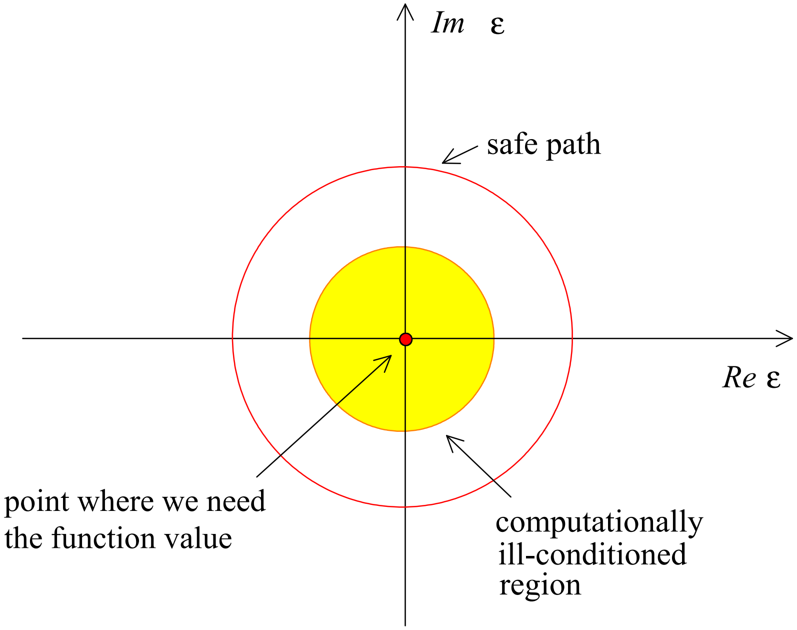
Difficulty: Numerical cancellation in forming $1-\cos\varepsilon$.


computationally ill-conditioned region
Point where we need the function value
ε

Solution: Think of $f(\varepsilon)$ as an analytic function of a complex variable ε.

Then $\varepsilon = 0$ is just a removable singularity.

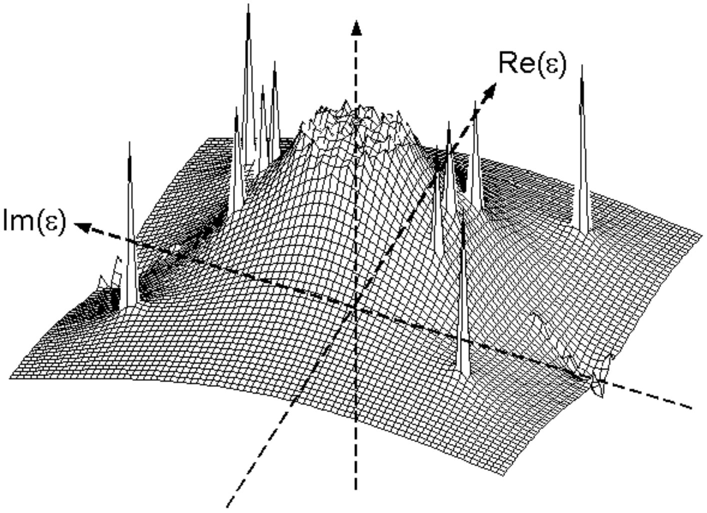
We can compute $f(0)$ as the average value of f around a circle, centered at the origin in a complex ε - plane.


Im ε
safe path
Re ε
point where we need the function value
computationally ill-conditioned region

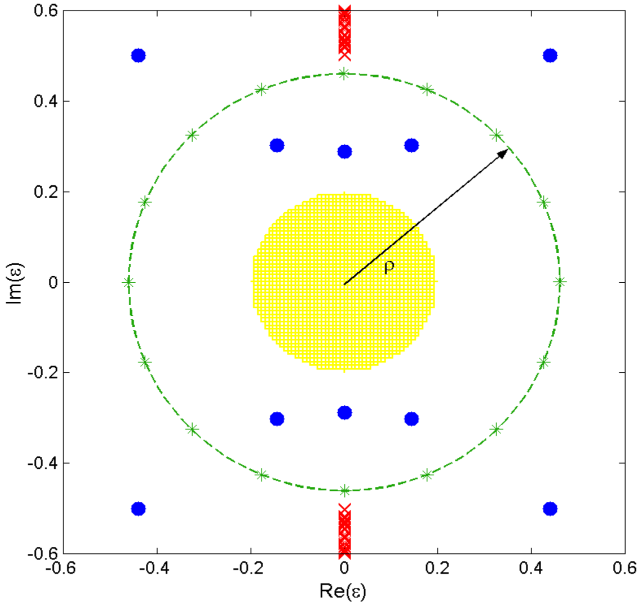
Contour integration - applied to the RBF case

RBF interpolant $s(\underline{x},\varepsilon)$ in complex ε - plane:

Typical condition number for A-matrix:



Possible computational path:



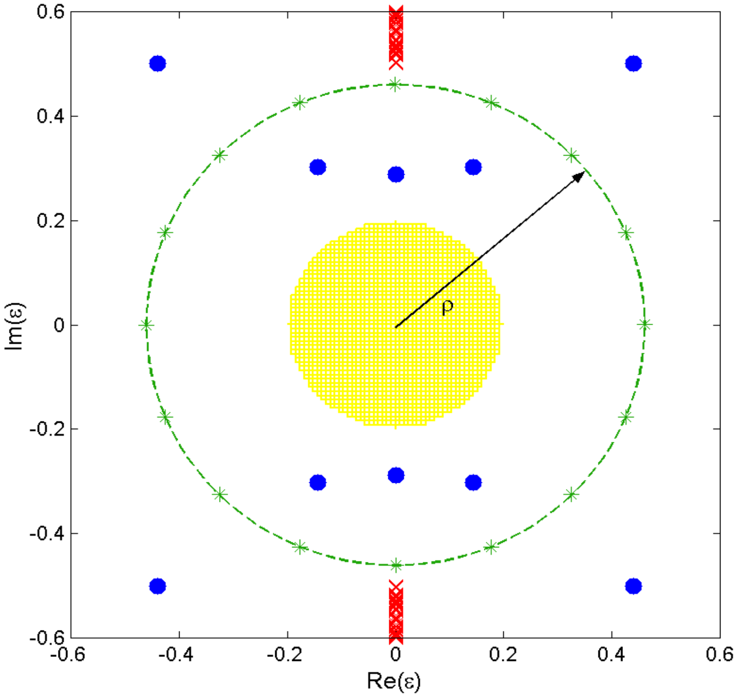
The interpolant $s(\underline{x},\varepsilon)$ is a meromorphic function of a complex variable ε.

Although $A(\boldsymbol{\varepsilon})$ is highly singular for $\boldsymbol{\varepsilon}$ near zero,

$\boldsymbol{\varepsilon} = 0$ is almost always a removable singularity for $s(\underline{x}, \boldsymbol{\varepsilon})$; else it is a low order pole.

The flat limit ε = 0 corresponds simply to a removable singularity in a complex ε-plane for the for the RBF interpolant $s(x,\varepsilon)$.

TASK: Given data f_k at nodes $\underline{x}_k$, $k = 0,1,...,N$, evaluate interpolant $s(\underline{x},\varepsilon)$ for very small or zero values of ε.



SOLUTION:

Evaluate instead $s(\underline{x},\varepsilon)$ around a larger circular path in the ε-plane. A complex FFT on the function values gives the coefficients of a Laurent expansion in a vicinity of the path:

$$s(\underline{x},\varepsilon) = \ldots + \varepsilon^{-6} c_{-6} + \varepsilon^{-4} c_{-4} + \varepsilon^{-2} c_{-2} + c_0 + \varepsilon^2 c_2 + \varepsilon^4 c_4 + \varepsilon^6 c_6 + \ldots$$

Convert the part with negative powers by Padé's method to rational form:

$$s(\underline{x},\varepsilon) = \{\text{rational expression}\} + c_0 + \varepsilon^2 c_2 + \varepsilon^4 c_4 + \varepsilon^6 c_6 + \ldots$$

We can now evaluate $s(\underline{x},\varepsilon)$ for any value of ε inside the contour.

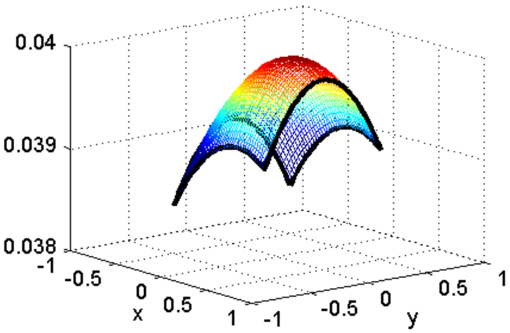
- Algorithm computationally stable up to a few hundred nodes in 2-D, several hundreds in 3-D

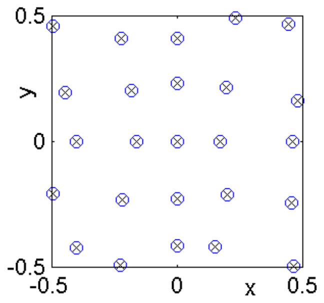
- Before this algorithm, it was wrongly believed that computing with an RBF basis (especially near-flat RBF) was intrinsically ill-conditioned ("uncertaincy principle" by Schaback).

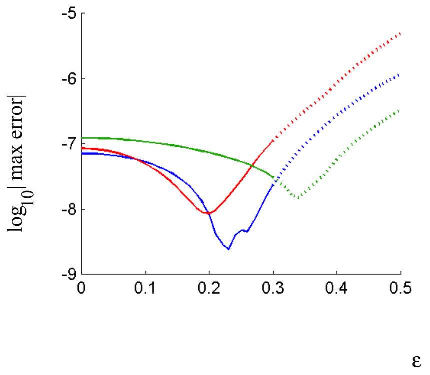
Obtaining the interpolant s(x) by means of $A\underline{\lambda} = \underline{f}$ followed by $s(\underline{x}) = \Sigma_{k=0}^{N} \lambda_k \phi(\|\underline{x} - \underline{x}_k\|)$ is merely an unstable algorithm for a stable problem.

Heuristic explanation of error curve for small ε

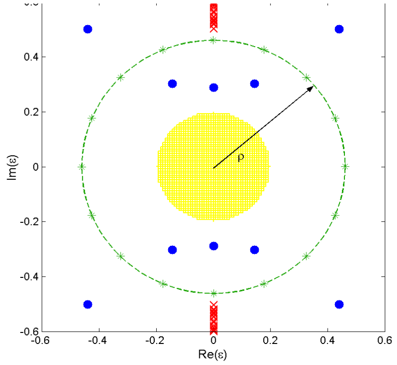
Recall 1: Function $f(x,y)=\frac{1}{25+(x-\frac{1}{5})^2+2y^2}$







Recall 2: $s(\underline{x},\varepsilon)=\{\text{rational expression}\}+c_0+\varepsilon^2 c_2+\varepsilon^4 c_4+\varepsilon^6 c_6+\ldots$



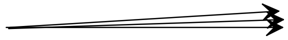
Observation: When ε is sufficiently small (inside all poles), then $s(\underline{x},\varepsilon)$ deviates from its $\varepsilon\to 0$ limit as $O(\varepsilon^2)$.

Describes the typical error curve shape for small ε.

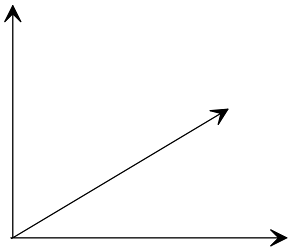
The concept for the RBF-QR method

Recognize that the existence of an ill-conditioned **basis** does not imply that the spanned **space** is bad.

Ex. 1: 3-D space

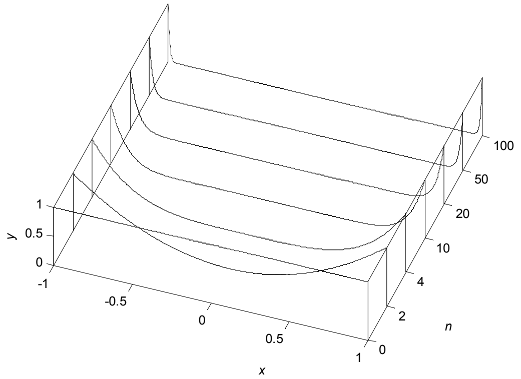


Bad Basis



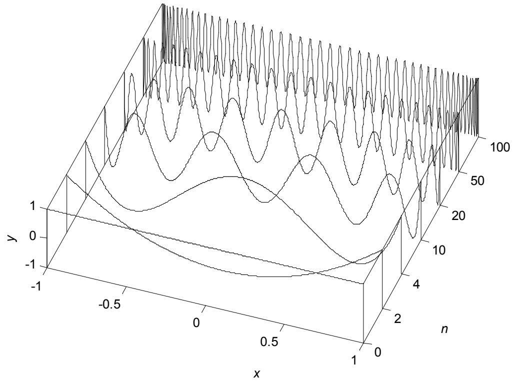
Good Basis

Ex. 2: Polynomials of degree ≤ 100



x^n , $n = 0, 1, \ldots, 100$

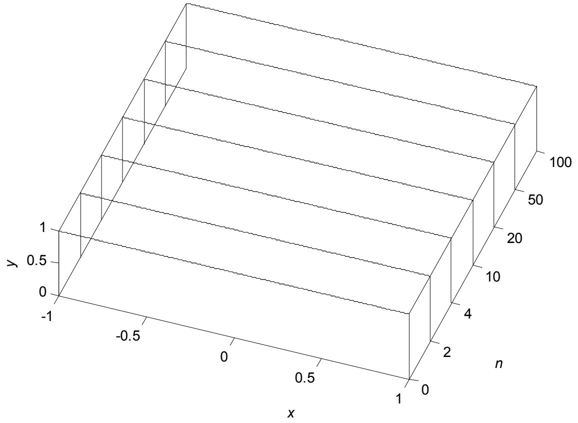
Bad Basis



$T_n(x)$, $n = 0, 1, \ldots, 100$

Good Basis

Ex. 3: Space spanned by RBFs in their flat $\varepsilon \to 0$ limit



Bad Basis

- The spanned space turns out to be excellent for computational work - just the basis that is bad.
- Is there any Good Basis in exactly the same space?
- RBF-QR finds such a basis through some analytical expansions, leading to numerical steps that all remain completely stable even in the flat basis function limit.

Background to RBF-QR for spheres: Spherical Harmonics (SPH)

Spherical harmonics: Restriction to surface of unit sphere of simple polynomials in *x*, *y*, *z*:

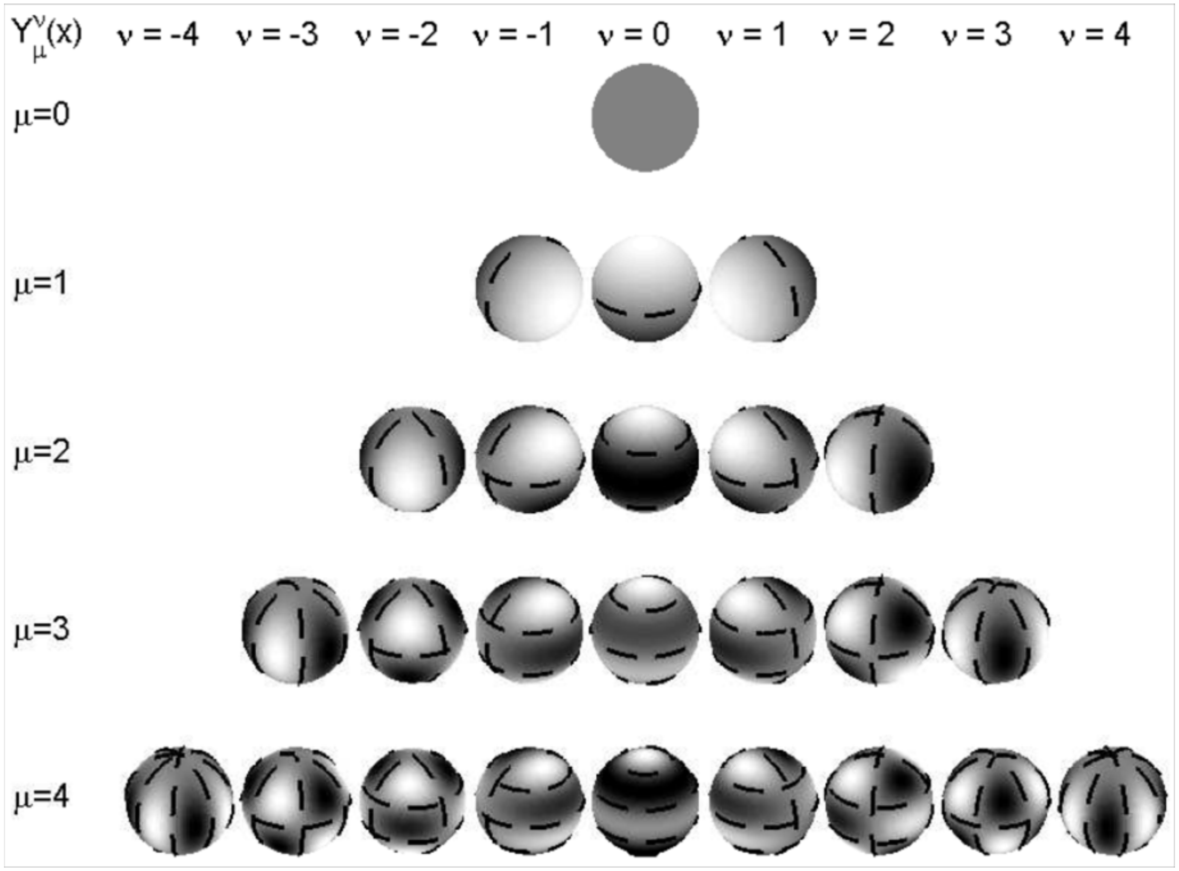
$\nu=$

		...	-2	-1	0	1	2	...
	$\mu=0$				$\frac{1}{2\sqrt{\pi}}1$			
$Y_\mu^\nu(x,y,z)$	1			$-\frac{1}{2}\sqrt{\frac{3}{2\pi}}\,y$	$\frac{1}{2}\sqrt{\frac{3}{\pi}}\,z$	$-\frac{1}{2}\sqrt{\frac{3}{2\pi}}\,x$		
	2		$\frac{1}{2}\sqrt{\frac{15}{2\pi}}\,xy$	$-\frac{1}{2}\sqrt{\frac{15}{2\pi}}\,zy$	$\frac{1}{4}\sqrt{\frac{5}{\pi}}\,(3z^2-1)$	$-\frac{1}{2}\sqrt{\frac{15}{2\pi}}\,zx$	$\frac{1}{4}\sqrt{\frac{15}{2\pi}}\,(x^2-y^2)$	
	...	...	...	...	...	...	...	...

- Counterpart to Fourier modes around periphery of unit circle
- Orthogonal
- Uniform resolution over surface
- Spectral accuracy for PDEs

but

- Not associated with any particular node set
- No counterpart to FFT
- No opportunities for variable resolution



Expansions of RBFs in terms of SPH

RBFs, centered on the surface of the unit sphere, can be expanded in SPH as follows:

$$\phi(\|\underline{x}-\underline{x}_i\|)=\sum_{\mu=0}^{\infty}\ \sum_{\nu=-\mu}^{\mu}{}' \left\{\varepsilon^{2\mu} c_{\mu,\varepsilon}\ Y_\mu^\nu(\underline{x}_i)\right\}\ Y_\mu^\nu(\underline{x})$$

where

MQ: $\phi(r)=\sqrt{1+(\varepsilon r)^2}$ $\qquad c_{\mu,\varepsilon}=\dfrac{-2\pi(2\varepsilon^2+1+(\mu+\frac{1}{2})\sqrt{1+4\varepsilon^2})}{(\mu+\frac{3}{2})(\mu+\frac{1}{2})(\mu-\frac{1}{2})}\left(\dfrac{2}{1+\sqrt{4\varepsilon^2+1}}\right)^{2\mu+1}$

IMQ: $\phi(r)=\dfrac{1}{\sqrt{1+(\varepsilon r)^2}}$ $\qquad c_{\mu,\varepsilon}=\dfrac{4\pi}{(\mu+\frac{1}{2})}\left(\dfrac{2}{1+\sqrt{4\varepsilon^2+1}}\right)^{2\mu+1}$

GA: $\phi(r)=e^{-(\varepsilon r)^2}$ $\qquad c_{\mu,\varepsilon}=\dfrac{4\pi^{3/2}}{\varepsilon^{2\mu+1}}e^{-2\varepsilon^2}\ I_{\mu+\frac{1}{2}}(2\varepsilon^2)$

Notes:

- Σ' denotes halving the term for $v = 0$.
- The singularity at $\varepsilon = 0$ in GA case is removable.

Key points of the RBF-QR algorithm:

- There is no loss of accuracy in computing $c_{\mu,\varepsilon}\ Y_\mu^\nu(\underline{x}_i)$, even if $\varepsilon\to 0$.
- The factors $\varepsilon^{2\mu}$ will be *analytically* kept out of the numerical algorithm.

RBF-QR method on a sphere:

Fornberg and Piret (2007)

The original ill-conditioned basis, and its expansion in terms of successive SPH functions:

$$\left\{\begin{array}{llllll}
\phi(\|\underline{x}-\underline{x}_1\|) & = c_{0,\varepsilon}Y_0^0(\underline{x}_1)Y_0^0(\underline{x}) & + \ \varepsilon^2 c_{1,\varepsilon}Y_1^{-1}(\underline{x}_1)Y_1^{-1}(\underline{x})+\varepsilon^2 c_{1,\varepsilon}Y_1^0(\underline{x}_1)Y_1^0(\underline{x})+\varepsilon^2 c_{1,\varepsilon}Y_1^1(\underline{x}_1)Y_1^1(\underline{x}) & + \ \varepsilon^4\{\ldots\} & + \ \varepsilon^6\{\ldots\}+\ldots \\
\phi(\|\underline{x}-\underline{x}_2\|) & = c_{0,\varepsilon}Y_0^0(\underline{x}_2)Y_0^0(\underline{x}) & + \ \varepsilon^2 c_{1,\varepsilon}Y_1^{-1}(\underline{x}_1)Y_1^{-1}(\underline{x})+\varepsilon^2 c_{1,\varepsilon}Y_1^0(\underline{x}_1)Y_1^0(\underline{x})+\varepsilon^2 c_{1,\varepsilon}Y_1^1(\underline{x}_1)Y_1^1(\underline{x}) & + \ \varepsilon^4\{\ldots\} & + \ \varepsilon^6\{\ldots\}+\ldots \\
\cdots & = \ \cdots & \quad \cdots & \quad \cdots & \quad \cdots \\
\phi(\|\underline{x}-\underline{x}_n\|) & = c_{0,\varepsilon}Y_0^0(\underline{x}_n)Y_0^0(\underline{x}) & + \ \varepsilon^2 c_{1,\varepsilon}Y_1^{-1}(\underline{x}_1)Y_1^{-1}(\underline{x})+\varepsilon^2 c_{1,\varepsilon}Y_1^0(\underline{x}_1)Y_1^0(\underline{x})+\varepsilon^2 c_{1,\varepsilon}Y_1^1(\underline{x}_1)Y_1^1(\underline{x}) & + \ \varepsilon^4\{\ldots\} & + \ \varepsilon^6\{\ldots\}+\ldots
\end{array}\right.$$

In matrix x vector form:

$$\begin{bmatrix} \phi(\|\underline{x}-\underline{x}_1\|) \\ \phi(\|\underline{x}-\underline{x}_2\|) \\ \cdots \\ \phi(\|\underline{x}-\underline{x}_n\|) \end{bmatrix} = \begin{array}{c} \begin{array}{cccccccccc} \overbrace{\qquad\qquad}^{O(1)} & \multicolumn{3}{c}{\overbrace{\qquad\qquad\qquad\qquad\qquad\qquad}^{O(\varepsilon^2)}} & \multicolumn{5}{c}{\overbrace{\qquad\qquad\qquad\qquad\qquad\qquad}^{O(\varepsilon^4)}} & \cdots \end{array} \\ \begin{bmatrix}
c_{0,\varepsilon}Y_0^0(\underline{x}_1) & \varepsilon^2 c_{1,\varepsilon}Y_1^{-1}(\underline{x}_1) & \varepsilon^2 c_{1,\varepsilon}Y_1^0(\underline{x}_1) & \varepsilon^2 c_{1,\varepsilon}Y_1^1(\underline{x}_1) & \varepsilon^4\ldots & \varepsilon^4\ldots & \varepsilon^4\ldots & \varepsilon^4\ldots & \varepsilon^4\ldots & \cdots \\
c_{0,\varepsilon}Y_0^0(\underline{x}_1) & \varepsilon^2 c_{1,\varepsilon}Y_1^{-1}(\underline{x}_2) & \varepsilon^2 c_{1,\varepsilon}Y_1^0(\underline{x}_2) & \varepsilon^2 c_{1,\varepsilon}Y_1^1(\underline{x}_2) & \varepsilon^4\ldots & \varepsilon^4\ldots & \varepsilon^4\ldots & \varepsilon^4\ldots & \varepsilon^4\ldots & \cdots \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
c_{0,\varepsilon}Y_0^0(\underline{x}_1) & \varepsilon^2 c_{1,\varepsilon}Y_1^{-1}(\underline{x}_n) & \varepsilon^2 c_{1,\varepsilon}Y_1^0(\underline{x}_n) & \varepsilon^2 c_{1,\varepsilon}Y_1^1(\underline{x}_n) & \varepsilon^4\ldots & \varepsilon^4\ldots & \varepsilon^4\ldots & \varepsilon^4\ldots & \varepsilon^4\ldots & \cdots
\end{bmatrix} \end{array}
\begin{bmatrix} Y_0^0(\underline{x}) \\ Y_1^{-1}(\underline{x}) \\ Y_1^0(\underline{x}) \\ Y_1^1(\underline{x}) \\ Y_2^{-2}(\underline{x}) \\ Y_2^{-1}(\underline{x}) \\ Y_2^0(\underline{x}) \\ Y_2^1(\underline{x}) \\ Y_2^2(\underline{x}) \\ \cdots \\ \cdots \\ \cdots \end{bmatrix}$$

Factorize the matrix as $Q \cdot E \cdot R$

$$\begin{bmatrix} & & \\ & Q & \\ & & \end{bmatrix}
\begin{bmatrix} 1 & & & & & \\ & \varepsilon^2 & & & & \\ & & \varepsilon^2 & & & \\ & & & \varepsilon^2 & & \\ & & & & \varepsilon^4 & \\ & & & & & \ddots \end{bmatrix}
\underbrace{\begin{bmatrix}
\bullet & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdots \\
 & \bullet & \bullet & \bullet & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdots \\
 & & \bullet & \bullet & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdots \\
 & & & \bullet & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdots \\
 & & & & \bullet & \bullet & \bullet & \bullet & \bullet & \cdot & \cdots \\
 & & & & & \ddots & \cdots & \cdots & \cdots & \cdots & \cdots
\end{bmatrix}
\begin{bmatrix} \\ \\ \\ \\ \\ \\ \end{bmatrix}}_{\text{well conditioned basis, spanning the same space}}$$

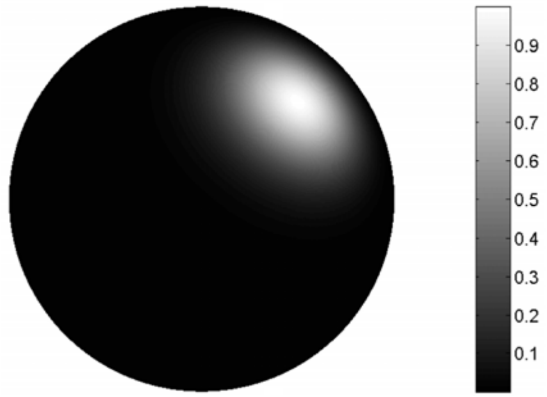
New basis is computed entirely without numerical cancellations (even if ε is very small), and it spanns exactly the same space as the original base - hence it gives the same interpolants, etc.

Critical "Counting" - issue: The number of eigenvalues of sizes O(ε^0),O(ε^2),O(ε^4), ... exactly matched the number of terms of matching powers in the RBF expansion (viz. 1,3,5,7,...) - else the method's rate of improvement would have fallen short of the problem's rate of worsening.

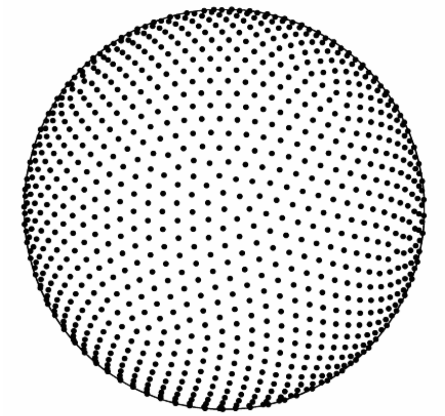
Interpolation on a sphere - RBF-Direct vs. RBF-QR

Test function:

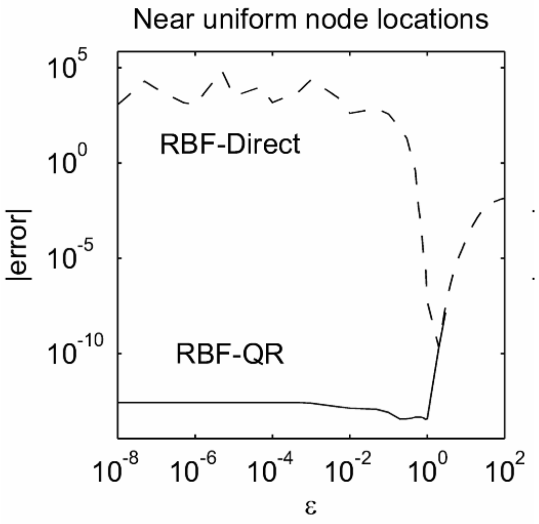
$$f(\underline{x}) = e^{-7(x+\frac{1}{2})^2-8(y+\frac{1}{2})^2-9(z-\frac{1}{\sqrt{2}})^2}$$



1849 minimal energy nodes



Errors as function of ε



RBF-Direct:

$s(\underline{x}) = \sum_{k=1}^{N} \lambda_k \, \phi(\| \underline{x} - \underline{x}_k \|)$ with the λ_k from solving $A\,\underline{\lambda} = \underline{f}$; Cond($A$) = $O(\varepsilon^{-84})$.

Lowering ε by a factor of 100 increases cond(A) a factor of $100^{84} = 10^{168}$. Maintaining >10 digits of accuracy by means of extended precision arithmetic requires the numerical precision to be raised from standard 16 digits to around 180 digits.
Extended precision generally not a cost effective approach.

RBF-QR:

With the new basis functions in exactly the same approximation space, cond(A') remains $O(1)$. Cost increase over RBF-Direct about 6 times, no matter how small ε is used.

RBF-QR in 1-D, 2-D and 3-D

Fornberg, Larsson, Flyer (2011)

Concept: Same as on the sphere (but so far limited to GA and BE RBFs); illustrated here in 2-D:

In the case of GA RBFs:

$$\phi(\|\underline{x}-\underline{x}_i\|) = e^{-\varepsilon^2((x-x_i)^2+(y-y_i)^2)} = e^{-\varepsilon^2({x_i}^2+{y_i}^2)} \cdot e^{-\varepsilon^2(x^2+y^2)} \cdot e^{2\varepsilon^2(xx_i+yy_i)}$$

The first factor is an irrelevant scalar multiplier; We Taylor expand the last factor

$$e^{2\varepsilon^2(xx_i+yy_i)} = 1 + 2\varepsilon^2(xx_i+yy_i) + \frac{2^2\varepsilon^4}{2!}(xx_i+yy_i)^2 + \frac{2^3\varepsilon^6}{3!}(xx_i+yy_i)^3 + \ldots$$

which leads to the new basis functions

$$e^{-\varepsilon^2(x^2+y^2)} \cdot \{\{1\},\ \{x,y\},\ \{x^2,xy,y^2\},\ \{x^3,x^2y,xy^2,y^3\},\ldots\} \qquad (1)$$

Counting correct: The groups contain {1,2,3,4,...} entries; matches the eigenvalue pattern for *A*-matrix. In *d*-D, $d \ge 1$, the same procedure gives always perfect 'counting' agreement.

Additional technicalities: RBF-QR based on the new basis does not degrade as $\varepsilon \to 0$, but the basis functions are reminiscent of increasing degree monomials. Problems associated with that can be bypassed by two further steps:

- Transform to polar coordinates: Replace (1) by

$$e^{-\varepsilon^2 r^2} \cdot \{\{1\},\ r\{\cos\theta,\sin\theta\},\ r^2\{1,\cos 2\theta,\sin 2\theta\},\ r^3\{\cos\theta,\sin\theta,\cos 3\theta,\sin 3\theta\},\ldots\}$$

- In 3-D, use Chebyshev radially and SPH angularly rather than Chebyshev and trig functions
- Replace powers of *r* by Chebyshev polynomials radially.

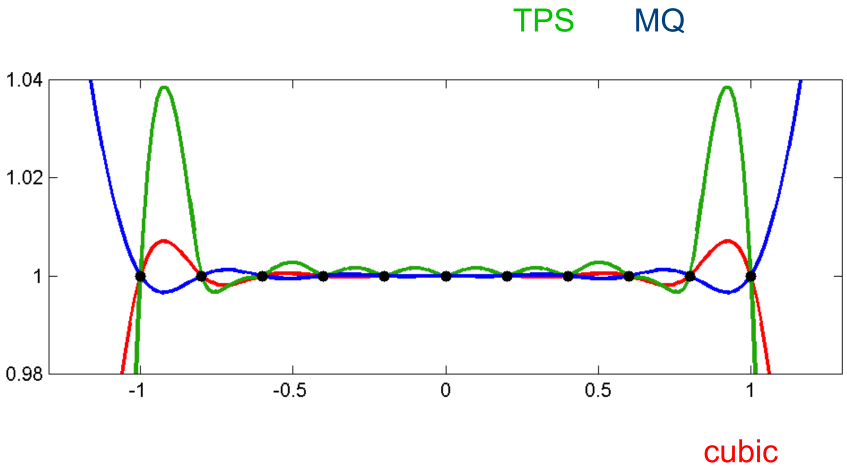
Summary:

- About 6-10 times the cost for RBF-Direct.
- Numerically stable all the way into the $\varepsilon \to 0$ limit.
- Can use *N* - values in the thousands, in 1-D, 2-D, 3-D.

One idea for reduction of RP caused by edge effects

Fornberg, Driscoll, Wright, Charles (2002)

Applied to constant data, RBF interpolants do NOT become constant functions:



Cubic RBF is a case of cubic splines - with the totally weird extra boundary conditions

$$\begin{cases} s''(\ 1) = 2s'(1) - \ s'(-1) \ - \tfrac{3}{2}(s(1)+s(-1)) \\ s''(-1) = \ s'(1) - 2s'(-1) \ - \tfrac{3}{2}(s(1)+s(-1)) \end{cases}$$

For standard cubic splines, two much better boundary treatments are routinely used:

1. Natural spline: $s''(x) = 0$ at $x = \pm 1$

2. Not-a-Knot: Disallow jump in third derivative one node in from each edge

Both approaches carry immediately over, first to cubic RBFs in 1-D, and then to any RBFs in *d*-D:

RBF equivalents for two BC options: Scattered points in 1-D

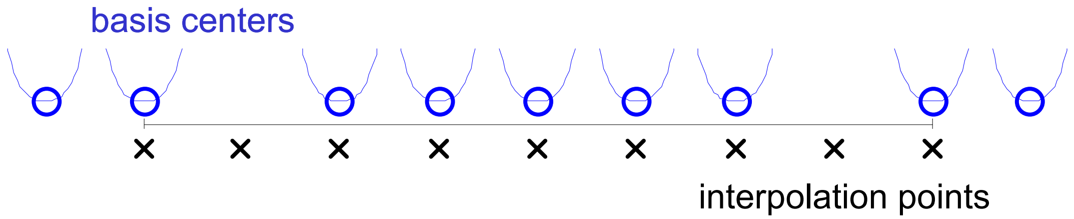
1. Natural spline:

Instead of $s(x) = \sum_{k=1}^{n} \lambda_k \phi(|x - x_k|)$, use $\begin{cases} s(x) = a + bx + \sum_{k=1}^{n} \lambda_k \phi(|x - x_k|) \\ \text{with constraints } \sum \lambda_k = 0,\ \sum \lambda_k x_k = 0 \end{cases}$.

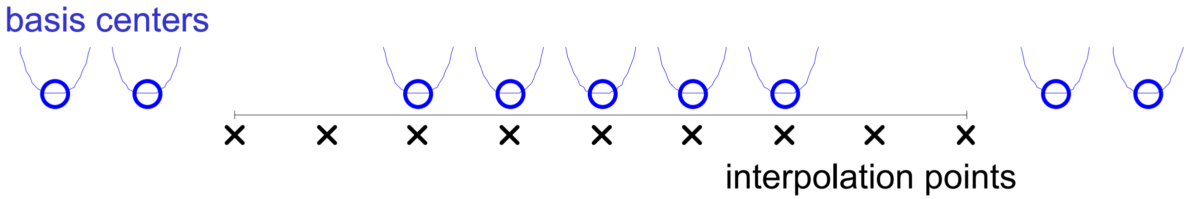
Proof: For $x \ge x_n$, $s(x) = \sum_{k=1}^{n} \lambda_k (x - x_k)^3$ and $s''(x) = \frac{d^2}{dx^2} \sum_{k=1}^{n} \lambda_k (x - x_k)^3 = 6x \sum_{k=1}^{n} \lambda_k - 6x \sum_{k=1}^{n} \lambda_k x_k = 0$. Similarly, $s''(x) = 0$ also for $x \le x_1$.

2. Not-a-Knot:

Separate interpolation points from RBF (basis) centers



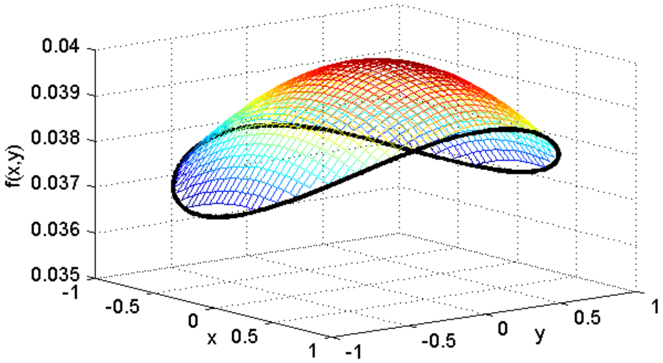
Carry it out one step further - we get '*super-not-a-knot*'



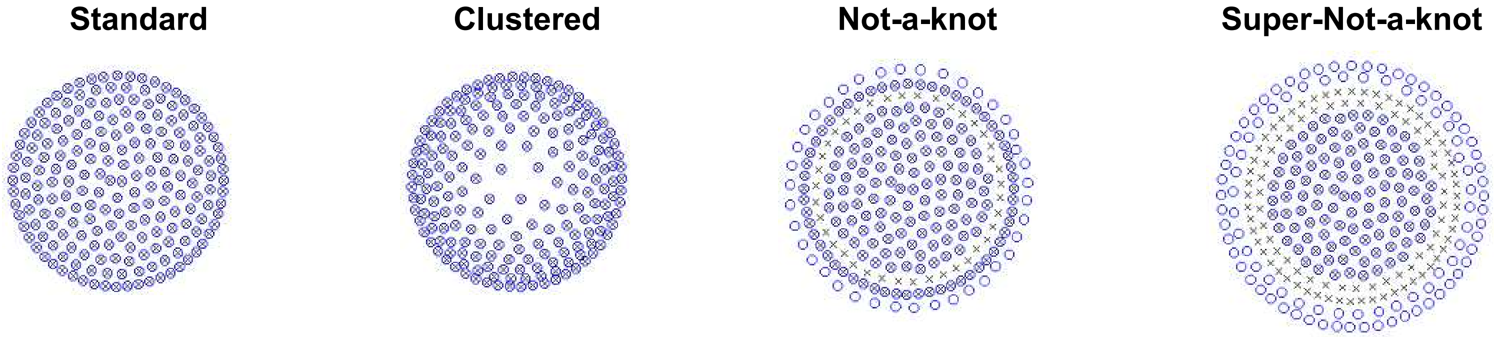
Both ideas carry over to any RBF-type over any multi-D domain:

... and for scattered points in 2-D

Test function: $f(x,y) = \dfrac{1}{25 + (x - \frac{1}{5})^2 + 2y^2}$



Different edge treatments:

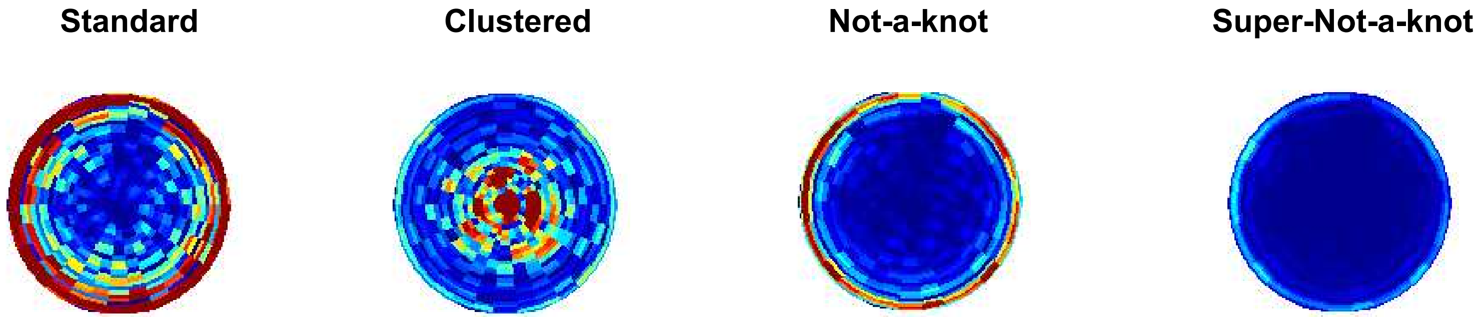

Standard

Clustered

Not-a-knot

Super-Not-a-knot

Errors 0 to 10^{-6}


Standard

Clustered

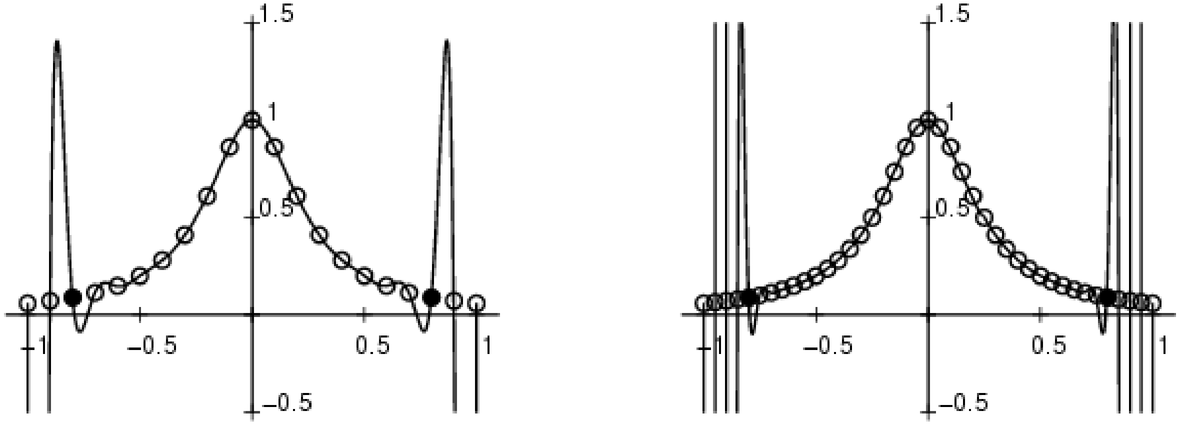
Not-a-knot

Super-Not-a-knot

RBFs, the Runge Phenomenon, and spatially variable ε

Fornberg and Zuev (2007)

Runge phenomenon (RP) for polynomials:



(a) $n = 21$ (b) $n = 41$

For equispaced polynomial interpolation on [-1,1]:

Error: $E(z,n) \approx e^{n(\psi(z_0)-\psi(z))}$ with $\psi(z) = -\frac{1}{2}\,\mathrm{Re}[(1-z)\ln(1-z)-(-1-z)\ln(-1-z)]$

- Function $f(x)$ enters only through z_0 - location of nearest singularity

- Typically 'controlled' by Chebyshev-type node refinement at edges (more options available with RBFs).

The Runge Phenomenon

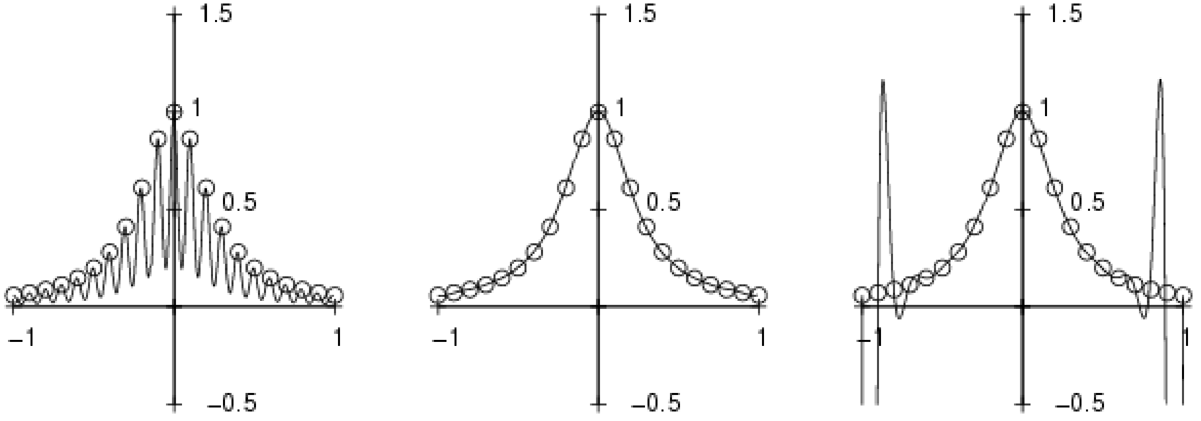
Runge phenomenon (RP) for RBFs   Two main causes:

- Edge effect, as $\varepsilon \to 0$

RBF interpolant then becomes increasingly 'polynomial-like'

Figures show GA interpolation of

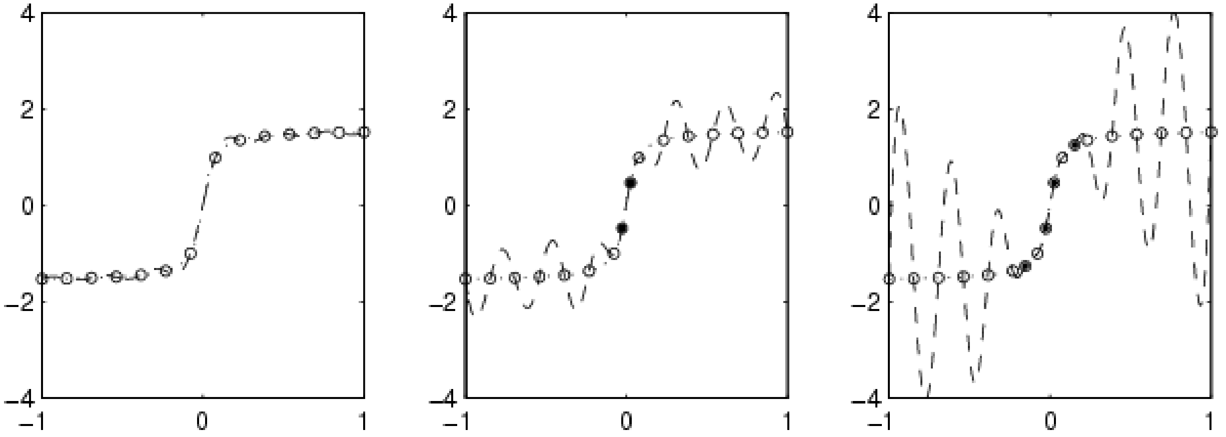
$$f(x) = \frac{1}{1+16x^2}$$



(a) $\varepsilon = 30$ (b) $\varepsilon = 3.5$ (c) $\varepsilon = 0.5$

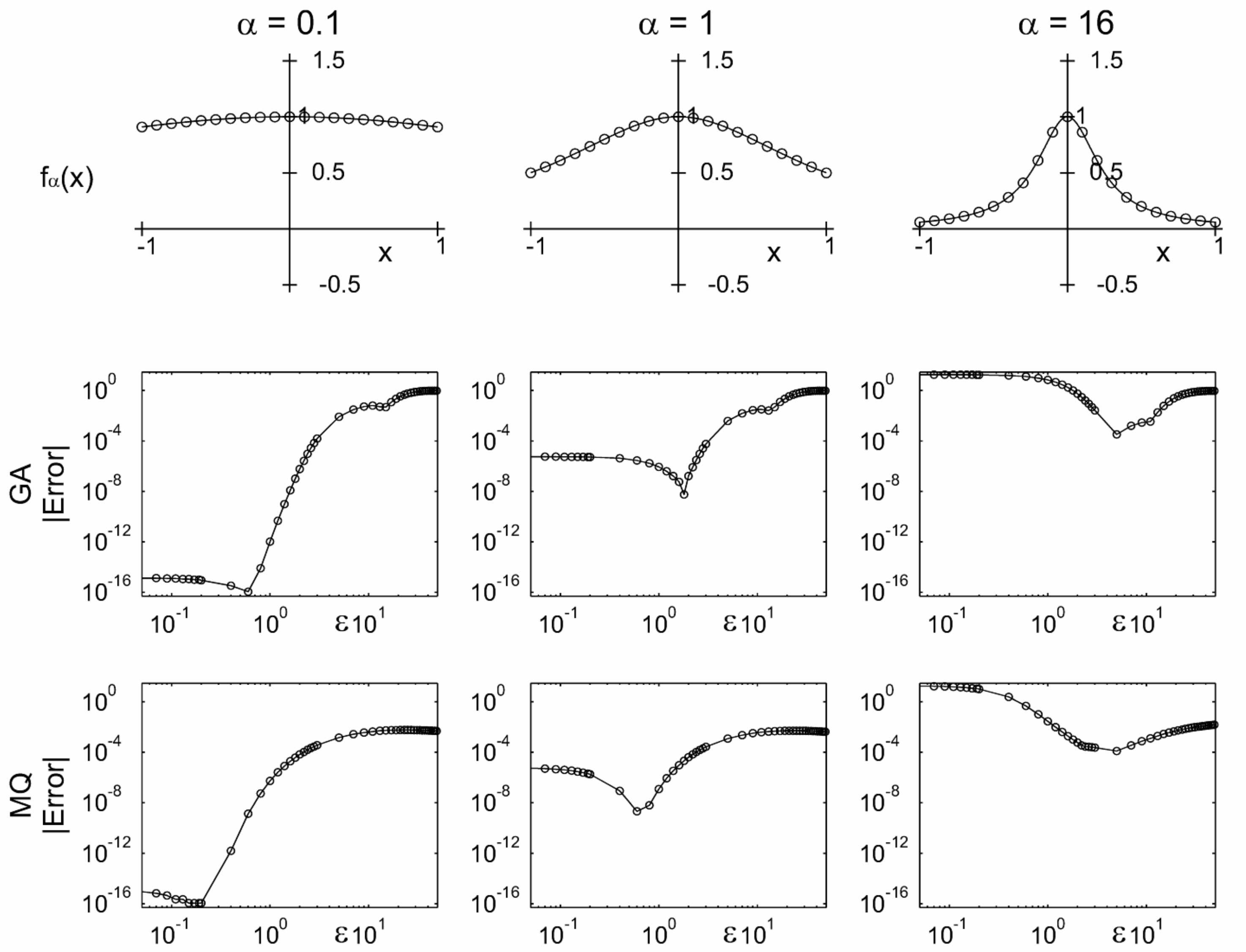
- Result of local mesh refinement

MQ RBF interpolation of $f(x) = \arctan(20x)$



(a) 14 nodes. (b) 16 nodes. (c) 18 nodes.

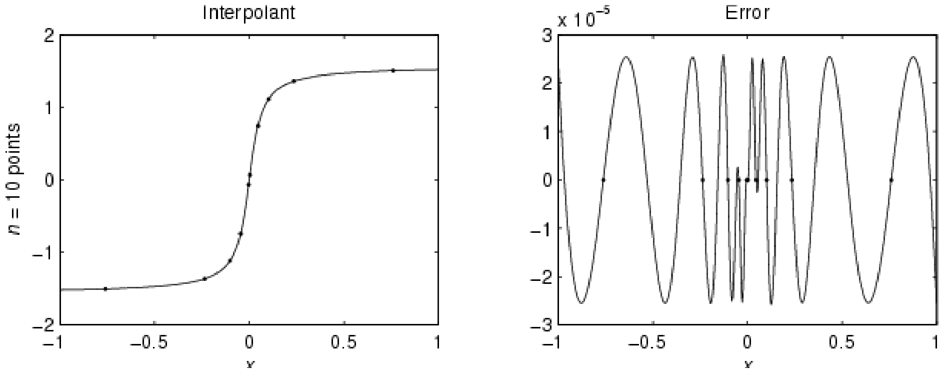
Reversal of error trend as $\varepsilon \to 0$ is due to the RP



- Rightmost column shows same test case as the top one on the previous slide
- Error degradation for small ε is less severe whenever the polynomial RP is smaller
- Error increase as $\varepsilon \to 0$ is due to RP. Key goal: Reduce or eliminate the RP!

Opportunities with variable x_i , ε_i RBF interpolation

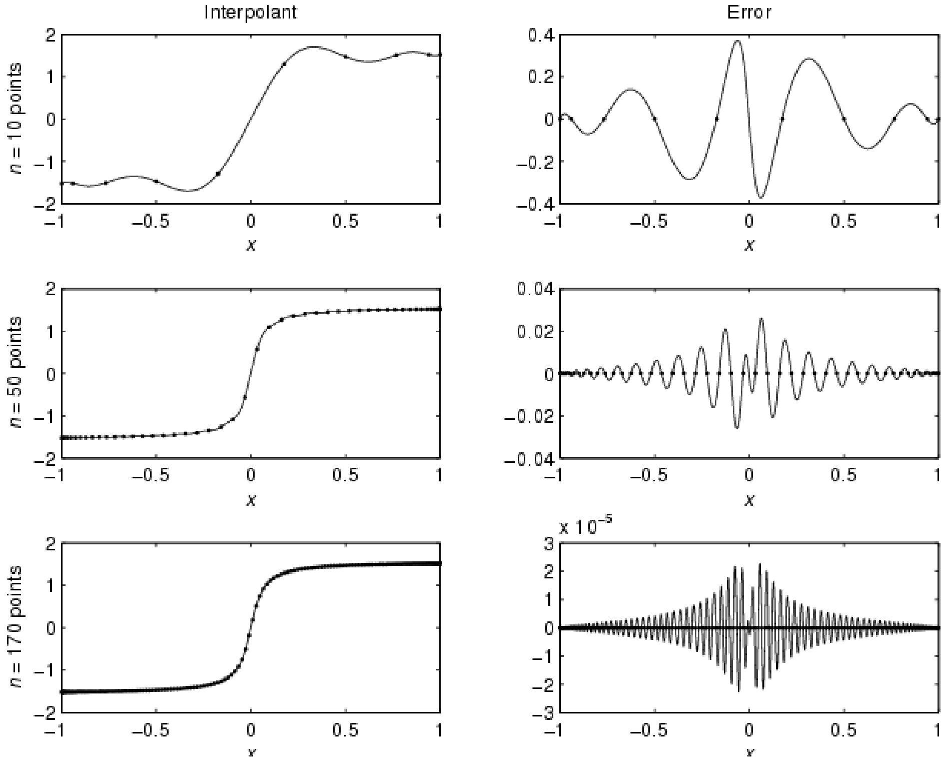
10-point MQ variable x_i , ε_i interpolation



Result above obtained by numerical optimization (genetic algorithm).

Too expensive for routine use, but
illustrates great potential, if one can find much faster strategies for finding good x_i and ε_i .

Chebyshev interpolation needs 170 points for matching accuracy



Polynomial Reproduction

Exact 'reproduction' of low order polynomials over unbounded node sets for certain RBF types; first shown on lattices, then for arbitrary infinite node distributions (Buhmann, Powell), later for BE (Flyer)

Type of basis function	Radial function $\phi(r)$	Reproduced degree in *d*-D
BE	$J_{k/2-1}/(\varepsilon r)^{k/2-1}$	all
cubic	r^3	*d* + 2
TPS	$r^2 \log r$	*d* + 1
linear	r	*d*
MQ	$\sqrt{1+(\varepsilon r)^2}$	*d*
IMQ	$1/\sqrt{1+(\varepsilon r)^2}$	*d* - 2
IQ	$1/(1+(\varepsilon r)^2)$	*d* - 3
GA	$e^{-(\varepsilon r)^2}$	none
Compact RBFs (Wendland, etc.)	[many formulas]	none

Not hard to spot: BE exceptional (as in many other situations; result not true in BE → GA limit).

The degree for polynomial that can be reproduced is an immediate reflection of the rate of growth for $\phi(r)$ as $|r| \to \infty$.

Applications of Polynomial Reproduction

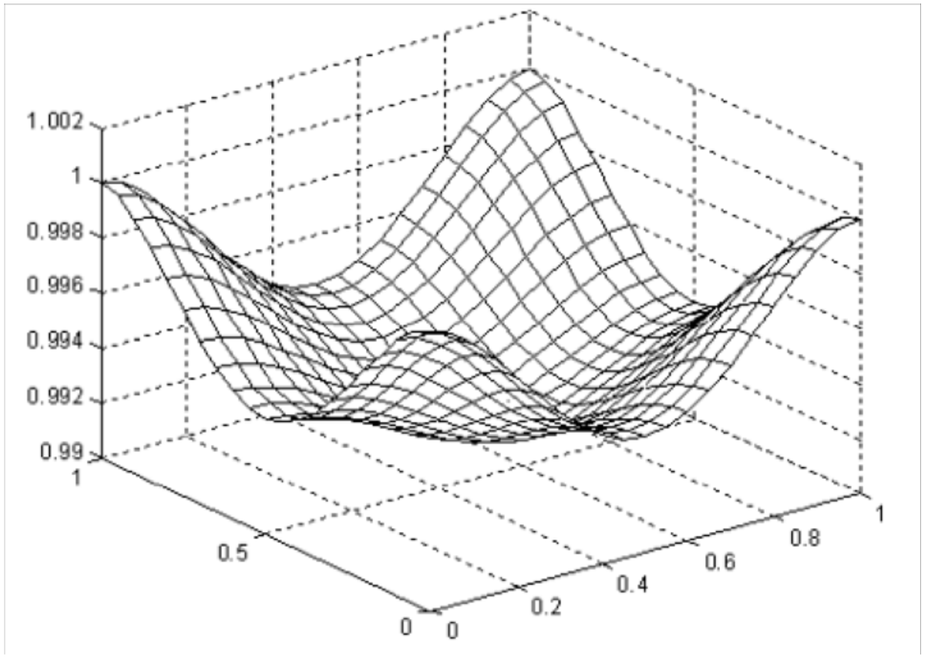
Analysis of RBF convergence rates on infinite lattices:

This often gives misleading results... Approach is based on low degree polynomials, and might not reflect spectral convergence properties that are typical of RBF approximations over bounded regions.

Stationary / Saturation errors:

With most numerical methods, increased node density translates to increased accuracy.
When using RBF-Direct, cond(*A*) will increase, suggesting using $\varepsilon = a/h$ (with a constant; *h* some typical node separation). Errors will than often 'saturate' when $h \to 0$ (approach constant for GA, decrease as $O(h^2)$ for MQ). In $h \to 0$ limit, the issue is reduced to the reproduction of a constant.

Example: Consider a *h*-spaced 2-D lattice, and the Wendland RBF $\phi_{3,1}(r) = (1-\varepsilon r)_+^4(4\varepsilon r + 1)$ with $\varepsilon = \dfrac{1}{3h}$.
Since the RBF is compact, it cannot reproduce a constant. Illustration shows $s(x)$ over $[0,h] \times [0,h]$ in the $h \to 0$ limit; center value 0.9912, i.e. about 1% error.



Possible remedies against saturation errors:

1. Don't increase ε at the high rate $\varepsilon = a/h$.
   Use a stable algorithm if needed; good chance for spectral convergence (if smooth RBF).

2. Choose some RBF implementation that reproduces as high order polynomials as is practical; provides algebraic convergence.

Analysis of RBFs on lattices

WARNING: **The approach frequently produces results that have little or no relevance to RBFs on irregularly spaced nodes or over finite domains.**

Example: The only error estimate that is quoted in the Scholarpedia article on RBFs is that MQ is accurate to $O(h^{d+1})$ on an infinite lattice with spacing h in d-D.

Notations and some general formulas:

Use convention: 1-D: $u(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \hat{u}(\xi)\, e^{i\xi x} d\xi$, $\hat{u}(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} u(x)\, e^{-i\xi x} dx$

In d-D, let: $r = \sqrt{x_1^2 + x_2^2 + \ldots + x_d^2}$, $\rho = \sqrt{\xi_1^2 + \xi_2^2 + \ldots + \xi_d^2}$

Hankel Transform:

$$\hat{\phi}(\rho) = \frac{1}{(2\pi)^{d/2}} \int_{-\infty}^{\infty} \ldots \int_{-\infty}^{\infty} \phi(\|\underline{x}\|)\, e^{-i\,\underline{\xi}\cdot\underline{x}} d\underline{x}$$

$$= \frac{1}{\rho^{(d-2)/2}} \int_0^{\infty} \phi(r)\, r^{d/2}\, J_{(d-2)/2}(r\,\rho)\, dr$$

If $d = 2m+1$ odd: $\hat{\phi}(\rho) = (-2)^m \sqrt{\frac{2}{\pi}} \frac{d^m}{d(\rho^2)^m} \int_0^{\infty} \phi(r) \cos(r\,\rho)\, dr$

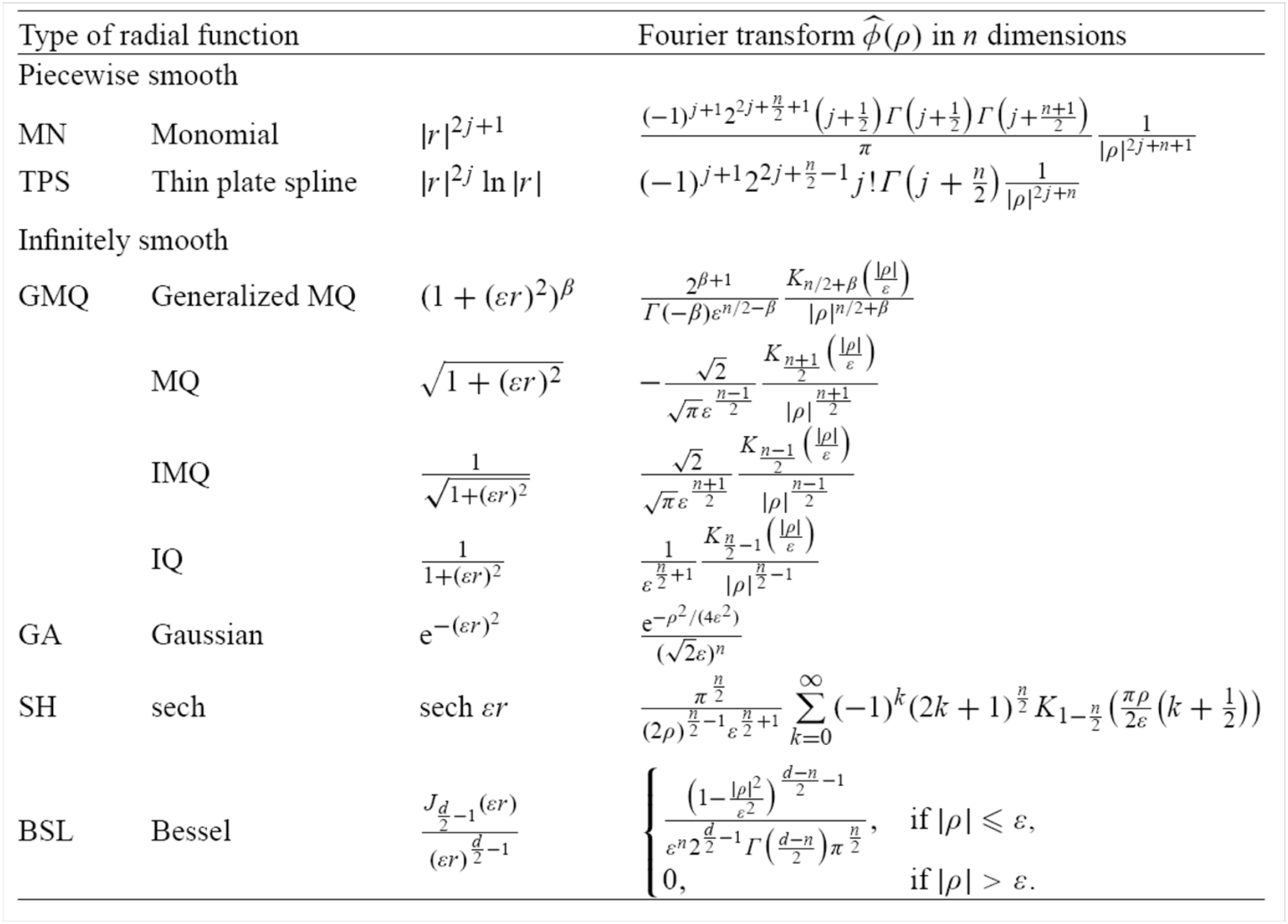
If $d = 2m+2$ even: $\hat{\phi}(\rho) = (-2)^m \frac{d^m}{d(\rho^2)^m} \int_0^{\infty} \phi(r)\, r\, J_0(r\,\rho)\, dr$

Generalized Fourier Transform:

Example: Immediate calculation of the FT of $\phi(r) = |r|^3$ fails due to divergence. However,

$$\hat{\phi}(r) = \lim_{a\to 0} \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} |r|^3 e^{-a|r|}\, e^{-i\xi r} dr = \lim_{a\to 0} \sqrt{\frac{1}{2\pi}}\, \frac{12(\xi^4 - 6\xi^2 a^2 + a^4)}{(\xi^2 + a^2)^2} = \sqrt{\frac{1}{2\pi}}\, \frac{12}{\xi^4}\,.$$

Fourier transforms of some common RBFs

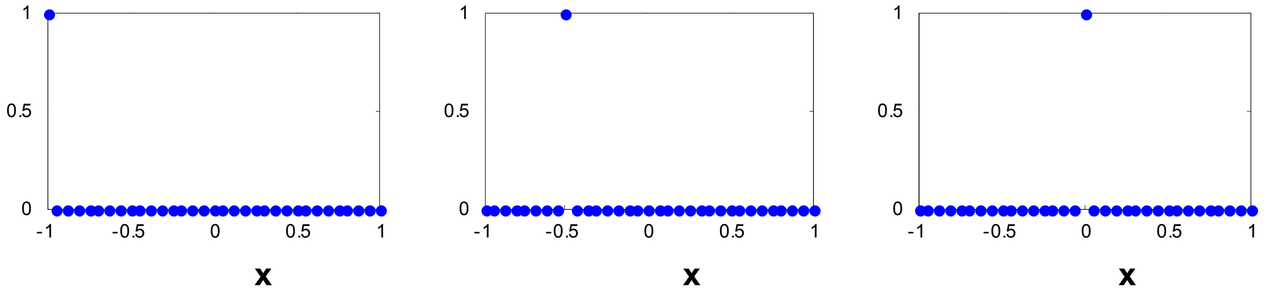


Type of radial function			Fourier transform $\widehat{\phi}(\rho)$ in n dimensions
Piecewise smooth			
MN	Monomial	$\lvert r\rvert^{2j+1}$	$\frac{(-1)^{j+1}2^{2j+\frac{n}{2}+1}\left(j+\frac{1}{2}\right)\Gamma\left(j+\frac{1}{2}\right)\Gamma\left(j+\frac{n+1}{2}\right)}{\pi}\frac{1}{\lvert\rho\rvert^{2j+n+1}}$
TPS	Thin plate spline	$\lvert r\rvert^{2j}\ln\lvert r\rvert$	$(-1)^{j+1}2^{2j+\frac{n}{2}-1}j!\,\Gamma\left(j+\frac{n}{2}\right)\frac{1}{\lvert\rho\rvert^{2j+n}}$
Infinitely smooth			
GMQ	Generalized MQ	$(1+(\varepsilon r)^2)^\beta$	$\frac{2^{\beta+1}}{\Gamma(-\beta)\varepsilon^{n/2-\beta}}\frac{K_{n/2+\beta}\left(\frac{\lvert\rho\rvert}{\varepsilon}\right)}{\lvert\rho\rvert^{n/2+\beta}}$
	MQ	$\sqrt{1+(\varepsilon r)^2}$	$-\frac{\sqrt{2}}{\sqrt{\pi}\,\varepsilon^{\frac{n-1}{2}}}\frac{K_{\frac{n+1}{2}}\left(\frac{\lvert\rho\rvert}{\varepsilon}\right)}{\lvert\rho\rvert^{\frac{n+1}{2}}}$
	IMQ	$\frac{1}{\sqrt{1+(\varepsilon r)^2}}$	$\frac{\sqrt{2}}{\sqrt{\pi}\,\varepsilon^{\frac{n+1}{2}}}\frac{K_{\frac{n-1}{2}}\left(\frac{\lvert\rho\rvert}{\varepsilon}\right)}{\lvert\rho\rvert^{\frac{n-1}{2}}}$
	IQ	$\frac{1}{1+(\varepsilon r)^2}$	$\frac{1}{\varepsilon^{\frac{n}{2}+1}}\frac{K_{\frac{n}{2}-1}\left(\frac{\lvert\rho\rvert}{\varepsilon}\right)}{\lvert\rho\rvert^{\frac{n}{2}-1}}$
GA	Gaussian	$\mathrm{e}^{-(\varepsilon r)^2}$	$\frac{\mathrm{e}^{-\rho^2/(4\varepsilon^2)}}{(\sqrt{2}\varepsilon)^n}$
SH	sech	$\operatorname{sech}\varepsilon r$	$\frac{\pi^{\frac{n}{2}}}{(2\rho)^{\frac{n}{2}-1}\varepsilon^{\frac{n}{2}+1}}\sum_{k=0}^{\infty}(-1)^k(2k+1)^{\frac{n}{2}}K_{1-\frac{n}{2}}\left(\frac{\pi\rho}{2\varepsilon}\left(k+\frac{1}{2}\right)\right)$
BSL	Bessel	$\frac{J_{\frac{d}{2}-1}(\varepsilon r)}{(\varepsilon r)^{\frac{d}{2}-1}}$	$\begin{cases}\frac{\left(1-\frac{\lvert\rho\rvert^2}{\varepsilon^2}\right)^{\frac{d-n}{2}-1}}{\varepsilon^n 2^{\frac{d}{2}-1}\Gamma\left(\frac{d-n}{2}\right)\pi^{\frac{n}{2}}}, & \text{if } \lvert\rho\rvert\leqslant\varepsilon, \\ 0, & \text{if } \lvert\rho\rvert>\varepsilon.\end{cases}$

Concept of 'locality'

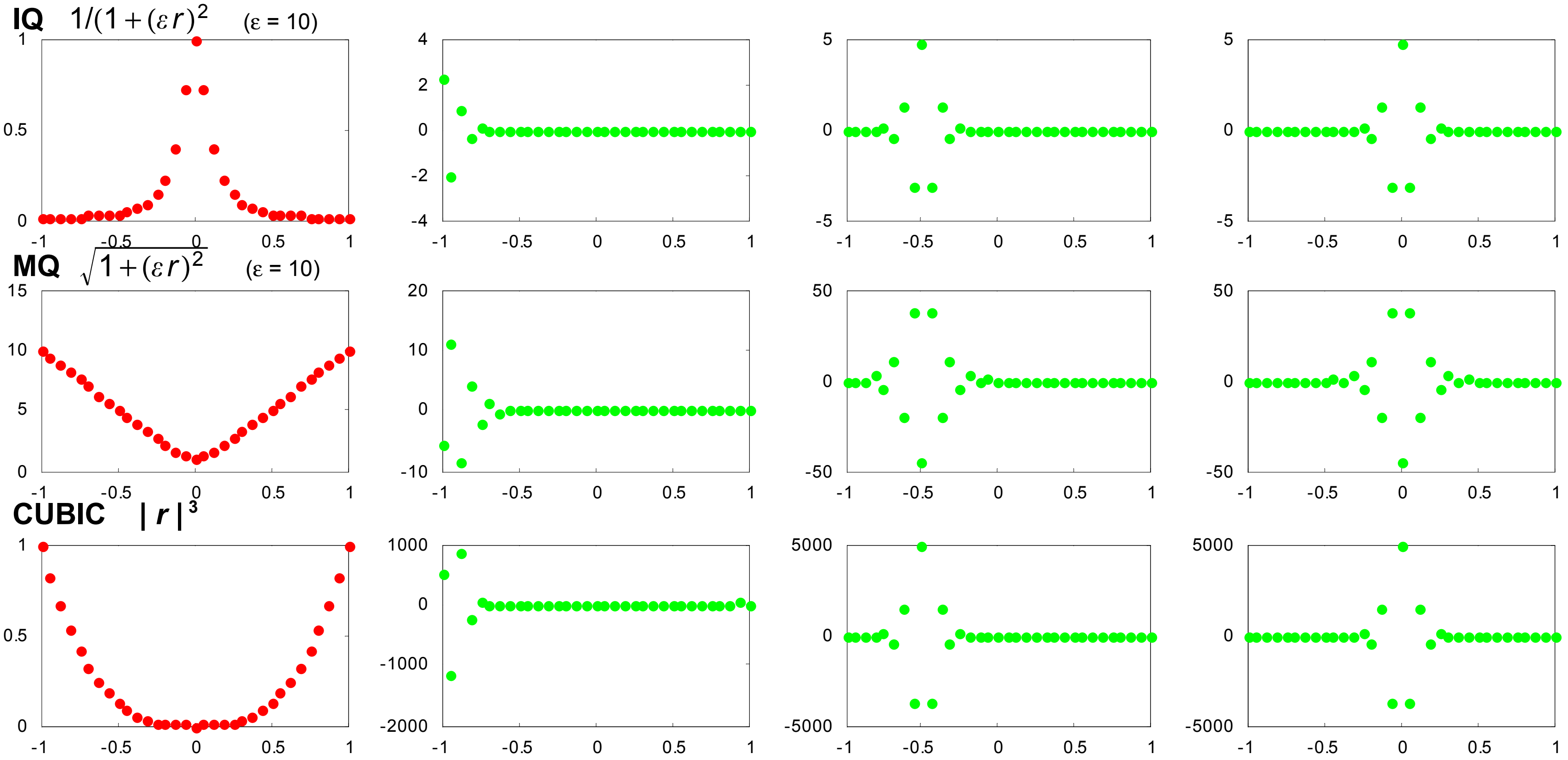
Fornberg, Flyer, Hovde, Piret (2008)

EXAMPLES OF CARDINAL DATA (over [-1,1])



EXPANSION COEFFICIENTS

RBF **Expansion coefficients for the three cardinal data cases above**



IQ $1/(1+(\varepsilon r)^2$ (ε = 10)

MQ $\sqrt{1+(\varepsilon r)^2}$ (ε = 10)

CUBIC $|\,r\,|^3$

Locality of cardinal coefficients on infinite 1-D lattice

For cardinal data, needs to hold:

$$\sum_{k=-\infty}^{\infty} \lambda_k \phi(n-k) = \begin{cases} 1 & n=0 \\ 0 & n \neq 0 \end{cases}$$

LHS discrete convolution:

Let $\begin{cases} \Lambda(\xi) = \sum_{k=-\infty}^{\infty} \lambda_k e^{ik\xi} \\ \Xi(\xi) = \sum_{k=-\infty}^{\infty} \phi(k) e^{ik\xi} \end{cases}$

Both 2π - periodic functions

Then:

$$\Lambda(\xi) \cdot \Xi(\xi) = 1$$

The fomulas above provide a way to compute λ_K:

$$\phi(r) \quad \Rightarrow \quad \Xi(\xi) \quad \Rightarrow \quad \Lambda(\xi) = 1/\Xi(\xi) \quad \Rightarrow \quad \lambda_k = \frac{1}{2\pi} \int_0^{2\pi} \Lambda(\xi)\, e^{-ik\xi}\, d\xi$$

First step may diverge; if so, use generalized Fourier transform and Poisson's summation formula:

$$\phi(r) \Rightarrow \hat{\phi}(\xi) \Rightarrow \Xi(\xi) = \sqrt{2\pi} \sum_{k=-\infty}^{\infty} \hat{\phi}(\xi + 2\pi k)$$

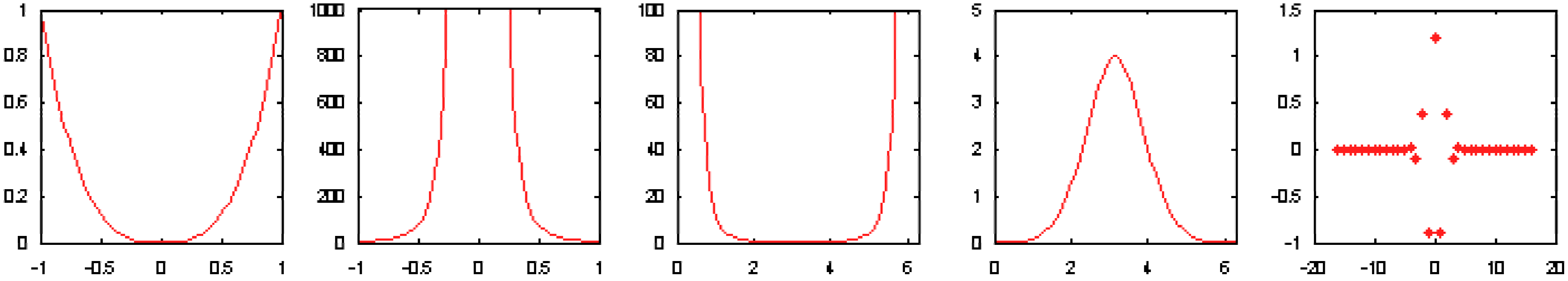
Task becomes to carry out asymptotic analysis on this integral, for *k* increasing

Standard task by contour integration

Examples of steps in finding λ_k

Cubic RBF:

$$\phi(r)=|\,r\,|^3 \quad\Rightarrow\quad \hat{\phi}(\xi)=\frac{12}{\sqrt{2\pi}}\,\frac{1}{\xi^4} \quad\Rightarrow\quad \Xi(\xi)=\frac{2+\cos\xi}{4\,\sin^4\frac{\xi}{2}} \quad\Rightarrow\quad \Lambda(\xi)=\frac{4\,\sin^4\frac{\xi}{2}}{2+\cos\xi} \quad\Rightarrow\quad \lambda_k=\begin{cases} -4+3\sqrt{3} & k=0\\ \frac{19}{2}-6\sqrt{3} & k=1\\ \dfrac{(-1)^k 3\sqrt{3}}{(2+\sqrt{3}\,)^k} & k\geq 2\end{cases}$$



Closed form expressions are available also in some more RBF cases:

GA: $\phi(r)=e^{-(\varepsilon\,r)^2} \quad\Rightarrow$

$$\lambda_k=\frac{e^{(\varepsilon k)^2}}{2}\,\frac{\sum_{j=k}^{\infty}\,(-1)^j\,e^{-\varepsilon^2(j+\frac{1}{2})^2}}{\sum_{j=0}^{\infty}\,(-1)^j\,(j+\frac{1}{2})\,e^{-\varepsilon^2(j+\frac{1}{2})^2}}\,,\qquad k\in Z.$$

SH: $\phi(r)=\text{sech}\,(\varepsilon\,r) \Rightarrow$

$$\lambda_k=\frac{1}{\sum_{j=0}^{\infty}\,(-1)^j\,\text{sech}^2(\varepsilon j)}\,(-1)^k\,\text{sech}\,(\varepsilon\,k),\qquad k\in Z.$$

Much more generally applicable approach:

Find the analytic expression for $\Lambda(\xi)$, and then estimate $\lambda_k=\dfrac{1}{2\pi}\displaystyle\int_0^{2\pi}\Lambda(\xi)\;e^{-ik\xi}\,d\xi$

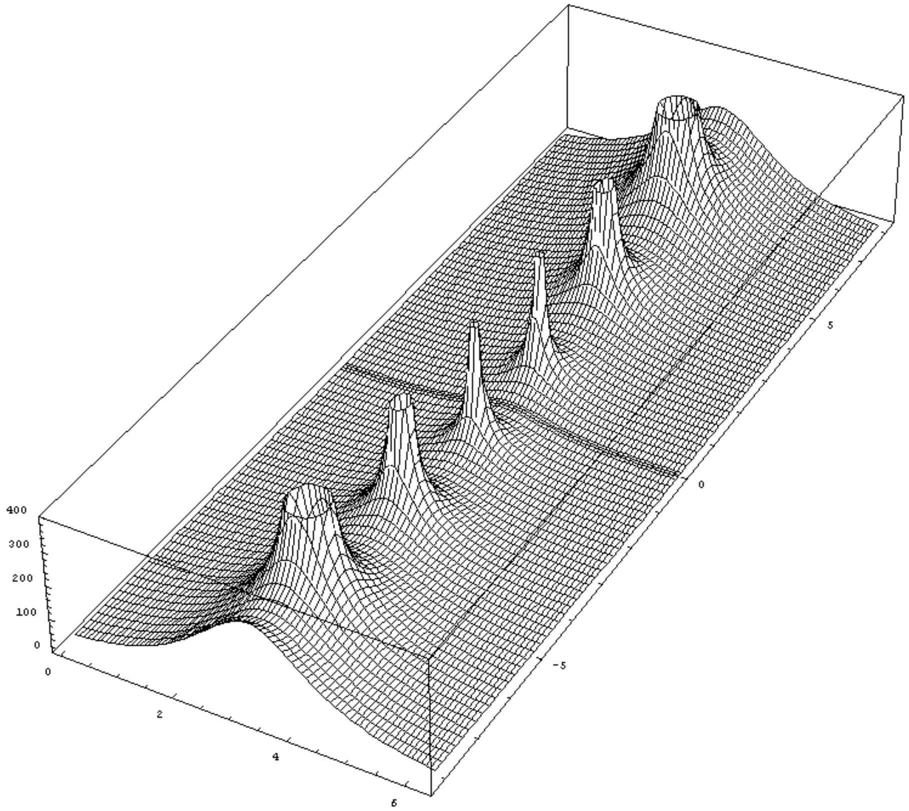
Analysis in the case of MQ ($\varepsilon = 1$)

$$\lambda_k = \frac{1}{2\pi}\int_0^{2\pi} \Lambda(\xi)\, e^{ik\xi}\, d\xi \qquad \text{where}$$

$$\Lambda(\xi) = -\frac{1}{2}\left\{\frac{1}{\sum_{j=0}^{\infty}\frac{K_1(2\pi j+\xi)}{2\pi j+\xi} - \sum_{j=1}^{\infty}\frac{K_1(2\pi j-\xi)}{2\pi j-\xi}}\right\}$$

Extend to complex plane in strip $0 \le \mathrm{Re}\,\xi \le 2\pi$

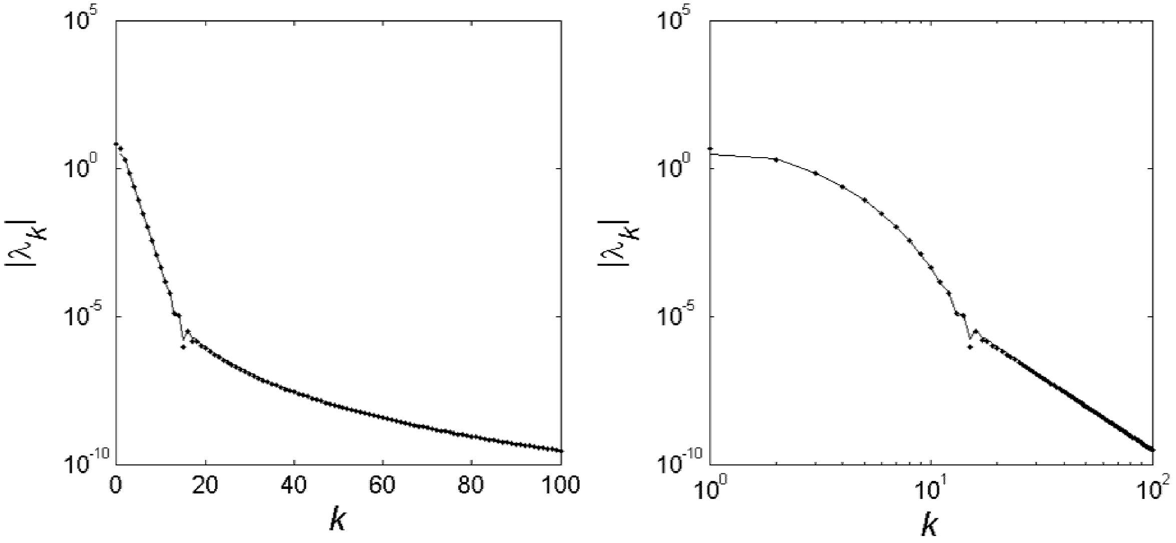
Magnitude shown ⇒



Change contour:

$$\Rightarrow \qquad \lambda_k \approx \underbrace{17.433\,(-1)^k\, e^{-1.0566\,k} + \dots}_{\substack{\text{exponential part}\\ \text{(from first pole)}}} \; \underbrace{-\frac{3}{k^5} + \dots}_{\substack{\text{algebraic part}\\ \text{(from branch points)}}}$$

Comparison between correct values for $|\lambda_k|$ (dots) and the asymptotic formula (solid curve):



Initial exponential decay:

- present for all RBFs

Ultimate algebraic decay:

- present for TPS, MQ, IQ, ...
- absent for GA, SH, cubic, ...

Close connection between λ_k and the cardinal interpolant $s(x)$

Example: **IQ:** $\phi(r) = \frac{1}{1+(\varepsilon r)^2}$

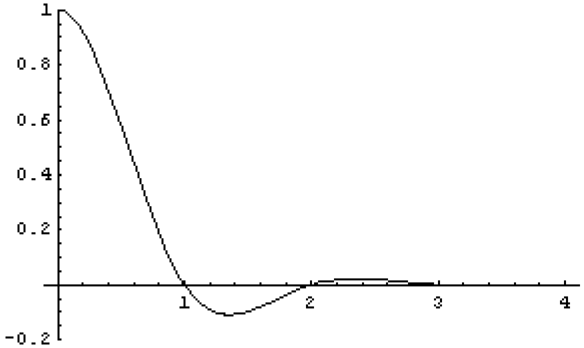
$$\begin{cases} \lambda_k = \dfrac{(-1)^k \varepsilon \sinh \frac{\pi}{\varepsilon}}{\pi^2} \displaystyle\int_0^{\pi} \dfrac{\cos k\xi}{\cosh \frac{\xi}{\varepsilon}}\, d\xi \\ s(x) = \dfrac{2 \sinh \frac{\pi}{\varepsilon} \sin \pi x}{\pi x (\cosh \frac{2\pi}{\varepsilon} - \cos 2\pi x)} \displaystyle\int_0^{\pi} \dfrac{\cos x\xi}{\left(\cosh \frac{\xi}{\varepsilon}\right)^2} d\xi \end{cases}$$

Choose $\varepsilon = 1$

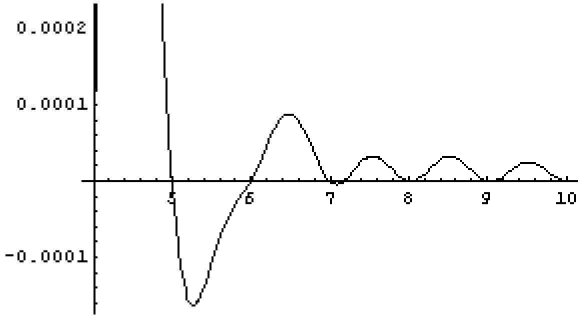
Expansion coefficients

k	λ_k
0	1.736969
1	-0.783036
2	0.138379
3	-0.043102
4	0.000939
5	-0.005300
6	-0.002424
7	-0.002075
8	-0.001535
9	-0.001230
10	-0.000996
11	-0.000825
...	...

Cardinal interpolant



[0, 4]



[4,10]

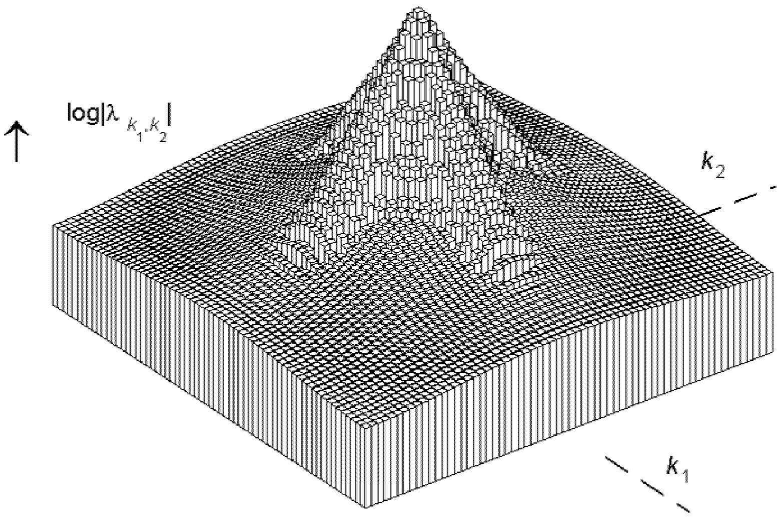
- Conceptual similarity follows from general integral formulations:

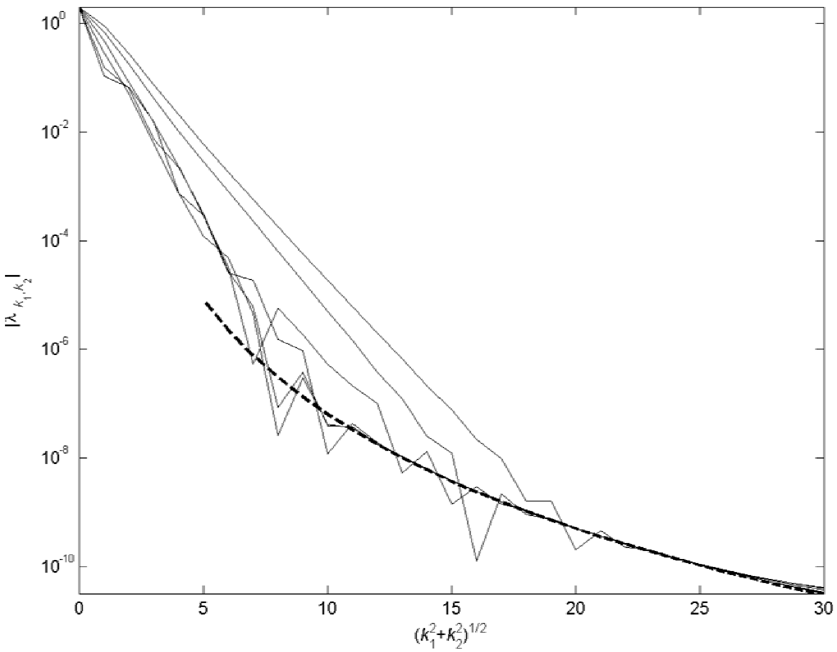
$$\lambda_k = \frac{1}{(2\pi)^{3/2}} \int_0^{2\pi} \frac{e^{ik\xi}}{\sum_{j=-\infty}^{\infty} \hat{\phi}(\xi + 2\pi j)} d\xi, \qquad s(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \frac{\hat{\phi}(\xi)\, e^{ix\xi}}{\sum_{j=-\infty}^{\infty} \hat{\phi}(\xi + 2\pi j)} d\xi\ .$$

- Both formulas above generalize immediately to lattices in n -D.
- By adding translates, get full description of Gibbs' phenomenon for RBFs.

Generalize to 2-D

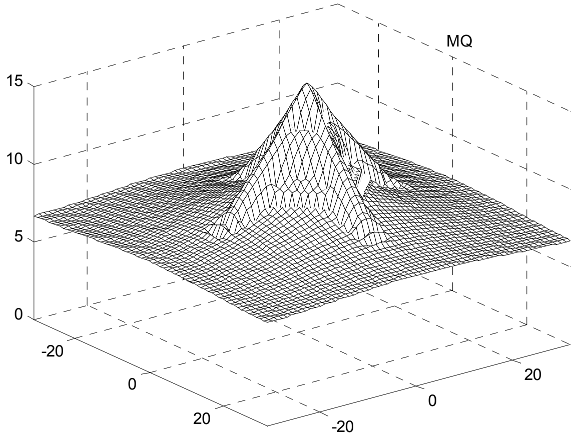
CUBIC RBF

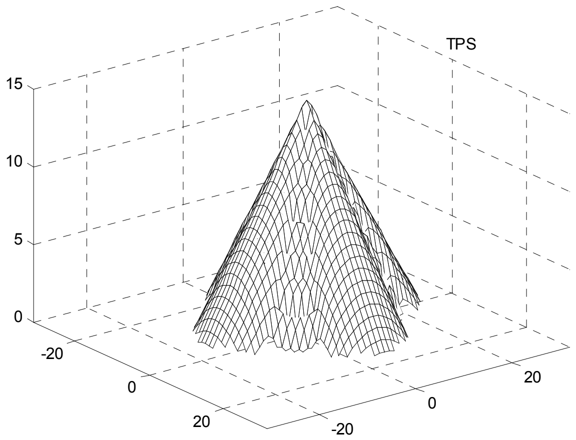


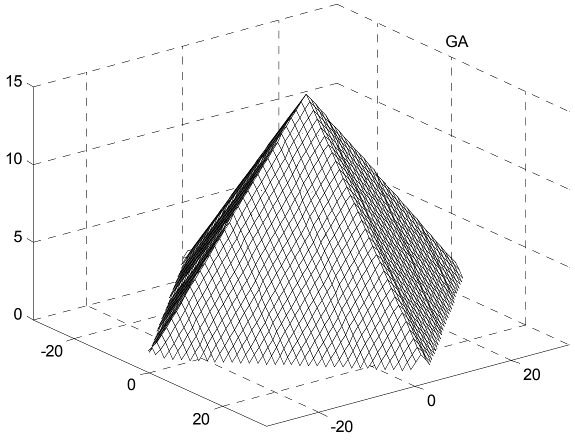


For other RBFs, get different coefficient patterns:

Dashed curve $-(\frac{5}{2\pi})^2 \frac{1}{|k|^7}$





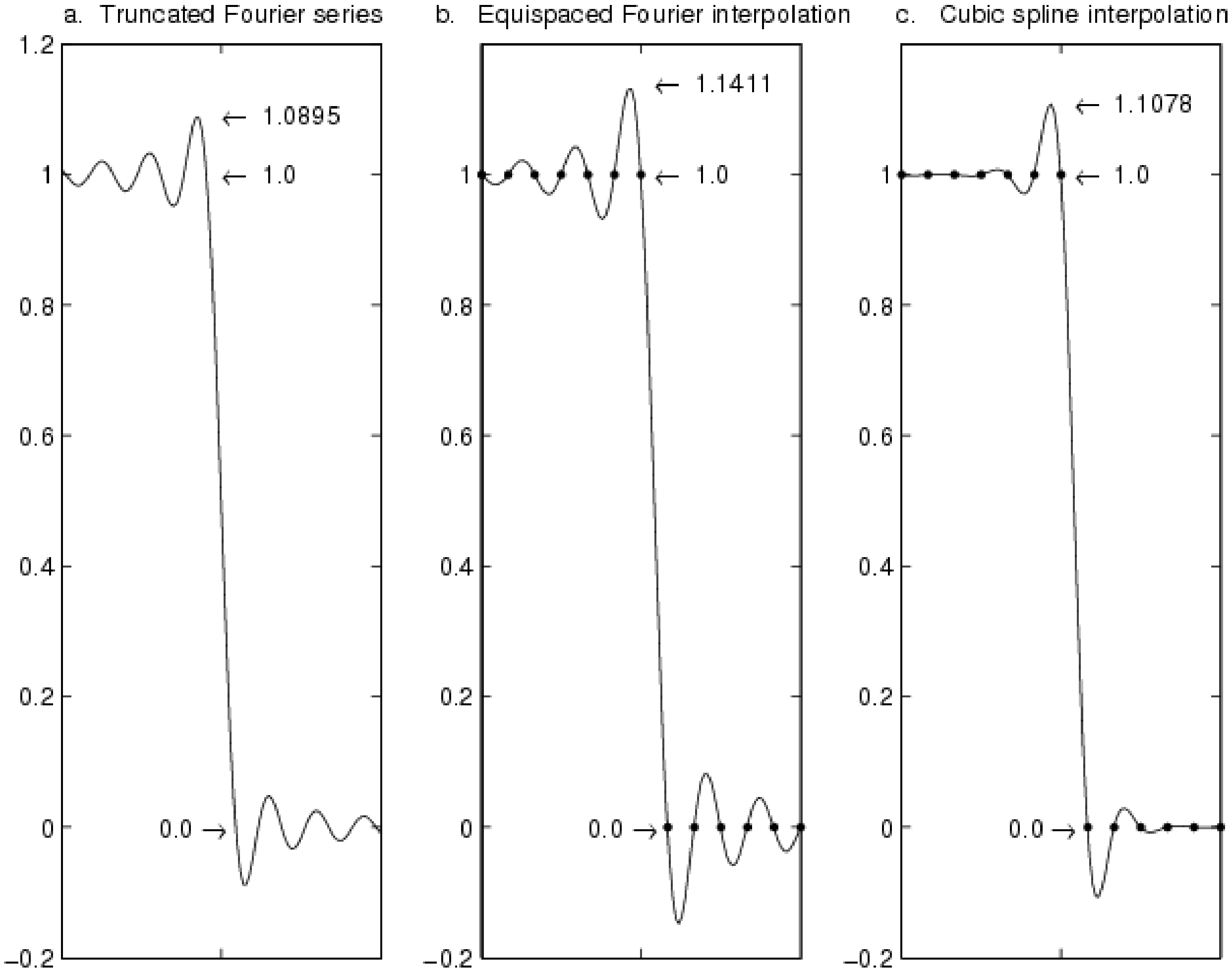


MQ **TPS** **GA**

RBF and the Gibbs Phenomenon

Fornberg and Flyer (2011)

Some examples of the Gibbs phenomenon (GP):



$$\frac{1}{2}+\frac{1}{\pi}\int_0^{\pi}\frac{\sin t}{t}dt \approx 1.0895$$

$$\max_{0<\xi<1}\left\{\frac{\sin \pi x}{\pi}\sum_{k=0}^{\infty}\frac{(-1)^k}{\xi-k}\right\}\approx 1.1411$$

$$\frac{1}{6}(8-2\sqrt{2}-\sqrt{3}-\sqrt{6})\approx 1.1078$$

Some closed-form expressions

With radial function $\phi(r)$:

$$\widehat{\phi}(\xi) = \int_{-\infty}^{\infty} \phi(x)\, e^{-i\xi x}\, dx \qquad \text{Fourier transform}$$

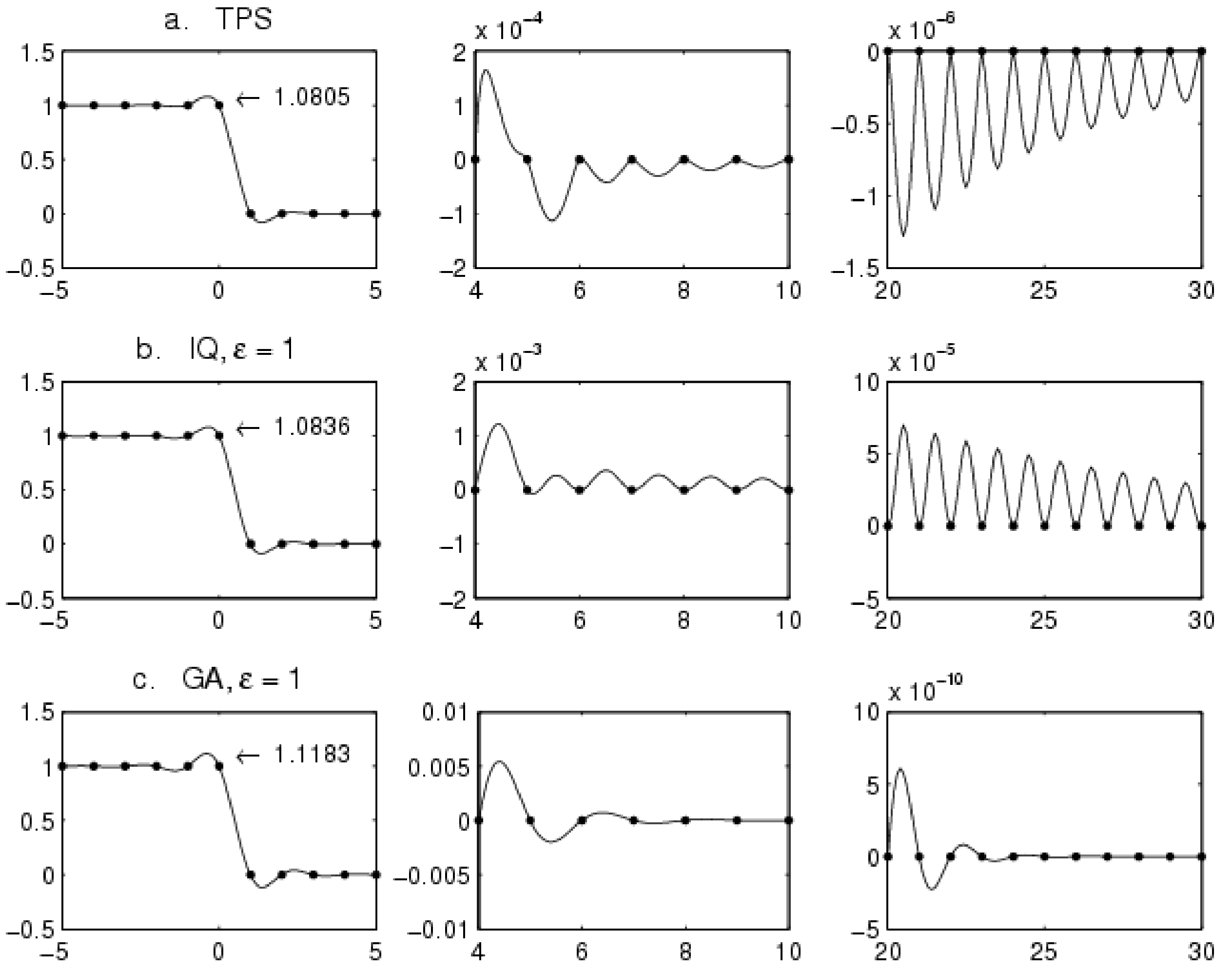
$$\Xi(\xi) = \sum_{k=-\infty}^{\infty} \phi(k)\, e^{ik\xi} = \sum_{j=-\infty}^{\infty} \widehat{\phi}(\xi + 2\pi j) \qquad \text{Poisson sum}$$

the RBF interpolant to step data ... 1 1 1 1 1 0 0 0 0 ... becomes

$$s_G(x) = \sum_{j=0}^{\infty} \frac{1}{2\pi} \int_{-\infty}^{\infty} \frac{\widehat{\phi}(\xi)}{\Xi(\xi)} \cos((x+j)\xi)\, d\xi = \frac{1}{2} - \frac{1}{4\pi} \int_{-\infty}^{\infty} \frac{\widehat{\phi}(\xi)}{\Xi(\xi)} \frac{\sin((x-\frac{1}{2})\,\xi)}{\sin\frac{\xi}{2}}\, d\xi$$

The interpolant can be evaluated numerically, or be estimated asymptotically by contour integration

Illustrations of the Gibbs phenomenon for RBF interpolants



a. TPS

b. IQ, $\varepsilon = 1$

c. GA, $\varepsilon = 1$

- Oscillations becoming one-sided with algebraic decay, or featuring exponential decay, dependent on whether $\hat{\phi}(\xi)$ has a branch point or not at $\xi = 0$.

- If there is a transition for an infinitely smooth $\phi(r)$- case, it moves out to infinity when $\varepsilon \to \infty$.

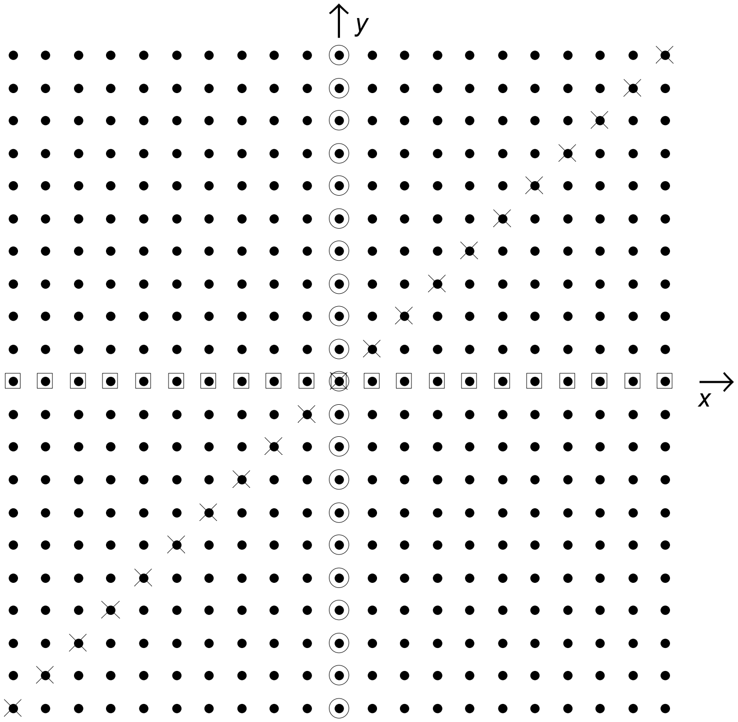
PS vs. RBF derivative approximations

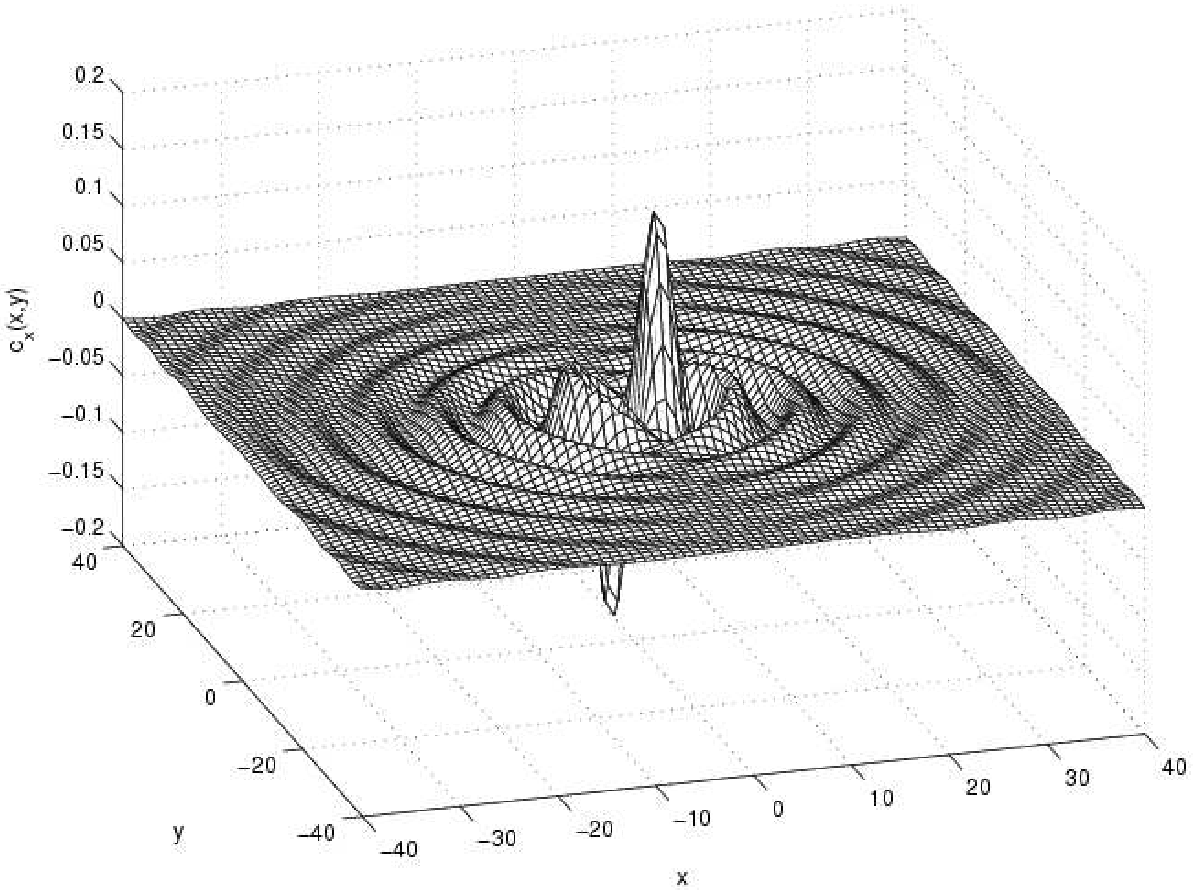
Fornberg, Flyer, Russell (2008)

There is something strange about how FD and PS methods approximate $\frac{1}{\sqrt{2}}\left(\frac{\partial}{\partial x}+\frac{\partial}{\partial y}\right)$

Where should $\frac{\partial}{\partial x}$ pick up its data from?

Answer: $\frac{x}{8}\,{}_0F_1(3,-\tfrac{1}{4}(x^2+y^2))$

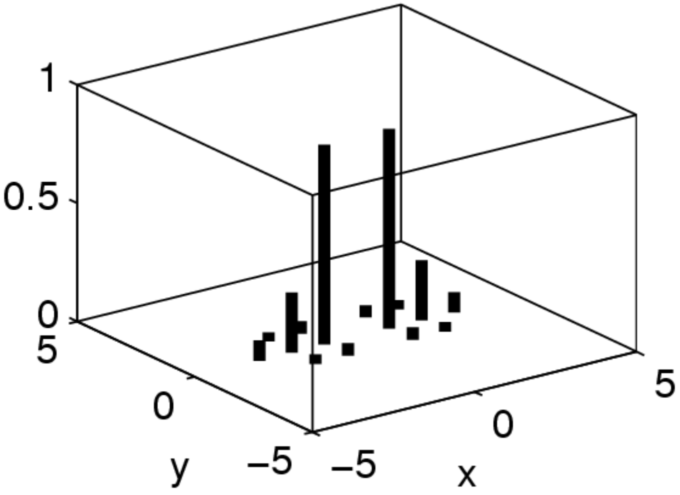




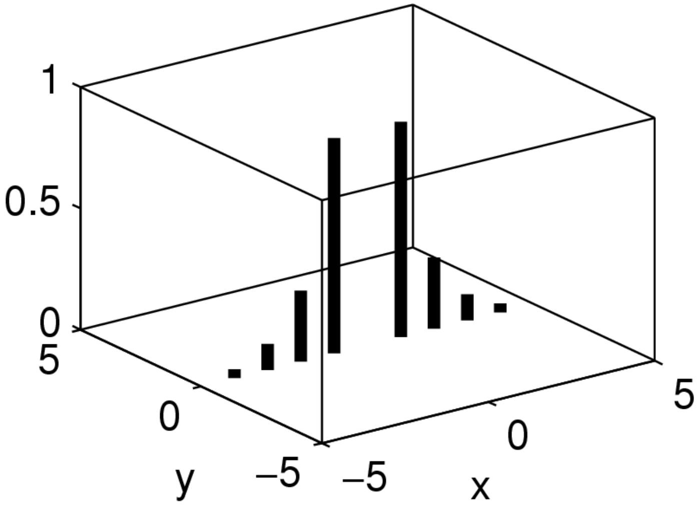
Turns out: RBFs on hexagonal or Halton (scattered) node types can be better conditioned and give higher accuracy than PS or RBFs on Cartesian lattices.

Where do RBF approximations pick up derivative information from?

Cartesian grid: MQ, ε = 1.

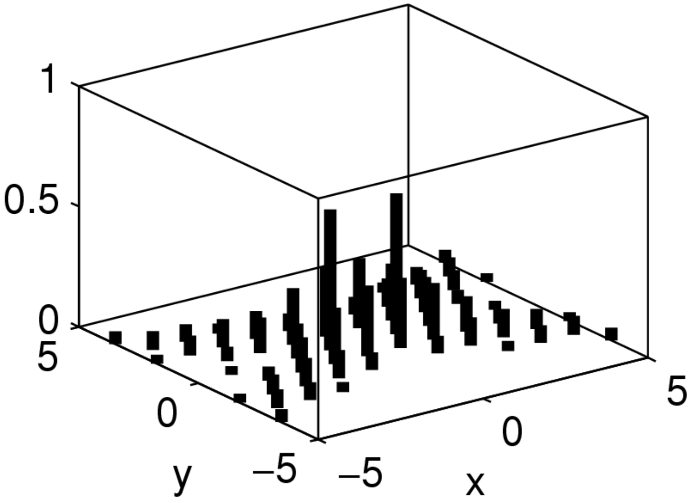


In the limit of ε→ 0

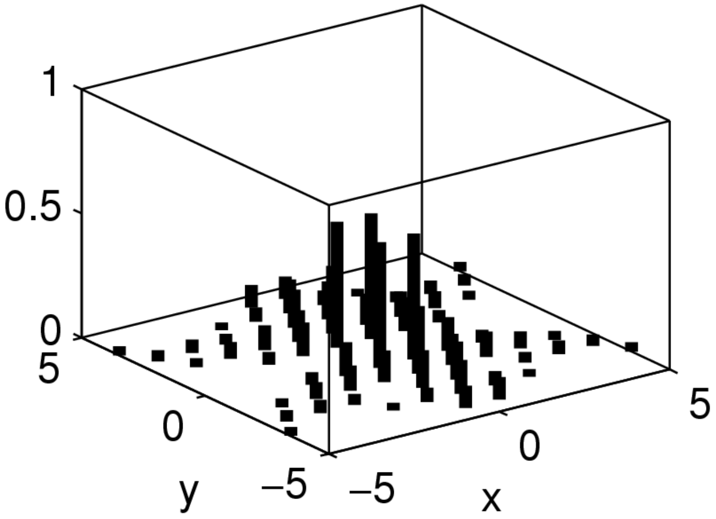


Hexagonal grid: Nontrivial stencil shapes also in the limit of ε→ 0

$\frac{\partial}{\partial x}$

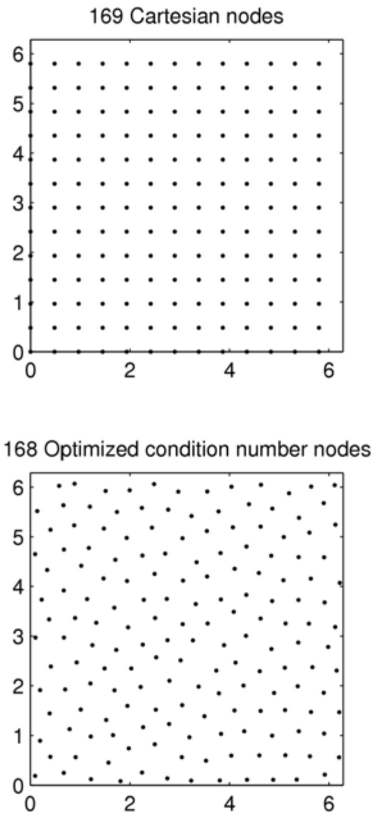


$\frac{\partial}{\partial y}$



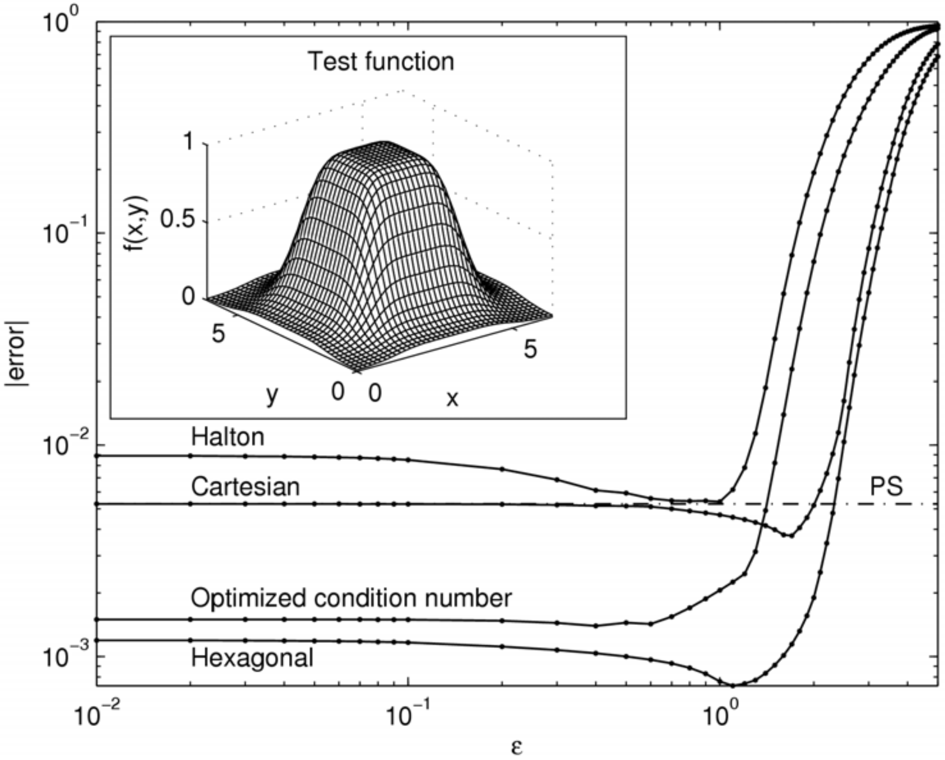
Comparison between grid types

Four different types of node layouts:



Interpolation errors for the test function

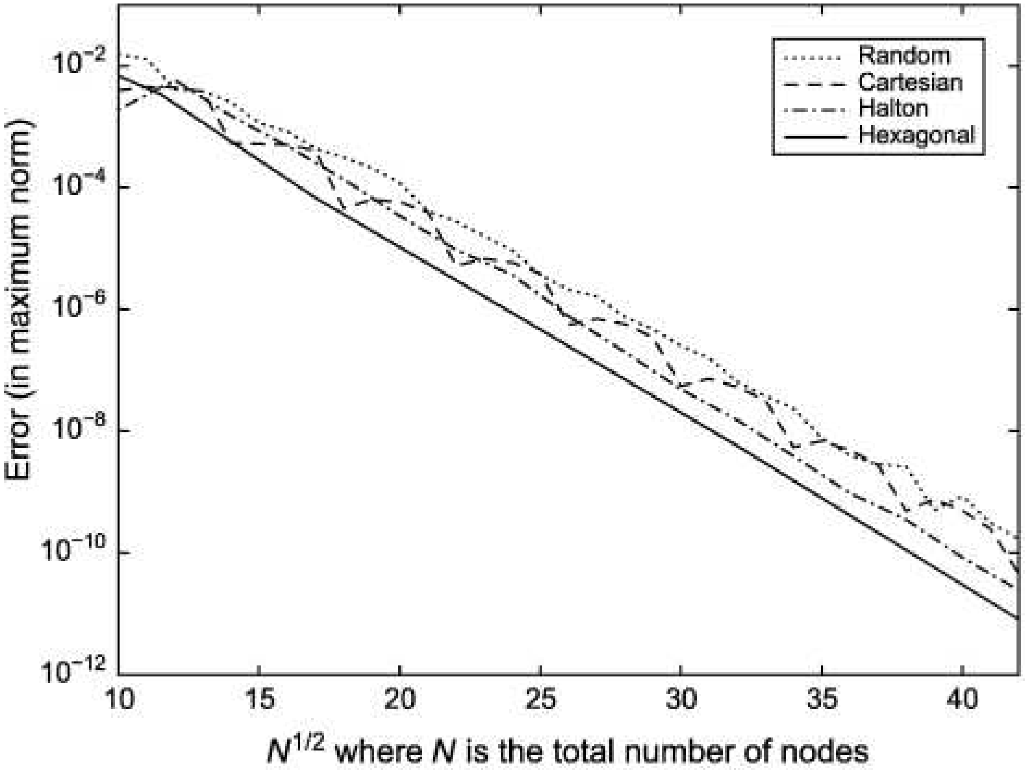
$$f(x,y) = 1/\left(1+\left(\sin\frac{\pi-x}{2}\right)^6+\left(\sin\frac{\pi-y}{2}\right)^6\right)^4$$



The test example (*N*=169 or 196) is too small for seeing any trends as node densities increase

Comparison using increasing values for *N*.

Four different types of node layouts; Gaussian RBFs:



CONCLUSION: Even in periodic settings, equispaced grids are NOT optimal in 2-D (or above)

Halton nodes: Typically not optimal, but often better than Cartesian lattices, and easy to generate in *d*-D:

```
function H = halton(numpts,ndims)
% This routine creates the identical sequences as the routine
% haltonseq that is available from matlab Central
%   numpts  (scalar) number of points to generate in Halton sequence
%   ndims   (scalar) number of dimensions, should be <=6
if ndims > 6; error('ndims > 6'); end
p = [2 3 5 7 11 13];
H = zeros(numpts,ndims);
for k = 1:ndims
    N = p(k); v1 = 0; v2 = 0:N-1; lv1 = 1;
    while lv1 <= numpts
        v2 = v2(1:max(2,min(N,ceil((numpts+1)/lv1))))/N;
        [x1,x2] = meshgrid(v2,v1);
        v1 = x1+x2; v1 = v1(:); lv1 = length(v1);
    end
    H(:,k) = v1(2:numpts+1);
end
```

Global RBFs for solving PDEs

Presented at NSF-CBMS Regional Research Conference
University of Massachusetts, Dartmouth, by

Bengt Fornberg and Natasha Flyer

Original proposal for using RBFs to solve PDEs

Two pioneering papers by Kansa (1990)

Multiquadrics - A scattered data approximation scheme with applications to computational fluid dynamics -

I. Surface approximations and partial derivative estimates
II. Solutions to parabolic, hyperbolic and elliptic partial differential equations

Demonstration examples included:

- 1-D von Neumann blast wave
- 2-D Poisson equation

Connection RBF-PS emerged around 2002-2003 (Driscoll, Fornberg, Fasshauer)

Theoretical connection via the theorem that the flat RBF-limit is of polynomial form

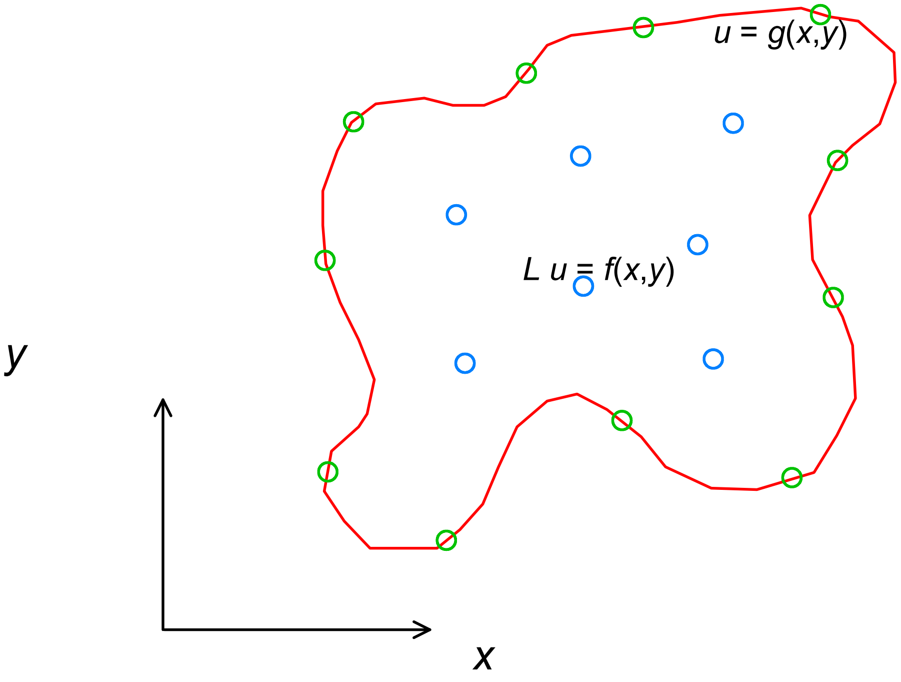
Solving Poisson's equation based on Kansa's formulation

Direct collocation (Method by Kansa):

On Boundary: $u = g(x, y)$

In interior: $L\,u \equiv \dfrac{\partial^2 u}{\partial x^2} + \dfrac{\partial^2 u}{\partial y^2} = f(x, y)$

Let $u(\underline{x}) = \sum_{j=1}^{n} \lambda_j\, \phi(\|\underline{x} - \underline{x}_j\|)$



Collocation gives:

$$\begin{bmatrix} \phi(\|\underline{x}-\underline{x}_j\|)|_{x=x_i} \\ -\ -\ -\ -\ -\ -\ -\ - \\ L\,\phi(\|\underline{x}-\underline{x}_j\|)|_{x=x_i} \end{bmatrix} \begin{bmatrix} \\ \lambda_j \\ \\ \end{bmatrix} = \begin{bmatrix} g \\ - \\ f \end{bmatrix}$$

Key features:

- Spectral accuracy (if smooth RBFs)
- Completely arbitrary geometry
- Code maybe 15 lines... (in Matlab, if using a direct solver)

Numerical tests for Poisson's equation

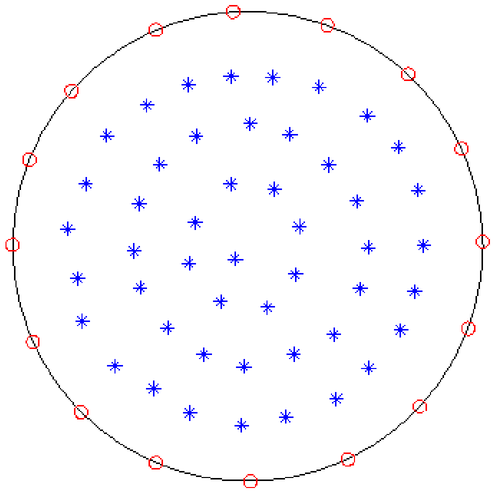
(Larsson and Fornberg, 2003)

$$\begin{cases} \dfrac{\partial^2 u}{\partial x^2} + \dfrac{\partial^2 u}{\partial y^2} = f(x,y) & \text{in interior,} \\ u = g(x,y) & \text{on boundary} \end{cases}$$

The RHS functions are chosen so that $u(x,y) = \dfrac{100}{100 + (x-0.2)^2 + 2y^2}$

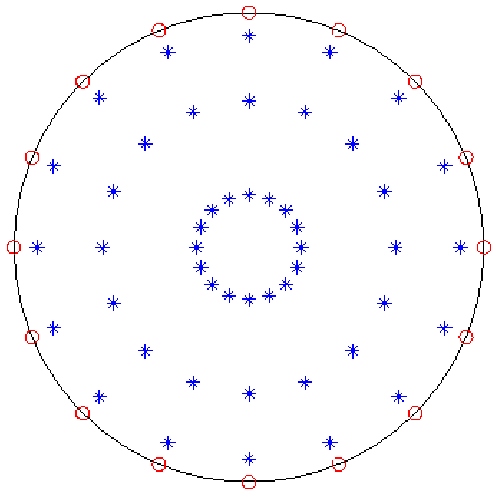
Direct RBF collocation

Test with different $\phi(r)$ and ε



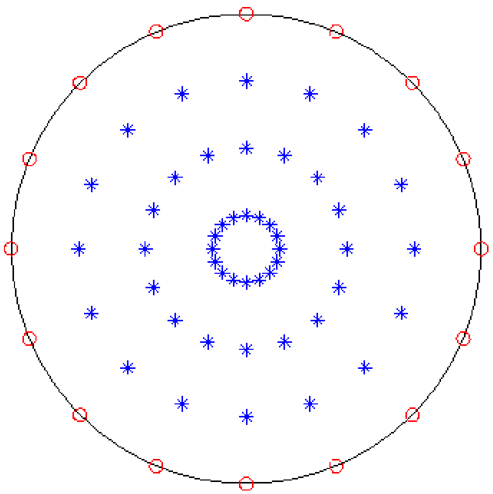
Pseudospectral (PS)

Fourier angularly
Chebyshev radially



Finite differences (FD2)

Centered FD2 in both directions



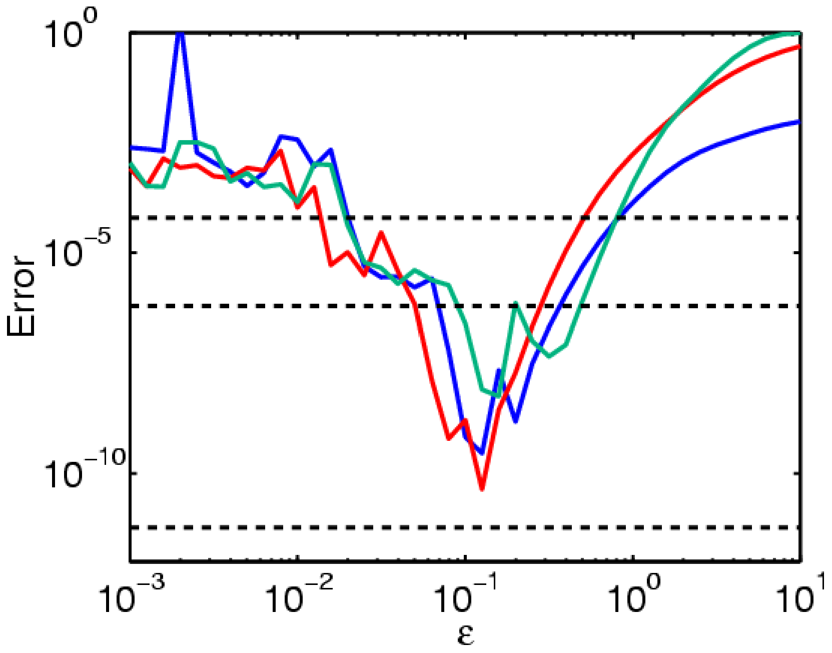
In all cases 48 nodes in the interior, 16 nodes on the boundary

Comparison between RBF, PS, and FD2 approximations

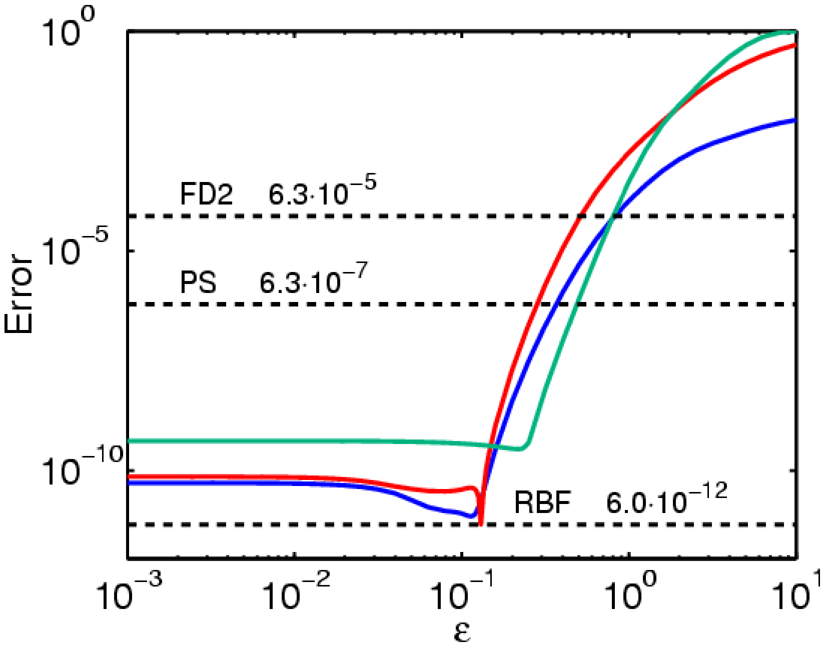
RBF approximations:	Gaussians	(GA)	$\phi(r) = e^{-(\varepsilon r)^2}$
	Inverse quadratics	(IQ)	$\phi(r) = 1/(1+(\varepsilon r)^2)$
	Multiquadrics	(MQ)	$\phi(r) = \sqrt{1+(\varepsilon r)^2}$

Errors (max norm) vs. ε:

Using direct computation



Using the Contour-Padé algorithm



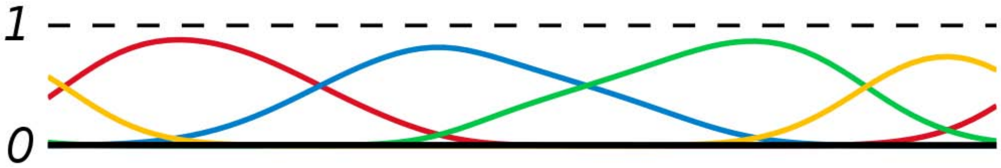
For FD2 in 2-D: Error is inversely proportional to number of points ⇒
RBF with 64 points similar accuracy to FD2 with $64 \cdot 10^{6}$ points

Several important topics that we will NOT cover in any detail:

Domain Decomposition:

- Defeat $O(N^3)$ operation count (or $O(N^2)$ per step when time stepping using pre-computed DMs)
- Implementation issues as with other methods (domain overlaps vs. edge-to-edge coupling)
- Concerns include RP at many artificially introduced interior boundaries

Partition of Unity:



- Carry out RBF interpolation separately over the different partition functions' support areas
- When 'weighing together' the separate interpolants, RP becomes very effectively suppressed.
- Approach originated in mid-1990s, but still relatively (surprisingly) little used in RBF contexts.

Fast algorithms:

Most algorithms focusing on fast evaluations of sums $s(\underline{x}) = \sum \lambda_i(\|\underline{x} - \underline{x}_i\|)$, as these arise both in evaluating RBF interpolants and in iterative algorithms for finding RBF expansion coefficients. Approaches include

- Multipole methods
- Krylov iteration-based algorithms
- Fast Gauss transforms

The implementations are often complex, and their effectiveness tend to decrease with lowered ε.
RBF-FD (and maybe Partition of Unity) approaches are likely to reduce or eliminate the need for 'Fast' algorithms.

Error estimates:

Generally, not essential for the effective practical use of RBF-type methods.

Greedy algorithms:

Based on interpolation residuals, recursively add nodes or exchange active nodes within some large node set. Often results in provable (but slow) convergence.

Other forms of dynamic node refinement:

Residual subsampling; e.g. for the 2-D Burger's eq. (Heryudono, Driscoll)

Refinement strategies can be 'borrowed' for ex. from the FEM literature.

In 2-D and ignoring boundary effects, hexagonal node sets often seem to work well. Such sets can for ex. be realized via electrostatic repulsion-type algorithms. If the character of the function suggests the need for local refinement, the electrostatic constant can be varied accordingly (Flyer and Lehto, 2010)

Automated strategies for choosing a (single) optimized shape parameter ε:

'Optimal' values often depend on whether an ill-conditioned (RBF-Direct) or a stable algorithm is used. Several heuristic formulas have been suggested based only on node densities.

- LOOCV ('Leave One Out Cross Validation') (Rippa, 1999)
  Repeatedly chaging the left-out node permits some cost savings (but still computationally expensive)

Next: **Five slides from the presentation**

A RBF Method for solving Parabolic Differential Equations on Surfaces

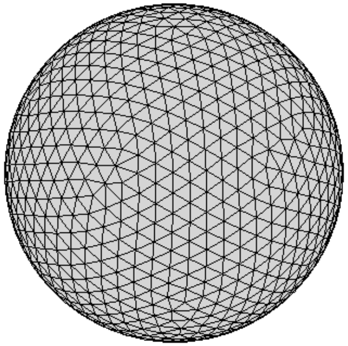
Grady B. Wright and Edward J. Fuselier

Presented at Int. Symp. Approx. Theory, Vanderbilt Univ. May 17-21, 2011

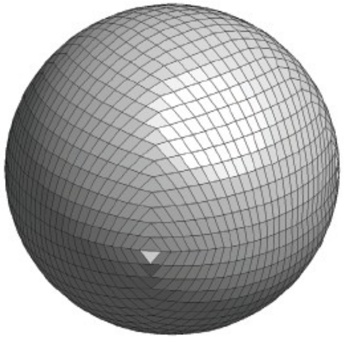
Current methods and the present approach

- Current numerical techniques can be split into 2 categories:

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

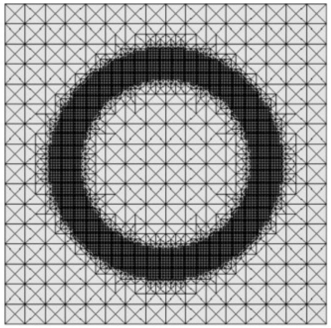


Triangulated Mesh
Dziuk (1988)
Stam (2003)
Xu (2004)
Dziuk & Elliot (2007)

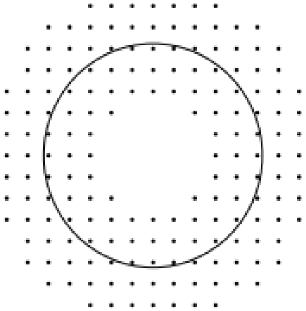


Logically rectangular grid
Calhoun and Helzel (2009)

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

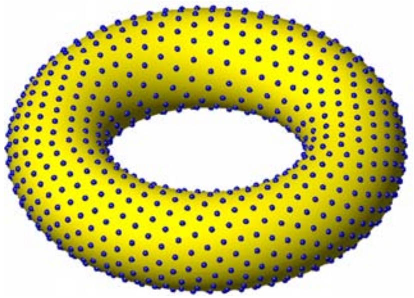


Level Set
Bertalmio *et al.* (2001)
Schwartz *et al.* (2005)
Greer (2006)
Sbalzarini *et al.* (2006)
Dziuk & Elliot (2010)



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

- Present kernel method:



- 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 manifold.
- Similarities to kernel methods for the unit sphere:
  Gia (2005); Flyer and Wright (2007, 2009); Wright *et al.* (2010); Flyer & Lehto (2010)

- Restriction: works for embedded submanifolds (defined implicitly or parametrically).

Example: 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)

$$\frac{\partial u}{\partial t} = D\delta\Delta_{\mathcal{S}} u + \underbrace{\alpha u(1-\tau_1 v^2) + v(1-\tau_2 u)}_{f_u(u,v)},$$

$$\frac{\partial v}{\partial t} = \delta\Delta_{\mathcal{S}} v + \underbrace{\beta v\left(1+\frac{\alpha\tau_1}{\beta}uv\right) + u(\gamma+\tau_2 v)}_{f_v(u,v)}.$$

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

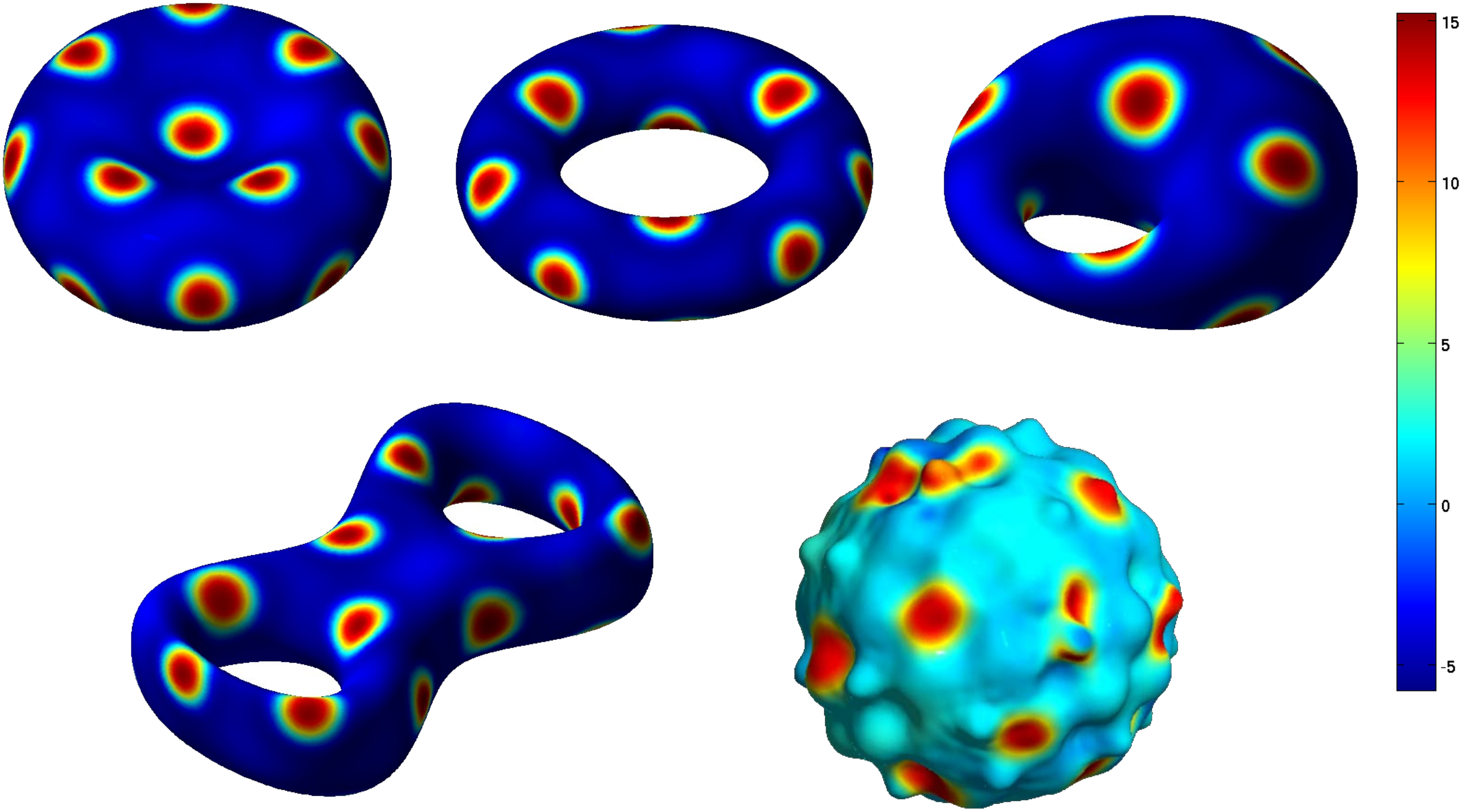
- These types of systems studied extensively in planar domains.

- More recent studies have focused on the sphere.

- Growing interest on more general surfaces.

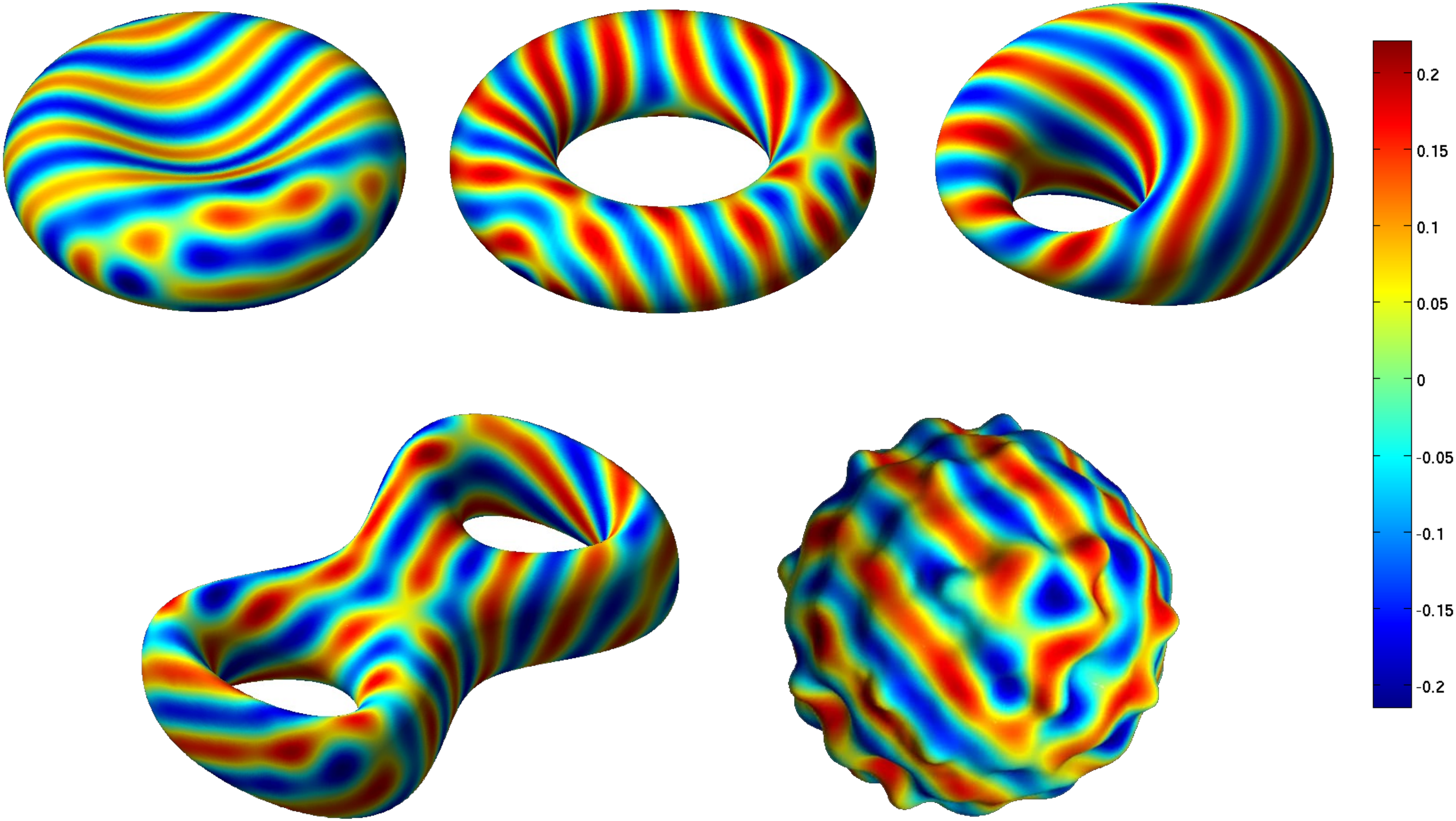
Example: Turing patterns

- Example of steady spot patterns computed with our kernel method:



Example: Turing patterns

- Example of steady stripe patterns computed with our kernel method:



Example: spiral waves in excitable media

- Example system: Barkley (1991)

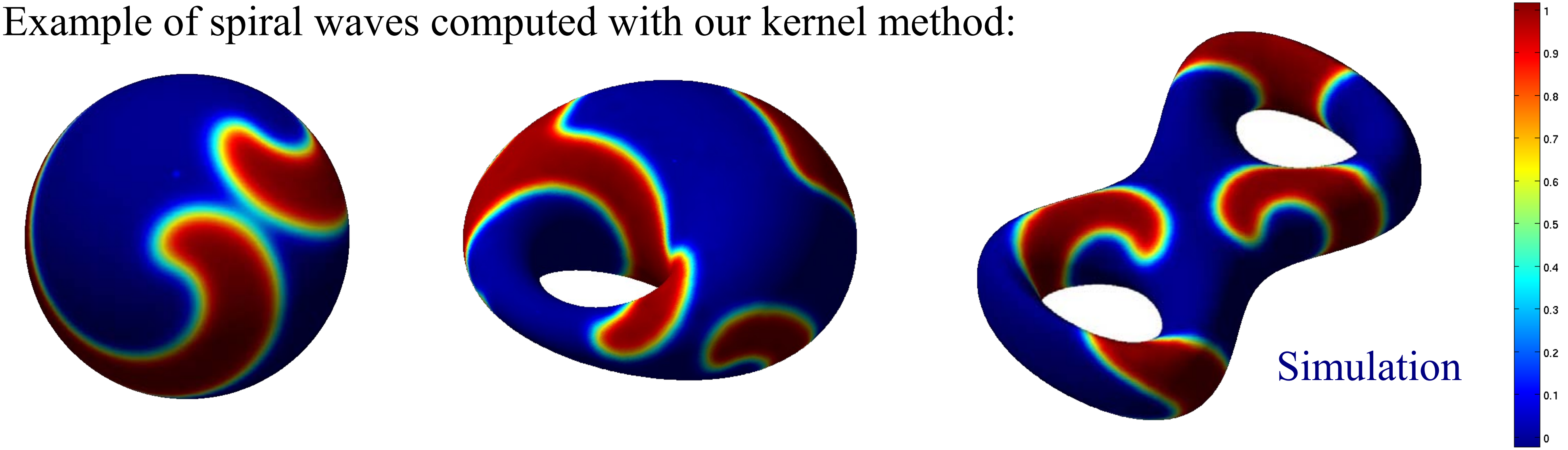
$$\frac{\partial u}{\partial t} = \delta_u \Delta_{\mathcal{S}} u + \underbrace{\frac{1}{\epsilon} u\left(1-u\right)\left(u - \frac{v+b}{a}\right)}_{f_u(u,v)},$$

$$\frac{\partial v}{\partial t} = \delta_v \Delta_{\mathcal{S}} v + \underbrace{u - v}_{f_v(u,v)},$$

u = activator species
v = inhibitor species

Simplification of FitzHugh-Nagumo model for a spiking neuron.

- Studied extensively on planar regions.
- Interest in these systems on surfaces since they are more physically relevant.
- Some studies already devoted to the case of spiral waves on the sphere.
- Example of spiral waves computed with our kernel method:



RBF-Generated FD (Finite Difference) Methods

Presented at NSF-CBMS Regional Research Conference
University of Massachusetts, Dartmouth, by

Bengt Fornberg and Natasha Flyer

RBF-FD origin and concept:

2000 Presented by Tolstykh at the 16th IMACS Conference, Lausanne.
2002 Noted in Driscoll & Fornberg, as a natural development based on the flat RBF limit theorem.
2003 Papers by Tolstykh & Shiribokov, by Shu, Ding & Yeo, and in Wright (Ph.D. thesis).

Concept: Illustrated below in the case of the Laplacian operator in 2-D

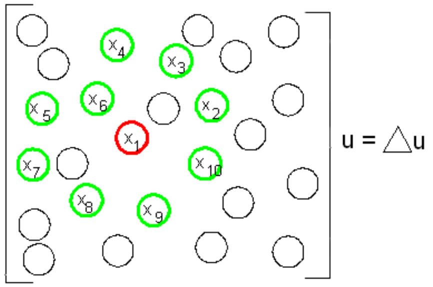
Explicit:

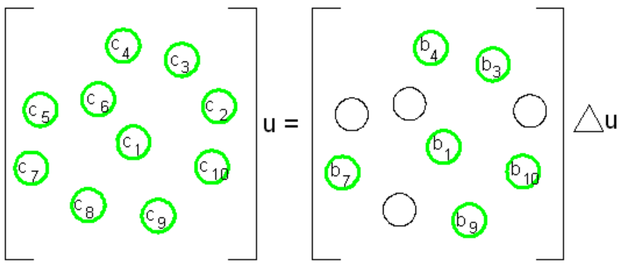
Compact (Implicit):

Regular FD: $\begin{bmatrix} & 1 & \\ 1 & -4 & 1 \\ & 1 & \end{bmatrix} \dfrac{u}{h^2} = \Delta u$ (2nd order)

$\begin{bmatrix} 1 & 4 & 1 \\ 4 & -20 & 4 \\ 1 & 4 & 1 \end{bmatrix} \dfrac{u}{6h^2} = \begin{bmatrix} & 1 & \\ 1 & 8 & 1 \\ & 1 & \end{bmatrix} \dfrac{\Delta u}{12}$ (4th order)

RBF-FD:





- On scattered node sets, all RBF-FD stencils different (i.e. we can't use one single stencil everywhere).
- One obtains weights by making the result exact for $\phi(\|x-x_k\|)$ centered at the u- nodes, as well as for $\Delta\phi(\|x-x_k\|)$ centered at Δu- nodes. (Slight modification: Use $s(x)=\sum \lambda_k \phi(\|x-x_k\|) + \alpha$ with constraint $\sum \lambda_k = 0$).
- When RBF-FD approach is applied on lattice-based node sets, $\varepsilon \to 0$ *often* reduces to regular FD case
- One can compute weights stably for all values of ε (e.g. with the Contour-Padé or RBF-QR algorithms)
- Diagonal dominance $(|c_1| \geq \sum_{k=2}^{n} |c_k|)$ may be possible also for relatively large stencils - advantageous both for stability and for iterative solution methods.

Application concepts:

Two different concepts with boundaries: Use FD at boundaries and RBF-FD inside - or other way round...

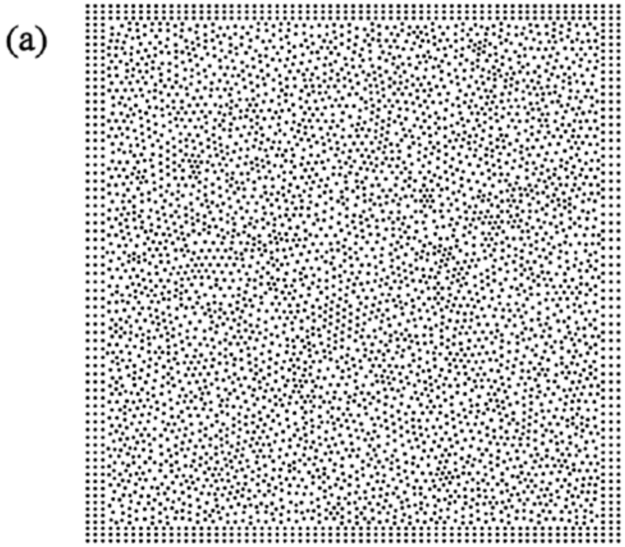


Fig. 6. A typical knot distribution for natural convection problem.

Shu, Ding & Yeo (2003)

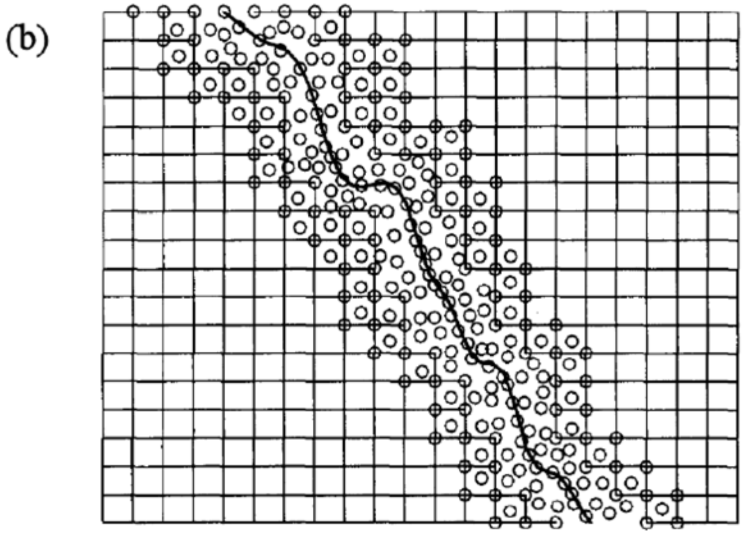


Figure 1. Example application for applying RBFs to approximate Maxwell's equations in lossless media at an irregular interface.

Fornberg, Driscoll, Wright, Charles (2002)

In domain interiors (e.g. on the surface of a sphere), one should be able to use strips with RBF-FD to 'patch together' areas that are based on FD lattices

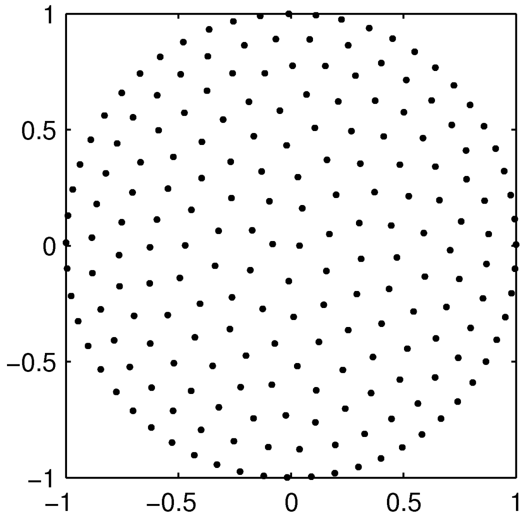
Example 1: Poisson's equation in unit circle

Wright and Fornberg (2006)

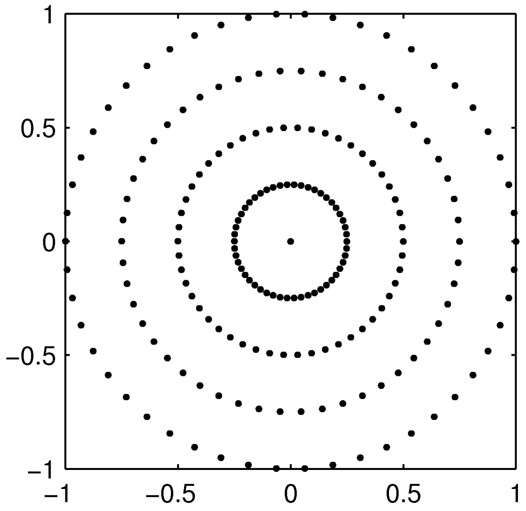
Test problem:

$\Delta u = f$ in unit circle; Dirichlet problem with solution $u(x,y) = \dfrac{25}{25 + (x-0.2)^2 + 2y^2}$.

Discretizations:

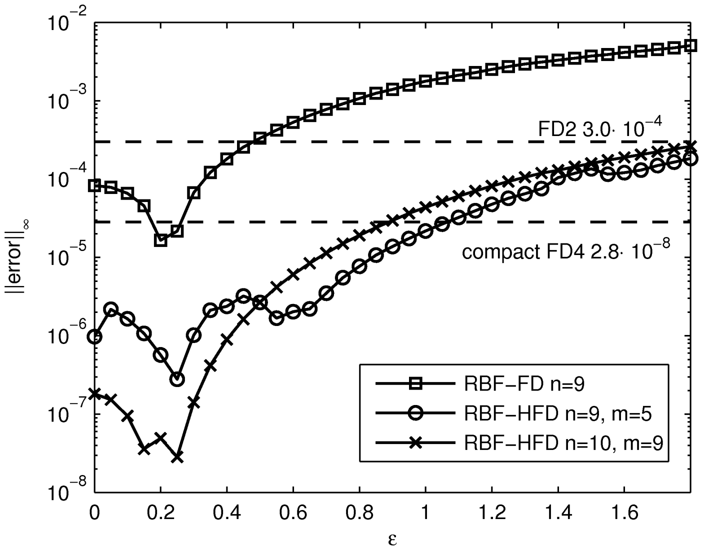


200 points unstructured



201 points structured

Max norm errors:



Some comparisons in the literature for this test problem:

Global RBF	$N = 50$	$4.6 \cdot 10^{-9}$
PS (Fourier-Chebyshev)	$N = 50$	$3.2 \cdot 10^{-6}$
RBF-FD (implicit, n = 10)	$N = 200$	$5 \cdot 10^{-8}$

Example 2

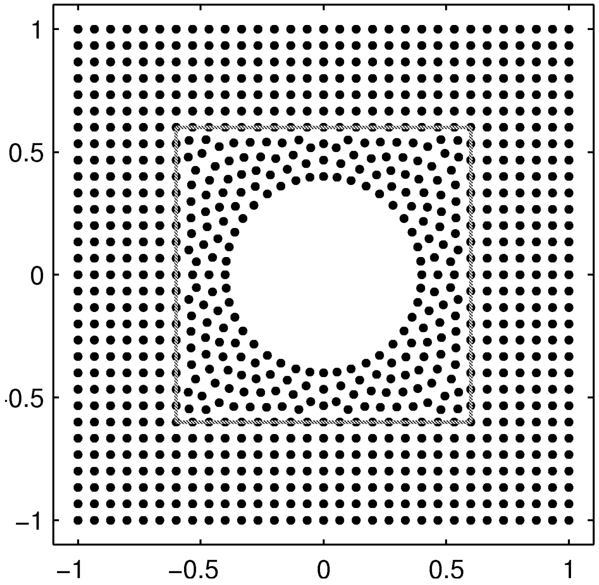
PDE: $\Delta u = e^{-2x} u^3$ in square outside central circle; Dirichlet problem with solution $u(x,y) = e^x \tanh \frac{y}{\sqrt{2}}$.

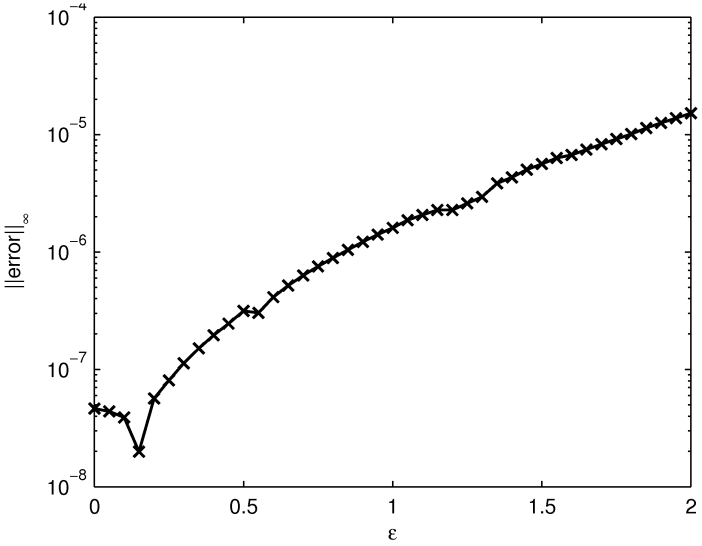
Hybrid discretization:

Max norm errors:

252 scattered nodes,
use (10/10 stencils)

482 regular nodes
use (9/5 stencils)



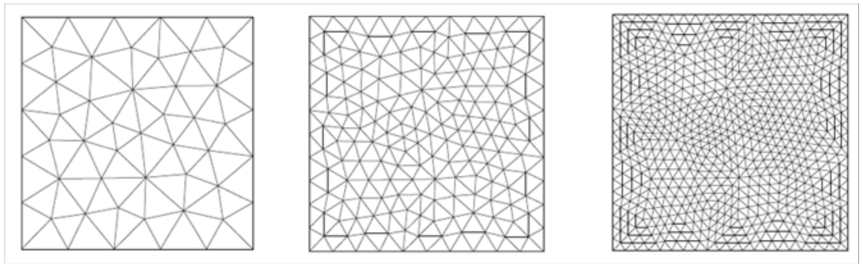


Implementation concept here: Use scattered nodes only where the geometry is complex.

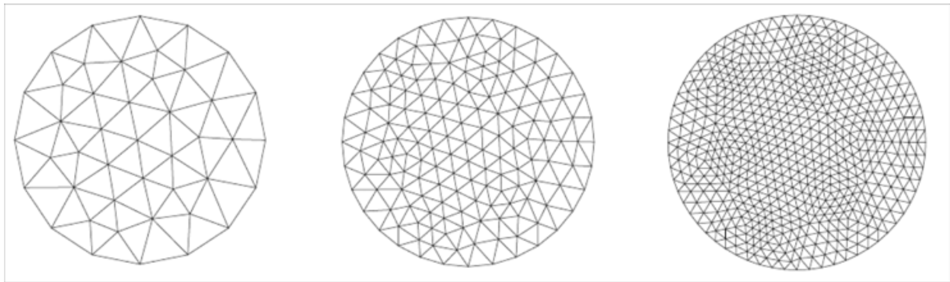
Optimal ε for fixed stencil size, increasing *N*

Davydov and Onah (2011)

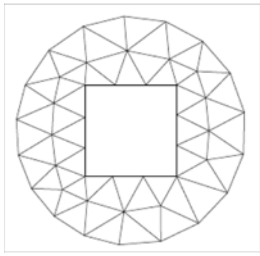
RBF-FD for Poisson's equation on various domains and different RHSs



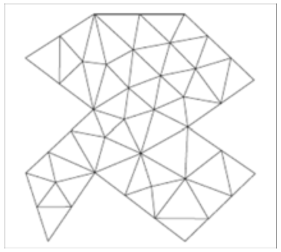
....



......



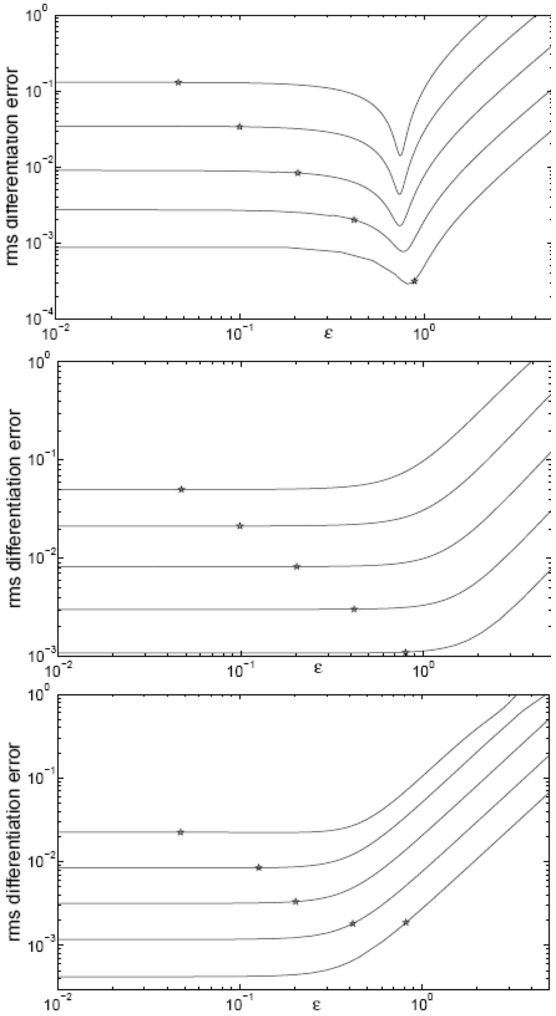
......



.....

Typical results of errors vs. ε when increasing *N*

⇒



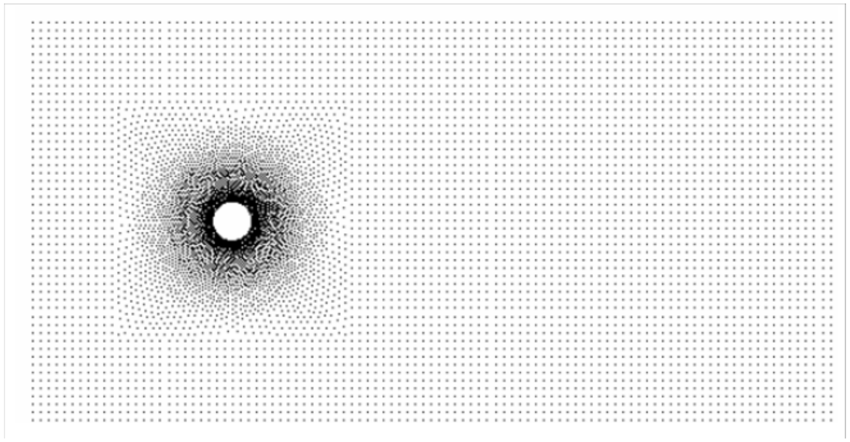
Dots mark lower reach of RBF-Direct; below these RBF-QR was used.

Some observations (based on small small stencil sizes only):

- Structure of the error curves may or may not have a distinctive dip
- The curve forms depend on the PDE's RHS, but hardly at all on domain type
- Increasing *N* essentially only shifts the curved downwards
- Use of a stable algorithm is essential for reaching good accuracies (possibly more so than in the case of global RBFs)

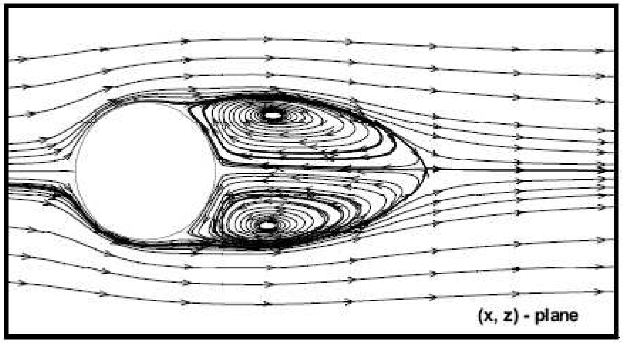
Illustrations from two RBF-FD implementations in fluid dynamics

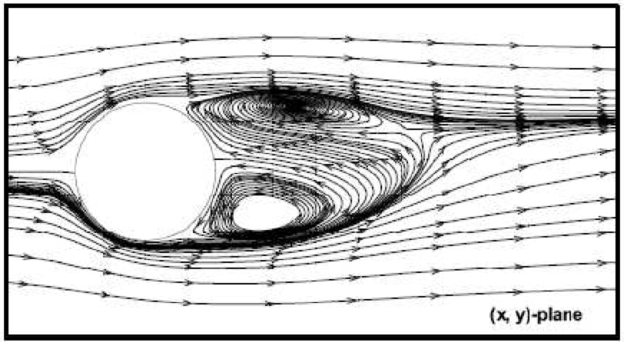
Shan, Shu, Lu (2008): Full 3-D simulation of **steady flows past a sphere**. Time dependent velocity component formulation integrated until steady state state has been reached. Steady solutions without axial symmetry are possible for *Re* $\gtrsim$ 211.



At left: Grid point distribution in the *x*,*y*-plane at *z* = 0.

At right: Example of a computed solution at *Re* = 250.





Chinchapatnam, Djidjeli, Nair, Tan (2009): **2-D driven cavity**. Tests "Randomly" vs. "Uniformly" distributed nodes.

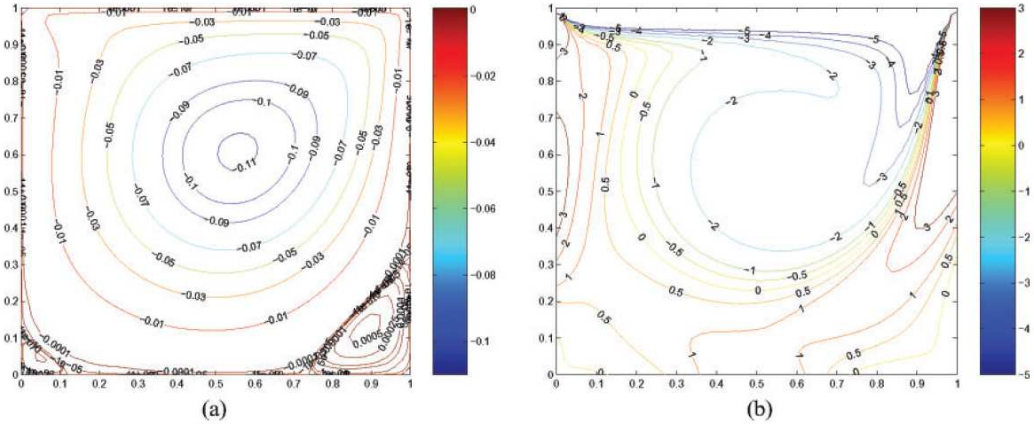


Fig. 6 Square driven cavity: *Re* = 400, contours of stream function and vorticity obtained using 51 × 51 random point distribution. (a) Stream function, (b) vorticity

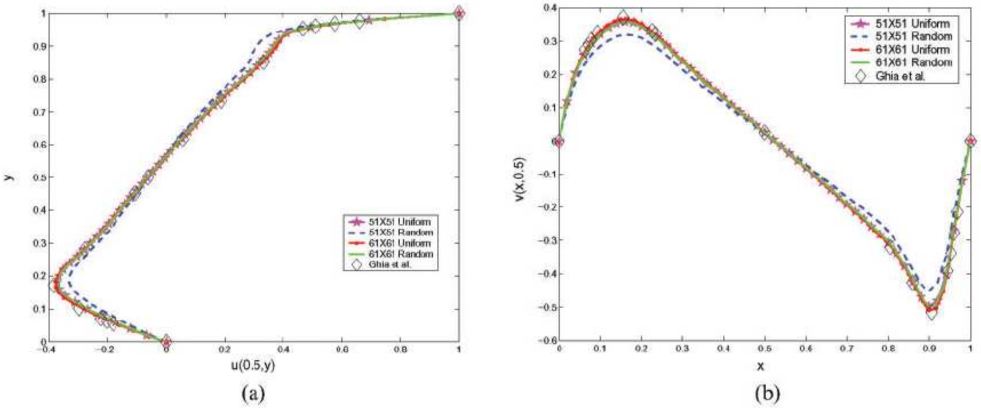


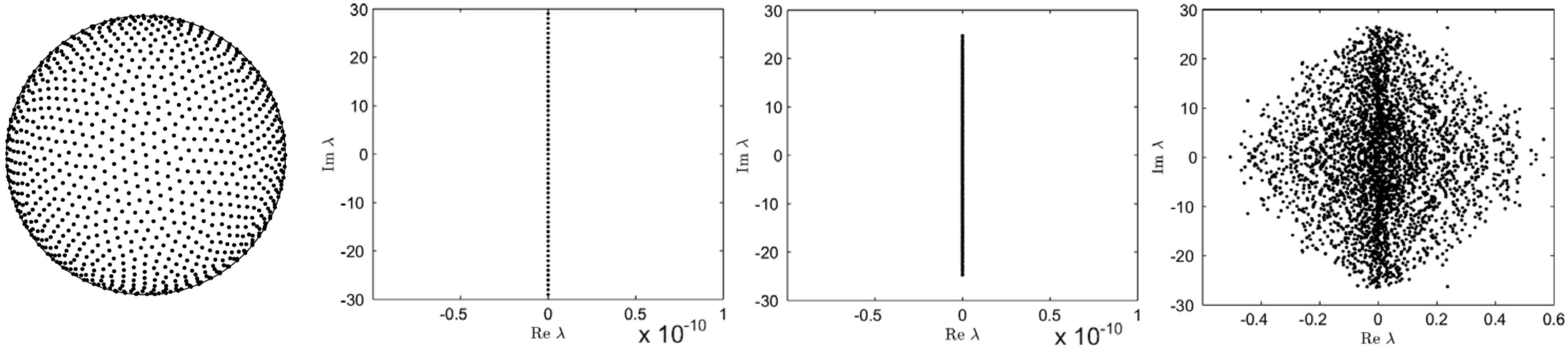
Fig. 9 Square driven cavity: comparison of velocity profiles obtained on the vertical and horizontal centre-lines using RBF method with reference [**21**] for *Re* = 1000. (a) Vertical centre-line, (b) horizontal centre-line

RBF-FD approximations for purely convective PDEs

Fornberg and Lehto (2011)

Complication: For PDEs with no natural damping, the DM *e*-values should be right on the imaginary axis. With RBF-FD approximations, eigenvalues corresponding to spurious (highly oscillatory) modes will be scattered off the axis, including into the RHP (right half plane), causing growth during time integration.

Illustration: Pure convection in an arbitrary direction around a sphere



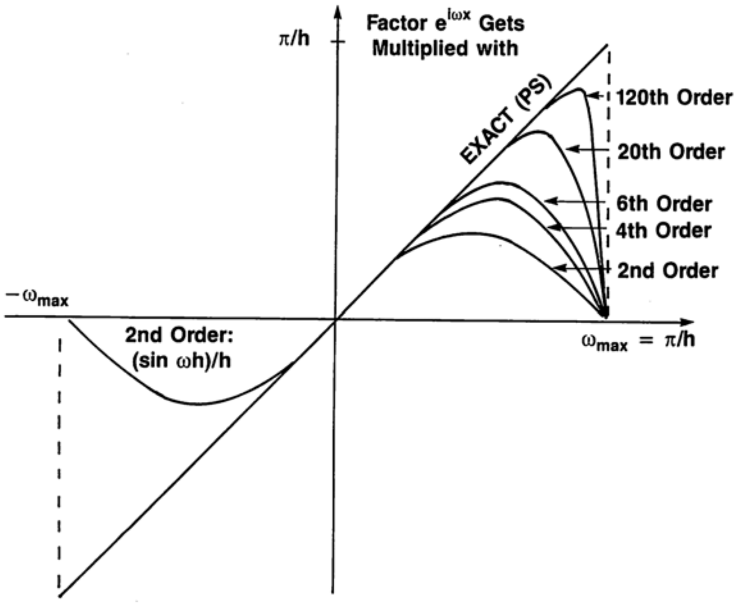
N = 1849 MD nodes   Global RBF: *N* = 900, ε = 0,   *N* = 900, ε = 5,   RBF-FD: *N* = 3600, *n* = 17, ε = 2.5

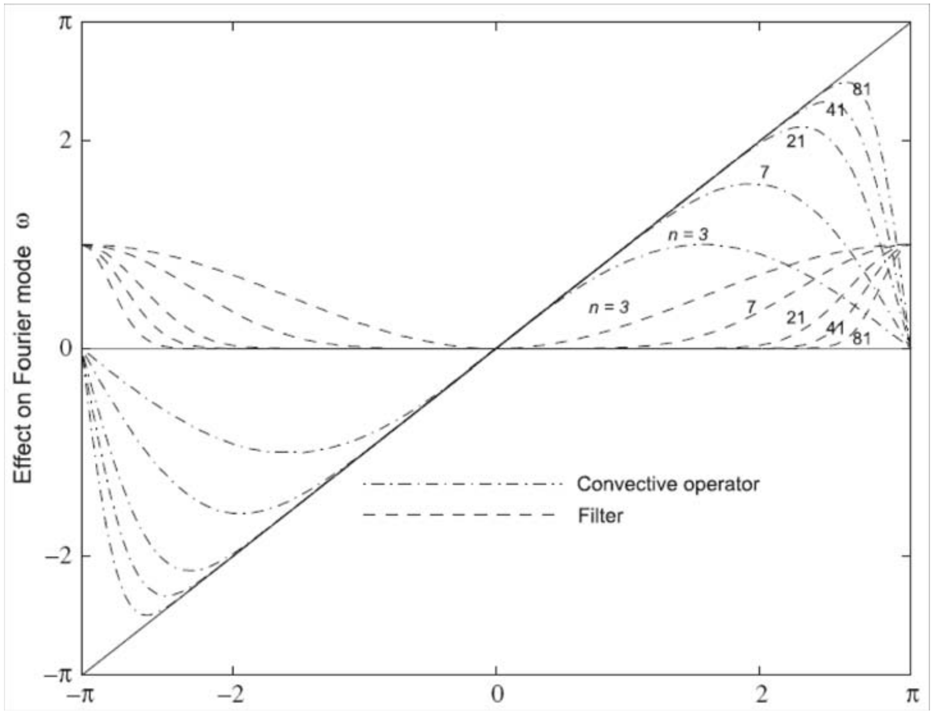
'Hidden' among the RBF-FD *e*-values are those with smooth *e*-functions, accurately represented and virtually right on the imaginary axis.

Task: Create 'filter' strategies, which leave accurate modes unchanged, but shifts spurious modes over to the LHP (left half plane).

Design of a local RBF-FD hyperviscosity filter

As background, recall the 'effect' in Fourier space of different order accurate FD approximations for d/dx:





Factors when approximating $\frac{d}{dx} e^{i\omega x} = i\omega e^{i\omega x}$ by centered FD formulas of increasing orders

Convective operator: Same curves as to the left; expressed in terms of stencil width *n*.

Filter: Damping achieved by approximating $\frac{d^{n-1}}{dx^{n-1}}$ to *lowest* (second) order

Using the same stencil size for the convective operator and the filter allows the modes that are convected accurately to be left essentially intact, while all the higher modes can be effectively damped out.

The filter is described as a hyperviscosity filter, due to its similarity with the same concept in numerical turbulence simulations.

Implementation of the RBF hyperviscosity filter in *d*-D

- Recall the relations used for calculating RBF-FD weights for an operator *L*:

$$\begin{bmatrix} & & \\ & A & \\ & & \end{bmatrix}\begin{bmatrix} w_1 \\ \vdots \\ \vdots \\ \vdots \\ w_n \end{bmatrix} = \begin{bmatrix} L\phi(\|\underline{x}-\underline{x}_1\|)|_{\underline{x}=\underline{x}_c} \\ \vdots \\ \vdots \\ \vdots \\ L\phi(\|\underline{x}-\underline{x}_n\|)|_{\underline{x}=\underline{x}_c} \end{bmatrix} \quad \text{or} \quad \left[\begin{array}{ccc|c} & & & 1 \\ & A & & \vdots \\ & & & 1 \\ \hline 1 & \dots & 1 & 0 \end{array}\right]\begin{bmatrix} w_1 \\ \vdots \\ w_n \\ - \\ w_{n+1} \end{bmatrix} = \begin{bmatrix} L\phi(\|\underline{x}-\underline{x}_1\|)|_{\underline{x}=\underline{x}_c} \\ \vdots \\ L\phi(\|\underline{x}-\underline{x}_n\|)|_{\underline{x}=\underline{x}_c} \\ - \\ L\,1|_{\underline{x}=\underline{x}_c} \end{bmatrix}$$

- When going from 1-D to *d*-D, replace high even order derivatives $\frac{d^{2k}}{dx^{2k}}$ by powers Δ^k of the Laplacian operator $\Delta = \frac{\partial^2}{\partial x_1^2} + \frac{\partial^2}{\partial x_2^2} + \ldots + \frac{\partial^2}{\partial x_d^2}$.

- With $L = \Delta^k$ and $\phi(r) = e^{-(\varepsilon r)^2}$, the required weights for the RBF-FD filter follow from the relation

$$L\phi(r) = \Delta^k \phi(r) = \varepsilon^{2k} p_{d,k}(r)\phi(r)$$

  where $p_{d,k}(r)$denotes (explicitly known, and thus easily calculated) generalized Laguerre polynomials.

- For the surface of a sphere (and likewise for other surfaces embedded in 3-D), each RBF-FD stencil is nearly flat, and thus one can use *d* = 2.

- For each RBF-FD stencil for the convective operator, we create a matching hyperviscosity RBF-FD stencil, and then apply the two together.

This type of hyperviscosity filter is used in all the following convective flow RBF-FD calculations.

Use of hyperviscosity allows the use of much larger/more accurate RBF-FD stencils than otherwise

In case of Global RBFs: The A^{-1}- filter is both simple and highly effective

Focus for now on GA RBFs; spherical geometry:

The DM: Smooth *e*-functions should be left intact; highly oscillatory (or generally jagged ones) should be damped.

The *A*-matrix: Positive definite, and has *e*-values: 1 $O(\varepsilon^0)$, 3 $O(\varepsilon^2)$, 5 $O(\varepsilon^4)$, 7 $O(\varepsilon^6)$, 9 $O(\varepsilon^8)$, ...
e-functions: very smooth ---> increasingly oscillatory

The A^{-1} matrix: This matrix is also positive definite, and it has exactly the same *e*-vectors (*e*-functions) as *A*.
The A^{-1}-matrix has *e*-values: 1 $O(\varepsilon^0)$, 3 $O(\varepsilon^{-2})$, 5 $O(\varepsilon^{-4})$, 7 $O(\varepsilon^{-6})$, 9 $O(\varepsilon^{-8})$, ...

This suggests: Instead of time stepping with the global RBF DM based on the convective PDE operator:

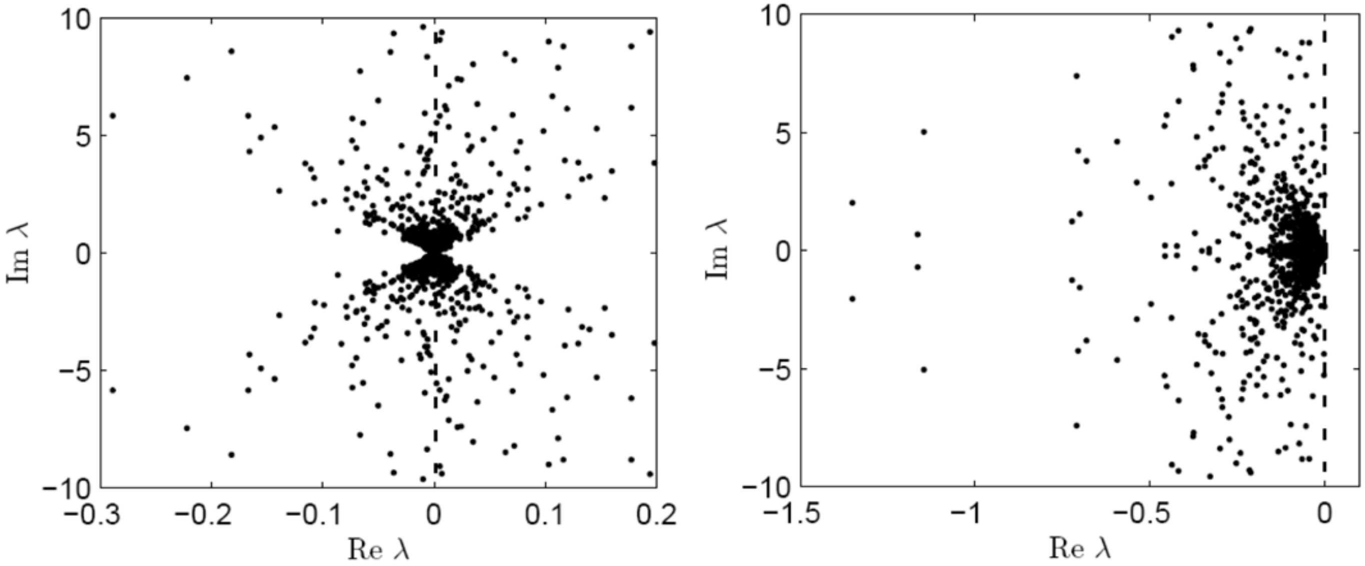
$$\frac{d}{dt}\left[\;\underline{u}\;\right] = \left[\quad DM \quad\right]\left[\;\underline{u}\;\right], \quad \text{use instead} \quad \frac{d}{dt}\left[\;\underline{u}\;\right] = \left\{\left[\quad DM \quad\right] - \gamma\left[\quad A \quad\right]^{-1}\right\}\left[\;\underline{u}\;\right]$$

Typical illustrations: Here shown in case of the vortex roll-up test case:

Left: No filter

Right: With the A^{-1}-filter

Note: No problem with *e*-values moving far out to the left; explicit time stepping still works just fine.


Im λ
Re λ
Im λ
Re λ