Generalized Aggregate Descriptor Approximations
and
Generalized Aggregate Product Descriptor Approximations

or

Construction of High Precision Surrogates

Chris Anderson
UCLA Dept. of Mathematics

Science at Extreme Scales:
Where Big Data Meets Large-Scale Computing

Workshop IV: New Architectures and Algorithms, Nov 26-30

DEPARTMENT OF Mathematics AT UCLA

Construction of high precision surrogates

Motivating application(s)/Context/Background

Abstract problem

1D Least squares splines

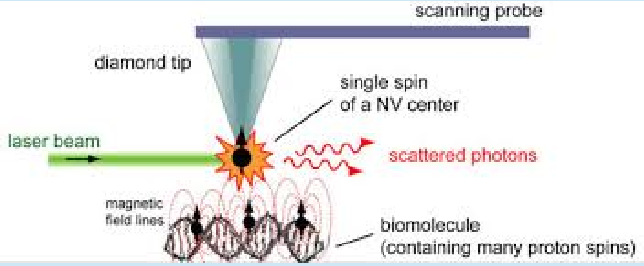
GADA and GAPDA approximations

Examples/Going forward

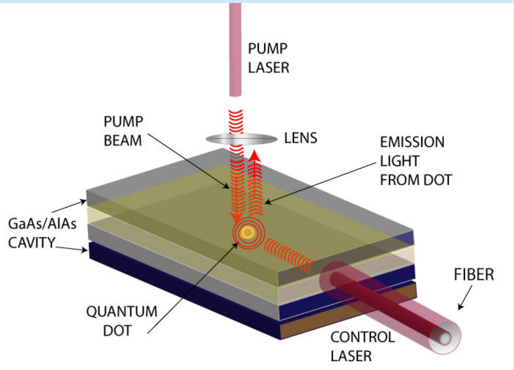
Motivation: Current and “next generation” of quantum devices

Search : Quantum Dots

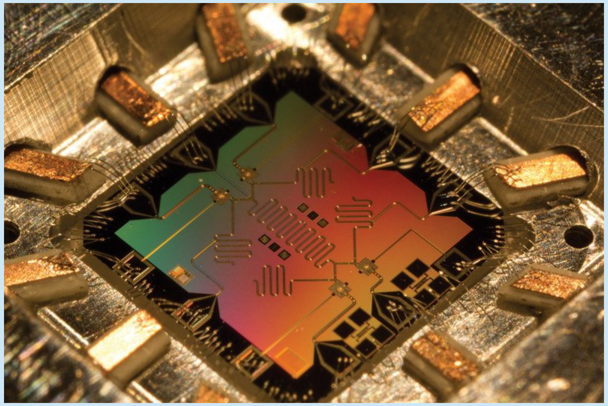
Quantum Sensing[1]



Quantum Dots in a Whole New Light[2]



The Future of Computing - Quantum & Qubits[3]



Quantum Cryptography[4]

1: http://web.mit.edu/degenlab/degenlab_techniques.html

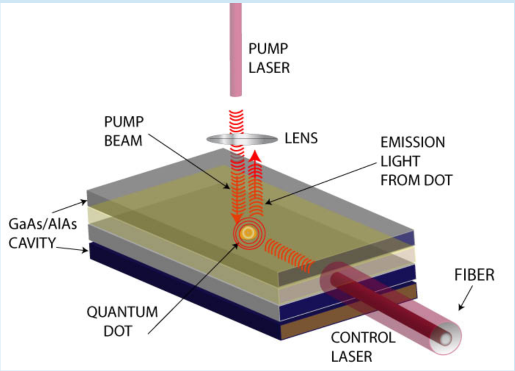
2: https://jqi.umd.edu/news/quantum-dots-whole-new-light

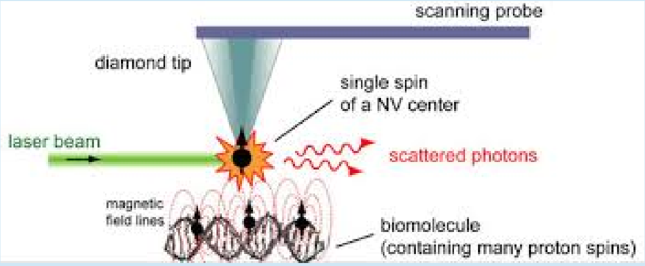
3 : https://www.autodesk.com/products/eagle/blog/future-computing-quantum-qubits/

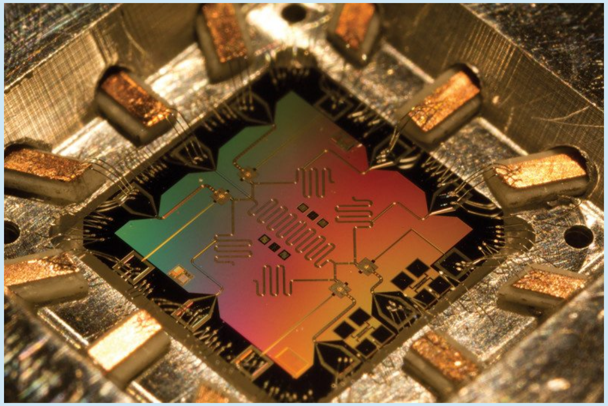
4: https://qutech.nl/academy_post/mooc-quantum-cryptography-starts-soon/screen-shot-2017-10-26-at-13-53-21/

Quantum mechanical system simulation challenges

- Complex local environment
- Multi-scale interaction with global environment
- Requires high precision approximations
- High computational cost

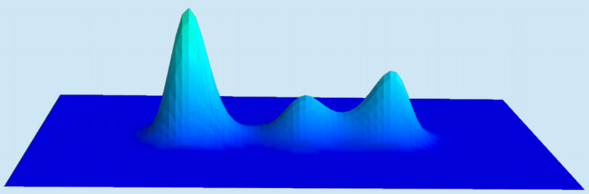






Sample Req'd Precision: 2-Dot FCI orbital convergence

# of orbitals		Ground State Energy (eV)
5	Energy_UD_1 :	8.720959134915952e-02
10	Energy_UD_1 :	8.718455204813846e-02
15	Energy_UD_1 :	8.718243856452054e-02
20	Energy_UD_1 :	8.718235456526482e-02
25	Energy_UD_1 :	8.718234263766753e-02
30	Energy_UD_1 :	8.718234124350170e-02
35	Energy_UD_1 :	8.718234065092317e-02
40	Energy_UD_1 :	8.718234052937261e-02



1.2155056e-10

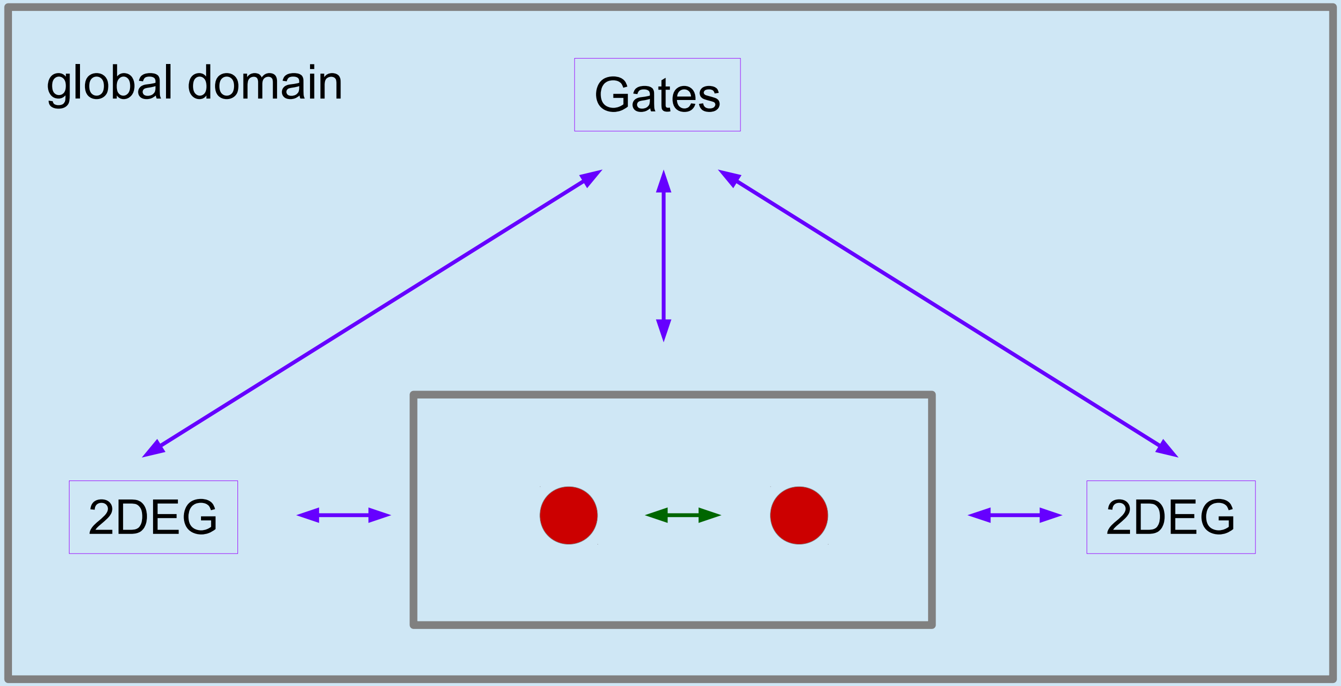
# of orbitals		1st excited state (eV)
5	Energy_UD_2 :	8.722103136190849e-02
10	Energy_UD_2 :	8.718685968161732e-02
15	Energy_UD_2 :	8.718246916998106e-02
20	Energy_DD_1 :	8.718239528836504e-02
25	Energy_DD_1 :	8.718235456614613e-02
30	Energy_DD_1 :	8.718235379265918e-02
35	Energy_DD_1 :	8.718235315055324e-02
40	Energy_DD_1 :	8.718235294294076e-02

Excited - Ground State Energy (eV)

5	1.1440e-5
10	2.3076e-6
15	3.0605e-8
20	4.0723e-8
25	1.1928e-8
30	1.2549e-8
35	1.2499e-8
40	1.2413e-8

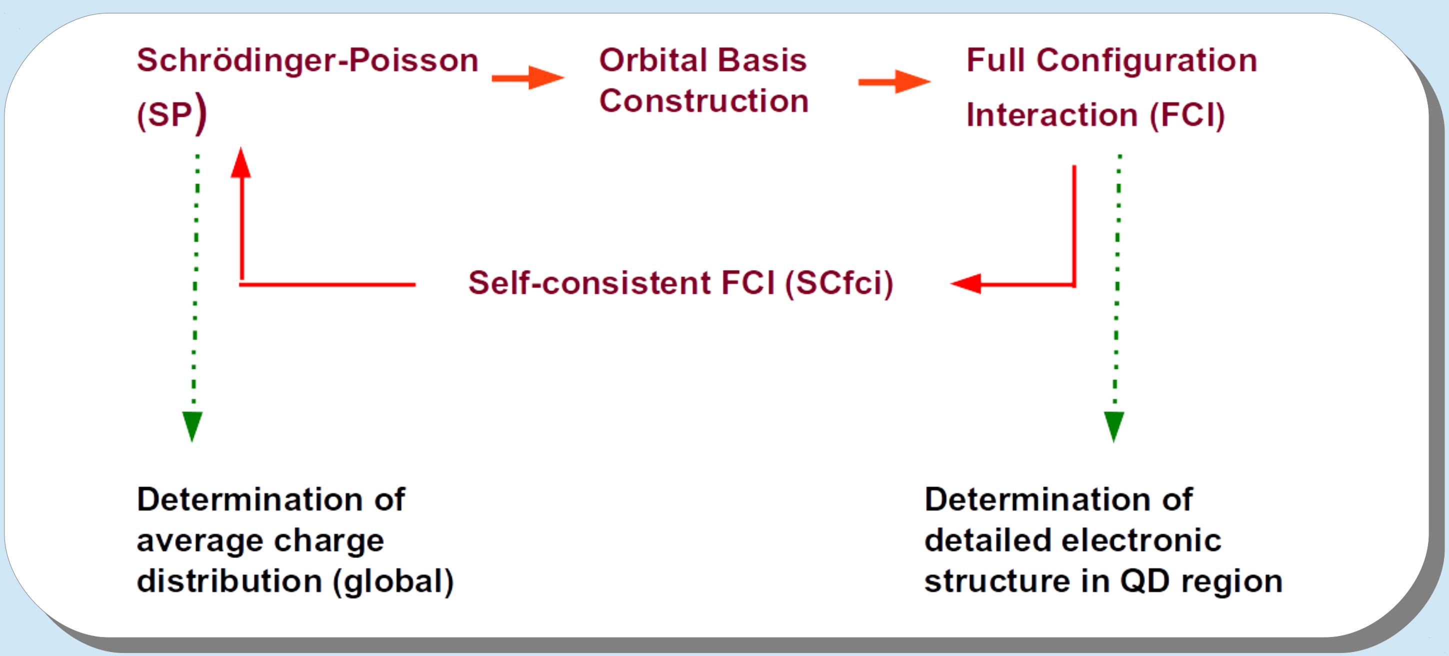
2.0761248e-10

Precision is required within a larger simulation...

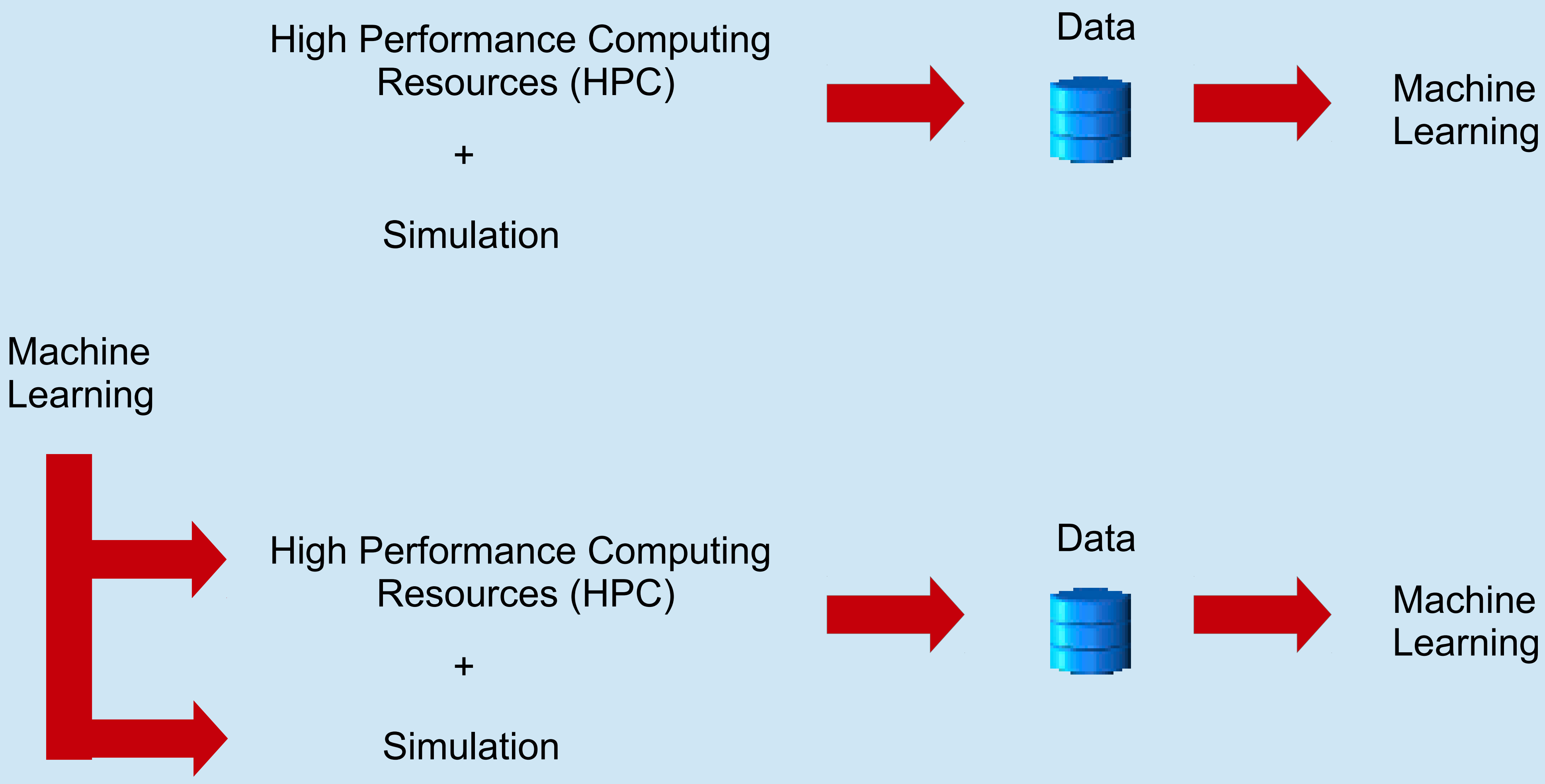


Average charge distributions interact electrostatically

Standard electronic quantum interaction in an external potential



Machine Learning and High Performance Computing



Machine Learning To Improve Simulation Performance

Machine Learning

High Performance Computing Resources (HPC)

+

Simulation

Data

Machine Learning

General motivation : Quantum system simulation

Replace CPU intensive tasks/simulations with surrogates

$$\vec{x} \in \Omega \subset \mathrm{R}^M \longrightarrow \mathrm{F} \longrightarrow \mathrm{R}$$

Ω = rectangular region

Computationally intensive task/simulation

$$\vec{x} \in \Omega \subset \mathrm{R}^M \longrightarrow \tilde{\mathrm{F}} \longrightarrow \mathrm{R}$$

Computationally inexpensive surrogate

General Problem

$$\vec{x} \in \Omega \subset \mathrm{R}^M \longrightarrow \mathrm{F} \longrightarrow \mathrm{R}$$

Ω = rectangular region

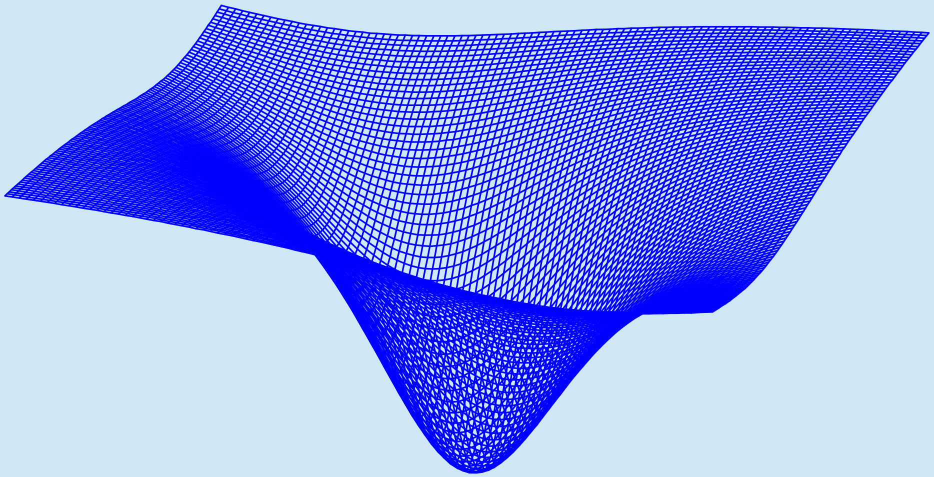
F = smooth function

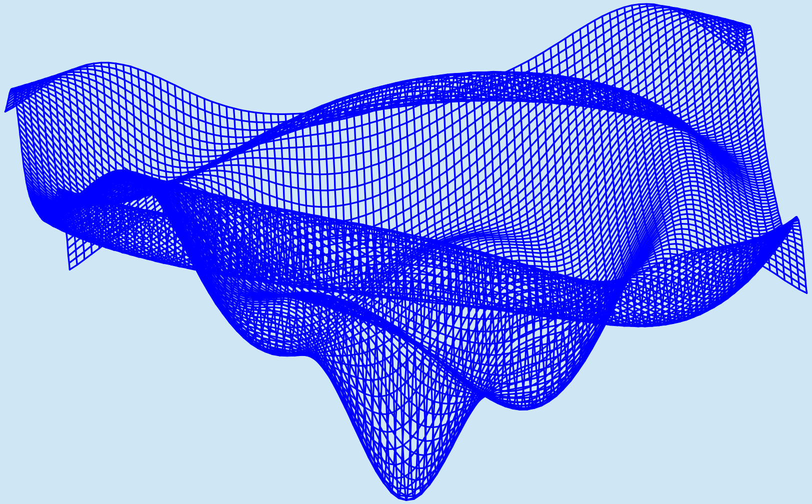
$$\vec{x} \in \Omega \subset \mathrm{R}^M \longrightarrow \tilde{\mathrm{F}} \longrightarrow \mathrm{R}$$

$\tilde{\mathrm{F}}$ = approximation with desirable properties

- Efficient to evaluate
- Derivatives approximate derivatives of F
- Well behaved convergence ($\| \ \|_\infty$) as training data size $\rightarrow \infty$
- Inclusion of “smoothing parameters”

Smooth functions





Training Data Assumptions

Training/validation data : $\{(\vec{x}_j, \mathrm{F}(\vec{x}_j))\}_{j=1}^{P} \in \Omega \subset \mathrm{R}^M$

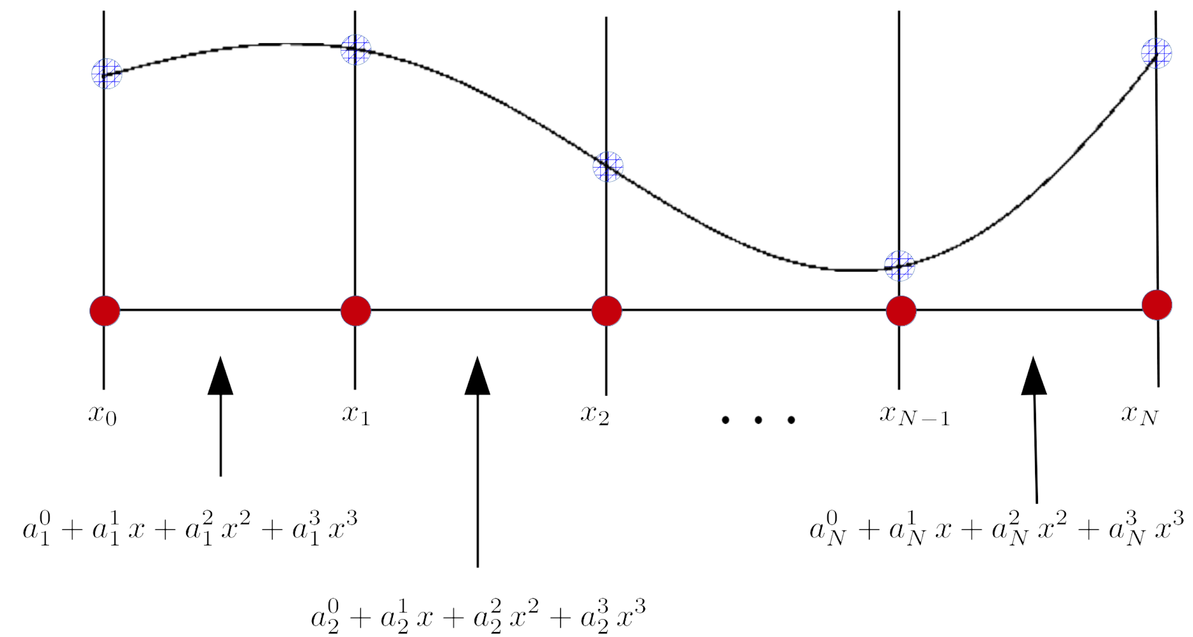
$$\vec{x} \in \Omega \subset \mathrm{R}^M \longrightarrow \tilde{\mathrm{F}} \longrightarrow \mathrm{R}$$

- All required data can be created as needed
- Data is created essentially without “noise”

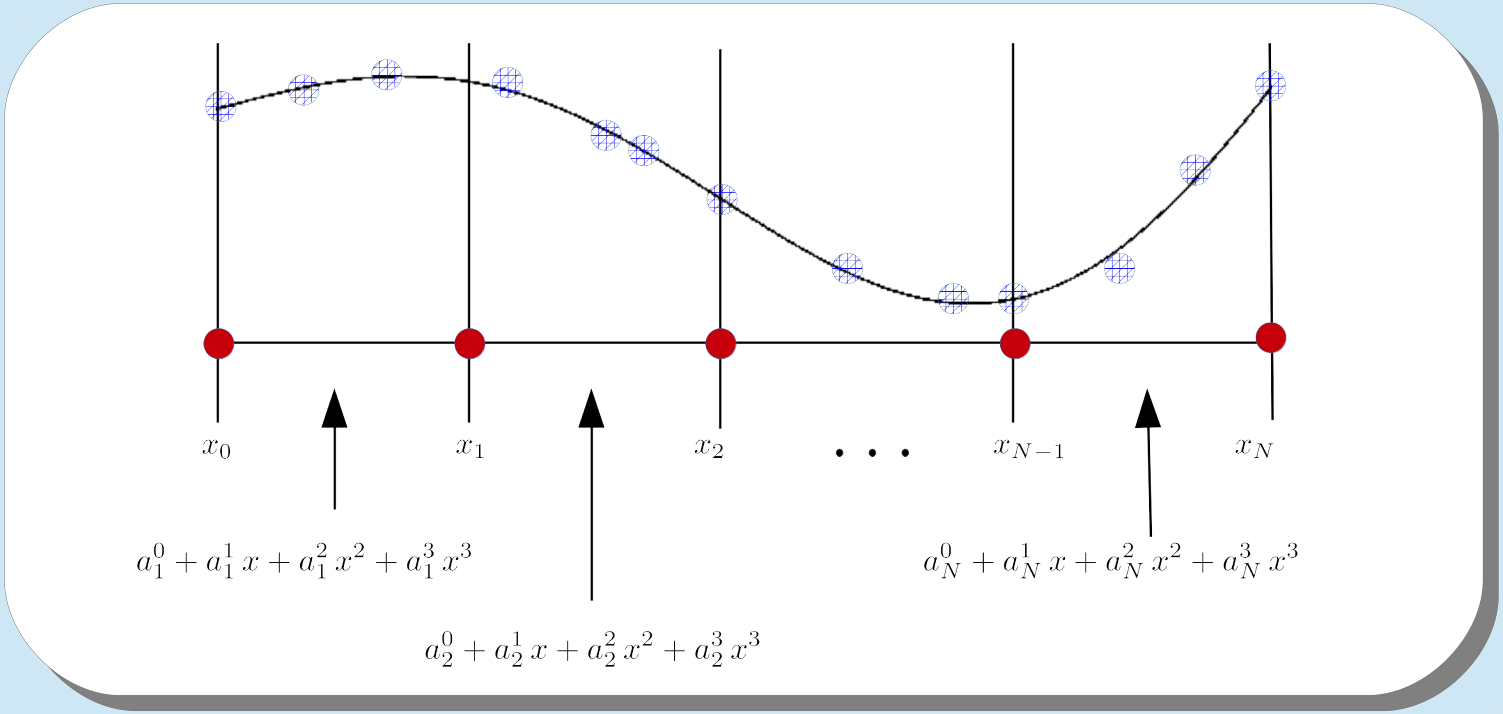
1D Spline Approximation

Smooth piecewise polynomial approximation

Cubic spline



1D Least Squares Spline Approximation

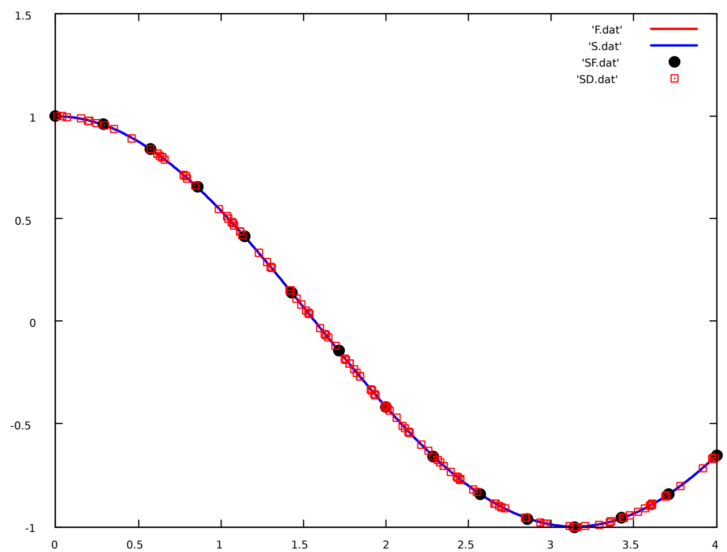


- A spline with pre-specified knots and a least squares fit to data at arbitrary locations.
- One can impose 2nd or 4th derivative “penalty” constraints:

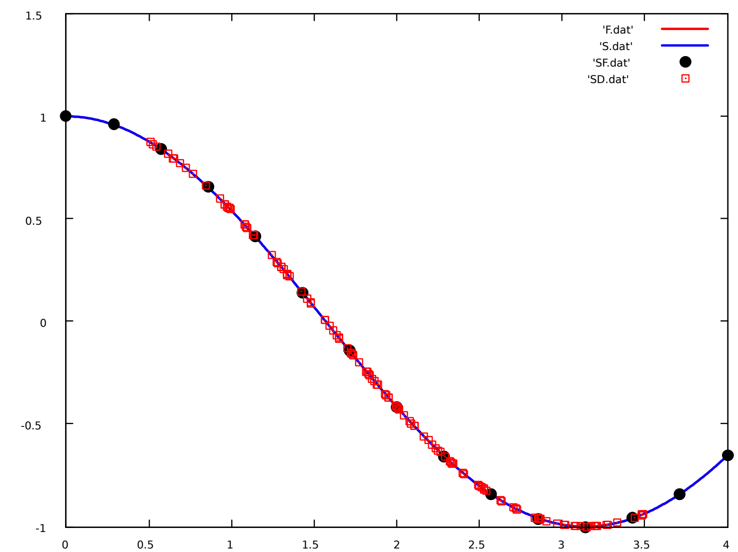
$$\epsilon\Delta S(x_j) = 0 \qquad \epsilon\Delta^2 S(x_j) = 0$$

Using Alternative Spline (AltSpline) construction

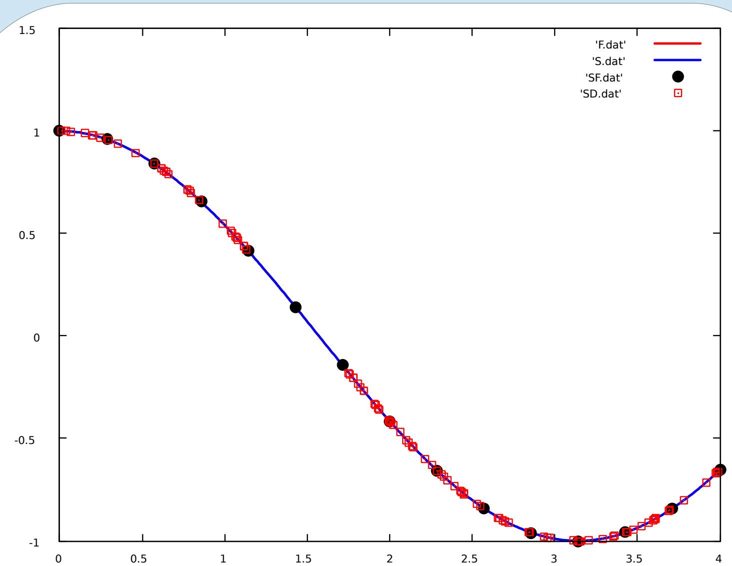
1D Least Squares Spline Approximation



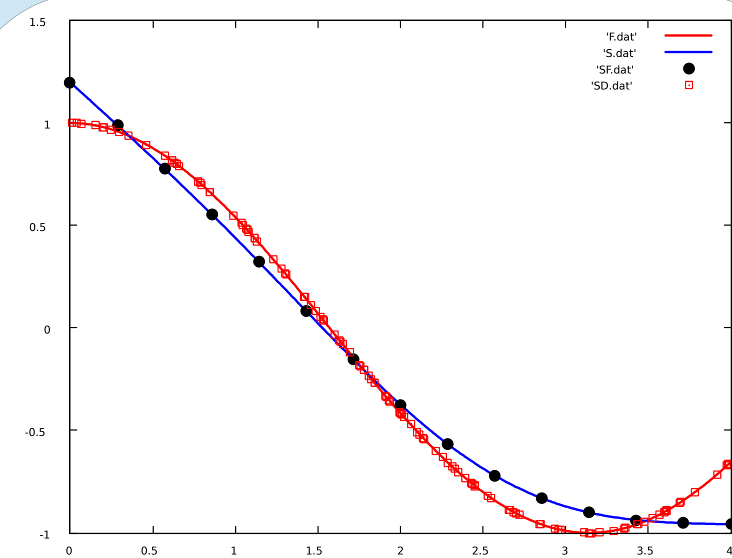
Standard LS spline fit



Extrapolation Test



Exclusion Test



Penalty Test

Construction of High Dimensional Approximations

Linear Model

$$\mathrm{F} \approx c + \sum_{i=1}^{M} a_i\, x_i$$

General Additive Model (GAM)

$$\mathrm{F} \approx \sum_{k=1}^{M} a_k\, f_k(\vec{x})$$

Approximations using "descriptors"

$$\{\phi_k(\vec{x})\}_{k=1}^{\mathrm{P}}$$

$$\phi_k(\vec{x}) : \mathrm{R}^M \to \mathrm{R}$$

Generalized Aggregate Descriptor Approximation

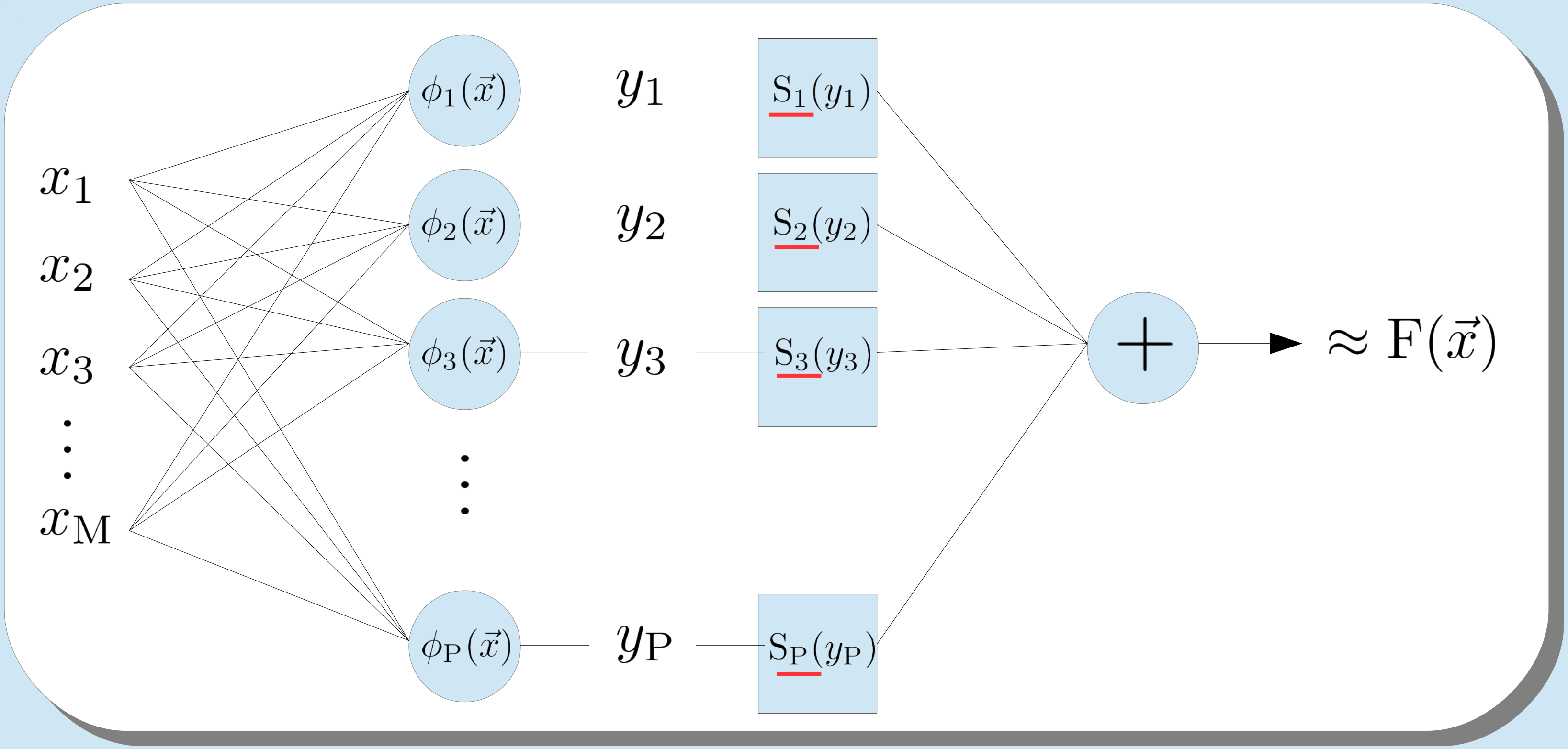
$$\mathrm{F}(\vec{x}) \approx \sum_{k=1}^{\mathrm{P}} \mathrm{S}_k(\phi_k(\vec{x})) = \sum_{k=1}^{\mathrm{P}} \mathrm{S}_k(y_k)$$

Least squares splines

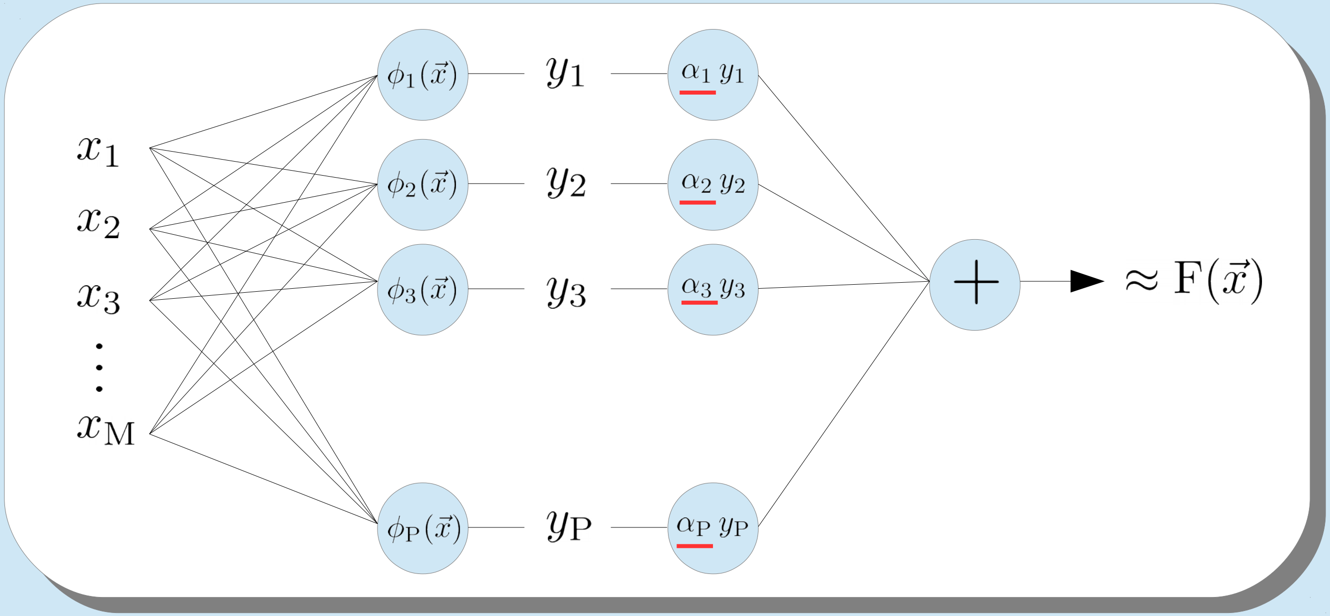
Generalized Aggregate Descriptor Approximation

A linear combination of least squares splines of descriptors

$$\{\phi_k(\vec{x})\}_{k=1}^{\mathrm{P}} \quad \mathrm{F}(\vec{x}) \approx \sum_{k=1}^{\mathrm{P}} \mathrm{S}_k(\phi_k(\vec{x})) = \sum_{k=1}^{\mathrm{P}} \mathrm{S}_k(y_k)$$



Typical least squares fitting of descriptors (and models)



α_k's determined** by minimizing

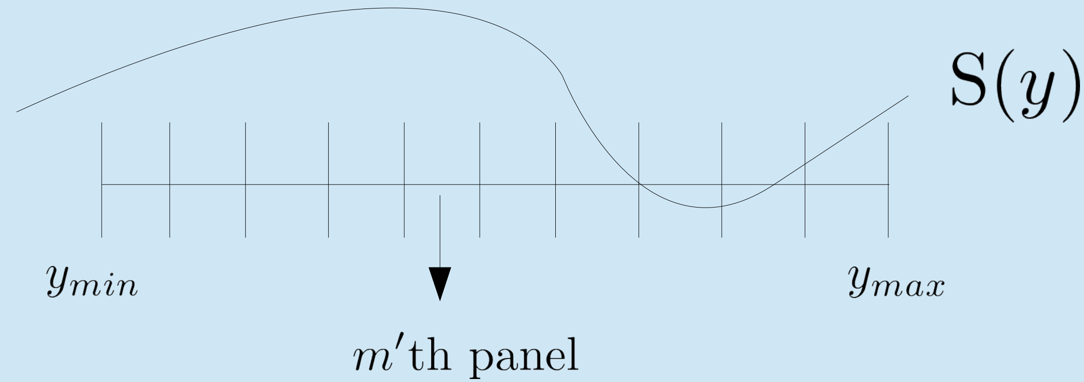
$$\sum_{i=1}^{i=N_t}\left(\mathrm{F}(\vec{x}_i) - \sum_{k=1}^{P}\alpha_k \phi_k(\vec{x}_i)\right)^2 = \sum_{i=1}^{i=N_t}\left(\mathrm{F}(\vec{x}_i) - \sum_{k=1}^{P}\alpha_k\, y_{(k,i)}\right)^2$$

$\{(\vec{x}_i\, \mathrm{F}(\vec{x}_i))\}_{i=1}^{N_t}$ = training data

A linear algebra problem

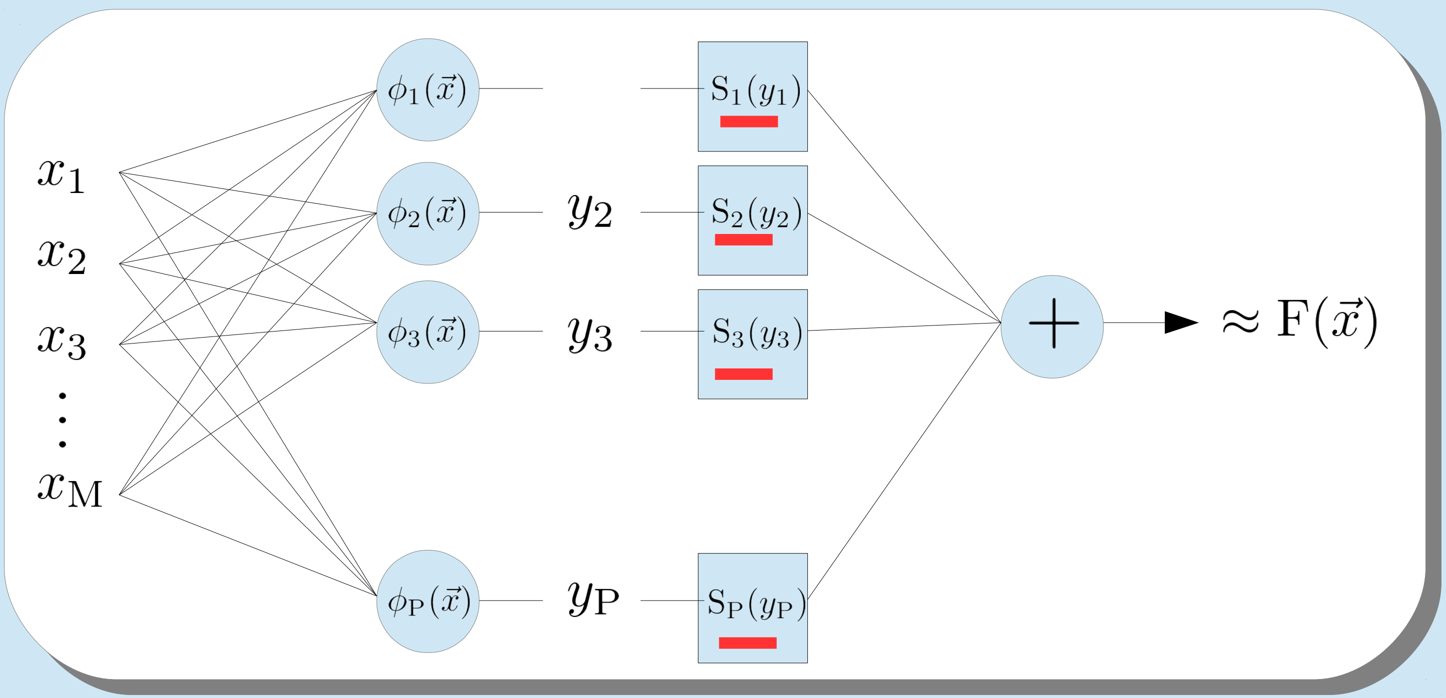
Least squares spline fitting of descriptors (and models)

$$\alpha\, y \to \mathrm{S}(y)$$



$$a_m + b_m y + c_m y^2 + d_m y^3 + \ldots$$

piecewise polynomial approximation



A "standard" linear least squares problem *

Choosing descriptors:

Suggested by the problem (physics) or just guessing

$$\{\phi_k(\vec{x})\} = \{x_1^2 + x_2^2, x_{x1} * x_2, \ldots\}$$

Determined by the data

Example: *Linear Combination descriptors*

$$\{\phi_k(\vec{x})\} = \{\vec{a}_k \cdot \vec{x}\}$$

$$||\vec{a}_k||_2 = 1$$

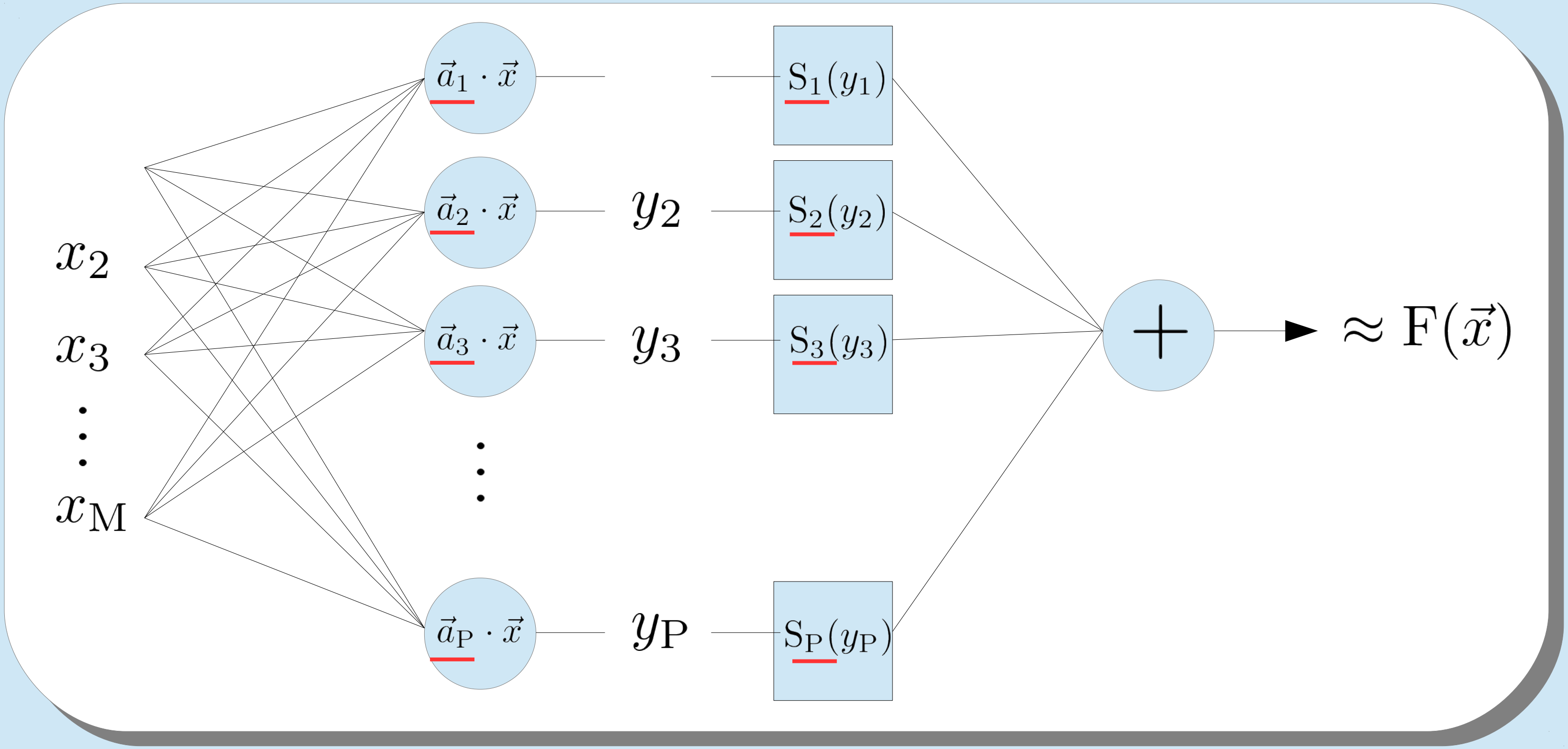
$$\mathrm{F}(\vec{x}) \approx \sum_{k=1}^{\mathrm{M}} \mathrm{S}_k(\vec{a}_k \cdot \vec{x})$$

Charles B. Roosen and Trevor J. Hastie,"Automatic Smoothing Spline Projection Pursuit". Journal of Computational and Graphical Statistics Vol. 3, No. 3 (Sep., 1994), pp. 235-248

+ Others....

GADA with LC Descriptors

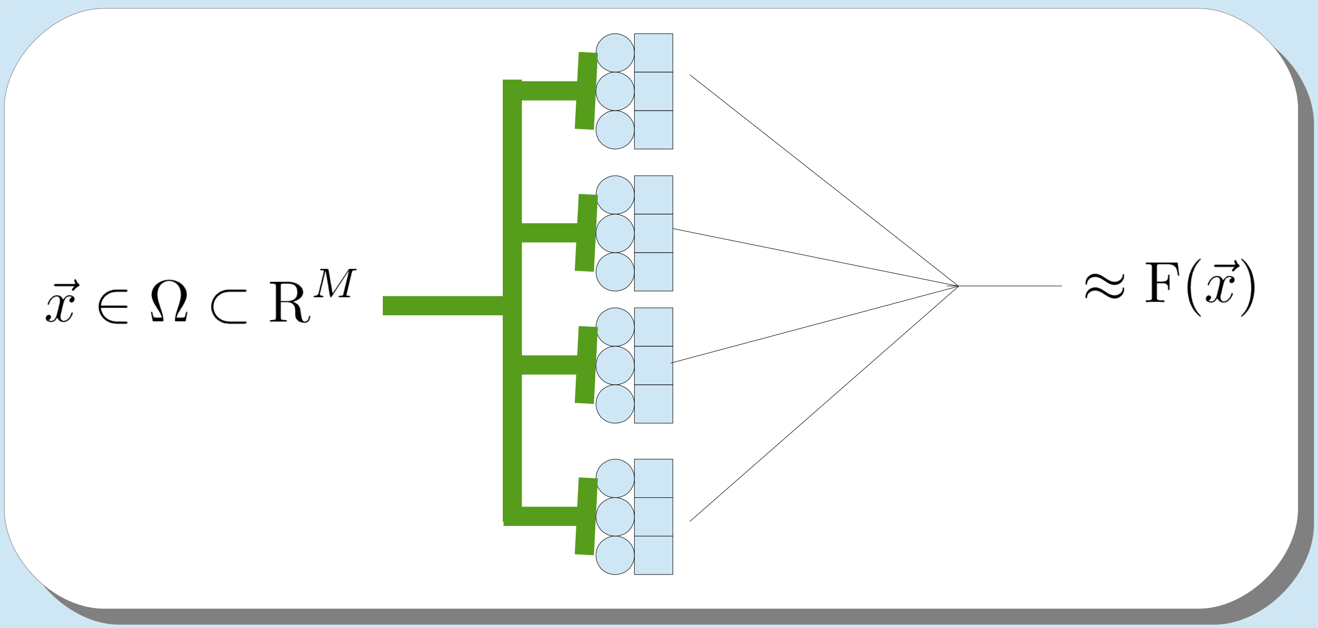
Determine $\vec{a}_k$ simultaneously with least squares spline coefficients.

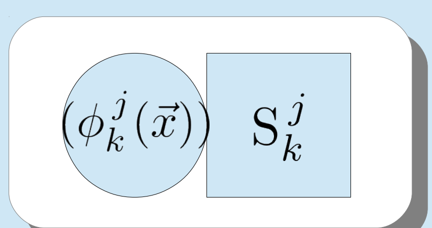


Generalized Aggregate Product Descriptors

$$\mathrm{F}(\vec{x}) \approx \sum_{k=1}^{\mathrm{Q}} \prod_{j=1}^{P} \mathrm{S}_k^j(\phi_k^j(\vec{x})) \qquad \left\{ \phi_k^j(\vec{x}) \right\}$$

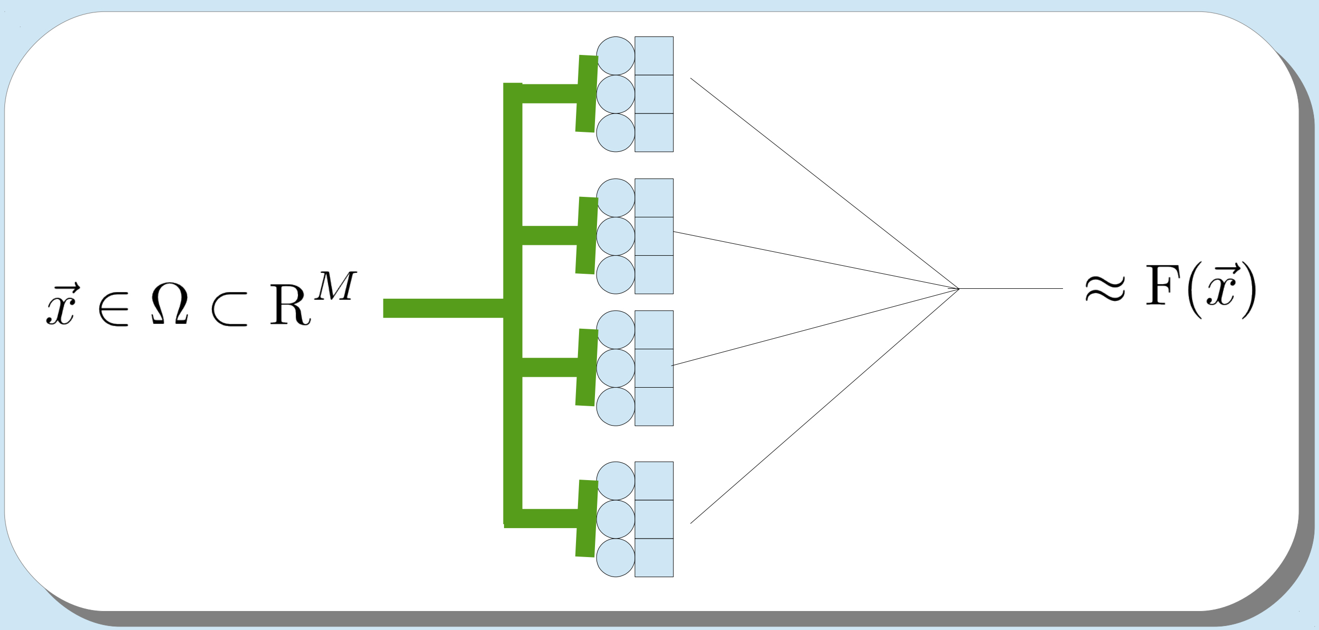
Sums of products of splined descriptors





Background/motivation

- A multi-dimensional product approximation
- Motivated (in part) by kernel ridge regression with Gaussian kernels
- Related ideas for discrete data : Canonical polyadic decomposition (CP) decomposition, (genaralization of SVD for high dimensional tensors), tensor-trains, tucker decomposition, etc. etc.
- Fitting strategy based upon those for CP methods.

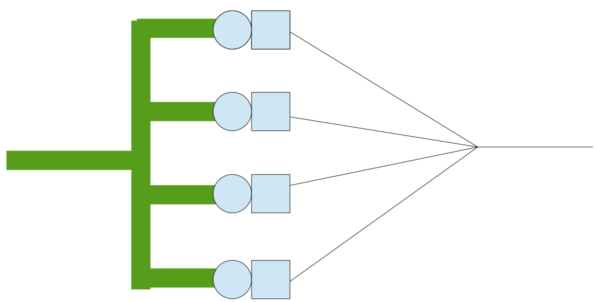


Approximation Framework

$$\vec{x} \in \Omega \subset \mathrm{R}^M \longrightarrow \tilde{\mathrm{F}} \longrightarrow \mathrm{R}$$

Approximations using "descriptors" $\{\phi_k(\vec{x})\}_{k=1}^{\mathrm{P}}$ $\phi_k(\vec{x}) : \mathrm{R}^M \to \mathrm{R}$

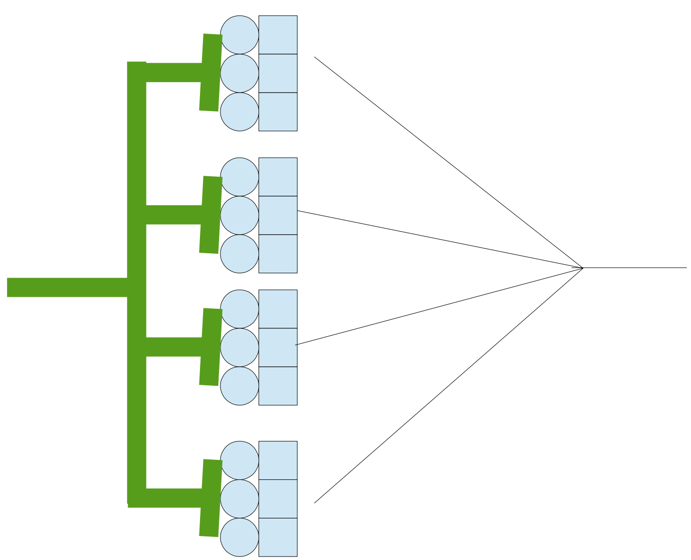
$\vec{x} \in \Omega \subset \mathrm{R}^M$



$\approx \mathrm{F}(\vec{x})$

Generalized Aggregate
Descriptor
(GADA)

$\vec{x} \in \Omega \subset \mathrm{R}^M$

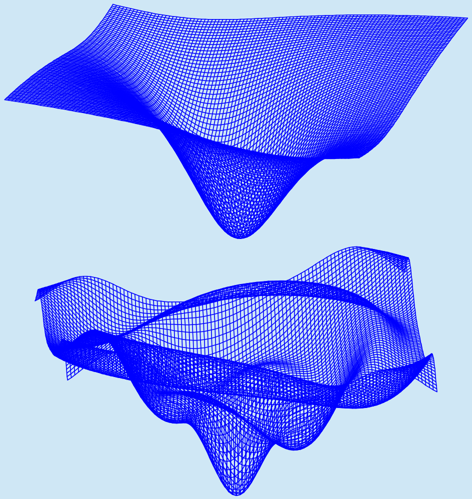


$\approx \mathrm{F}(\vec{x})$

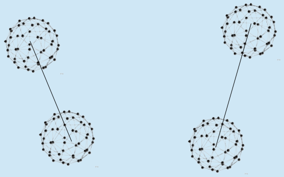
Generalized Aggregate
Product Descriptor
(GAPDA)

Examples

- User specified descriptors
- Adding a penalty
- Using GADA with adaptively determined LC descriptors
- Using GAPDA with coordinate descriptors



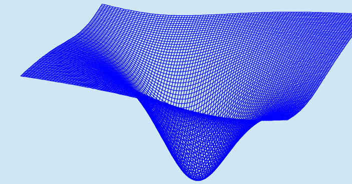
- Application to 2-electron integrals evaluation



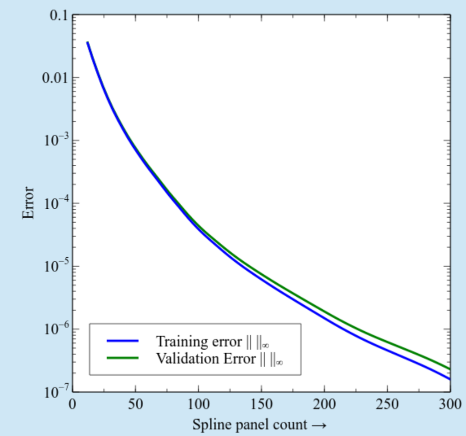
$$\mathrm{I}(A,B,C,D) = \int_{\mathrm{R}^3} \phi_A(\vec{s})\phi_B(\vec{s}) \int_{\mathrm{R}^3} \frac{\phi_C(\vec{r})\phi_D(\vec{r})}{||\vec{r}-\vec{s}||}$$

- GADA : User specified descriptors

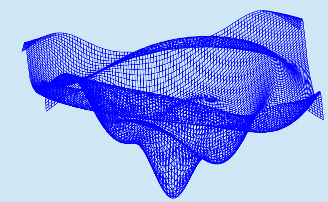
$$F(x,y) = -1/((\frac{x}{\sqrt{2}} + \frac{y}{\sqrt{2}})^2 + 4(\frac{x}{\sqrt{2}} - \frac{y}{\sqrt{2}})^2 + 0.2)$$



$$\phi_0(x,y) = (\frac{x}{\sqrt{2}} + \frac{y}{\sqrt{2}})^2 + 4(\frac{x}{\sqrt{2}} - \frac{y}{\sqrt{2}})^2 + 0.2$$

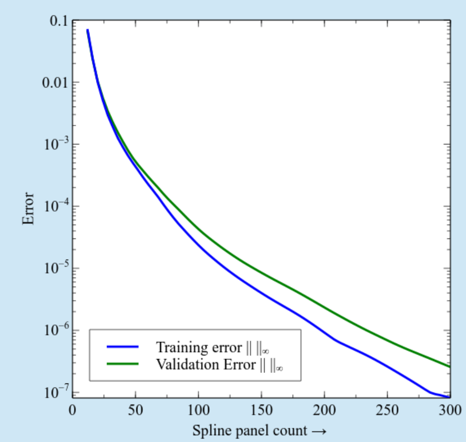


$$F(x,y) = -1/((\frac{x}{\sqrt{2}} + \frac{y}{\sqrt{2}})^2 + 4(\frac{x}{\sqrt{2}} - \frac{y}{\sqrt{2}})^2 + 0.2) - \frac{1}{2}(\cos(18\sqrt{x^2+y^2}))$$



$$\phi_0(x,y) = (\frac{x}{\sqrt{2}} + \frac{y}{\sqrt{2}})^2 + 4(\frac{x}{\sqrt{2}} - \frac{y}{\sqrt{2}})^2 + 0.2$$

$$\phi_1(x,y) = x^2 + y^2$$

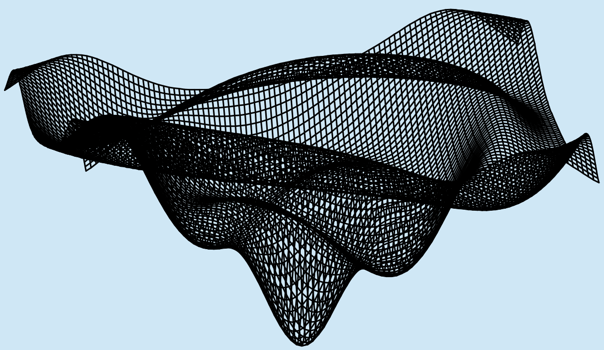


Cubic spline, 4000 T data, 1000 V data

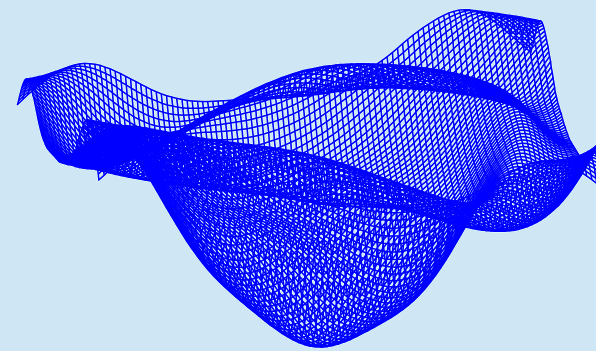
- GADA : LS spline fit with user specified descriptors and 2nd derivative penalty

$$\phi_0(x,y) = (\frac{x}{\sqrt{2}} + \frac{y}{\sqrt{2}})^2 + 4(\frac{x}{\sqrt{2}} - \frac{y}{\sqrt{2}})^2 + 0.2$$

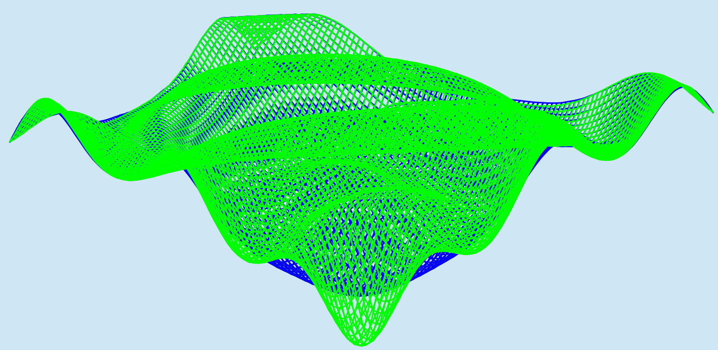
$$\phi_1(x,y) = x^2 + y^2$$



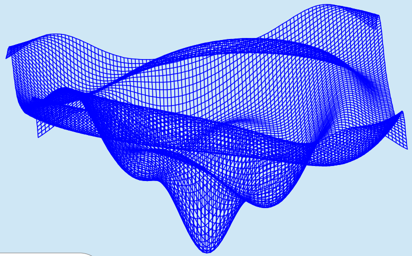
$\mathrm{F}(x,y)$



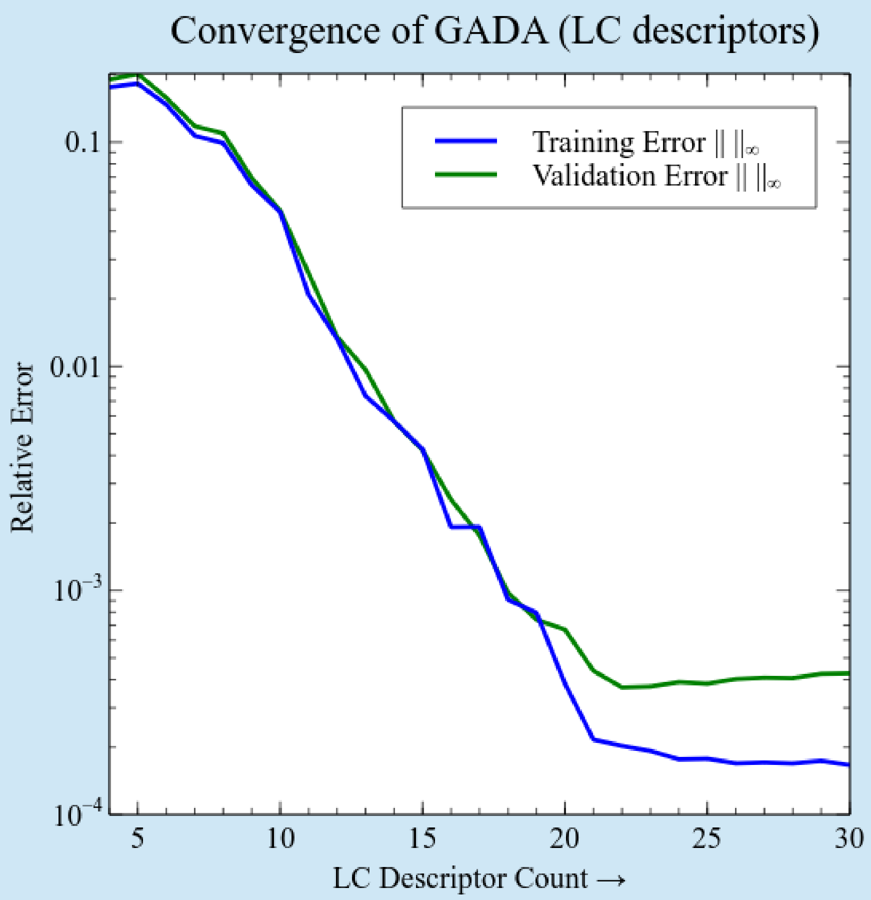
$\tilde{\mathrm{F}}(x,y)$



- Using GADA with adaptively determined LC descriptors



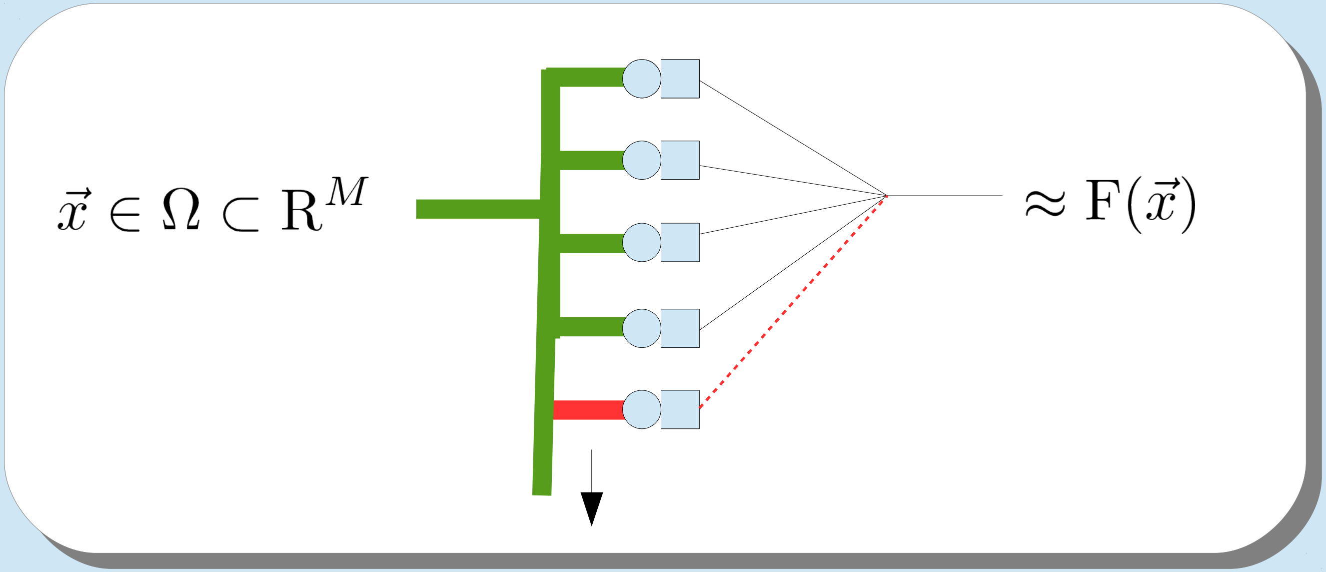
$$F(x,y) = -1/((\frac{x}{\sqrt{2}} + \frac{y}{\sqrt{2}})^2 + 4(\frac{x}{\sqrt{2}} - \frac{y}{\sqrt{2}})^2 + 0.2) - \frac{1}{2}(\cos(18\sqrt{x^2+y^2}))$$



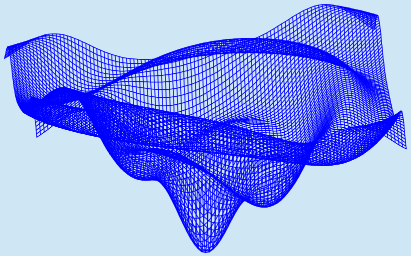
$$F(\vec{x}) \approx \sum_{k=1}^{M} S_k(\vec{a}_k \cdot \vec{x})$$

4000 T Data (random)
200x200 V Data (equispaced)

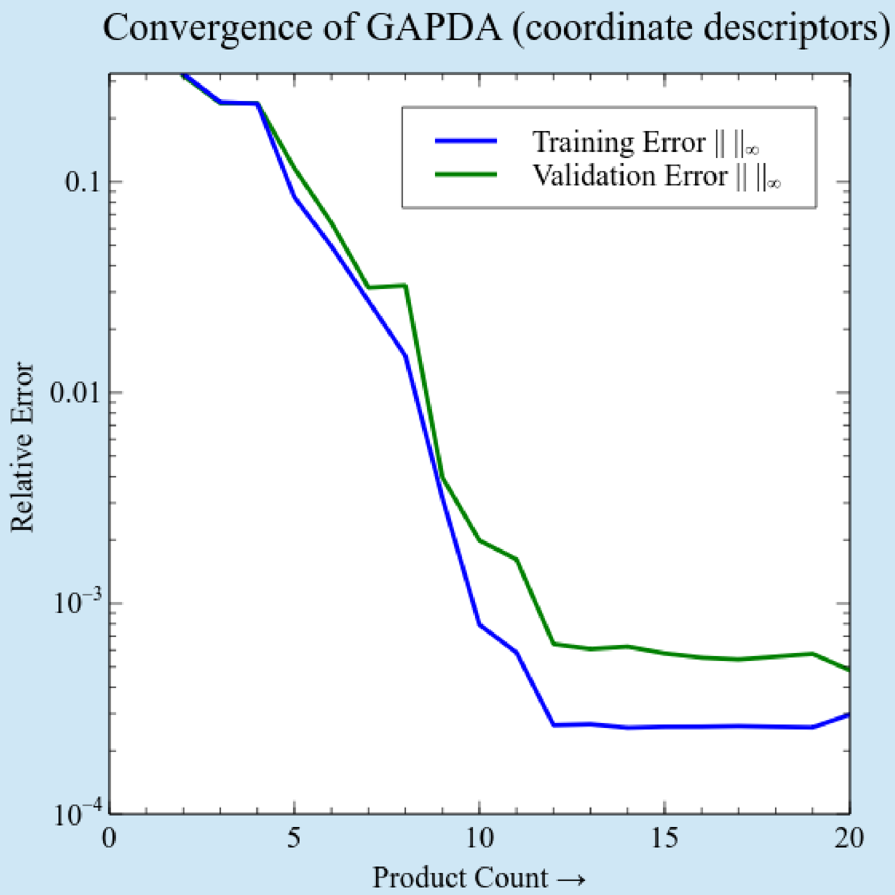
Convergence by adding terms: “broader not deeper”



- Using GAPDA with coordinate descriptors



$$F(x,y) = -1/((\frac{x}{\sqrt{2}} + \frac{y}{\sqrt{2}})^2 + 4(\frac{x}{\sqrt{2}} - \frac{y}{\sqrt{2}})^2 + 0.2) - \frac{1}{2}(\cos(18\sqrt{x^2+y^2}))$$



$$F(\vec{x}) \approx \sum_{k=1}^{Q} \prod_{j=1}^{2} S_k^j(x_j)$$

4000 T Data (random)
200x200 V Data (equispaced)

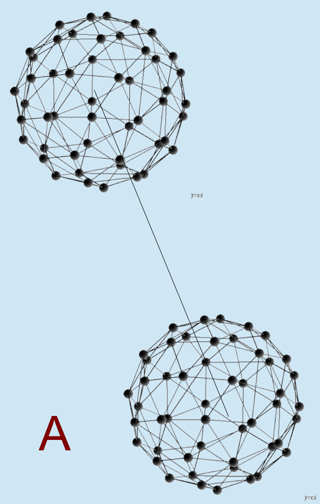
Application to 2-electron integrals evaluation: the task

$$\mathrm{I}(A,B,C,D) = \int_{\mathrm{R}^3} \phi_A(\vec{s})\phi_B(\vec{s}) \int_{\mathrm{R}^3} \frac{\phi_C(\vec{r})\phi_D(\vec{r})}{||\vec{r}-\vec{s}||}$$

$$\mathrm{I}(A,B,C,D) = \int_{\mathrm{R}^3} \phi_A(\vec{s})\phi_B(\vec{s})$$

$$\left[-4\pi\Delta^{-1}(\phi_C\phi_D) \right]$$

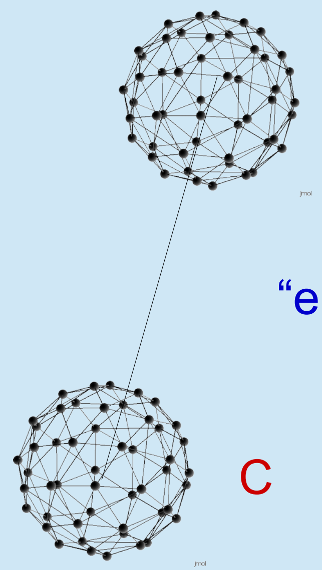
B



“receiver”

A

D



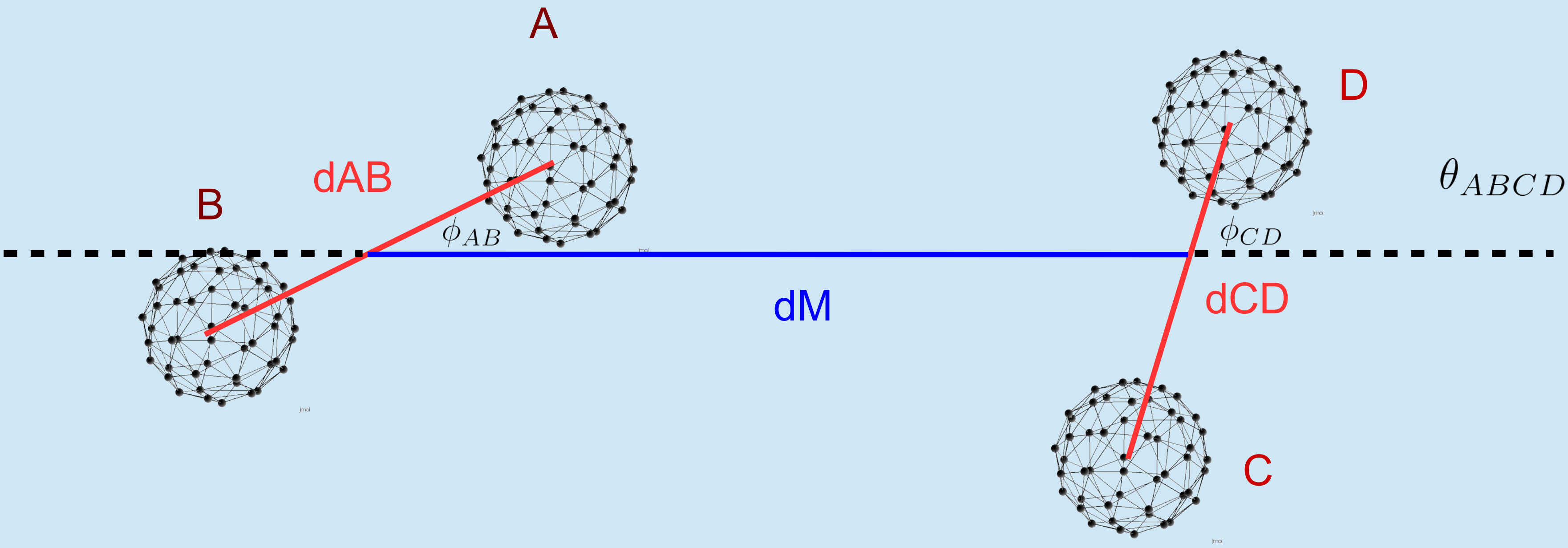
“electrostatic source”

C

Computational task: Evaluate an *average value* of the *potential* due a dipole source.

Where? Electronic structure computations

Coordinates (descriptors)



6 coordinates (descriptors based upon (A,B,C,D))

$$(\mathrm{dAB}, \mathrm{dCD}, \mathrm{dM}, \phi_{AB}, \phi_{CD}, \theta_{ABCD})$$

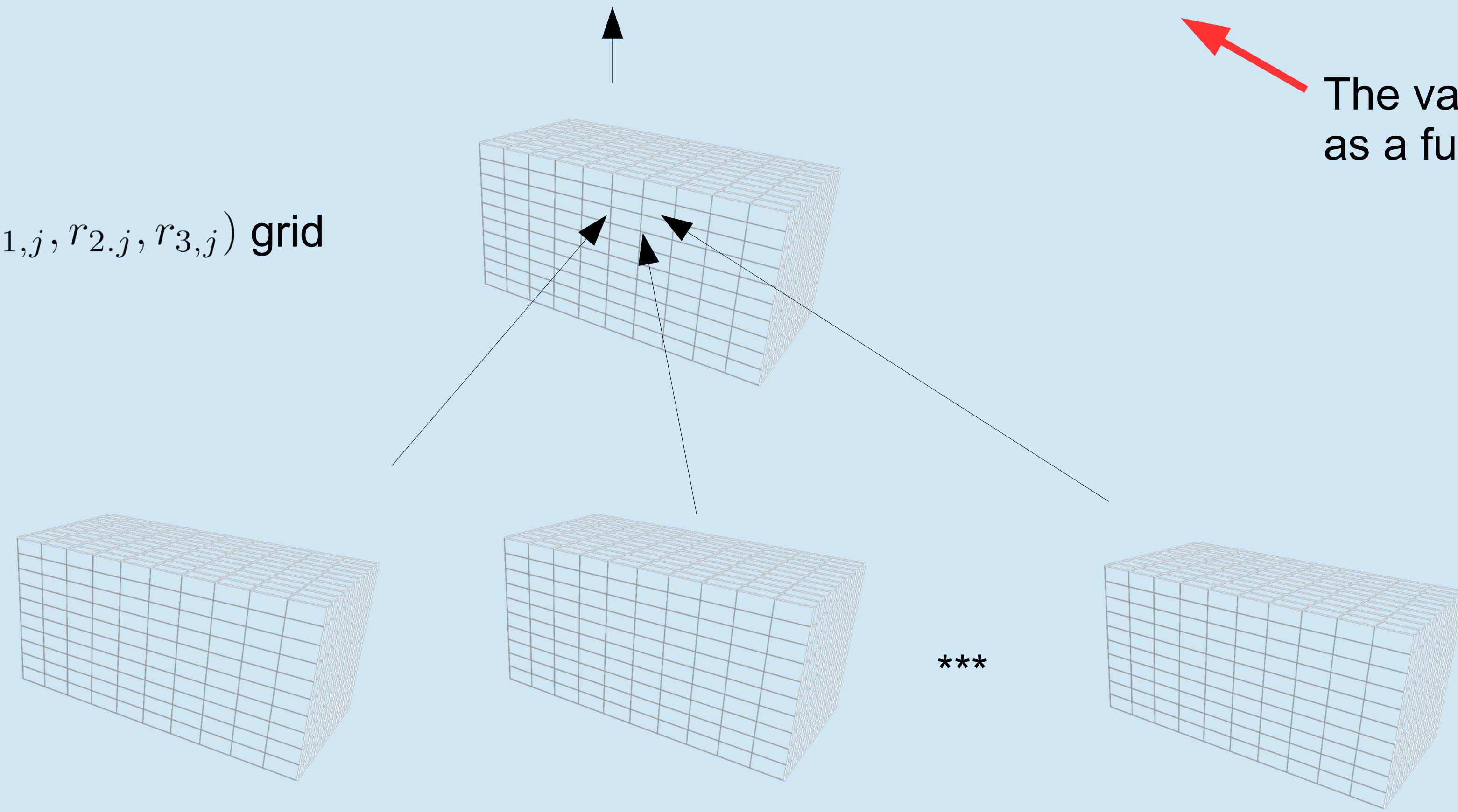
A classical approach (why not?)

Nested high order equidistant polynomial interpolation

$$\mathrm{I}(\vec{x}) \approx \sum \alpha_j \tilde{\mathrm{I}}_{(\theta_1,\theta_2,\theta_3)} \left(r_{1,j} r_{2.j}, r_{3,j}\right)$$

The value at $(r_{1,j}, r_{2.j}, r_{3,j})$
as a function of $(\theta_1, \theta_2, \theta_3)$

$(r_{1,j}, r_{2.j}, r_{3,j})$ grid



$(\theta_1, \theta_2, \theta_3)$ grids: one for each $(r_{1,j}, r_{2.j}, r_{3,j})$

Nested high-order interpolation results

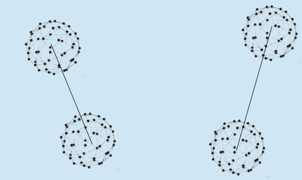
Convergence w.r.t. radial variables, keeping theta variables fixed.

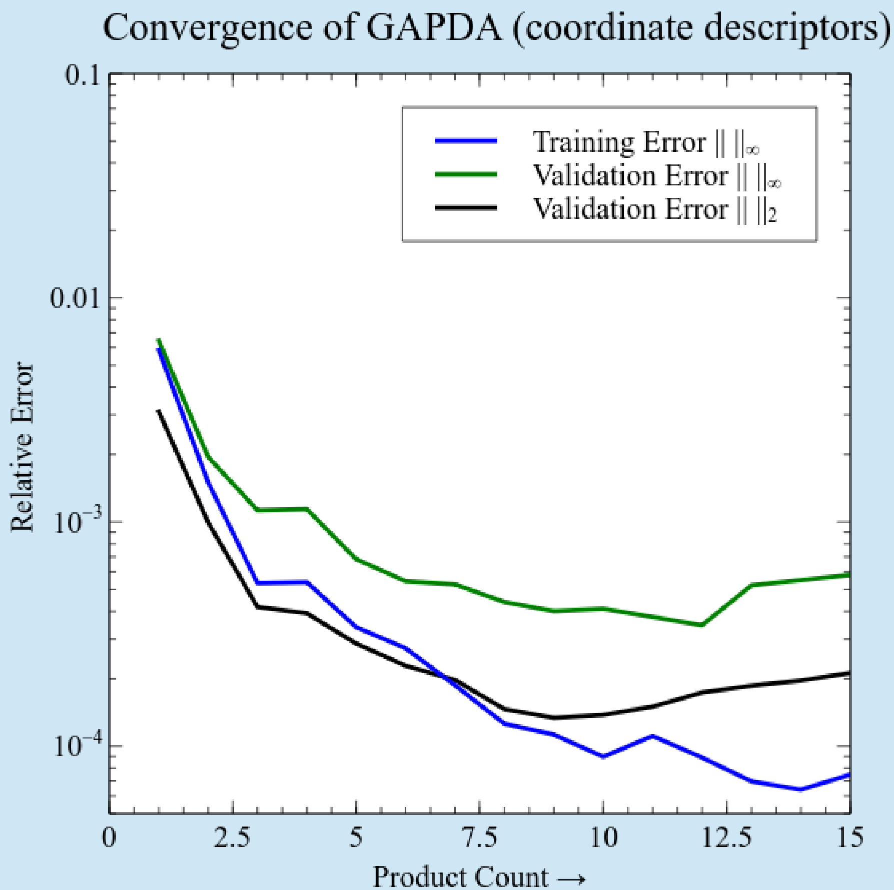
Number of samples	Maximal Error $\|\| * \|\|_{\infty}$	Data size
287K	0.0121622	2 MB
1M	0.000648798	8MB
2M	0.000398793	16MB

Generalized Aggregate Product Approximation Results

$$\mathrm{I}(A,B,C,D) = \int_{\mathrm{R}^3} \phi_A(\vec{s})\phi_B(\vec{s}) \int_{\mathrm{R}^3} \frac{\phi_C(\vec{r})\phi_D(\vec{r})}{||\vec{r}-\vec{s}||}$$

$$\mathrm{F} : \mathrm{R}^6 \rightarrow \mathrm{R}$$





$$\mathrm{F}(\vec{x}) \approx \sum_{k=1}^{Q} \prod_{j=1}^{6} \mathrm{S}_k^j(x_j)$$

Training data size : 4000
Validation data size : 4000
Quintic Splines

Model storage: 230 KB

2-electron Integral Approximation Comparison

Nested high-order interpolation

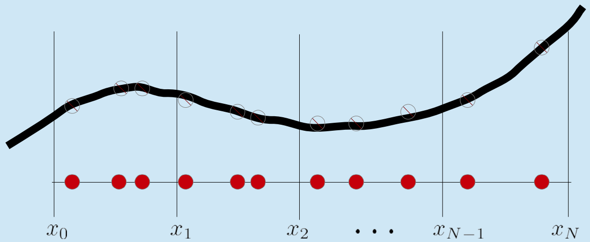
Number of samples	Maximal Error $\|\| * \|\|_\infty$	Data size
287K	0.0121622	2 MB
1M	0.000648798	8MB
2M	0.000398793	16MB

Generalized Aggregate Product
(4000 Samples)

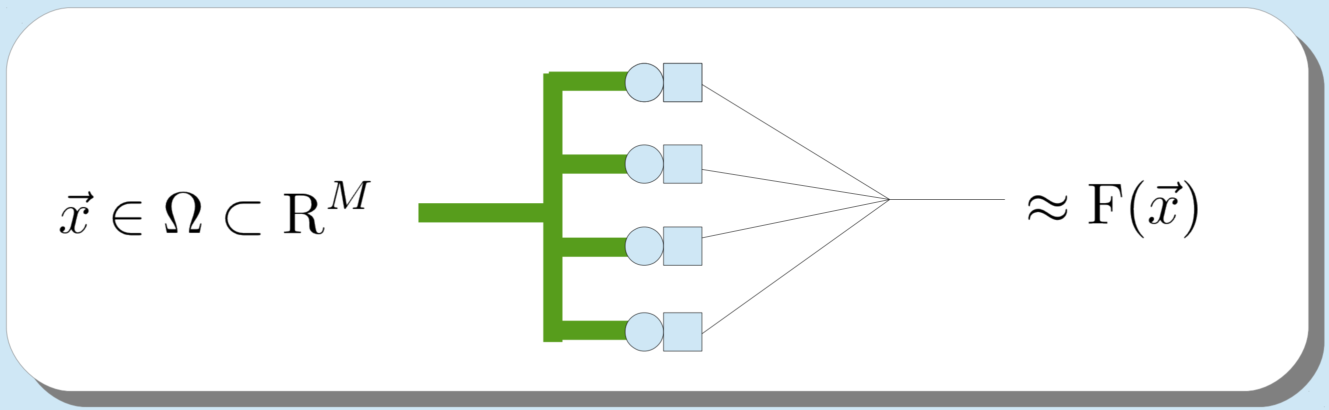
Product Count	Max Rel. Error	RMS Rel. Error
1	0.00644047	0.00311191
2	0.00195411	0.000996405
3	0.00112727	0.000417999
4	0.00114105	0.000391691
5	0.000682333	0.000286763
6	0.000544277	0.000228528
7	0.000527051	0.000197421
8	0.000439617	0.000146379
9	0.000400546	0.000133788
10	0.000410176	0.000137895
11	0.000377382	0.000150192
12	0.000346323	0.000173615

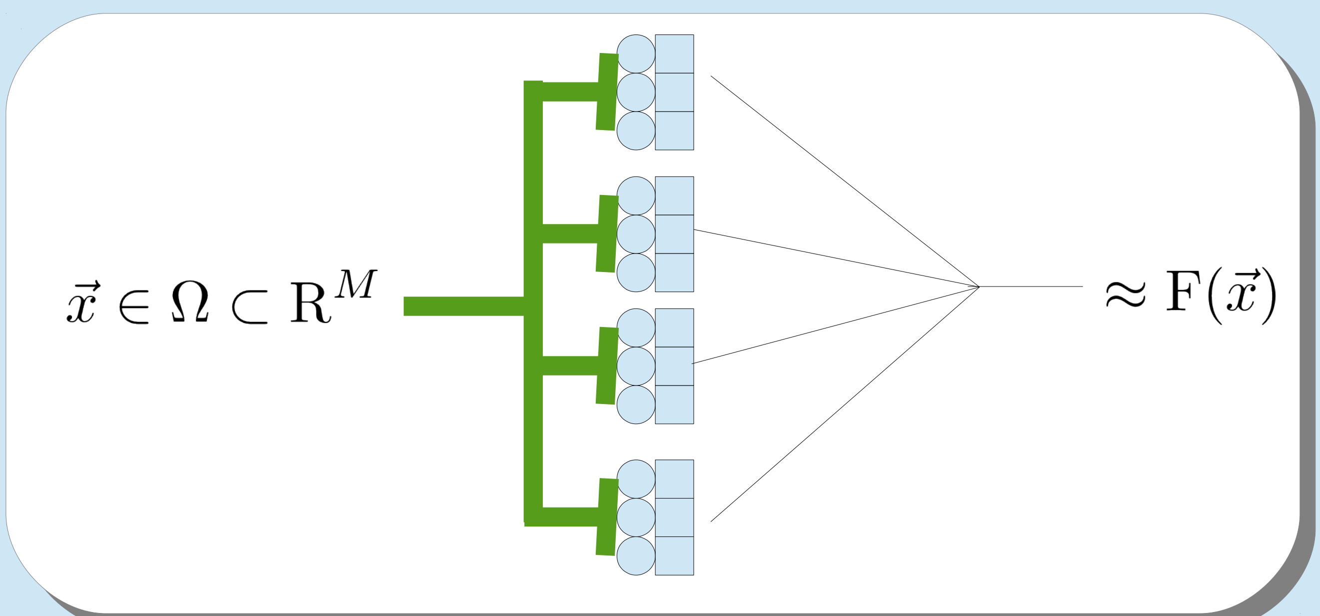
Descriptors + least squares splines + fitting procedures

$$\left\{ \phi_k^j(\vec{x}) \right\}$$

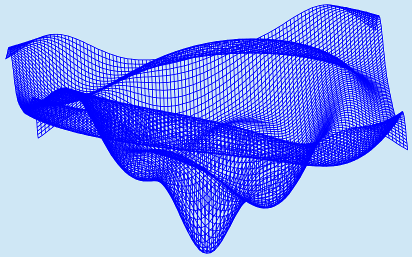


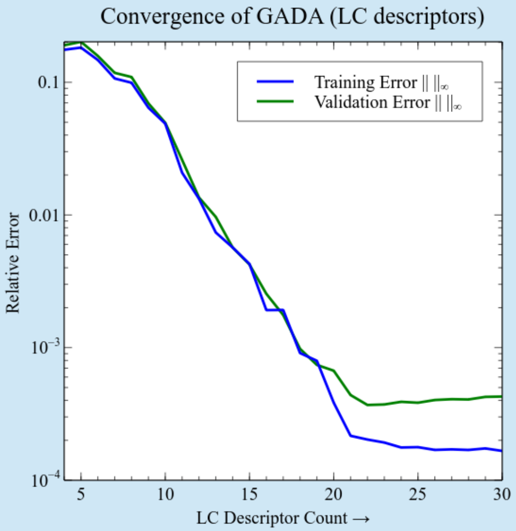

```
*
*
gadaUtility.setSVDverbose(svdVerboseFlag);
gadaUtility.setSolverMethod(solverMethod);
gadaUtility.setSVDcutoff(svdCutoff);

gadaUtility.updateFit(genAggDescriptor, Xdata, FX);

currentResidual     = gadaUtility.getResiduals(genAggDescriptor, Xdata, FX);
RconditionNumber = gadaUtility.getRconditionNumber();

*
*
```

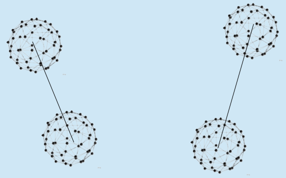


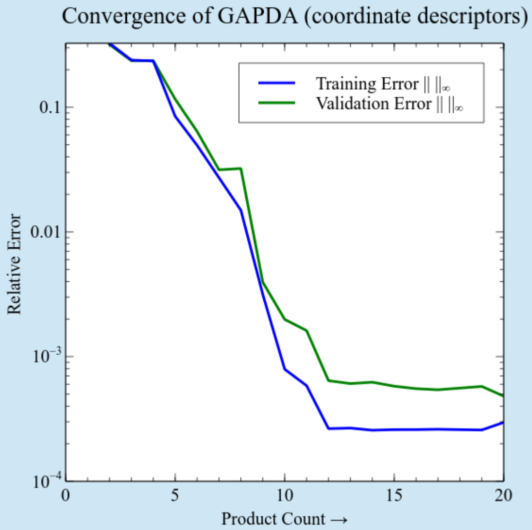


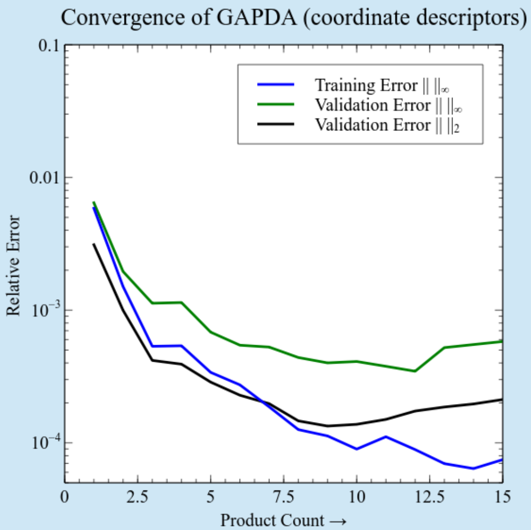
= appoximations of smooth functions



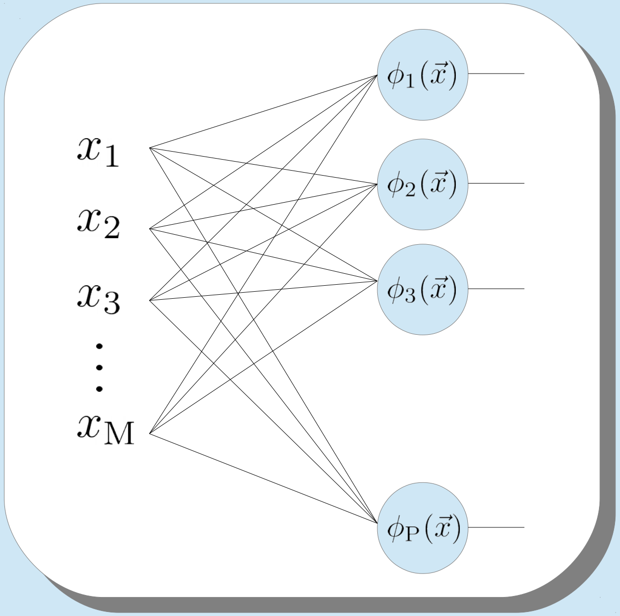






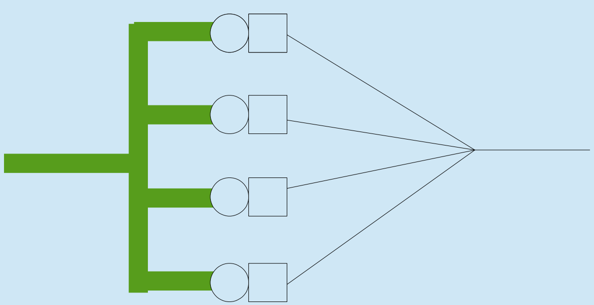


Finding physics



$$\mathrm{F}(\vec{x}) \approx \sum_{k=1}^{\mathrm{P}} \mathrm{S}_k(\phi_k(\vec{x}))$$

Spline identifies contribution of descriptor



Alternative to LASSO or L1?

“Drop in” replacement for KRR?

Kernel Ridge-Regression approximation

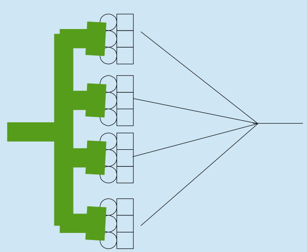
$$\mathrm{F}(\vec{x}) \approx \sum \alpha_j \, e^{\frac{-||\vec{x}-\vec{x}_j||^2}{\sigma^2}}$$

Gaussians are “separable”

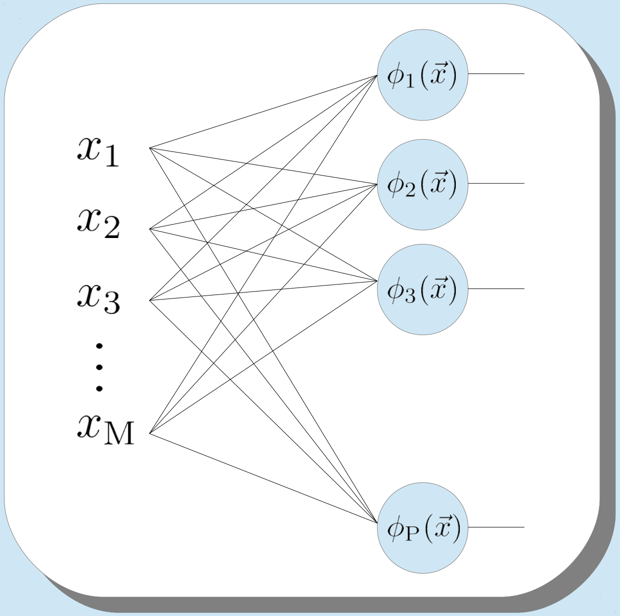
$$\mathrm{F}(\vec{x}) \approx \sum_{j=1}^{N} \alpha_j \, e^{\frac{-(x_1-x_{1,j})^2}{\sigma^2}} \; e^{\frac{-(x_2-x_{2,j})^2}{\sigma^2}} \; e^{\frac{-(x_3-x_{3,j})^2}{\sigma^2}}$$

$$\mathrm{F}(\vec{x}) \approx \sum_{k=1}^{Q} \mathrm{S}_k^1(x_1)\mathrm{S}_k^2(x_2)\mathrm{S}_k^3(x_3)$$

Q << N

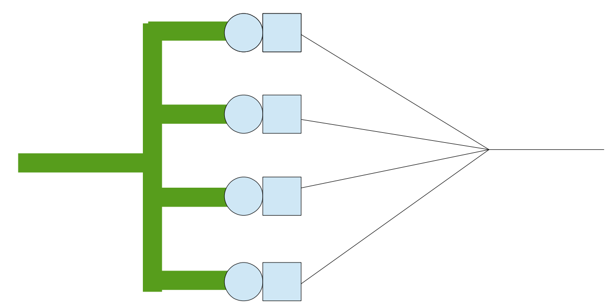


There are other ways to non-linearly stitch descriptors together

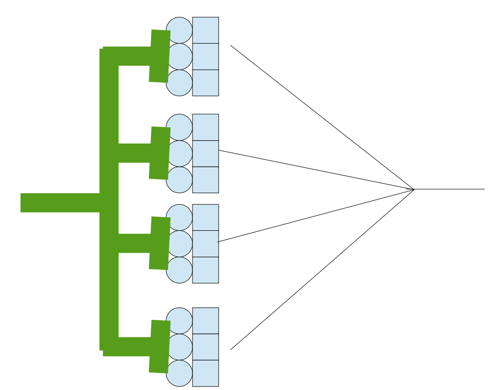


Neural Network

$$\mathrm{F}(\vec{x}) \approx \sum_{k=1}^{\mathrm{P}} \mathrm{S}_k(\phi_k(\vec{x}))$$



$$\mathrm{F}(\vec{x}) \approx \sum_{k=1}^{\mathrm{Q}} \prod_{j=1}^{P} \mathrm{S}_k^j(\phi_k^j(\vec{x}))$$



Others ….

Thanks!