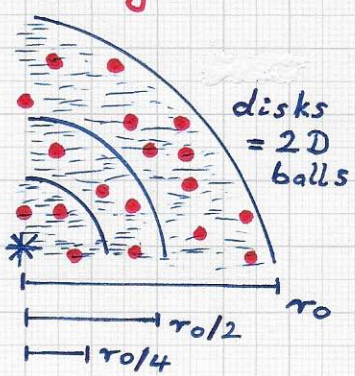
■ OBJECT AND MATERIAL EIGENFUNCTIONS - Cont'd.

• Laplacian eigenfunctions and neural networks: ...

● It is possible to adapt the described iterated stochastic Richardson extrapolation scheme to the estimation of a density value for a specific $\mathcal{S}$-location in the $\mathcal{S}$-domain — where one is given a finite set of $\mathcal{S}$-tuples $(S_1, \ldots, S_c)$ only. We first discuss density estimation more generally. We are interested in density in a d-dimensional ball. Since density is given by "mass/volume" (physical density) and our "masses" are (unit mass) points, density is simply defined as number of points per volume. Further, our volumes of interest are d-dimensional balls with a common center '*'. The volume of a d-dimensional ball (d-ball) with radius r is $V_d = c_d \cdot r^d$, where the dimension-dependent constants c_d are defined on p. 1, 9-10-2022. Thus, if N points lie inside a d-ball with volume V_d, then the resulting point density will be $\varrho = N / V_d$. The figure (above) shows (parts of) three concentric circles with radii r_0, $r_0/2$ and $r_0/4$ that are the boundary circles of three 2-balls (disks) with common center point '*'.



...

■ OBJECT AND MATERIAL EIGENFUNCTIONS - Cont'd.

• Laplacian eigenfunctions and neural networks: ...

• We can now perform Richardson extrapolation in this way:

i) We define the sequence of d-ball radius values as $r_0, r_0/2, r_0/4, \ldots$

ii) We calculate the corresponding d-ball volume value sequence, i.e., $V_d^0 = c_d r_0^d$,
$V_d^1 = c_d (r_0/2)^d = V_d^0/2^d$,
$V_d^2 = c_d (r_0/4)^d = V_d^0/4^d$,
$V_d^3 = c_d (r_0/8)^d = V_d^0/8^d$, ...

iii) We calculate the numbers $N_0, N_1, N_2, N_3, \ldots$, i.e., the numbers of points (5-tuples) inside the d-balls with radii $r_0, r_0/2, r_0/4, \ldots$

iv) We compute the resulting point densities, i.e., $\rho_0 = N_0/V_d^0$, $\rho_1 = N_1/V_d^1$, $\rho_2 = N_2/V_d^2, \ldots$

These steps and formulas can be summarized:

- Define radius r_0.
- Using the sequence $r_{i+1} = r_i/2$, $i = 0, 1, 2, \ldots$, i.e., $r_i = r_0/2^i$, $i = 0, 1, 2, \ldots$, calculate the volumes
$$V_d^i = c_d \left(\frac{r_0}{2^i}\right)^d, \quad i = 0, 1, 2, \ldots$$
- Determine the numbers $N_0, N_1, N_2, \ldots$ and the final density values
$$\rho_i = \frac{N_i}{c_d} \left(\frac{2^i}{r_0}\right)^d, \quad i = 0, 1, 2, \ldots$$

...

■ OBJECT AND MATERIAL EIGENFUNCTIONS - Cont'd.

• Laplacian eigenfunctions and neural networks: ...

• Example. We consider the 2D case ($d=2$), where a sequence of disks is used and the value of $c_d = c_2$ is π. One obtains:

$$\varrho_0 = N_0/\pi \cdot (2^0/r_0)^2 = N_0/(\pi r_0^2) = N_0/a_0 ,$$
$$\varrho_1 = N_1/\pi \cdot (2^1/r_0)^2 = N_1 \cdot 4/(\pi r_0^2) = 4N_1/a_0 ,$$
$$\varrho_2 = N_2/\pi \cdot (2^2/r_0)^2 = N_2 \cdot 16/(\pi r_0^2) = 16N_2/a_0 ,$$
$$\varrho_3 = N_3/\pi \cdot (2^3/r_0)^2 = N_3 \cdot 64/(\pi r_0^2) = 64N_3/a_0 , \ldots$$

Here, $a_0 = \pi r_0^2$ is the area of the disk with radius r_0.

For the 3D case ($d=3$), $c_d = c_3 = 4/3\pi$, and one obtains:

$$\varrho_0 = N_0/(4/3\pi) \cdot (2^0/r_0)^3 = \ldots = N_0/v_0 ,$$
$$\varrho_1 = N_1/(4/3\pi) \cdot (2^1/r_0)^3 = \ldots = 8N_1/v_0 ,$$
$$\varrho_2 = N_2/(4/3\pi) \cdot (2^2/r_0)^3 = \ldots = 64N_2/v_0 ,$$
$$\varrho_3 = N_3/(4/3\pi) \cdot (2^3/r_0)^3 = \ldots = 512N_3/v_0 , \ldots$$

Here, $v_0 = 4/3\pi r_0^3$ is the volume of the ball with radius r_0.

One must keep in mind the progression of the values of the $(2^i)^d$ terms that grow rapidly:

$d=2$: 1, 4, 16, 64, ...
$d=3$: 1, 8, 64, 512, ...
$d=4$: 1, 16, 256, 4096, ...

This rapid growth has two implications in the context of our application to material classification:

i.) The $\mathbb{S}$-domain should be low-dimensional.

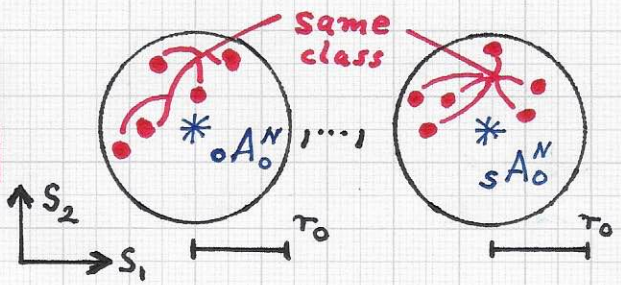
ii.) One should be prepared to use relatively large numbers of material samples.

...

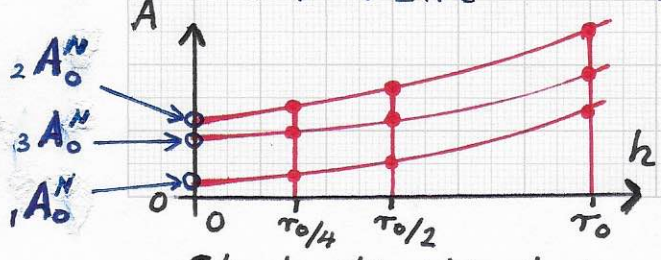
■ OBJECT AND MATERIAL EIGENFUNCTIONS - Cont'd.

• Laplacian eigenfunctions and neural networks:...

• We briefly discuss why one can understand iterated Richardson extrapolation as STOCHASTIC extrapolation in our classification setting: We compute one density estimate $\rho(*)$ based on one set of material samples. Thus, this ONE density estimate $\rho(*)$ is SAMPLE-DEPENDENT. A - generally - better density estimate could be generated by calculating several density estimates - each one based on its own set of material samples - and finally computing the average of all sample-specific density estimates. The figure (left) sketches this idea at an abstract level. Individual, sample-specific density estimates ${}_sA_o^N$, $s=1...S$, are calculated via standard iterated Richardson extrapolation. The different material sample sets used produce the different S-tuple sets shown as '•' data in the 2D disk with radius r_o, with center '*'.



The figure (left, bottom) captures the essential idea: Three sample sets ultimately produce three extrapolated values ${}_sA_o^o$, $s=1,2,3$, producing, after iteration, ${}_sA_o^N$. $\Rightarrow \rho(*) = \frac{1}{3}\sum_{s=1}^{3} {}_sA_o^N$.



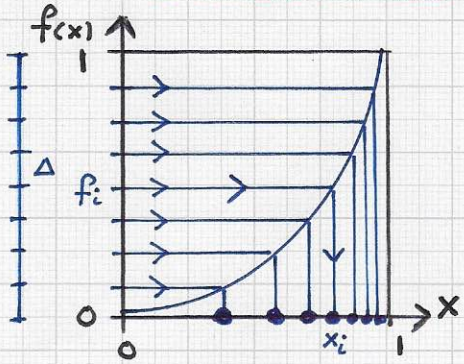
Stochastic iterated Richardson extrapolation

...

■ OBJECT AND MATERIAL EIGENFUNCTIONS - Cont'd.

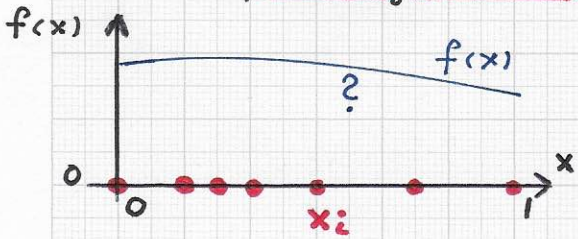
• Laplacian eigenfunctions and neural networks:...

Distribution function



Generating specific distribution / density of x_i-values along the x-axis. A uniform spacing Δ is used on the f-axis, defining function values f_i of the distribution function $f(x)$. The value of x_i is obtained by calculating the inverse value of function value f_i.

Generation of distribution / density function



In our application, we are concerned with the construction of a "high-quality" approximation of an unknown distribution / density function $f(x)$ when only samples x_i are given.

• **Note.** In numerical grid generation, one is concerned with the discretization of the continuum domain space of a function via point sets - usually connected point sets where points define vertices of certain finite elements - where "point density is defined by distribution functions." The figure (left, top) illustrates the principle of a distribution function: One needs to discretize the x-domain interval of some function to be approximated by (n+1) points that should be distributed in the x-domain interval in a specific way - via a "distribution function $f(x)$." This function is used for point generation as follows: One samples the function $f(x)$ uniformly / equidistantly along the f-axis and, for f-values $f_i = i/n$, $i=0...n$, one calculates the corresponding x_i-values, defining the desired point distribution and density on the x-axis.

...