Rekursive Verfahren und hierarchische Datenstrukturen in der numerischen Analysis

Hans-Joachim Bungartz

Unter Berücksichtigung von Vorlesungen von Christoph Zenger

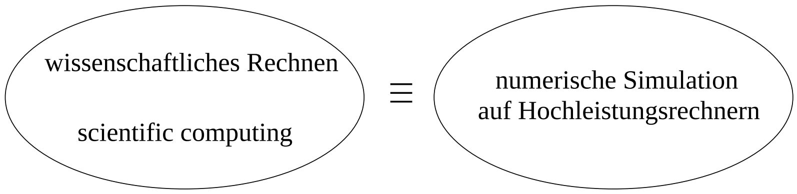
17. März 1999

Inhaltsverzeichnis

Kapitel 1

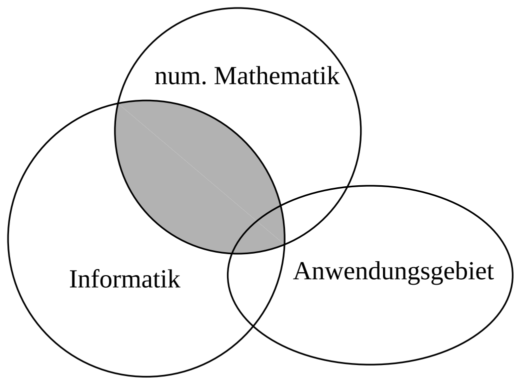
Einführung

1.1 Umfeld der Vorlesung, Literatur



1. Modellbildung:
    - Erstellung eines physikalischen bzw. mathematischen Modells („vereinfachtes Abbild der Realität“)
      i. d. R.: System gekoppelter gewöhnlicher Differentialgleichungen (ODE) bzw. partieller Differentialgleichungen (PDE)
2. Numerik:
    - Diskretisierung (Gebiet und Gleichungen)
    - Lösung (letztendlich: große lineare Gleichungssysteme)
3. Implementierung:
    - Datenstrukturen, Algorithmen
    - Zielarchitektur: Monoprozessor, Parallelrechner
4. Einbettung:
    - Schnittstelle Simulation $\longleftrightarrow$ CAD
    - Schnittstelle Simulation $\longleftrightarrow$ Experimente
5. Visualisierung:
    - graphische Aufbereitung und Darstellung der Ergebnisse
6. Validierung:
    - Rückkopplung zum Experiment
    - Benchmarks

In dieser Vorlesung befassen wir uns vor allem mit obigen Schritten 2 und 3 des wissenschaftlichen Rechnens und bewegen uns im Überlappungsbereich von numerischer Mathematik und Informatik.



Literatur:

ad 1.	Wirth, Algorithmen und Datenstrukturen, Teubner
	Mehlhorn, Data and Algorithms 1-3, Springer
	deutsch: Teubner
ad 2./3./5.	Primärliteratur, Diplomarbeiten und Dissertationen, keine Bücher
ad 4.	Braess, Finite Elemente, Springer

1.2 Dogmatik des wissenschaftlichen Rechnens

(bislang) 7 Gebote:

(I) rekursiv:

- Datenstrukturen (Bäume, Geflechte)
- Algorithmen (auf Bäumen etc.)

(II) hierarchisch:

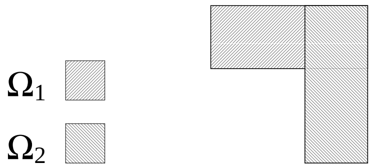
- Datenstrukturen (z. B. record, array und Baum):

```
type t    = record        s1: t1;
                          s2: t2;
                          s3: t3
            end;

type t    = array[1..10]  of real;

type tree = ↑ node;
type node = record        left_son: tree;
                          right_son: tree;
                          value: real
            end;
```

- Algorithmen (z. B. Gebietszerlegung für PDEs):



$Lu = f$ auf $\Omega = \Omega_1 \cup \Omega_2$;
löse abwechselnd auf Ω_1 und Ω_2
mit entsprechenden Randwerten

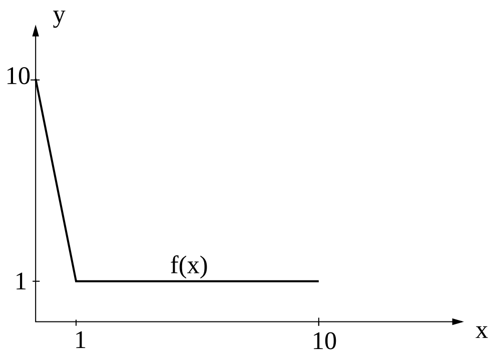
- Zahldarstellung:
  nicht hierarchisch: |||||||||||||| (Bierstriche)
  hierarchisch: 14 (Dezimalsystem)

(III) adaptiv:

- Die Auswahl eines effizienten Algorithmus' für ein numerisches Problem ist eine Optimierungsaufgabe unter einer Nebenbedingung:

$$\min_{\text{Aufwand} = \text{const}} \|u^{ex} - u^{num}\|.$$

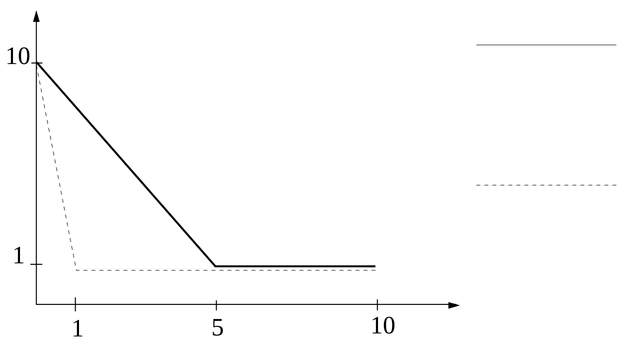
D. h.: Bei der Diskretisierung sind Punkte dort zu investieren, wo es das Problem erfordert.
Als Beispiel betrachten wir die numerische Quadratur:



$\int_0^{10} f(x)dx = 14.5$

Wir wollen drei Gitterpunkte investieren, dazwischen wird linear interpoliert. Den Näherungswert $\tilde{\int}$ für obiges Integral erhalten wir über den exakten Integralwert des Interpolanten:

$\rightarrow$ äquidistanter Fall: $x_0 = 0, x_1 = 5, x_2 = 10:$ $\tilde{\int} = 32.5$
$\rightarrow$ adaptiver Fall: $x_0 = 0, x_1 = 1, x_2 = 10:$ $\tilde{\int} = \int = 14.5$!



$\tilde{\int} = 32.5$ (äquidistant)

$\tilde{\int} = 14.5$ (adaptiv)

(IV) verteilt:

- Einsatz von Parallelrechnern
- Teamwork: Teilaufgaben auf Experten auslagern, z. B. bei Fluid-Struktur-Wechselwirkungen (Stausee-Damm)
  → nicht alles neu modellieren

(V) modular:

- Erkenntnis des Software-Engineering allgemein
- in der Numerik:
  - ⋆ PDEs:

    Geometriediskretisierung

    Gleichungsdiskretisierung

    iterativer Löser (cg, Gauß-Seidel, Mehrgitterverfahren)

    Grundbausteine aus linearer Algebra $A\,b$, $b^T\,u$

    → alles möglichst entkoppeln!
  - ⋆ Fluid-Struktur-Wechselwirkungen:
    Fluid- und Strukturteil weitmöglichst entkoppeln

(VI) interdisziplinär:

- ergibt sich aus dem Aufgabenspektrum des wissenschaftlichen Rechnens

(VII) strukturiert:

- Struktur vereinfacht Problemstellung i. d. R.; deshalb empfiehlt es sich, vorhandene Struktur auszunutzen, also beispielweise
  - ⋆ gekoppelte PDEs nicht einfach in einen Topf werfen und System lösen, sondern möglichst weit entkoppeln;
  - ⋆ nicht unnötig komplizierte Elemente verwenden (Tetraedergitter etc.);

(VIII) ???

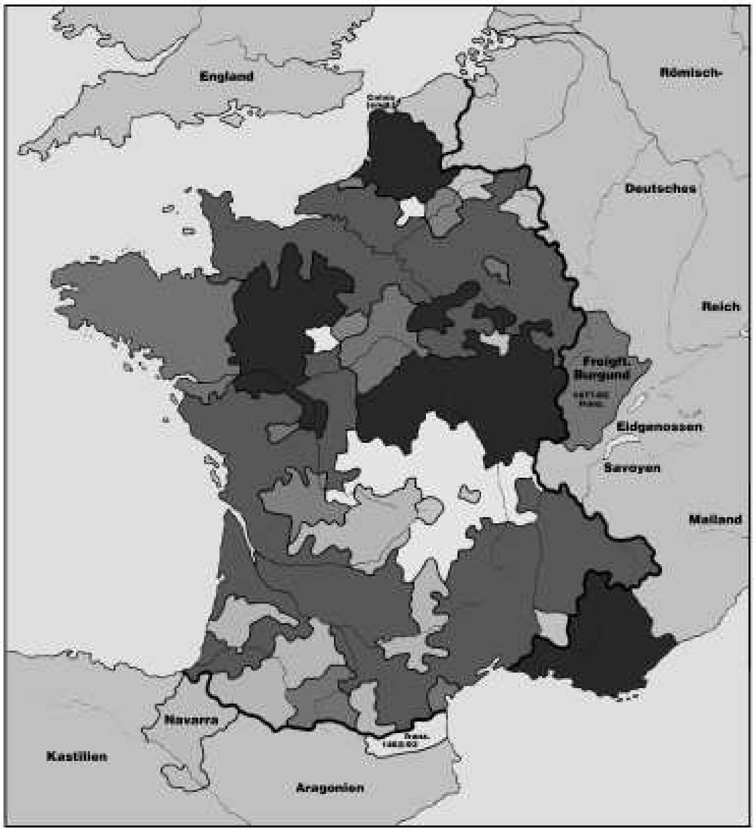
(IX) ???

(X) ???

<u>Leitgedanke des rekursiven bzw. hierarchischen Vorgehens:</u>

teile und herrsche — divide and conquer — divide et impera

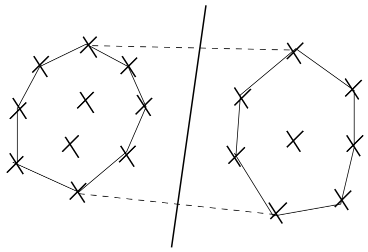
Der Ausspruch wird heute König Ludwig XI. von Frankreich zugeschrieben. Er wird oft zur Charakterisierung der Politik der „Befriedung“ im antiken Rom gebraucht, läßt sich aber erst seit der Renaissance belegen. Ludwig XI. war französischer König von 1461 bis 1483. Er festigte den noch an den Folgen des 1453 beendeten Hundertjährigen Krieges leidenden französischen Nationalstaat. Sein Hauptwidersacher hierbei war Karl der Kühne, der letzte der großen Herzöge Burgunds (Philipp der Kühne 1363–1404, Johann der Unerschrockene 1404–1419, Philipp der Gute 1419–1467, Karl der Kühne 1467–1477). Daneben war Frankreich noch von einer ganzen Reihe weiterer potentieller Feinde umgeben: Navarra, Kastilien, Aragon, England, das habsburgische Kaiserreich sowie die Eidgenossen. Stets versuchte deshalb Ludwig, unter seinen Nachbarn Zwietracht zu säen und so eine gegen Frankreich gerichtete Allianz zu verhindern: divide et impera!



Frankreich nach dem Hundertjährigen Krieg

Anwendungsbeispiele aus Informatik und Mathematik:

- Sortieren mittels mergesort:
  teile Menge in 2 Hälften, sortiere beide Hälften, mische die sortierten Teilfelder
- Berechnung der konvexen Hülle:



- Gebietszerlegung bei partiellen Differentialgleichungen
- numerische Quadratur, Spline-Interpolation:
  konstruiere Polynom mit gewünschten Eigenschaften auf jedem Teilintervall, klebe die (Teil-) Lösungen zusammen

1.3 Rekursive Datenstrukturen

i) arithmetischer Ausdruck:

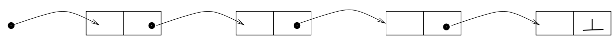
```
type  expr  =  record  op1: term;
                       op2: term;
                       op: operator
               end;
type  term  =  record  case t boolean of
                        true : (id: real)
                        false : (expression: expr)
               end;
```

Eine solche Struktur hat den Vorteil, daß die Größe der Objekte nicht a priori feststehen muß und sich im Laufe der Zeit ändern kann – die Struktur ist dynamisch.

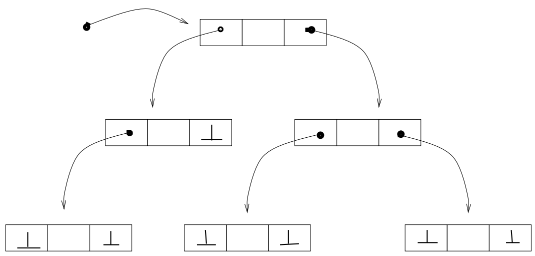
ii) Listen, Bäume und allgemeine Geflechte

- Realisierung über Zeiger (Pointer, Verweise)
- Abbruch über leeren Zeiger / nilpointer

lineare Liste:



Binärbaum:



Vor- und Nachteile von Geflechten:

+ dynamisch, geeignet für adaptive Finite-Elemente-Netze
– indirekte Adressierung: Verletzung des Lokalitätsprinzips möglich (schlecht für Parallelisierung, Speicherhierarchie (Cache))
– eingeschränkte Möglichkeit des Direktzugriffs auf Knoten
– Bäume können zu Listen degenerieren ⇒ Balancierung erforderlich (AVL-Bäume); balancierte Bäume sind wichtig etwa für die Parallelisierung (sonst ist die Last zu ungleich verteilt)

– organisatorischer Overhead (bei Tetraedernetzen z. T. einige kByte pro Knoten)

Die Nachteile sind hier vor allem aus Gründen der Fairness aufgeführt. Der Vorteil der Flexibilität wiegt natürlich sehr schwer!

iii) Alternative zu Geflechten: Hash-Tabellen

Die übliche Situation, bei der Hash- oder Streuspeicherverfahren zum Einsatz kommen, ist der Fall einer sehr großen Menge potentieller Elemente (Schlüssel), aus der aber i. a. nur wenige auftreten. Man denke etwa an Nachnamen in einer Personenkartei, an diskrete Gitterpunkte in einem Grundgebiet oder an die Nicht-Null-Einträge einer dünn besetzten Matrix. In allen Fällen wäre es irgendwo zwischen ungeschickt und unmöglich, für jedes potentielle Element einen Speicherplatz zu reservieren. Deshalb sehen wir einen statischen Speicherbereich zur Aufnahme der aktuellen Schlüssel vor und definieren eine geeignete Hash-Funktion H, die jedem Element bzw. Schlüssel seinen Platz in diesem Bereich zuordnet:

Gitterpunkt $x \in [0,1]^d \xrightarrow{H}$ Adresse i im array $A[0..k]$.

Dabei kann es natürlich vorkommen, daß zwei verschiedene Schlüssel denselben Platz zugewiesen bekommen. Da wir schließlich Platz sparen wollen, kann H sinnvollerweise nicht injektiv sein. Bei einer solchen Kollision müssen geeignete Maßnahmen ergriffen werden wie etwa die Einführung von Überlauflisten. Das kann aber nur eine gewisse Zeit gut gehen. Spätestens, wenn die Anzahl der aktuellen Schlüssel in die Nähe von k, der Größe unseres Speicherbereichs kommt, wird die Kollision zum Regelfall. Dann muß eine Neuorganisation mit neuem (d. h. größerem) k und neuem H erfolgen. Es ist klar, daß die Hash-Funktion H das Herzstück der ganzen Sache ist und mit Bedacht konstruiert werden muß.

1.4 Rekursive Algorithmen

Unter Rekursion versteht man bekanntlich die Rückführung einer gegebenen Aufgabe auf eine „einfachere“ Aufgabe desselben Typs. Diese Problemreduktion kann dabei sehr unterschiedlich aussehen.

Ein paar Beispiele:

- direkte Rekursion: Berechnung der Fakultät

  $fac(n) = n \cdot fac(n-1)$
  $fac(1) = 1$

- indirekte Rekursion: Test auf gerade/ungerade

  $iseven(n) = isodd(n-1)$
  $iseven(1) = FALSE$

  $isodd(n) = iseven(n-1)$
  $isodd(1) = TRUE$

Wann ist die Formulierung als rekursiver Algorithmus sinnvoll und wann nicht?

- egal: Fakultät

  Bei der Berechnung der Fakultät haben obige Rekursion sowie die naheliegende Iteration

$$fac := 1;\ i = 2..n:\ fac := fac \cdot i;$$

dieselbe Komplexität (sowohl im Hinblick auf das Niederschreiben als auch, was die Rechenzeit angeht).

- schädlich: Fibonacci-Zahlen ($a_0 = a_1 = 1,\ a_n = a_{n-1} + a_{n-2}$ für $n \geq 2$)

  Im rekursiven Programm

```
function fib(n: integer): integer;
  if n=0 then fib:=1 else
  if n=1 then fib:=1 else
           fib:= fib(n-1) + fib(n-2);
```

  wächst die Anzahl der Aufrufe exponentiell, es kommt zu Mehrfachaufrufen mit demselben Argument. Eine solche kaskadische Rekursion ist der einfachen Iteration

$$x_{alt} := 1;\ x_{neu} := 1;\ i = 2..n:\ x_{neu} := x_{alt} + x_{neu};\ x_{alt} := x_{neu} - x_{alt};$$

  hoffnungslos unterlegen!

- nützlich:

  Hier sind die Fälle rekursiv definierter Daten (z. B. Bäume) sowie rekursiv formulierte Aufgabenstellungen zu nennen:

  → Selbstähnlichkeit, Fraktale

  → Hilbert-Kurven zeichnen

  Ferner fallen in die Kategorie „nützlich" Problemstellungen mit starker „Eigendynamik":

  → rekursive Substrukturierung bei der Finite-Elemente-Methode in der Statik

  → adaptive Unterteilung und rekursive Substrukturierung bei Radiosity-Verfahren in der Computergraphik

  → Backtracking: finde einen Weg aus einem Labyrinth bzw. finde eine Springerzugfolge, die genau einmal über jedes Feld des Schachbretts führt

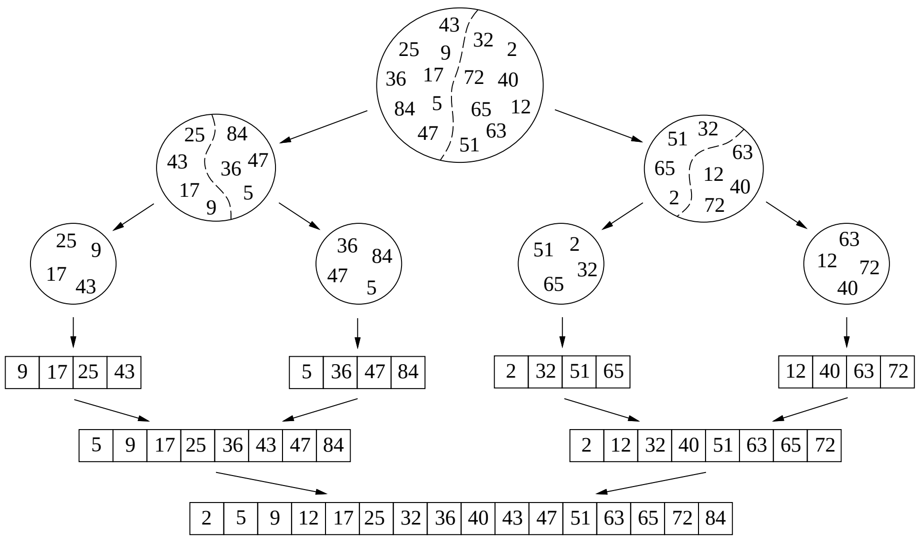
  Rekursion ist naturgemäß auch dann ein hilfreiches Prinzip, wenn die Ausgangsaufgabenstellung zu komplex ist und nicht direkt angegangen werden kann (→ divide & conquer)

  → Sortieren

Als Beispiele schauen wir uns zum Abschluß dieses Kapitels zwei wohlbekannte rekursive Sortieralgorithmen an, mergesort und quicksort.

procedure mergesort(S);
$n := |S|$;
$S =: S_1 \cup S_2, S_1 \cap S_2 = \emptyset, \big|\ |S_1| - |S_2|\ \big| \leq 1;$
mergesort(S_1);
mergesort(S_2);
mische die beiden Ergebnisfolgen;

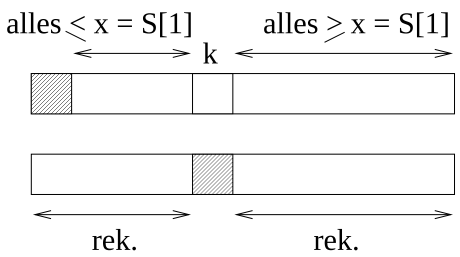
Komplexität: worst case und average $O(n \log n)$



Mergesort

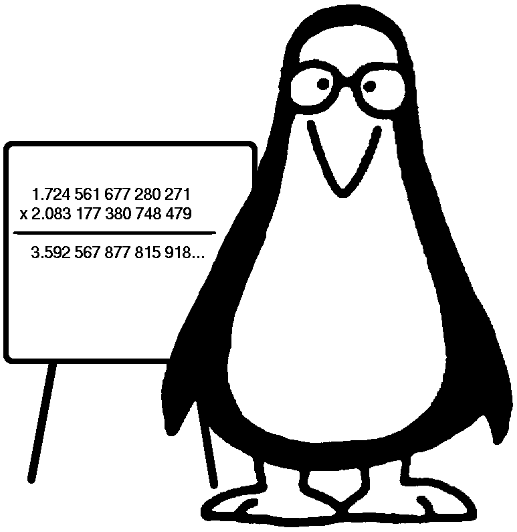
```
procedure quicksort(S,l,r);
          i := l;
          k := r+1;
          x := S[l];
          while i < k
              repeat i +=1 until S[i] ≥ x;
              repeat k −=1 until S[k] ≤ x;
              if k > i then vertausche (S[k],S[i]);
          vertausche (S[l],S[k]);
          if l < k-1 then quicksort(l,k-1);
          if k+1 < r then quicksort(k+1,r);
```

Der Startaufruf lautet quicksort(S,1,n).



Komplexität: worst case O(n^2), average O($n \log n$)

Bei einer worst case Betrachtung schneidet mergesort also besser ab. Daß quicksort dennoch von großer Bedeutung ist, liegt daran, daß es im mittleren Fall mit einer besseren Konstante vor dem Ordnungsterm $n \log n$ aufwarten kann.



Leider war der Autor zu XXXX, um hier noch weiter zu schreiben.

Kapitel 2

Numerische Quadratur nach Archimedes

In diesem Kapitel wollen wir uns mit der numerischen Quadratur, also der numerischen Berechnung des Integrals einer gegebenen Funktion f von d Veränderlichen, befassen. Wir folgen dabei einem sehr alten divide-et-impera-Prinzip, der Archimedischen Ausschöpfung. Für den Übergang vom eindimensionalen zum höherdimensionalen Fall stützen wir uns auf das Cavalierische Prinzip.

2.1 Der eindimensionale Fall

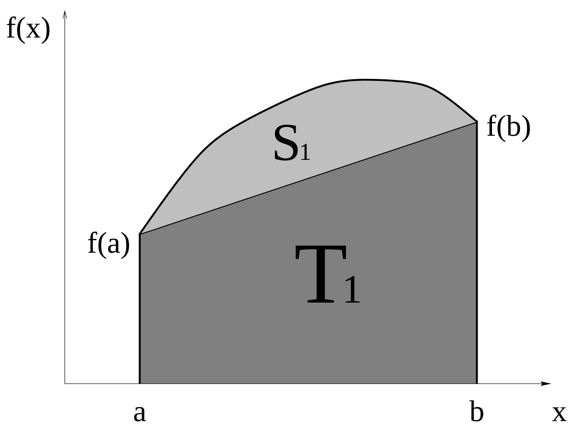
Berechnet werden soll das Integral

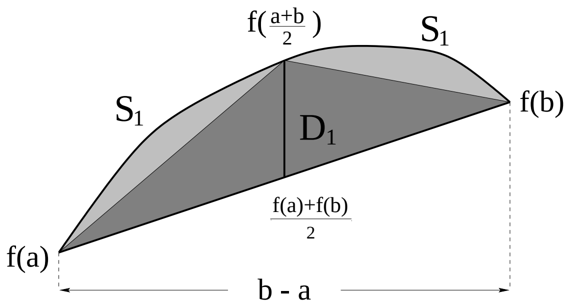
$$F_1(f(x), a, b) \ := \int_a^b f(x)\, dx \ .$$

Hierzu wird F_1 aufgespalten in ein Trapez T_1 und in ein Restsegment S_1:

$$F_1(f(x), a, b) \ = \ \underbrace{\frac{b-a}{2} \cdot (f(a) + f(b))}_{T_1} \ + \ S_1(f(x), a, b) \ .$$

S_1 wird zunächst durch ein einbeschriebenes Dreieck D_1 angenähert. Dann wird rekursiv fortgeschaltet:





$$S_1(f(x),a,b) \;=\; \underbrace{\left(f\left(\frac{a+b}{2}\right)-\frac{f(a)+f(b)}{2}\right)\cdot\frac{b-a}{2}}_{D_1} + S_1\left(f(x),a,\frac{a+b}{2}\right) + S_1\left(f(x),\frac{a+b}{2},b\right).$$

Die Formel für D_1 ergibt sich aus der Beziehung

$$\text{Fläche}_\triangle = \frac{1}{2}\,\text{Grundseite}\;\cdot\text{Höhe},$$

also für das linke und rechte Teildreieck von D_1 jeweils

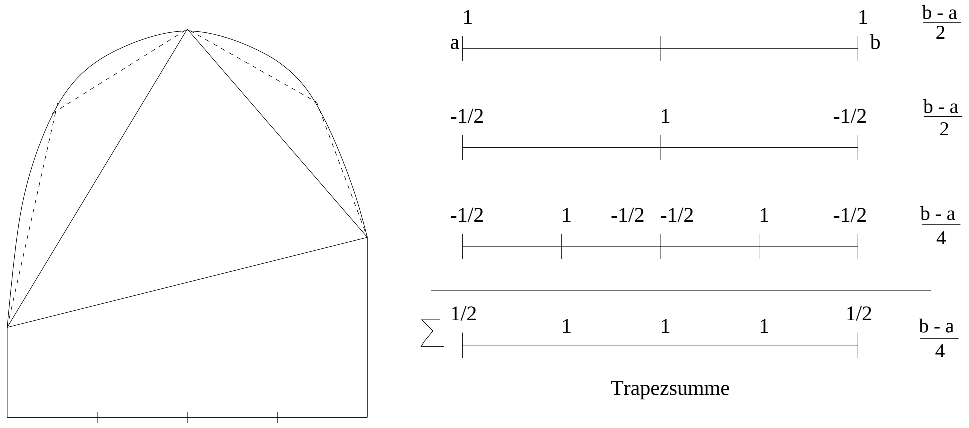
$$\frac{1}{2}\cdot\underbrace{\left(f\left(\frac{a+b}{2}\right)-\frac{f(a)+f(b)}{2}\right)}_{g}\cdot\underbrace{\frac{b-a}{2}}_{h}.$$

Die Rekursion erfordert ein Abbruchkriterium. Hier gibt es mehrere Möglichkeiten:

(i) feste Rekursionstiefe t:
Bei $t=1$ lautet der Abbruchfall

$$S_1(f(x),a,b) \;:=\; \left(f\left(\frac{a+b}{2}\right)-\frac{f(a)+f(b)}{2}\right)\cdot\frac{b-a}{2} \;=\; D_1(f(x),a,b)\;,$$

was gerade der bekannten Trapezsumme mit drei äquidistanten Stützstellen a, $(a+b)/2$ und b und der Maschenweite $h=(b-a)/2$ entspricht. Im Fall einer allgemeinen Tiefe t erhält man die Trapezsumme mit 2^t+1 äquidistanten Stützstellen und der Maschenweite $h=(b-a)/2^t$:



(ii) adaptiv: bis Erreichen einer vorgegebenen Genauigkeit ε:
Hier haben wir zwei Möglichkeiten. Man kann erstens abbrechen, wenn der Flächeninhalt des lokalen Dreiecks D_1 unter ε fällt, also

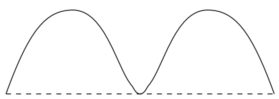
$$S_1 \;:=\; D_1 \qquad \text{falls} \qquad \left| f\left(\frac{a+b}{2}\right) - \frac{f(a)+f(b)}{2} \right| \cdot \frac{b-a}{2} \;<\; \varepsilon,$$

oder man bricht die Rekursion mit $S_1 := D_1$ ab, falls

$$\left| f\left(\frac{a+b}{2}\right) - \frac{f(a)+f(b)}{2} \right| \;<\; \varepsilon$$

erfüllt ist. Jetzt ist der Flächeninhalt des lokalen Dreiecks D_1 kleiner als $\varepsilon \cdot h$. Falls an allen Stellen auf derselben Tiefe t abgebrochen wird, bedeutet dies, daß ε jetzt eine obere Schranke für den Fehler ist, der auf der ganzen Stufe t entsteht (die Grundseiten der Dreiecke einer Tiefe t überdecken das Ausgangsgebiet nach Konstruktion gerade).

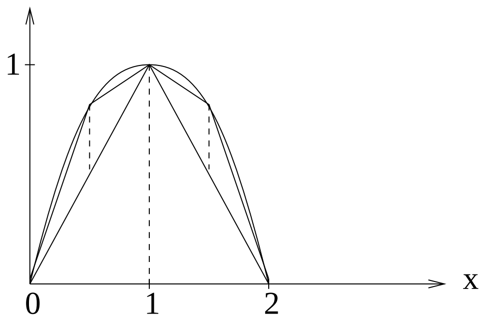
Im ersten Fall ist ε also ein Maß für den lokalen Fehler, im zweiten Fall – obwohl lokal gemessen – ein Maß für den globalen Fehler. Eine in beiden Fällen auftretende Problematik ist offensichtlich: Die Dreiecksfläche kann sich zu Null ergeben, obwohl das Integral nicht verschwindet. Hier kann man etwas Abhilfe schaffen, indem man erst abbricht, wenn die Abbruchbedingung auf mehreren Stufen hintereinander erfüllt ist!



$f(\frac{a+b}{2}) - \frac{f(a)+f(b)}{2} = 0$,
aber Integral nicht klein

Beispiel der Parabel:

$f(x) = x(2-x)$
$\int_0^2 f(x)dx = \frac{4}{3}$



Rekursions-tiefe	„Höhe“ der Dreiecke	Fläche der Dreiecke	Anzahl der Dreiecke	Summe dieser Tiefe zusammen
1	1	1	1	1
2	$\frac{1}{4}$	$\frac{1}{8}$	2	$\frac{1}{4}$
3	$\frac{1}{16}$	$\frac{1}{64}$	4	$\frac{1}{16}$
4	$\frac{1}{64}$	$\frac{1}{512}$	8	$\frac{1}{64}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
k	$\frac{1}{4^{k-1}}$	$\frac{1}{8^{k-1}}$	2^{k-1}	$\frac{1}{4^{k-1}}$

$$\sum = \sum_{i=0}^{\infty} 4^{-i} = \frac{4}{3}$$

Bei Tiefe 2 gibt es zwei Dreiecke, eines auf $[0,1]$ und eines auf $[1,2]$. Um die jeweilige „Höhe“ zu berechnen, subtrahieren wir von der Parabel (also von $x(2-x)$) die Funktion des Dreiecks der Tiefe 1, d. h. x auf $[0,1]$ und $2-x$ auf $[1,2]$:

auf [0,1] : $x(2-x) - x = x(1-x)$;
Wert in $x = \frac{1}{2}$ ist $\frac{1}{4}$, als Fläche ergibt sich $\frac{1}{8}$;

auf [1,2] : $x(2-x) - (2-x) = (x-1)(2-x)$;
Wert in $x = \frac{3}{2}$ ist $\frac{1}{4}$, als Fläche ergibt sich $\frac{1}{8}$.

Somit ergibt sich als Summe aller Flächeninhalte der Dreiecke der Tiefe 2 der Wert $\frac{1}{4}$.

Einfache Erhöhung der Ordnung:

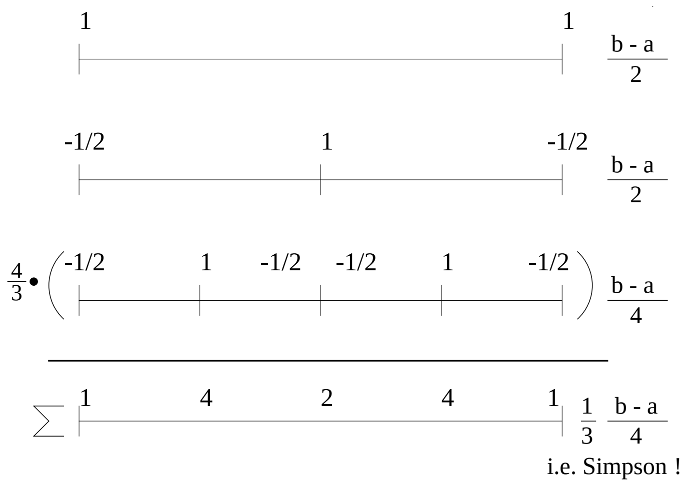
- bisheriges Vorgehen integriert lineare Funktionen exakt (Trapezsumme: Polygonzug)
- multipliziere die Dreiecksflächen auf letzter Stufe t mit $\frac{4}{3}$:

$$\left.\begin{array}{lll} t=4: & 1+\frac{1}{4}+\frac{1}{16}+\frac{1}{64}\cdot\frac{4}{3} & =\frac{4}{3} \\ t=3: & 1+\frac{1}{4}+\frac{1}{16}\cdot\frac{4}{3} & =\frac{4}{3} \\ t=2: & 1+\frac{1}{4}\cdot\frac{4}{3} & =\frac{4}{3} \\ t=1: & 1\cdot\frac{4}{3} & =\frac{4}{3} \end{array}\right\} \text{alles exakt!}$$

Die jetzige Vorgehensweise entspricht nicht mehr der Trapezregel bzw. Trapezsumme, sondern der Faßregel bzw. der Quadratur nach Simpson:

$$(b-a)\left[\frac{1}{6}f(a) + \frac{4}{6}f(\frac{a+b}{2}) + \frac{1}{6}f(b)\right]$$

Somit werden jetzt alle Polynome vom Grad 2 korrekt integriert. Wir schauen uns die neuen Gewichte genau an und sehen, daß sich in der Tat die Simpsonsche Quadraturformel ergibt:



Beispiel: $f(x) = e^x$ auf $[0,1]$

ε	# Dreiecke ($\lvert D_1\rvert < \varepsilon$)		$\lvert$Fehler(Trapez)$\rvert$		$\lvert$Fehler(Simpson)$\rvert$	
8^{-2}	5	$)\cdot 3.8$	$4.77 \cdot 10^{-3}$	$) : 11.6$	$5.79 \cdot 10^{-4}$	$) : 371.2$
8^{-4}	19	$)\cdot 4.4$	$4.13 \cdot 10^{-4}$	$) : 17.4$	$1.56 \cdot 10^{-6}$	$) : 284.7$
8^{-6}	83	$)\cdot 4.0$	$2.38 \cdot 10^{-5}$	$) : 15.9$	$5.48 \cdot 10^{-9}$	$) : 256.0$
8^{-8}	333	$)\cdot 4.0$	$1.50 \cdot 10^{-6}$	$) : 16.0$	$2.14 \cdot 10^{-11}$	$) : 253.6$
8^{-10}	1337	$)\cdot 4.0$	$9.38 \cdot 10^{-8}$	$) : 16.0$	$8.44 \cdot 10^{-14}$	$) : 190.1$
8^{-12}	5351	$)\cdot 4.0$	$5.86 \cdot 10^{-9}$	$) : 16.0$	$4.44 \cdot 10^{-16}$	
8^{-14}	21411		$3.66 \cdot 10^{-10}$			
	$\times 4$ entspricht $h : 4 \;\Rightarrow$		$O(h^2)$		$O(h^4)$	

Rekursives perl-Programm:

```
$integral = ar1 (0,1,0.001);
sub f   { my ($x) = @_;
    2 + $x * (1-$x);
 }
sub ar1 { my ($a,$b,$eps) = @_;
      (f($a)+f($b)) * ($b-$a)/2 + s1($a,$b,$eps);
 }
sub s1  { my($a,$b,$eps) = @_;
    my ($d);
    $d = (f(($a+$b)/2) - (f($a)+f($b))/2) * ($b-$a)/2;
    if (abs($d) ⟨ $eps) { $d * 4/3; }
    else  { $d + s1($a,($a+$b)/2,$eps) + s1(($a+$b)/2,$b,$eps);}
}
```

Rekursives Programm in PASCAL-ähnlicher Notation:

```
function ar1 (a,b,eps: real): real;
   begin
      ar1 := (f(a)+f(b)) * (b–a)/2 + s1(a,b,eps);
   end;

function s1 (a,b,eps: real): real;
   var d: real;
   begin
      d := (f((a+b)/2) – (f(a)+f(b)/2) * (b-a);
      if (abs(d) ⟨ eps);
         then s1 := d * 4/3
         else s1 := d + s1(a,(a+b)/2,eps) + s1((a+b)/2,b,eps);
   end;
```

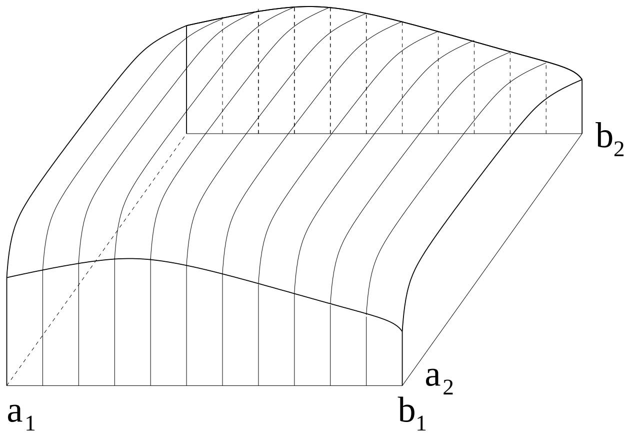
Anmerkung:

In beiden Programmen wird f wird an vielen Stellen mehrfach ausgewertet. Dies kann vermieden werden, wenn man zu a und b auch f(a) und f(b) übergibt. Dann wird immer nur $f(\frac{a+b}{2})$ neu ausgewertet.

2.2 Der mehrdimensionale Fall

2.2.1 Der zweidimensionale Fall

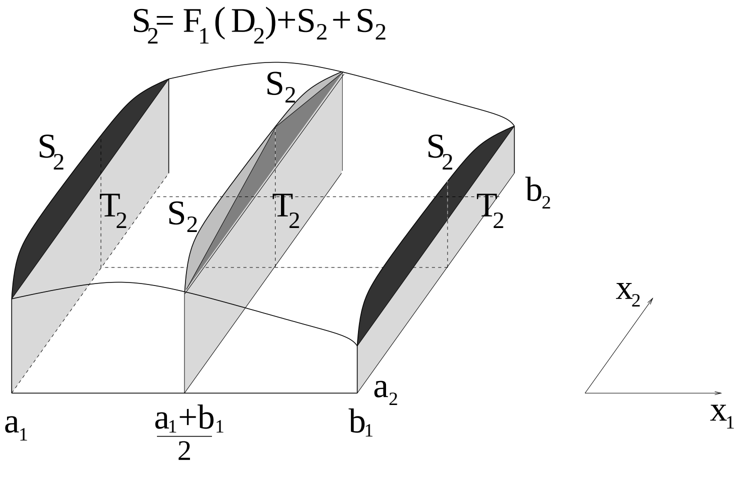
Übertragung der Archimedischen Ausschöpfung ins Zweidimensionale mittels des „Cavalierischen Prinzips": Zerlege Volumen in gleichmäßig dünne Scheiben, berechne die Fläche der Scheiben, summiere die Flächen auf und multipliziere mit der Scheibendicke.



Dies entspricht der Vorgehensweise im Satz von Fubini:

$$\int\limits_{\Omega^{(d)}} f(x_1,...,x_d)\; d(x_1,...,x_d) \;=\; \int\limits_{a_d}^{b_d} \Big(\int\limits_{\Omega^{(d-1)}(x_d)} f(x_1,...,x_d)\; d(x_1,...,x_{d-1}) \Big)\; dx_d$$

mit $\Omega^{(d-1)}(x_d) = \left\{ \xi \in \Omega^{(d)} : \xi_d = x_d \right\}$.



Berechnet werden soll das zweidimensionale Integral

$$F_2(f(x_1,x_2),a_1,a_2,b_1,b_2) \;:=\; \int\limits_{a_1}^{b_1}\int\limits_{a_2}^{b_2} f(x_1,x_2)\; dx_1 dx_2\;.$$

Analog zum eindimensionalen Fall zuvor spalten wir wieder auf:

$$F_2(f(x_1,x_2),a_1,a_2,b_1,b_2) \;=\; F_1(T_2(x_1),a_1,b_1) \;+\; S_2(f(x_1,x_2),a_1,a_2,b_1,b_2)\ ,$$

wobei

$$T_2(x_1) \;:=\; \underbrace{\frac{b_2-a_2}{2}\cdot(f(x_1,a_2)+f(x_1,b_2))}_{\text{Trapez als Funktion von } x_1}$$

und

$$\begin{aligned} S_2(f(x_1,x_2),a_1,a_2,b_1,b_2) \;&:=\; F_1(D_2(x_1),a_1,b_1) \\ &\quad+\; S_2(f(x_1,x_2),a_1,a_2,b_1,\frac{a_2+b_2}{2}) \\ &\quad+\; S_2(f(x_1,x_2),a_1,\frac{a_2+b_2}{2},b_1,b_2) \end{aligned}$$

mit

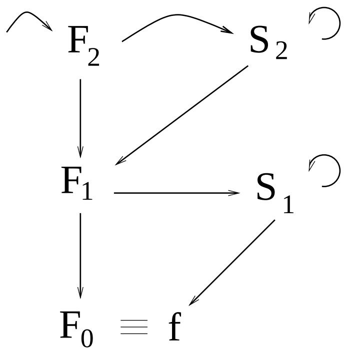
$$D_2(x_1) \;:=\; \frac{b_2-a_2}{2}\cdot\left(f\left(x_1,\frac{a_2+b_2}{2}\right)-\frac{f(x_1,a_2)+f(x_1,b_2)}{2}\right)\ .$$

Wir brechen ab, wenn der durch Aneinanderkleben der Dreiecke D_2 in x_1-Richtung entstehende Körper ein Volumen $\leq \varepsilon$ hat:

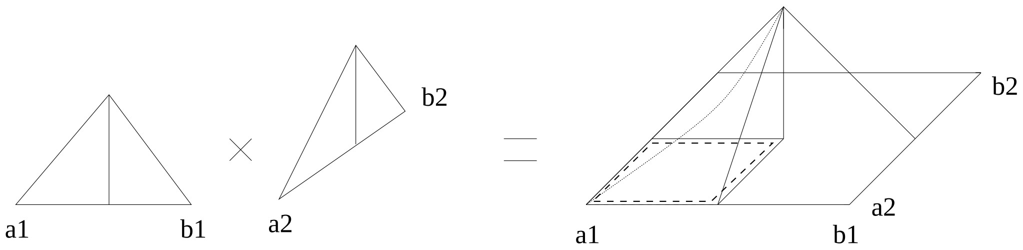
$$\begin{aligned} S_2(f(x_1,x_2),a_1,a_2,b_1,b_2) \;&:=\; F_1(D_2(x_1),a_1,b_1)\ , \\ &\quad\text{falls}\quad |F_1(D_2(x_1),a_1,b_1)| \;\leq\; \varepsilon\ . \end{aligned}$$

Der Körper, dessen Volumen zu berechnen ist, wird also unterteilt in einen Basiskörper (Aneinanderkleben in x_1- Richtung aller x_2-Trapeze) und einen Deckel (Aneinanderkleben in x_1- Richtung aller x_2-Hauben). Der Deckel wird dann rekursiv unterteilt in einen Dreieckskörper und zwei neue (halb so breite) Deckel.

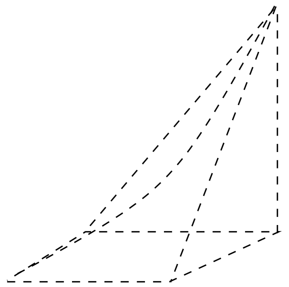
Aufrufhierarchie der Rekursion:



Folglich wird ein dreidimensionaler Körper durch eine Folge von Grundvolumen approximiert, das Integral F_2 durch Aufsummieren deren Volumen. Im Eindimensionalen waren die Grundvolumen Dreiecke. Welche Gestalt haben sie jetzt? Offensichtlich handelt es sich um „Produkte“ zweier Dreiecke in x_1- und x_2-Richtung:



„bilineare Funktionen“: Linearkombinationen von 1, x_1, x_2 und $x_1 \cdot x_2$
auf



etwa vom Typ $x_1 \cdot x_2$

<u>Beispiel:</u> $f(x) = 2 + x(1-x)y(1-y)$ auf $[0,1]^2$

ε	# Pagoden		Fehler	
8^{-2}	6		$4.34 \cdot 10^{-3}$	
		$) \cdot 8.3$		$)\ : 10.0$
8^{-4}	50		$4.34 \cdot 10^{-4}$	
		$) \cdot 6.4$		$)\ : 11.6$
8^{-6}	322		$3.73 \cdot 10^{-5}$	
		$) \cdot 5.6$		$)\ : 12.6$
8^{-8}	1794		$2.97 \cdot 10^{-6}$	
		$) \cdot 5.1$		$)\ : 13.2$
8^{-10}	9218		$2.25 \cdot 10^{-7}$	
		$) \cdot 4.9$		$)\ : 13.6$
8^{-12}	45058		$1.66 \cdot 10^{-8}$	
		$) \cdot 4.7$		$)\ : 13.9$
8^{-14}	212994		$1.19 \cdot 10^{-9}$	
		$) \cdot 4.6$		$)\ : 14.2$
8^{-16}	982042		$8.41 \cdot 10^{-11}$	

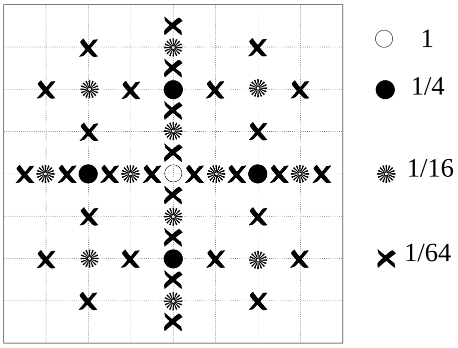
1D Archimedes:
$\# \triangle \cdot 2 \Rightarrow \frac{\text{Fehler}}{4}$

2D Archimedes:
$\# \text{Pagoden} \cdot 4 \Rightarrow \frac{\text{Fehler}}{16}$

<u>Effizienzgewinn gegenüber der zweidimensionalen Trapezsumme:</u>

Maschenweite $h = 2^{-t}$ (d. h. Tiefe t)	Genauigkeit Ordnung	Genauigkeit $t = 10$	# Pagoden Ordnung	# Pagoden $t = 10$
2D Trapezsumme	$O(h^2)$	10^{-6}	$O(h^{-2})$	10^6
2D Archimedes	$O(h^2 \lvert \log_2 h \rvert)$	10^{-5}	$O(h^{-1} \lvert \log_2 h \rvert)$	10^4

Beispiel: $f(x) = 16 \cdot x(1-x)y(1-y)$



Zur Erzielung einer Genauigkeit ε reichen weniger Pagoden/Punkte als bei der zweidimensionalen Trapezsumme aus (volles $N \times N$ Gitter).

Rekursives perl-Programm:

```
$integral = ar2(t2,d2,0,0,1,1,$eps);

sub f  { my ($x) = @_;   exp($x+$y);}

sub s1 { my ($f,$a,$b,$eps,$l,$r) = @_;
    my ($d);
    $d = (&$f(($a+$b)/2,$l,$r) - (&$f($a,$l,$r)+&$f($b,$l,$r))/2) * ($b-$a)/2;

    if (abs($d) ⟨ $eps)  { $d;}
    else { $d + s1($f,$a,($a+$b)/2,$eps,$l,$r) + s1($f,($a+$b)/2,$b,$eps,$l,$r);}
 }

sub ar1 { my ($f,$a,$b,$eps,$l,$r) = @_;
      (&$f($a,$l,$r)+&$f($b,$l,$r)) * ($b-$a)/2 + s1($f,$a,$b,$eps,$l,$r);
 }

sub t2 { my ($x,$l,$r) = @_;
     (f($x,$l)+f($x,$r)) * ($r-$l)/2;
 }

sub d2 { my ($x,$l,$r) = @_;
     (f($x,($l+$r)/2) - (f($x,$l)+f($x,$r))/2) * ($r-$l)/2;
 }

sub s2 { my ($f, $a1,$a2,$b1,$b2,$eps) = @_;
     my ($d);
     $d = ar1($f, $a1,$b1,$eps,$a2,$b2);
     if (abs($d) ⟨ $eps)  { $d;}
     else { $d + s2($f,$a1,$a2,$b1,($a2+$b2)/2,$eps);
                 +  s2($f,$a1,($a2+$b2)/2,$b1,$b2,$eps); }

 }

 sub ar2 {  my ($ft,$fd,$a1,$a2,$b1,$b2,$eps) = @_;
     ar1($ft,$a1,$b1,$eps,$a2,$b2) + s2($fd,$a1,$a2,$b1,$b2,$eps);
 }
```

zur Erinnerung:

$$
\begin{array}{llll}
F_2 & \text{„=“} & F_1(T_2) \; + \; S_2 & \hat{=} \text{ ar2} \\
S_2 & \text{„=“} & F_1(D_2) \; + \; S_2 \; + S_2 & \hat{=} \text{ s2} \\
 & & & \\
F_1 & \text{„=“} & T_1 \; + \; S_1 & \hat{=} \text{ ar1} \\
S_1 & \text{„=“} & D_1 \; + \; S_1 \; + \; S_1 & \hat{=} \text{ s1}
\end{array}
$$

2.2.2 Allgemeine Dimensionalität d

Hier geht's ganz analog mit steigendem Effizienzgewinn gegenüber der Trapezsumme. Es ist möglich, den d-dimensionalen Archimedes dimensionsunabhängig zu programmieren, d. h. mit einer festen Anzahl von Prozeduren und d als Parameter. Solche Programme werden wir später betrachten.

2.3 Erhöhung der Genauigkeit durch Extrapolation

Wir haben bereits gesehen, daß die eindimensionale Quadratur nach Archimedes im Ergebnis gerade der Trapezsumme $T(h; f)$ entspricht:

$$T(h; f) \; := \; h \cdot \left(\frac{1}{2} f(a) + f(a+h) + f(a+2h) + \cdots + f(b-h) + \frac{1}{2} f(b) \right)$$

mit $h = \frac{b-a}{n}$. Für die Trapezsumme gilt aber bekanntlich die Euler-Maclaurinsche Summenformel:

$$T(h; f) \; = \; I(f) + \tau_1(f) \cdot h^2 + \tau_2(f) \cdot h^4 + \ldots + \tau_k(f) \cdot h^{2k} + O(h^{2k+2}),$$

falls $f \in \mathcal{C}^{2k+1}([a, b])$. Hierbei bezeichnet $I(f)$ den exakten Integralwert. Das bedeutet, daß $T(h; f)$ eine Approximation der Ordnung $O(h^2)$ darstellt. Dies kann man sich zunutze machen:

$$
\begin{array}{rlll}
 & T(h) & = \; \tau_0 + \tau_1 \cdot h^2 + \tau_2 \cdot h^4 + \ldots & | \cdot (-\frac{1}{3}) \\
+ & T(\frac{h}{2}) & = = \; \tau_0 + \tau_1 \cdot \frac{h^2}{4} + \tau_2 \cdot \frac{h^4}{16} + \ldots & | \cdot (+\frac{4}{3}) \\
\hline
 & T'(h) & = \; \frac{4}{3} T(\frac{h}{2}) - \frac{1}{3} T(h) & \\
 & & = \; \tau_0 + \tau_2' \cdot h^4 + \ldots . &
\end{array}
$$

D. h.,

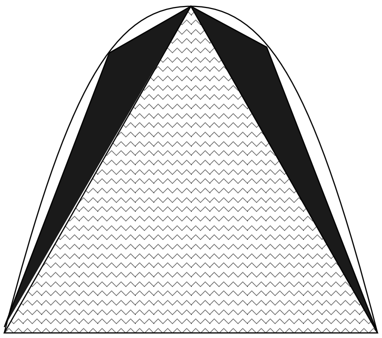
$$T'(h) \; = \; \tau_0 + O(h^4) \; = \; I(f) + O(h^4) \; .$$

Nimmt man $T(\frac{h}{4})$ hinzu, erhält man $T''(h) = I(f) + O(h^6)$. In jedem solchen Approximationsschritt erhöht sich also die Approximationsordnung um zwei h-Potenzen.

Diese Erhöhung der Genauigkeit deuten wir jetzt im hierarchischen Sinne um:

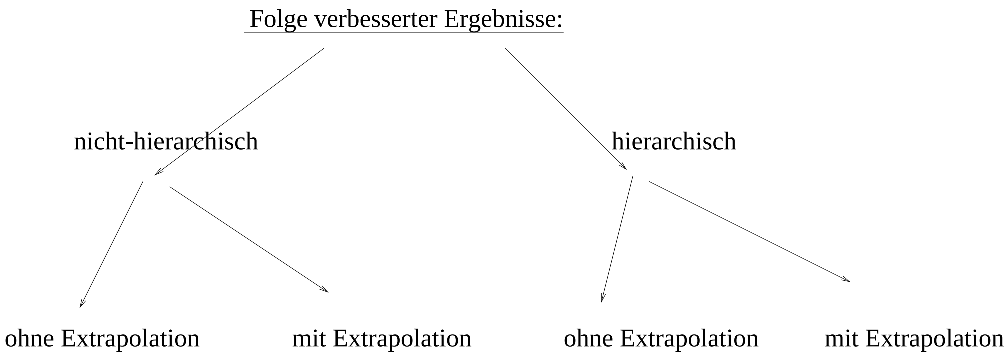
$$T'(h) = \frac{4}{3} \cdot T\left(\frac{h}{2}\right) - \frac{1}{3} \cdot T(h) \; = \; T(h) + \frac{4}{3} \cdot \left(T\left(\frac{h}{2}\right) - T(h) \right) \; .$$

Dabei ist $T(\frac{h}{2}) - T(h)$ gerade der hierarchische Beitrag des neuen Levels mit Maschenweite $h/2$. Wir wissen bereits, daß im Falle einer quadratischen Funktion $ax^2 + bx + c$ das Multiplizieren dieses Beitrags mit $4/3$ zum exakten Ergebnis führt. Im sonstigen (glatten) Fall wird die Approximation besser und ist jetzt von der Ordnung $O(h^4)$.



$T(\frac{b-a}{2})$

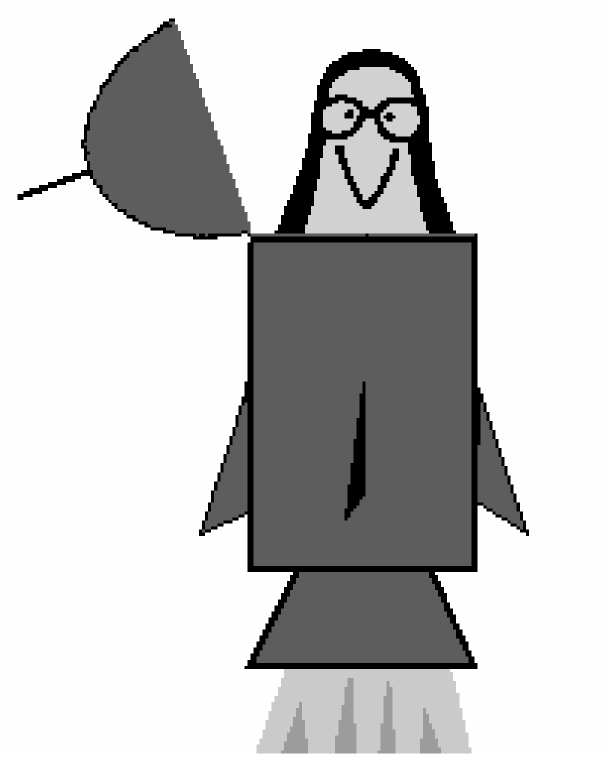
$T(\frac{b-a}{4}) - T(\frac{b-a}{2})$



$T(h)$	$T(h)$	$T(h)$		$T(h)$
$T(\frac{h}{2})$	$T'(h)$	$T(\frac{h}{2}) - T(h)$	$\}$ Inkremente	$T'(h) - T(h)$
$T(\frac{h}{4})$	$T''(h)$	$T(\frac{h}{4}) - T(\frac{h}{2})$	$\}$	$T''(h) - T'(h)$
$T(\frac{h}{8})$	$T'''(h)$	$\vdots$		$\vdots$
$\vdots$	$\vdots$	$\vdots$		$\vdots$
$\vdots$	$\vdots$	$\vdots$		$\vdots$
$\downarrow$	$\downarrow$	$\downarrow$		$\downarrow$
τ_0	τ_0	0		0

Als wesentlichen Vorteil der hierarchischen Vorgehungsweise halten wir fest, daß wir hier über ein natürliches Abbruchkriterium ohne Kenntnis der exakten Lösung verfügen. Man bricht ab, wenn der Beitrag eines Levels unter eine bestimmte Schranke fällt.

Natürlich stellt die Extrapolation erhöhte Glattheitsanforderungen an die Funktion f. Dementsprechend hilft Extrapolation in Gebieten, in denen f glatt ist. Wo dies nicht der Fall ist, muß das Gitter adaptiv verfeinert werden, um eine größere Genauigkeit zu erhalten.



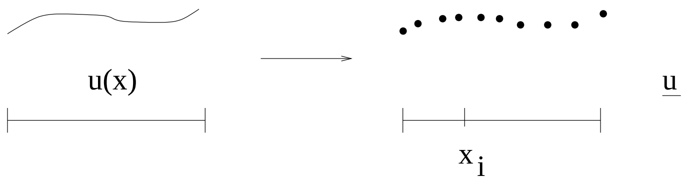
Leider war der Autor zu XXXX, um hier noch weiter zu schreiben.

Kapitel 3

Funktionsdarstellung und Interpolation

3.1 Hierarchische Basen

- Unsere Aufgabe ist es nun, eine kontinuierlich gegebene Funktion $u : [0,1] \longrightarrow \mathbb{R}$ im Rechner, d. h. diskret darzustellen. Dies erfolgt üblicherweise mittels eines Vektors $\underline{u} \in \mathbb{R}^n$, dessen Koeffizienten u_i gerade die Funktionswerte $u(x_i)$ an n Stützstellen $x_i \in [0,1],\ 1 \le i \le n$, repräsentieren.

  Offensichtlich ist zur näherungsweisen Darstellung bzw. Rekonstruktion abhängig von der Glattheit von u eine gewisse Mindestzahl von diskreten Punkten erforderlich. (Shannonsches Abtasttheorem).

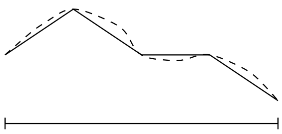
  Manchmal sind die diskreten Samples auch der Ausgangspunkt, d.h., eine geschlossene kontinuierliche Darstellung von u liegt gar nicht vor.

- Für viele Anwendungen ist auch die Auswertung zwischen einzelnen Stützstellen gefragt (Interpolation, numerische Quadratur als exakte Integration eines Interpolanten, Lösung einer partiellen Differentialgleichung etc.).

  Deshalb konstruieren wir einen Interpolanten $u^I(x) : [0,1] \to \mathbb{R}$ zu u bzgl. der Stützstellen x_i, also

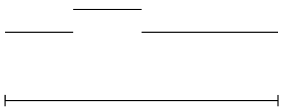
$$u^I(x_i) = u(x_i),\ 1 \le i \le n.$$

Einige einfache Beispiele:



stückweise linear

kubischer Spline

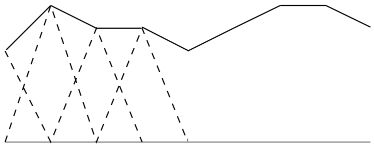


stückweise konstant

Allgemein bezeichnen wir das Grundgebiet (hier das Intervall $[0,1]$) mit Ω, und $\Omega^n := \{x_i \in \Omega, i = 1, \ldots, n\}$ ist die Menge der Stützstellen in Ω, das Gitter. Die Stützstellen x_i können äquidistant sein, müssen dies aber nicht.

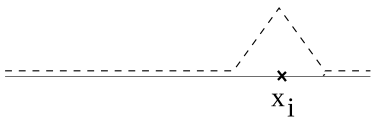
- Man kann den Interpolanten u^I auch über eine Linearkombination von Basisfunktionen des zugehörigen diskreten endlich-dimensionalen Vektorraums definieren, also etwa des Raums der bzgl. Ω^n stückweise linearen Funktionen. Einige Beispiele:

α) Standard-Hutfunktionen



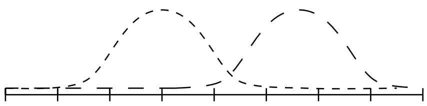
$u^I(x) = \sum_{i=1}^{n} u_i \cdot \varphi_i(x)$

φ_i : Basisfunktion zu x_i



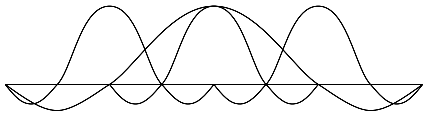
$\Rightarrow$ Raum der stückweise linearen Funktionen auf Ω bzw. Ω^n

β) B-Spline-Basis



$\Rightarrow$ Raum der stückweisen Polynome vom Grad k, $\in \mathcal{C}^{k-1}$ auf Ω bzw. Ω^n

γ) sonstige: Wavelet-Basis etc.



$\Rightarrow$ Raum hängt ab vom Typ der verwendeten Wavelets

Somit haben wir nur 3 Darstellungen für u:

$$\begin{array}{ccccc} u(x) & \longleftrightarrow & u^I(x) = \sum u_i \cdot \varphi_i(x) & \longleftrightarrow & \{u_i\} \hat{=} \underline{u} \\ \text{kontinuierlich} & & \text{Interpolant} & & \text{diskret} \\ & & & u^I \equiv \underline{u} & \end{array}$$

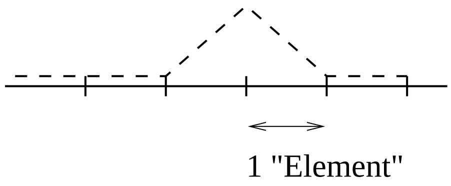
Hierbei werden oft u^I und $\underline{u}$ identifiziert.

- Insbesondere für die numerische Lösung partieller Differentialgleichungen mittels der Finite-Elemente-Methode (siehe Kapitel 4) sind dabei die stückweise linearen Funktionen bzw. d-dimensionale Verallgemeinerungen davon von großer Bedeutung. Stückweise lineare Basisfunktionen sind einfach zu definieren:

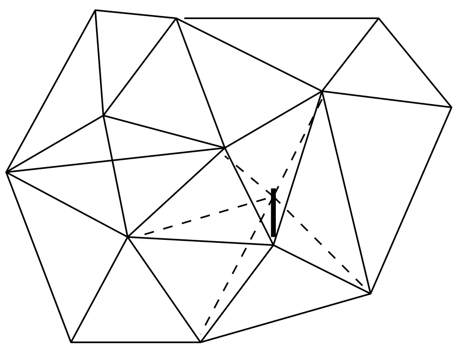
  → 1D: Hier ist die sogenannte „Hutfunktion" der Platzhirsch:

  $$\varphi_i(x) = \begin{cases} 1 & : \ x_i \\ 0 & : \ x_j \neq x_i \end{cases}, \qquad \text{linear dazwischen.}$$

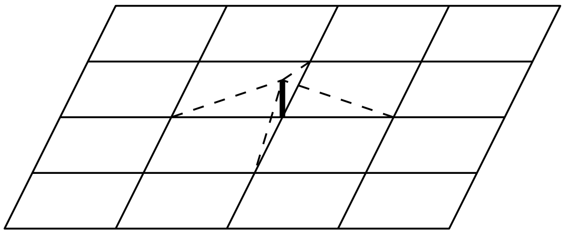
  Die Gesamtheit aller Hutfunktionen im Gitter Ω^n spannt offensichtlich den Raum der bzgl. Ω^n stückweise linearen Funktionen auf. Ein finites Element ist hier ein Teilintervall zwischen zwei benachbarten Stützstellen. Man fühle sich angesichts der Hutform an die Dreiecke bei der Quadratur nach Archimedes in Kapitel 2 erinnert!

  → 2D: Die erste Möglichkeit einer Verallgemeinerung auf den zweidimensionalen Fall führt auf dreieckige Elemente. Auf jedem Element hat man drei natürliche Freiheitsgrade (die Funktionswerte in den Eckpunkten). Mit den (elementlokalen) Basisfunktionen 1, x und y kann man auf jedem Element also eine beliebige lineare Funktion darstellen. Als (globale) Basisfunktionen φ_i verwendet man wieder die stückweise linearen Funktionen, die in allen bis auf einen Gitterpunkt verschwinden. Diese Basisfunktionen spannen offensichtlich gerade den Raum der stückweise linearen Funktionen auf.

  Es geht aber auch rechteckig: Jetzt haben wir in jedem rechteckigen Element vier Freiheitsgrade. Mit den (elementlokalen) Basisfunktionen 1, x, y und $x \cdot y$ können wir somit eine beliebige bilineare Funktion darstellen. Die schon vertraute Definition der (globalen) Basisfunktionen φ_i (1 in einem Gitterpunkt, 0 in den anderen) führt jetzt auf die Pagodenfunktionen, die wir schon bei der Quadratur nach Archimedes kennengelernt hatten. Diese spannen nun den Raum der stückweise bilinearen Funktionen auf.



Offensichtlich bilden die jeweiligen Funktionen in allen drei Fällen eine Basis. Der einer Basisfunktion zugeordnete Koeffizient (u_i in $u_i \cdot \varphi_i(x)$) stellt dabei den Wert von $u(x)$ an der Stelle x_i dar. Deshalb wird eine solche Basis Stützpunktbasis bzw. Nodal Point Basis genannt. Stützpunktbasen bieten

- keinen direkten Anhaltspunkt zur lokalen Verfeinerung (im glatten Fall sind alle Koeffizienten etwa gleich groß);
- keine Möglichkeit der Wiederverwertbarkeit von Basisfunktionen bei lokaler Verfeinerung (beim Übergang von n zu $2 \cdot n$ sind alle Basisfunktionen und somit auch alle Koeffizienten u_i neu zu berechnen).

Deshalb machen wir uns auf die Suche nach einem hierarchischen Konzept.

- Dazu erinnern wir uns an Fourier-Reihen bzw. Fourier-Polynome für $f: [-\pi,\pi] \to \mathbb{R}$:

$$\left.\begin{array}{c} a_k \\ b_k \end{array}\right\} = \frac{1}{\pi} \cdot \int_{-\pi}^{\pi} f(x) \cdot \left\{\begin{array}{c} \sin(kx) \\ \cos(kx) \end{array}\right\} dx \qquad \text{für} \qquad \left\{\begin{array}{l} k = 0,1,2,\ldots \\ k = 1,2,3,\ldots \end{array}\right.$$

$$Sf(x) = \frac{a_0}{2} + \sum_{k=1}^{\infty} (a_k \cos(kx) + b_k \sin(kx)) \qquad \text{Fourier-Reihe}$$

$$S_n f(x) = \frac{a_0}{2} + \sum_{k=1}^{n} (a_k \cos(kx) + b_k \sin(kx)) \qquad \text{Fourier-Polynom}$$

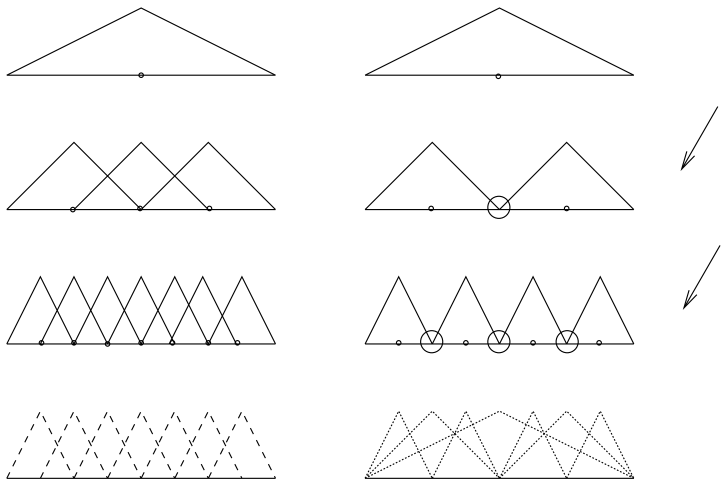
Die Fourier-Koeffizienten a_k,b_k klingen – abhängig von der Glattheit von f – ab. Man kann Aussagen zeigen der Art:

$$|a_k|\ ,\ |b_k| \ \leq\ c \cdot k^{-l} \text{ für } f \in \mathcal{C}^l[-\pi,\pi].$$

Eine Erhöhung der Genauigkeit der Darstellung (d. h. eine Verringerung des Fehlers $\|f - S_n f\|$) erfolgt durch Hinzunahme höherfrequenter Anteile bzw. Basisfunktionen. Somit erhalten wir über die Größe der Fourier-Koeffizienten eine Art Fehlermaß. Außerdem können wir von der Wiederverwertbarkeit profitieren: Beim Übergang von n zu $n+1$ oder zu $2 \cdot n$ bleiben die schon berechneten Koeffizienten unverändert. Es kommen lediglich neue Koeffizienten als hierarchisches Inkrement hinzu.

- Diese Überlegungen führen von der Stützpunktbasis zur hierarchischen Basis. Der Koeffizient v_i einer Basisfunktion $\phi_i(x)$ ist jetzt nicht mehr der Funktionswert, sondern die Differenz aus Funktionswert und schon Vorhandenem (d. h. hierarchisch Höherem): v_i ist ein hierarchisches Inkrement und wird hierarchischer Überschuß genannt.

→ 1D: hierarchische Basis aus Hutfunktionen



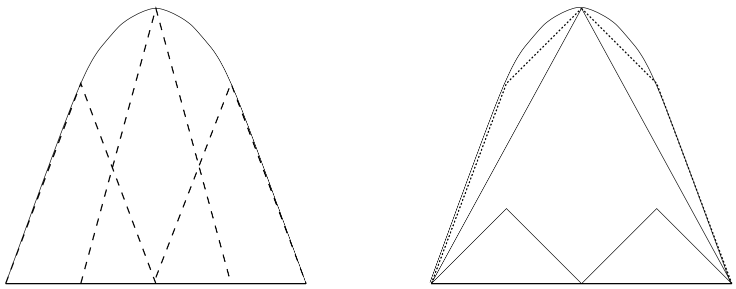
Immer nur die neuen Punkte

an groben Stellen wird Altes übernommen

Stützpunktbasis (alle Basisfunktionen stammen vom feinsten Level)

Hierarchische Basis (Mischung unterschiedlicher Level)

Als Beispiel betrachten wir wieder die Parabel $f(x) := x(2-x)$.



Koeffizienten 3/4, 1, 3/4

Koeffizienten 1/4, 1, 1/4

Bei gleicher Auflösung bzw. Maschenweite bzw. gleichem Level ergeben sich dieselben Interpolanten – schließlich handelt es sich ja nur um eine andere Basisdarstellung im selben Raum. Man kann die Basisdarstellung natürlich jederzeit ändern, wie man das vom Basiswechsel aus der linearen Algebra kennt:

$$\begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{pmatrix} \longrightarrow \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{pmatrix}$$

$$\begin{array}{rcl} \text{Stützpunktkoeffizienten} & & \text{hierarchische Koeffizienten} \\ u^I(x) = \sum_{i=1}^{n} u_i \cdot \varphi_i(x) & = & u^I(x) = \sum_{i=1}^{n} v_i \cdot \phi_i(x) \\ H \cdot \underline{u} = & \underline{v} & \text{Basiswechsel} \end{array}$$

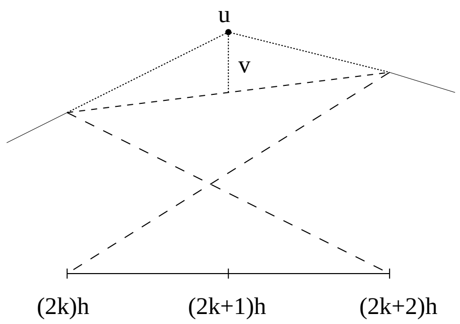
Um den Basiswechsel und somit die Matrix H explizit angeben zu können, muß man nur die Definition des hierarchischen Überschusses als Differenz von Funktionswert und Interpolant aus allen hierarchisch höheren Basisfunktionen heranziehen:

$$v_{2k+1} = u_{2k+1} - \frac{u_{2k} + u_{2k+2}}{2} \; .$$

In jeder Zeile von H findet sich also das Muster

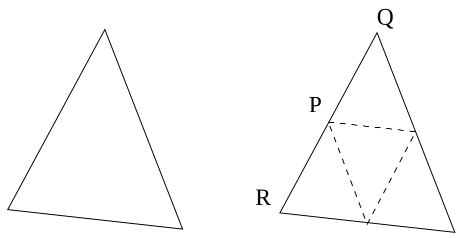
$$\left(-\frac{1}{2}, 1, -\frac{1}{2} \right) ,$$

jedoch in unterschiedlicher „Breite“.



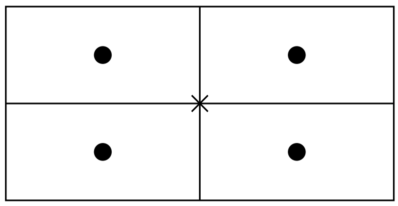
Als natürliche Datenstruktur erhalten wir im Falle der Stützpunktbasis den array $u[1..n]$, im hierarchischen Fall jedoch einen Binärbaum, weil dieser gerade die Hierarchie (d. h. die Vater-Sohn-Beziehung) widerspiegelt.

→ 2D: Zunächst betrachten wir wieder Dreieckselemente:

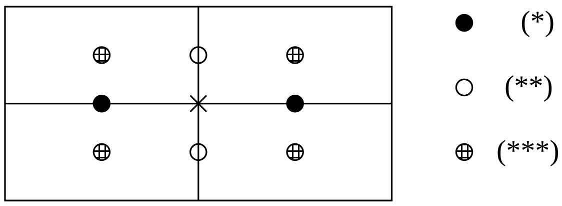


Im Falle der Stützpunktbasis ist der Wert in P der Funktionswert $u(P)$, in hierarchischer Darstellung ist der Wert in P der Überschuß $u(P) - \frac{u(Q)+u(R)}{2}$.

Bei Rechteckselementen gibt es zwei verschiedene Strategien. Die univariate Vorgehensweise verfeinert stets in beiden Richtungen zugleich. Dadurch haben alle Elemente dasselbe Verhältnis von Seitenlängen.



Die multivariate Vorgehensweise gestattet dagegen die separate Verfeinerung jeder Koordinatenrichtung. Somit gibt es zusätzlich zu oben Nachfahren, die aus einer Verfeinerung in x-Richtung entstanden sind (*), und solche, bei denen in y-Richtung verfeinert wurde (**). Die Nachfahren im univariaten Fall entstehen bei zwei aufeinanderfolgenden Verfeinerungsschritten in x- und y-Richtung. Die Verfeinerung nur in einer Richtung ermöglicht Elemente mit stark unterschiedlichem Verhältnis der Seitenlängen.



Vorteile der hierarchischen Vorgehensweise:

+ Multilevel-Ansätze werden unterstützt, also
  - ⋆ Betrachtung auf verschiedenen Diskretisierungsniveaus (Levels)
  - ⋆ schnelle Lösung der bei PDEs entstehenden linearen Gleichungssysteme („hierarchische Basis verbessert Kondition“)
+ adaptive lokale Verfeinerung wird unterstützt
+ man gewinnt Einblick in Problemstruktur (was ist wichtig, was ist weniger wichtig: der $\frac{1}{4}$-Abfall gilt zwar nur für $ax^2 + bx + c$ exakt, die (asymptotische) Gesetzmäßigkeit gilt aber viel allgemeiner).

All das eröffnet wie bei der Quadratur nach Archimedes Perspektiven für eine effizientere Gitterpunktwahl.

Nachteile:

– Rekonstruktion/Auswertung des Interpolanten wird aufwendiger:
  Addition der Beiträge jedes Levels $\rightarrow$ $|\log h|$-Komplexität bei einem Einzelzugriff auf die Baumstruktur; dies ist jedoch kein Nachteil, wenn an allen Stellen ausgewertet werden soll.

– Die Matrizen der bei PDEs entstehenden linearen Gleichungssysteme werden voller, weil durch die starke Überlappung der Elemente mehr Elemente miteinander koppeln. I. d. R. führt dies dazu, daß die Algorithmen trickreicher werden (müssen), ein Komplexitätsverlust kann jedoch i. a. vermieden werden.

3.2 Tensorprodukträume

Ziel ist nun die Übertragung der hierarchischen Basis ins Mehrdimensionale. Dabei wollen wir uns möglichst eng an die eindimensionale Konstruktion anlehnen und die Verallgemeinerung dimensionsrekursiv machen, d. h. „dD = 1D ∘ (d-1)D“ . Dazu müssen wir Elemente und Basisfunktionen definieren.

(i) Zu interpolierende Funktionen:

$$\Omega :=]0,1[^d \quad , \quad \bar{\Omega} = [0,1]^d,$$

$$X(\bar{\Omega}) := \left\{ u := \bar{\Omega} \to \mathbb{R},\ \frac{\partial^{k_1+\cdots+k_d} u}{\prod_{j=1}^{d} \partial x_j^{k_j}} \in \mathcal{C}^0(\bar{\Omega}),\ k_j \in \{0,1,2\} \right\},$$

$$X_0(\bar{\Omega}) := \left\{ u \in X(\bar{\Omega}) \ :\ u|_{\partial\Omega} = 0 \right\}.$$

Wir beziehen uns also auf das d-dimensionale Einheitsintervall und betrachten hierauf Funktionen mit stetigen gemischten Ableitungen bis zur Ordnung 2 in jeder Richtung. Ferner definieren wir zwei Halbnormen auf $X_0(\bar{\Omega})$:

$$|u|_\infty := \left\| \frac{\partial^{2d} u}{\prod_{j=1}^{d} \partial x_j^2} \right\|_\infty = \max_{\underline{x} \in \bar{\Omega}} \left| \frac{\partial^{2d} u(\underline{x})}{\prod_{j=1}^{d} \partial x_j^2} \right| \qquad (L_\infty\text{-Norm}),$$

$$|u|_2 := \left\| \frac{\partial^{2d} u}{\prod_{j=1}^{d} \partial x_j^2} \right\|_2 = \left(\int_{\bar{\Omega}} \left| \frac{\partial^{2d} u(\underline{x})}{\prod_{j=1}^{d} \partial x_j^2} \right|^2 d\Omega \right)^{\frac{1}{2}} \qquad (L_2\text{-Norm}),$$

wobei $\underline{x} = (x_1, \ldots, x_d) \in \bar{\Omega}$. Es handelt sich in der Tat nur um Halbnormen, da $|u|_\infty = 0$ oder $|u|_2 = 0$, aber $u \neq 0$ möglich ist (z. B. für konstantes u).

(ii) Teilraumzerlegung

Multiindex $\underline{l} = (l_1, \ldots, l_d) \in \mathbb{N}^d$

Gitter $\Omega_{\underline{l}}$ mit Maschenweite $\underline{h_l} := (h_{l_1}, \ldots, h_{l_d}) = (2^{-l_1}, \ldots, 2^{-l_d})$

d. h.:

- l_j gibt den Level und damit die Auflösung des Gitters in j-Richtung an;
- $h_{l_j} = 2^{-l_j}$ gibt die Maschenweite in j-Richtung an;
- das Gitter $\Omega_{\underline{l}}$ ist äquidistant in jeder Koordinatenrichtung, aber es sind unterschiedliche Maschenweiten in den verschiedenen Koordinatenrichtungen möglich.

Die Gitterpunkte von $\Omega_{\underline{l}}$ werden mit $x_{\underline{l},\underline{i}} = (x_{l_1,i_1}, \ldots, x_{l_d,i_d})$ bezeichnet, wobei $x_{l_j,i_j} = i_j \cdot h_{l_j} = i_j \cdot 2^{-l_j}$, $i_j = 0, \ldots, 2^{l_j}$.

Jetzt definieren wir den Raum der bzgl. den inneren Gitterpunkten von $\Omega_{\underline{l}}$ stückweise d-linearen Funktionen:

$$V_{\underline{l}} \;:=\; \text{span}\,\{\phi_{\underline{l},\underline{i}}\,,\; i_j = 1, \ldots, 2^{l_j} - 1,\; j = 1, \ldots, d\}\;,$$

wobei die d-dimensionalen Basisfunktionen $\phi_{\underline{l},\underline{i}}(\underline{x})$ als Produkt eindimensionaler Basisfunktionen eingeführt werden:

$$\begin{aligned}
\phi_{\underline{l},\underline{i}}(\underline{x}) \;&:=\; \prod_{j=1}^{d} \phi_{l_j,i_j}(x_j)\;, \\
\phi_{l_j,i_j}(x_j) \;&=\; \phi\left(\frac{x_j - i_j h_{l_j}}{h_{l_j}}\right), \\
\text{support}\;(\phi_{l_j,i_j}) \;&=\; \left[x_{l_j,i_j} - h_{l_j}, x_{l_j,i_j} + h_{l_j}\right]\;.
\end{aligned}$$

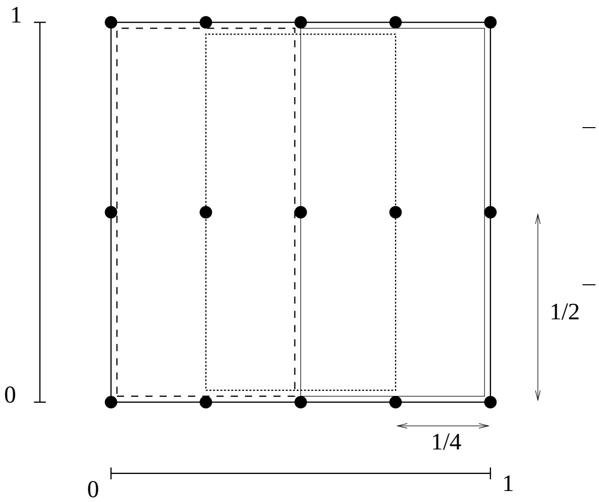
Die Funktion $\phi(x)$ übernimmt dabei die Rolle der „Mutter aller Basisfunktionen“:

$$\phi(x) \;:=\; \begin{cases} 1 - |x| & \text{auf } [-1,1] \\ 0 & \text{sonst} \end{cases}\;.$$

Die Basisfunktion $\phi_{\underline{l},\underline{i}}$ ist also im Gitterpunkt $x_{\underline{l},\underline{i}}$ „aufgehängt“.

Als Beispiel betrachten wir das Gitter $\Omega_{\underline{l}}$ mit $d = 2$, $\underline{l} = (2,1)$, und folglich $\underline{h} = (\frac{1}{4}, \frac{1}{2})$:

- Gitter:



- 15 Gitterpunkte

$$x_{\underline{l},\underline{i}} \;:\; \begin{array}{l} l_1 = 2 \Rightarrow i_1 = 0, \ldots, 4 \\ l_2 = 1 \Rightarrow i_2 = 0, 1, 2 \end{array}$$

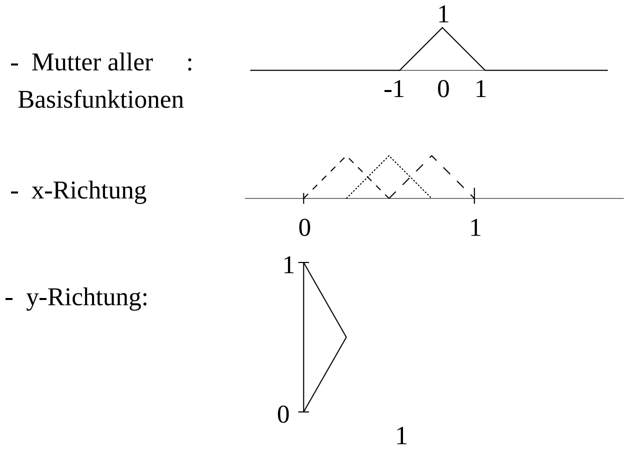
- 3 innere Punkte

$$x_{\underline{l},\underline{i}} \;:\; \begin{array}{l} l_1 = 2 \Rightarrow i_1 = 1, 2, 3 \\ l_2 = 1 \Rightarrow i_2 = 1 \end{array}$$

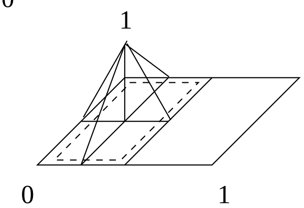
- $V_{(2,1)}$ ergibt sich als der von den in den inneren Gitterpunkten lebenden Basisfunktionen aufgespannte lineare Raum:

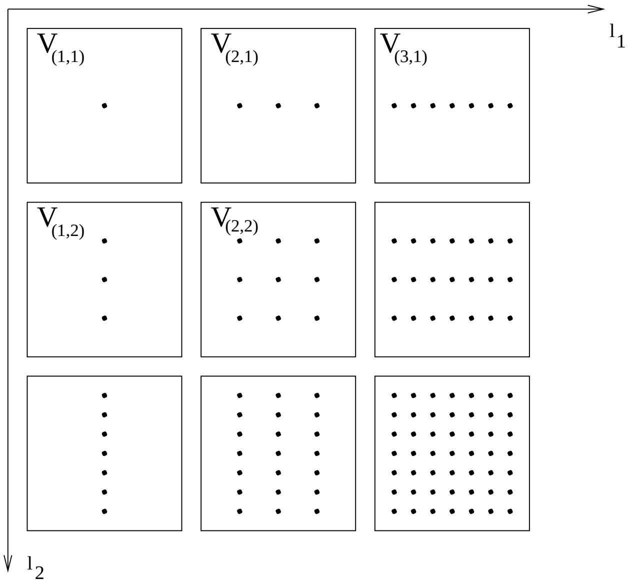
$$V_{(2,1)} \;:=\; \text{span}\;\{\,\phi_{(2,1),(1,1)},\;\phi_{(2,1),(1,2)},\;\phi_{(2,1),(1,3)}\,\}$$

– Basisfunktionen:



d. h. ϕ





– die Räume $V_{\underline{l}}$ sind geschachtelt (nested), d. h. $V_{\underline{l}} \subseteq V_{\underline{k}}$ für $\underline{l} \leq \underline{k}$ (komponentenweise);

– definiere $V_n := V_{n \cdot \underline{1}} = V_{(n,n,\ldots,n)}$

$\Rightarrow$ $V_n \subset V_{n+1}$, und mit $\sum_{n=1}^{\infty} V_n =: V$

(Vektorraum aller – endlichen – Linearkombinationen) gilt:

$\overline{V} = H_0^1$ Sobolevraum zum Index 1

(Vervollständigung von $\mathcal{C}_0^\infty$ bzgl. der H^1-Norm, wichtig bei PDEs)

Wegen $X_0 \subset H_0^1$ ist folglich jedes $u \in X_0$ beliebig genau approximierbar in V und kann als $\sum_1^\infty u_n,\ u_n \in V_n$, dargestellt werden.

(iii) Hierarchische Teilraumzerlegung

Das Schema der $V_{\underline{l}}$ ist hierarchisch geschachtelt, V bildet aber keine direkte Summe:

$$V = \sum_{n=1}^{\infty} V_n \overset{V_n \subset V_{n+1}}{=} \bigcup_{n=1}^{\infty} V_n \overset{V_n \subset V_{n+1}}{=:} \lim_{n\to\infty} V_n \quad \text{(Grenzwert existiert).}$$

Dementsprechend ist die Darstellung $u = \sum_1^\infty u_n,\ u_n \in V_n$, nicht eindeutig. Ferner gilt

$$V_n^{(\infty)} := V_n = V_{n\cdot\underline{1}} = \sum_{|\underline{l}|_\infty \le n} V_{\underline{l}} \qquad \text{mit} \qquad |\underline{l}|_\infty = \max_{1\le j\le d} l_j .$$

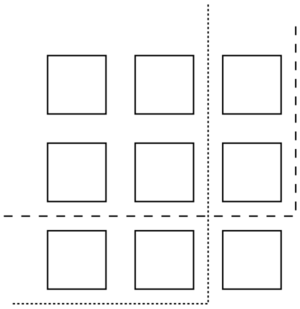
V_n hat den oberen Index (∞) bekommen, um die Definition über die diskrete l_∞-Norm herauszustellen.

Um zu einer direkten Summe (und somit eindeutigen Zerlegung) zu gelangen, gehen wir über zum Schema der Differenzräume $W_{\underline{l}}$,

$$W_{\underline{l}} := V_{\underline{l}} - \sum_{j=1}^{d} V_{\underline{l}-\underline{e_j}},$$

wobei $\underline{e_j}$ den j-ten Einheits-Multiindex bezeichnet.

Der Vollständigkeit halber: $V_{\underline{l}} = \{0\}$, falls ein $l_j = 0$;



V(3,3)

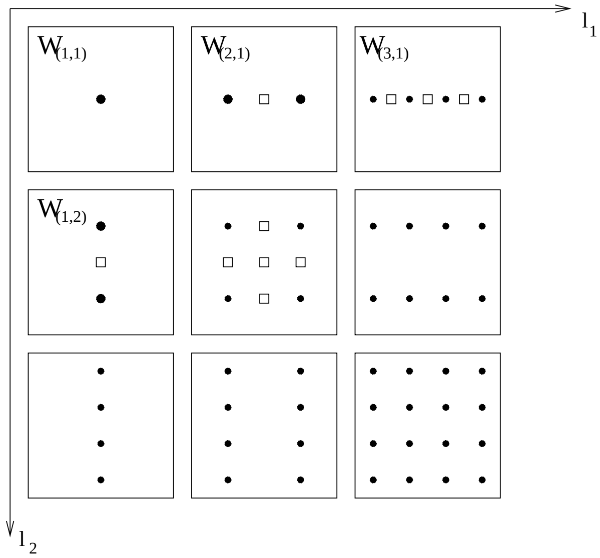
d. h. in jeder Richtung fällt jeder zweite Punkt heraus

Damit gilt: $$V_n^{(\infty)} = \sum_{|\underline{l}|_\infty \le n} V_{\underline{l}} = \bigoplus_{|\underline{l}| \le n} W_{\underline{l}}$$

und $$V = \bigoplus_{\underline{l}\in\mathbb{N}^d} W_{\underline{l}} .$$

Ausgehend von den hierarchischen Differenzräumen $W_{\underline{l}}$ haben wir in der hierarchischen Teilraumzerlegung jetzt direkte Summen. Als hierarchisches Inkrement bringt jeder Teilraum $W_{\underline{l}}$ jetzt etwas Neues.

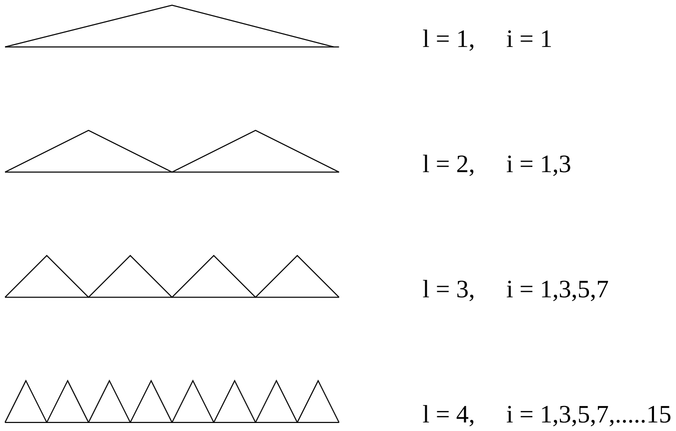
Betrachten wir das Beispiel $n = 3$:



Hier gilt offensichtlich $\sum\limits_{|\underline{l}|_\infty \leq 3} W_{\underline{l}} \; = \; V_{(3,3)} \; = \; V_3^{(\infty)}$, und für $V_{\underline{l}}$ und $W_{\underline{l}}$ erhalten wir folgende Darstellung:

$$
\begin{aligned}
V_{\underline{l}} \quad &= \quad \operatorname{span}\left\{\phi_{\underline{l},\underline{i}},\; i_j = 1, \ldots, 2^{l_j} - 1,\; j = 1, \ldots, d\right\}, \\
W_{\underline{l}} \quad &= \quad \operatorname{span}\left\{\phi_{\underline{l},\underline{i}},\; \underbrace{i_j = 1, \ldots, 2^{l_j} - 1,\; i_j \text{ ungerade},\; j = 1, \ldots, d}_{=:\, I_{\underline{l}}}\right\} \\
&= \quad \operatorname{span}\left\{\phi_{\underline{l},\underline{i}},\; \underline{i} \in I_{\underline{l}}\right\} \; .
\end{aligned}
$$

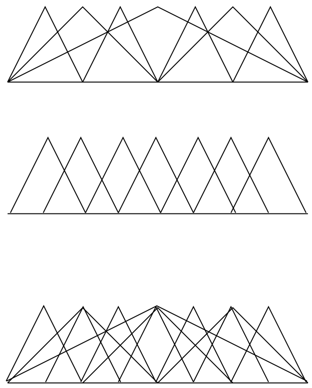
Wir machen uns nochmals die Numerierung klar:



<u>Bemerkung:</u>

- Die Basisfunktionen von $V_{n\cdot\underline{1}}$ bilden eine Stützpunktbasis (SPB) von $V_n^{(\infty)}$.
- Die Basisfunktionen aus $W_{\underline{l}}$, $|\underline{l}|_\infty \leq n$, bilden eine hierarchische Basis (HB) von $V_n^{(\infty)}$.
- Alle Basisfunktionen aus $V_{\underline{l}}$, $|\underline{l}|_\infty \leq n$, bilden keine Basis, sondern ein sogenanntes Erzeugendensystem (EZS) von $V_n^{(\infty)}$ (Mehrdeutigkeit der Darstellung).
- Es gilt: $\text{span}\{\text{SPB}_n\} = \text{span}\{\text{HB}_n\} = \text{span}\{\text{EZS}_n\} = V_n^{(\infty)}$ sowie im eindimensionalen Fall $\text{EZS}_n = \bigcup_{l=1}^{n} \text{SPB}_l$.

1D

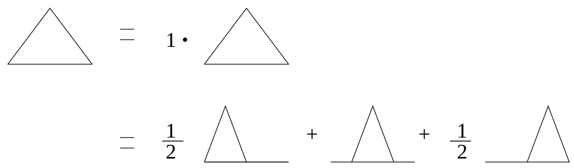


HB

SPB

alles: EZS

- Die mehrdeutige Darstellung einer Funktion im Erzeugendensystem hat Vorteile für die Mehrgitterlösung von PDEs.



(iv) <u>Interpolation/Darstellung:</u>

Jetzt benützen wir die hierarchische Teilraumzerlegung aus (iii), um Interpolanten zu konstruieren. Jedes $u \in X_0(\bar{\Omega})$ läßt sich eindeutig darstellen als

$$u(\underline{x}) \;=\; \sum_{\underline{l}} u_{\underline{l}}(\underline{x})$$

mit $u_{\underline{l}} \in W_{\underline{l}}$ und

$$u_{\underline{l}}(\underline{x}) = \sum_{\underline{i}\in I_{\underline{l}}} v_{\underline{l},\underline{i}} \cdot \phi_{\underline{l},\underline{i}}(\underline{x})\,.$$

Das ist die Darstellung in hierarchischer Basis mit hierarchischen Koeffizienten bzw. Überschüssen $v_{\underline{l},\underline{i}}$.

Im folgenden Abschnitt geht es um die Frage, wie gut $V_n^{(\infty)}$ ein $u \in X_0(\bar{\Omega})$ approximiert und wieviel das kostet. Wir betrachten also $|V_n^{(\infty)}|$, die Anzahl der Punkte im $V_n^{(\infty)}$ zugrundeliegenden Gitter, und $||u-u_n^{\infty}||$, den Interpolationsfehler des Interpolanten $v_n^{(\infty)} \in V_n^{(\infty)}$ von u in $V_n^{(\infty)}$. Den Fehler messen wir in drei Normen:

a) Maximumsnorm bzw. L_∞-Norm: $||u||_\infty := \max_{\underline{x}\in\bar{\Omega}} |u(\underline{x})|$,

b) L_2-Norm: $||u||_2 := \left(\int_{\bar{\Omega}} |u(\underline{x})|^2 d\underline{x}\right)^{\frac{1}{2}}$,

c) Energienorm: $||u||_E := \left(\int_{\bar{\Omega}} |\nabla u|^2 d\underline{x}\right)^{\frac{1}{2}}$.

3.3 Approximationstheorie

Es folgen fünf Lemmata samt Beweis:

Lemma 1:

Für die Anzahl der Punkte bzw. Basisfunktionen im Teilraum $W_{\underline{l}}$ gilt:

$$|W_{\underline{l}}| \; := \; \dim W_{\underline{l}} \; = \; 2^{|\underline{l}-\underline{1}|_1} \; ,$$

wobei $|\underline{l}|_1 \; = \; \sum_{j=1}^{d} l_j$ die diskrete l_1-Norm bezeichnet.

Beweis:

$$\begin{aligned}
W_{\underline{l}} &= \text{span}\,\{\phi_{\underline{l},\underline{i}},\; \underline{i} \in I_{\underline{l}}\}, \\
I_{\underline{l}} &= \{\underline{i} \;:\; i_j = 1,\ldots,2^{l_j-1},\; i_j \text{ ungerade},\; j=1,\ldots,d\} \\
&\Rightarrow \text{ in jeder Dimension: } \; 2^{l_j-1} \text{ Stück} \\
&\Rightarrow \quad |W_{\underline{l}}| \; = \; \prod_{j=1}^{d} 2^{l_j-1} \; = \; 2^{|\underline{l}|_1-d} \; = \; 2^{|\underline{l}-\underline{1}|_1} . \qquad \square
\end{aligned}$$

Lemma 2:

Für eine Basisfunktion $\phi_{\underline{l},\underline{i}}$ gilt:

$$\begin{aligned}
||\phi_{\underline{l},\underline{i}}||_\infty &= 1, \\
||\phi_{\underline{l},\underline{i}}||_2 &= \left(\frac{2}{3}\right)^{d/2} \cdot 2^{-|\underline{l}|_1/2} \\
&= \left(\frac{2}{3}\right)^{d/2} \cdot \left(\prod_{j=1}^{d} h_{l_j}\right)^{1/2}, \\
||\phi_{\underline{l},\underline{i}}||_E &= \sqrt{2}\cdot\left(\frac{2}{3}\right)^{(d-1)/2} \cdot 2^{-|\underline{l}|_1/2} \cdot \left(\sum_{j=1}^{d} 4^{l_j}\right)^{1/2} \\
&= \sqrt{2}\cdot\left(\frac{2}{3}\right)^{(d-1)/2} \cdot \left(\prod_{j=1}^{d} h_{l_j}\right)^{1/2} \cdot \left(\sum_{j=1}^{d} h_{l_j}^{-2}\right)^{1/2} .
\end{aligned}$$

Beweis:

Die erste Gleichung folgt unmittelbar aus der Definition der Basisfunktionen. Bzgl. der L_2-Norm zeigen wir den eindimensionalen Fall. Die Verallgemeinerung auf beliebiges d erfolgt dann mittels Fubini.

$$
\begin{aligned}
\left(\int\limits_{-h_{l_1}}^{+h_{l_1}} |\phi_{l_1,i_1}(x_1)|^2 \; dx_1 \right)^{1/2} &= \left(\int\limits_{-h_{l_j}}^{0} \left| \frac{x_j + h_{l_j}}{h_{l_j}} \right|^2 \; dx_1 \; + \int\limits_{0}^{h_{l_j}} \left| \frac{h_{l_j} - x_j}{h_{l_j}} \right|^2 \; dx_1 \right)^{1/2} \\
&= \left(\frac{x_j + h_{l_j}{}^3}{3\,h_{l_j}} \Big|_{-h_{l_j}}^{0} - \frac{h_{l_j} - x_j{}^3}{3\,h_{l_j}} \Big|_{0}^{h_{l_j}} \right)^{1/2} \\
&= \left(\frac{2}{3} h_{l_j} \right)^{1/2} .
\end{aligned}
$$

In der Energienorm gilt

$$
\begin{aligned}
\|\phi_{\underline{l},\underline{i}}\|_E^2 &= \int\limits_{-h_{l_1}}^{+h_{l_1}} \cdots \int\limits_{-h_{l_d}}^{+h_{l_d}} \sum_{j=1}^{d} \left(\frac{\partial \phi_{\underline{l},\underline{i}}}{\partial x_j} \right)^2 \; dx_d \ldots dx_1 \\
&= \frac{2^d}{h_{l_1}{}^2 \ldots h_{l_d}{}^2} \cdot \int\limits_{0}^{h_{l_1}} \cdots \int\limits_{0}^{h_{l_d}} \sum_{j=1}^{d} \frac{x_1^2 \ldots x_d^2}{x_j^2} \; dx_d \ldots dx_1 \\
&= \frac{2^d}{h_{l_1}{}^2 \ldots h_{l_d}{}^2} \cdot \sum_{j=1}^{d} \frac{h_{l_1}{}^3 \ldots h_{l_d}{}^3}{3^{d-1} \cdot h_{l_j}^2} \\
&= 2 \cdot \left(\frac{2}{3} \right)^{d-1} \cdot 2^{-|\underline{l}|_1} \cdot \sum_{j=1}^{d} 4^{l_j} \qquad \Box
\end{aligned}
$$

Lemma 3:

Sei

$$
\begin{aligned}
\psi_{l_j,i_j}(x_j) &:= 2^{-l_j+1} \cdot \phi_{l_j,i_j}(x_j), \\
\psi_{\underline{l},\underline{i}}(\underline{x}) &:= \prod_{j=1}^{d} \psi_{l_j,i_j}(x_j) \; .
\end{aligned}
$$

Dann gilt für die hierarchischen Koeffizienten $v_{\underline{l},\underline{i}}$ folgende Integraldarstellung:

$$
v_{\underline{l},\underline{i}} := \int\limits_{\bar{\Omega}} \psi_{\underline{l},\underline{i}}(\underline{x}) \cdot \frac{\partial^{2d} u(\underline{x})}{\prod_{j=1}^{d} \partial x_j^2} \; d\Omega \; .
$$

Beweis:

für 1D, 2D einfache Übungsaufgabe (partielle Integration!) $\Box$

Lemma 4:

Für die Größenordnung der hierarchischen Koeffizienten gilt:

$$\begin{aligned}
|v_{\underline{l},\underline{i}}| &\leq 2^{-d} \cdot 2^{-2|\underline{l}|_1} \cdot |u|_\infty, \\
|v_{\underline{l},\underline{i}}| &\leq 2^{-d} \cdot \left(\frac{2}{3}\right)^{d/2} \cdot 2^{-\frac{3}{2}|\underline{l}|_1} \cdot \left|u\ |_{\phi_{\underline{l},\underline{i}}}\right|_2 .
\end{aligned}$$

(„Abfall der hierarchischen Koeffizienten“)

Beweis:

Die erste Abschätzung folgt direkt aus Lemma 3:

$$\begin{aligned}
|v_{\underline{l},\underline{i}}| &= \Big| \int_{\bar{\Omega}} \psi_{\underline{l},\underline{i}}(\underline{x}) \frac{\partial^{2d} u(\underline{x})}{\partial x_1^2 \ldots \partial x_d^2}\, d\Omega \Big| \\
&\leq |u|_\infty \cdot \left| \int_{\bar{\Omega}} \psi_{\underline{l},\underline{i}}(\underline{x})\, d\Omega \right| \\
&= |u|_\infty \cdot 2^{-d} \cdot 2^{-|\underline{l}|_1} \cdot \left| \int_{\bar{\Omega}} \phi_{\underline{l},\underline{i}}(\underline{x})\, d\Omega \right| \\
&= |u|_\infty \cdot 2^{-d} \cdot 2^{-2|\underline{l}|_1} .
\end{aligned}$$

Die Cauchy-Schwarzsche Ungleichung und Lemma 2 führen zur zweiten Abschätzung:

$$\begin{aligned}
|v_{\underline{l},\underline{i}}| &\leq \left\|\psi_{\underline{l},\underline{i}}\right\|_2 \cdot \left\| \frac{\partial^{2d} u\ |_{\phi_{\underline{l},\underline{i}}}(\underline{x})}{\partial x_1^2 \ldots \partial x_d^2}\, d\Omega \right\|_2 \\
&= 2^{-d} \cdot 2^{-|\underline{l}|_1} \cdot \left\|\phi_{\underline{l},\underline{i}}\right\|_2 \cdot \left|u\ |_{\phi_{\underline{l},\underline{i}}}\right|_2 \\
&= 2^{-d} \cdot \left(\frac{2}{3}\right)^{d/2} \cdot 2^{-\frac{3}{2}|\underline{l}|_1} \cdot \left|u\ |_{\phi_{\underline{l},\underline{i}}}\right|_2 . \qquad \square
\end{aligned}$$

Jetzt betrachten wir den Beitrag des ganzen Teilraums $W_{\underline{l}}$ zum Interpolanten.

Lemma 5:

Sei $u \in X_0(\bar{\Omega})$ gegeben in hierarchischer Darstellung, $u = \sum u_{\underline{l}},\ u_{\underline{l}} \in W_{\underline{l}}$. Dann gilt für den Beitrag $u_{\underline{l}}$ des Teilraums $W_{\underline{l}}$:

$$\begin{aligned}
\|u_{\underline{l}}\|_\infty &\leq 2^{-d} \cdot 2^{-2|\underline{l}|_1} \cdot |u|_\infty, \\
\|u_{\underline{l}}\|_2 &\leq 3^{-d} \cdot 2^{-2|\underline{l}|_1} \cdot |u|_2, \\
\|u_{\underline{l}}\|_E &\leq \frac{1}{2 \cdot 12^{(d-1)/2}} \cdot 2^{-2|\underline{l}|_1} \cdot \left(\sum_{j=1}^{d} 4^{l_j} \right)^{1/2} \cdot |u|_\infty .
\end{aligned}$$

D. h.: Bei hinreichender Glattheit ($u \in X_0(\bar{\Omega})$) fällt die Bedeutung der $W_{\underline{l}}$ mit wachsendem $\underline{l}$ rasch ab!

Beweis:

Zunächst zur Maximumsnorm:

Die Träger aller $\phi_{\underline{l},\underline{i}}$ sind disjunkt ($\underline{l}$ fest). Damit sowie mit Lemma 2 und Lemma 4 folgt

$$\|u_{\underline{l}}\|_\infty = \Big\|\sum_{\underline{i}\in I_{\underline{l}}} v_{\underline{l},\underline{i}}\cdot\phi_{\underline{l},\underline{i}}\Big\|_\infty \le 2^{-d}\cdot 2^{-2|\underline{l}|_1}\cdot|u|_\infty .$$

Dieselben Argumente sowie die Tatsache, daß die Träger der $\phi_{\underline{l},\underline{i}}$ bei festem $\underline{l}$ Ω disjunkt überdecken, führen zu

$$\begin{aligned}
\|u_{\underline{l}}\|_2^2 &= \sum_{\underline{i}\in I_{\underline{l}}} |v_{\underline{l},\underline{i}}|^2\cdot\|\phi_{\underline{l},\underline{i}}\|_2^2 \\
&\le \sum_{\underline{i}\in I_{\underline{l}}} \left(\frac{1}{6}\right)^d\cdot 2^{-3|\underline{l}|_1}\cdot\left|u\ |_{\phi_{\underline{l},\underline{i}}}\right|_2^2\cdot\left(\frac{2}{3}\right)^d\cdot 2^{-|\underline{l}|_1} \\
&= 9^{-d}\cdot 2^{-4|\underline{l}|_1}\cdot|u|_2^2 .
\end{aligned}$$

Für die Energienorm folgt wiederum mit dem Disjunktheitsargument

$$\begin{aligned}
\|u_{\underline{l}}\|_E^2 &= \sum_{\underline{i}\in I_{\underline{l}}} \|v_{\underline{l},\underline{i}}^2\cdot\phi_{\underline{l},\underline{i}}\|_E^2 \\
&\le \sum_{\underline{i}\in I_{\underline{l}}} |v_{\underline{l},\underline{i}}|^2\cdot 2\cdot\left(\frac{2}{3}\right)^{d-1}\cdot 2^{-|\underline{l}|_1}\cdot\sum_{j=1}^d 4^{l_j} .
\end{aligned}$$

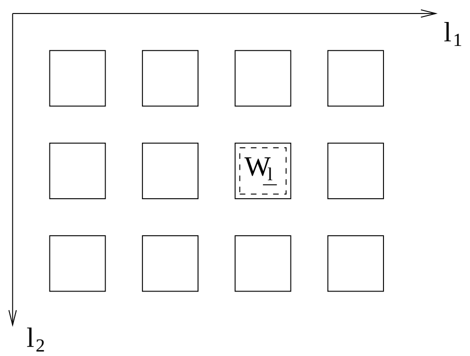
Für $|v_{\underline{l},\underline{i}}|$ kann man beide Abschätzungen aus Lemma 4 vewenden; wir wählen diejenige über die ∞-Halbnorm:

$$\begin{aligned}
&\le \sum_{\underline{i}\in I_{\underline{l}}} 2^{-2d}\cdot 2^{-4|\underline{l}|_1}\cdot|u|_\infty^2\cdot 2\cdot\left(\frac{2}{3}\right)^{d-1}\cdot 2^{-|\underline{l}|_1}\cdot\sum_{j=1}^d 4^{l_j} \\
&= \frac{1}{2\cdot 6^{d-1}}\cdot|u|_\infty^2\cdot 2^{-5|\underline{l}|_1}\cdot\sum_{j=1}^d 4^{l_j}\cdot\underbrace{\sum_{\underline{i}\in I_{\underline{l}}} 1}_{=(\star)} \\
&= \frac{1}{4\cdot 12^{d-1}}\cdot|u|_\infty^2\cdot 2^{-4|\underline{l}|_1}\cdot\sum_{j=1}^d 4^{l_j},
\end{aligned}$$

wobei $(\star)$: $|W_{\underline{l}}| = 2^{|\underline{l}-\underline{1}|_1} = 2^{|\underline{l}|_1}\cdot 2^{-d}$ nach Lemma 1. $\square$

Zusammenfassung des bisher Gezeigten:

- zurück zum hierarchischen Teilraumschema:



$u \in X_0(\bar{\Omega})$: $\quad u = \sum\limits_{\underline{l}} u_{\underline{l}} \qquad u_{\underline{l}} \in W_{\underline{l}}$ $\qquad$ eindeutig

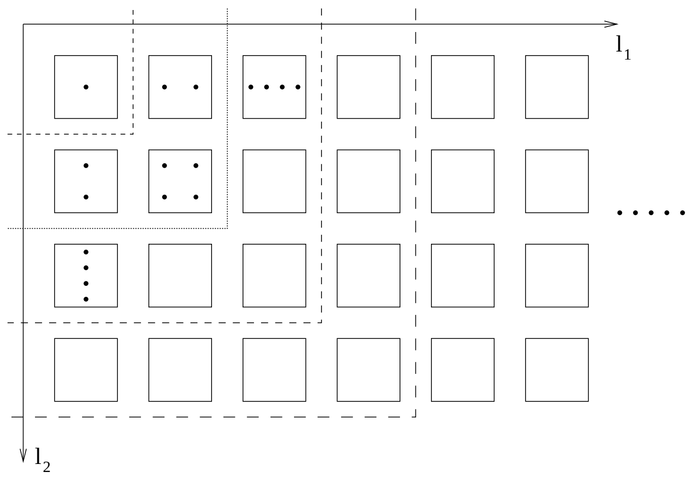
$\qquad\qquad u_{\underline{l}} = \sum\limits_{\underline{i} \in I_{\underline{l}}} v_{\underline{l},\underline{i}} \cdot \phi_{\underline{l},\underline{i}}(x_{\underline{l}})$ $\qquad$ eindeutig

- Lemma 1: $|W_{\underline{l}}| = 2^{|\underline{l}-\underline{1}|_1}$
  $\rightarrow$ Anzahl der Gitterpunkte/Basisfunktion in $W_{\underline{l}}$
  $W_{\underline{l}}$ „teurer“ mit wachsendem $\underline{l}$
- Lemma 2: Normabschätzungen der Basisfunktionen $\phi_{\underline{l},\underline{i}}$ $\quad (\|.\|_\infty,\ \|.\|_2,\ \|.\|_E)$
  $\rightarrow$ was steuert eine Basisfunktion größenordnungsmäßig bei?
- Lemma 3: Integraldarstellung der hierarchischen Koeffizienten / Überschüsse $v_{\underline{l},\underline{i}}$
- Lemma 4: Größenordnung der hierarchischen Überschüsse $v_{\underline{l},\underline{i}}$
- Lemma 5: Lemma 2 und Lemma 4: Normabschätzungen von $u_{\underline{l}}$
  $\rightarrow$ was steuert ein Teilraum $W_{\underline{l}}$ größenordnungsmäßig bei?
  Lemma 1 zeigt die Kosten eines Teilraums $W_{\underline{l}}$ an, Lemma 5 seinen Nutzen.

In der Praxis ist man nicht an einer unendlichen Reihendarstellung von $u \in X_0(\bar{\Omega})$ interessiert, sondern an einer endlichen Approximation bzw. an einem Interpolanten in einem endlich dimensionalen Teilraum von V bzw. $H_0^1(\bar{\Omega})$:

$$\text{statt } u = \sum_{\underline{l} \in \mathbb{N}^d} u_{\underline{l}} \qquad \text{jetzt } u^L = \sum_{\underline{l} \in L} u_{\underline{l}} \qquad \text{mit} \qquad L \subset \mathbb{N}^d,\ |L| < \infty\ .$$

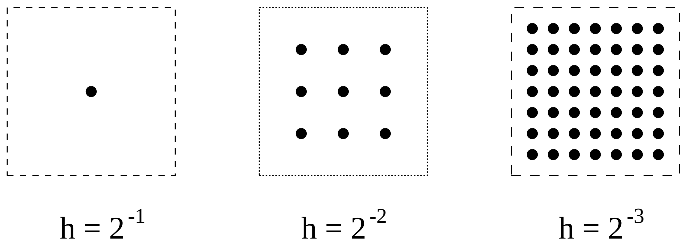
Prinzipiell sind die L beliebig wählbar, praktisch soll aber eine sukzessive Verfeinerung möglich sein. Das Teilraumschema legt rechteckige bzw. quadratische Fenster (d. h. Ausschnitte aus dem Schema) nahe:



Die resultierenden Appoximationsräume sind gerade die bereits eingeführten $V_n^{(\infty)}$:

$$V_n^{(\infty)} \;=\; \bigoplus_{|\underline{l}|_\infty \leq n} W_{\underline{l}} \; .$$

Die entsprechenden Gitter sind die äquidistanten Gitter der Maschenweite $h = 2^{-n}$ in jeder Koordinatenrichtung:



Kosten / Nutzen von $V_n^{(\infty)}$:

Offensichtlich gilt

$$|V_n^{(\infty)}| \;=\; (2^n - 1)^d \;=\; O(2^{d \cdot n}) \;=\; O(h^{-d}) \; .$$

Somit wächst die Anzahl der Punkte im Gitter zu $V_n^{(\infty)}$ bzw. die Anzahl der Basisfunktionen in $V_n^{(\infty)}$ exponentiell in d. Dieser höchst unerfreuliche Zusammenhang wird auch als Fluch der Dimensionalität bezeichnet. Dies ist ein Grund, warum Numerik bzw. Simulation im Höherdimensionalen nach wie vor ein großes Problem darstellen.

Lemma 6:

Seien $u \in X_0(\bar{\Omega})$ und $u_n^{(\infty)} \in V_n^{(\infty)}$ sein Interpolant in $V_n^{(\infty)}$.
Dann gilt für den Interpolationsfehler $u - u_n^{(\infty)}$:

$$\begin{aligned}
\|u - u_n^{(\infty)^{3)}}\|_{\infty^{1)}} \;&\leq\; \frac{d}{6^d} \cdot 2^{-2n} \cdot |u|_{\infty^{2)}} \;=\; O(h^2) \; , \\
\|u - u_n^{(\infty)}\|_2 \;&\leq\; \frac{d}{9^d} \cdot 2^{-2n} \cdot |u|_2 \;=\; O(h^2) \; , \\
\|u - u_n^{(\infty)}\|_E \;&\leq\; \frac{d^{3/2}}{2 \cdot 3^{(d-1)/2} \cdot 6^{d-1}} \cdot 2^{-n} \cdot |u|_\infty \;=\; O(h) \; .
\end{aligned}$$

Man beachte, daß der Index „∞“ dreimal auftritt und dabei für drei verschiedene Bedeutungen steht: $^{1)}$ für die Verwendung der L_∞-Norm, $^{2)}$ für die Verwendung der L_∞-Halbnorm und $^{3)}$ als Symbol für den Approximationsraum $V_n^{(\infty)}$.

Lemma 6 zeigt, daß alle von d abhängigen Terme in den Schranken von n unabhängig sind. Erhöhen wir n, d. h., setzen wir die Maschenweite $h = 2^{-n}$ herab, so erzeugt d keine (nicht schon vorher dagewesenen) Probleme. h-asymptotisch ist der Fehler $O(h^2)$ bzw. $O(h)$, also von d unabhängig. Der Aufwand, um eine solche Genauigkeit zu realisieren, wächst aber wegen $|V_n^{(\infty)}| = O(h^{-d})$ exponentiell in d!

Beweis:

Wir beginnen mit der Abschätzung in der Maximumsnorm, wobei wir die Abschätzung für $\|u_{\underline{l}}\|_\infty$ aus Lemma 5 benutzen:

$$\begin{aligned}
\|u - u_n^{(\infty)}\|_\infty &= \Big\| \sum_{\underline{l}} u_{\underline{l}} - \sum_{|\underline{l}|_\infty \le n} u_{\underline{l}} \Big\|_\infty \\
&\le \sum_{|\underline{l}|_\infty > n} \|u_{\underline{l}}\|_\infty \\
&\le \frac{1}{2^d} \cdot |u|_\infty \cdot \sum_{|\underline{l}|_\infty > n} 2^{-2|\underline{l}|_1} \\
&= \frac{1}{2^d} \cdot |u|_\infty \cdot \Big(\sum_{\underline{l}} 4^{-|\underline{l}|_1} - \sum_{|\underline{l}|_\infty \le n} 4^{-|\underline{l}|_1} \Big) \\
&= \frac{1}{2^d} \cdot |u|_\infty \cdot \Big(\big(\sum_{k=1}^{\infty} 4^{-k} \big)^d - \big(\sum_{k=1}^{n} 4^{-k} \big)^d \Big) \\
&= \frac{1}{2^d} \cdot |u|_\infty \cdot \Big(\big(\frac{1}{3}\big)^d - \big(\frac{1}{3}\big)^d (1 - 4^{-n})^d \Big) \\
&= \frac{1}{6^d} \cdot |u|_\infty \cdot \Big(1 - (1 - 4^{-n})^d \Big) \\
&\le \frac{d}{6^d} \cdot |u|_\infty \cdot 4^{-n}
\end{aligned}$$

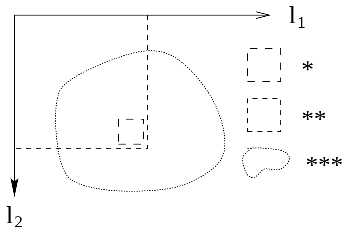
Für die L_2-Norm geht's ganz analog! Im Falle der Energieabschätzung erhalten wir wieder mit Lemma 5:

$$\begin{aligned}
\|u - u_n^{(\infty)}\|_E &\le \sum_{|\underline{l}|_\infty > n} \|u_{\underline{l}}\|_E \\
&\le \frac{1}{2 \cdot 12^{(d-1)/2}} \cdot |u|_\infty \cdot \sum_{|\underline{l}|_\infty > n} 2^{-2|\underline{l}|_1} \cdot \big(\sum_{j=1}^{d} 4^{l_j} \big)^{1/2} \\
&\le \frac{\sqrt{d}}{2 \cdot 12^{(d-1)/2}} \cdot |u|_\infty \cdot \sum_{|\underline{l}|_\infty > n} 2^{-2|\underline{l}|_1} \cdot \max_{1 \le j \le d} 2^{l_j} \\
&\le \frac{d^{3/2}}{2 \cdot 12^{(d-1)/2}} \cdot |u|_\infty \cdot \sum_{l_1 = |\underline{l}|_\infty > n} 2^{-2|\underline{l}|_1} \cdot 2^{l_1} \\
&\le \frac{d^{3/2}}{2 \cdot 12^{(d-1)/2}} \cdot |u|_\infty \cdot \sum_{l_1 > n} 2^{-l_1} \cdot \big(\sum_{l_j=1}^{l_1} 4^{-l_j} \big)^{d-1} \\
&= \frac{d^{3/2}}{2 \cdot 12^{(d-1)/2}} \cdot |u|_\infty \cdot \frac{1}{3^{d-1}} \cdot 2^{-n} \qquad \square
\end{aligned}$$

Zusammenfassung der Approximationseigenschaften von $V_n^{(\infty)}$:

- regelmäßiges Gitter der Maschenweite $h = 2^{-n}$
- $O(h^{-d})$ Punkte
- L_∞- bzw. L_2-Norm: Fehler $O(h^2)$
  Energienorm: Fehler $O(h)$

All dies ist natürlich keine neue Erkenntnis, die gezeigten Resultate sind Standard bei der Interpolation. Die hierarchische Teilraumzerlegung eröffnet aber den Weg zu besseren (d. h. effizienteren) Gittern, die gleiche Genauigkeit bei weniger Punkten bieten.



Die Standard-Approximationstheorie betrachtet $V_n^{(\infty)}$ nur als einen Teilraum $V_{\underline{l}}$ im $V_{\underline{l}}$-Schema (*). Für uns ist $V_n^{(\infty)}$ ein rechteckiges Fenster, also eine ganze Menge von Teilräumen $W_{\underline{l}}$ im $W_{\underline{l}}$-Schema (**). Daraus erwächst eine Optimierungsmöglichkeit: Andere Mengen von Teilräumen $W_{\underline{l}}$ (***) können in Betracht gezogen werden. Das werden wir in Kapitel 5 tun.

3.4 Datenstrukturen und Algorithmen

3.4.1 Datenstrukturen

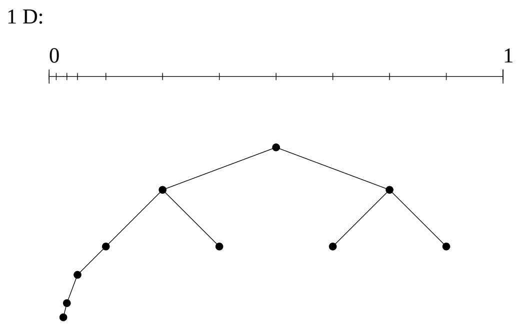
Solange mit herkömmlichen regulären Gittern gearbeitet wird (d. h. z. B. $V_n^{(\infty)}$), können arrays als Datenstruktur verwendet werden:

$$u\,[1\,..\,2^n-1, 1\,..\,2^n-1, 1\,..\,2^n-1]$$

für eine Funktion $u_n^{(\infty)}$ von drei Veränderlichen. Die Nachteile dieses Vorgehens sind offensichtlich:

- die hierarchische Struktur spiegelt sich nicht wider;
- nicht dimensionsrekursiv, sondern fest für ein d;
- lokale adaptive Verfeinerung bzw. allgemeinere Gitterstrukturen (vgl. die Bemerkung zuvor) sind problematisch.

Deshalb geht man über zu Baumstrukturen:



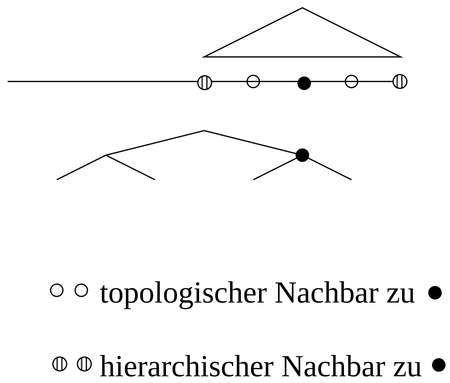
- Realisierung über Zeiger/Pointer/Verweise;
- Hierarchie der Teilraumzerlegung steckt in der Hierarchie der Datenstruktur;
- lokale Adaption leicht möglich.

Auch hier gibt's Nachteile:

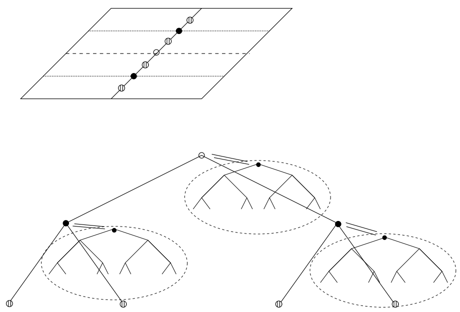
- bei starker Adaption Degenerierung zu Listen;
- Overhead durch Organisation des Geflechts (Zeiger definieren, umsetzen etc.):
  $\rightarrow$ klein bei Tensorproduktgittern
  $\rightarrow$ groß bei beliebig unstrukturierten Gittern
- keine Datenlokalität:
  Ein array liegt als Block im Speicher, Nachbarn liegen nebeneinander. Ein Geflecht liegt beliebig verteilt, Nachbarn (topologische und hierarchische) liegen u. U. weit entfernt im Speicher. Dies ist schlecht etwa für die Effizienz des Caches.

Nachbarschaftsrelationen:

Es gibt topologische Nachbarn (nebeneinanderliegende Gitterpunkte) und hierarchische Nachbarn bzw. Vorfahren (Randpunkte des Trägers der lokalen Basisfunktionen). Hierarchische Nachbarn sind Randpunkte des Trägers der aktuellen Basisfunktion und immer hierarchisch höher als der aktuelle Punkt.



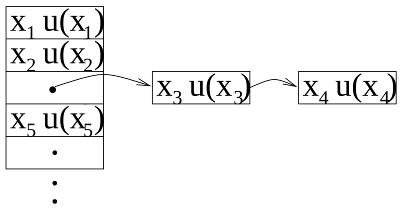
Die Verallgemeinerung auf Funktionen von d Veränderlichen erfolgt rekursiv: Jeder Knoten im d-dimensionalen Baum enthält einen (d-1)-dimensionalen Baum. Dies ist eine dimensionsrekursive Betrachtungsweise: Ein d-dimensionaler Baum ist ein eindimensionaler Baum von (d-1)-dimensionalen Bäumen, ein nulldimensionaler Baum ist eine Zahl.



Eine Alternative zu Baumstrukturen stellen Hash-Tabellen dar. Sie überwinden den Nachteil der fehlenden Datenlokalität und vermeiden zumindest teilweise den Overhead der Verwaltung. Hash-Techniken werden eingesetzt, wenn aus einer großen Menge potentieller Grundelemente (Schlüssel) wahrscheinlich nur wenige auftreten werden, so daß das Reservieren von Speicherplatz, der der Schlüsselmenge angemessen wäre, unangebracht ist. Bei uns sind die Schlüssel gerade alle möglichen Gitterpunkte in allen denkbaren Gittern Ω^n, $n \in \mathbb{N}$. Die Hash-Funktion H bildet diese Schlüsselmenge auf einen möglichst kompakten Speicherbereich ab:

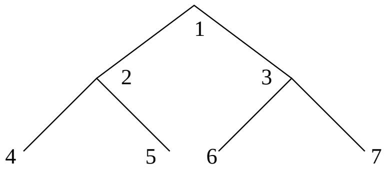
$$\{\text{ Menge aller potentiellen Gitterpunkte }\} \quad \xrightarrow{H} \quad \text{array } m[0..N] \; .$$

Wie wir schon im Einführungskapitel gesehen haben, wird H i. a. nicht injektiv sein: Wir wollen ja Speicherplatz sparen. Somit kann es zu Kollisionen kommen, etwa $H(x_3) = H(x_4)$ für zwei Gitterpunkte x_3 und x_4. Die Kollisionsbehandlung kann z. B. über lineare Überlauflisten erfolgen:



Zur Identifizierung der Gitterpunkte müssen wir eine Numerierung einführen. Zwei Möglichkeiten liegen nahe:

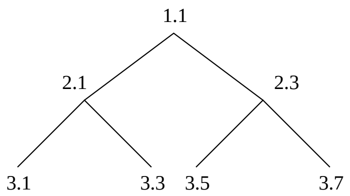
- hierarchische Numerierung (Level implizit erkennbar)



Vater(k) = k *div* 2
Söhne(k) = $2k,\ 2k+1$
Level(k) = # Bits_in_Dualdarstellung

  Im d-dimensionalen Fall erhält man als „Nummer" eines Gitterpunkts ein d-Tupel, also z. B. $(17, 6, 9, 153)$.

- hierarchische Numerierung (Level explizit erkennbar)



  Hier wird der Gitterpunkt $x_{\underline{l},\underline{i}}$ also über den Index $\underline{l}$, der den Level angibt, und über den Index $\underline{i}$, der die fortlaufende Nummer des Punkts innerhalb eines Levels angibt, identifiziert.

  Im d-dimensionalen Fall erhält man als „Nummer" eines Gitterpunkts $x_{\underline{l},\underline{i}}$ ein $(2 \cdot d)$-Tupel $(l_1, i_1, l_2, i_2, \ldots, l_d, i_d)$.

Die Konstruktion von H ist das Kernstück jedes Hash-Verfahrens. Hier gibt es konkurrierende Ziele. Einerseits soll die Darstellung möglichst kompakt sein, d. h., N ist möglichst klein zu wählen. Andererseits sollen Kollisionen vermieden werden, weil diese den Aufwand erhöhen und die Hash-Charakteristik (Direktzugriff auf alle Elemente) stören. Deshalb darf N auch nicht zu klein gewählt werden, bzw. es muß bei Bedarf heraufgesetzt werden (was natürlich dann zu einer Neubestimmung von H etc. führt).

Man muß jedoch nicht alles neu programmieren bzw. erfinden, da es zu Hash-Techniken eine Reihe von Standard-Lösungen gibt (C++ Templates, Perl etc.).

3.4.2 Algorithmen

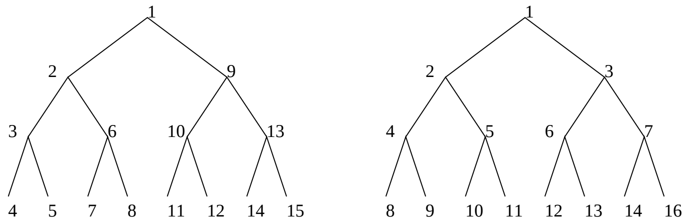
wesentliche Operationen:

- hierarchisieren (Transformation Stützpunktbasis $\to$ hierarchische Basis,
  Funktionswerte $\mapsto$ hierarchische Überschüsse)
- dehierarchisieren (Umkehrtransformation)
- arithmetische Operatoren („Algebra“):

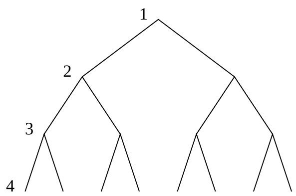
$$\begin{array}{rcl} u_{1,n}^{(\infty)} & + & u_{2,n}^{(\infty)} \\ u_{1,n}^{(\infty)} & - & u_{2,n}^{(\infty)} \\ \alpha & \cdot & u_{n}^{(\infty)} \\ u_{1,n}^{(\infty)} & \cdot & u_{2,n}^{(\infty)} \\ u_{1,n}^{(\infty)} & : & u_{2,n}^{(\infty)} \end{array}$$

Hauptbaustein hierzu ist der Baumdurchlauf, für den es mehrere Strategien gibt:

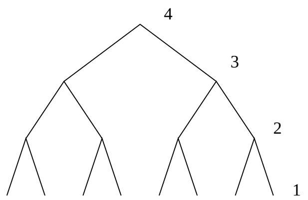
(i) depth first / breadth first (Besuchsreihenfolge)



(ii) top down / bottom up (Arbeitsreihenfolge)



„Söhne nach Vater“



„Vater nach Söhnen“

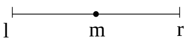
Eine Standard-Rekursion sieht dann beispielsweise wie folgt aus:

```
proc durchlauf (tree t)
  if t= nil then skip ;
  else
      ⟨top-down-Arbeit⟩;
      durchlauf (left_son(t));
      durchlauf (right_son(t));
      ⟨bottom-up-Arbeit⟩;
```

Dieser Algorithmus folgt dem depth-first-Prinzip. Alle Anweisungen im Block „top-down-Arbeit“ werden vor dem weiteren Abstieg erledigt, also top-down. Dementsprechend werden die Anweisungen im Block „bottom-up-Arbeit“ erst nach Rückkehr von den Söhnen ausgeführt.

Hierarchisierung:

$$v_m \;:=\; u_m - \frac{1}{2}(u_l + u_r)$$



x_m: aktueller Punkt mit Basisfunktion ϕ_m
x_l, x_r: hierarchische Nachbarn von x_m (Randpunkte des Trägers von ϕ_m)

Die Hierarchisierung ist als top-down- oder als bottom-up-Operation realisierbar. Wenn u und v allerdings alternativ verwandt werden sollen (d. h. nur ein Speicherplatz pro Gitterpunkt, v überschreibt u), so muß bottom-up-mäßig vorgegangen werden, weil die hierarchischen Nachbarn in Stützpunktbasisdarstellung benötigt werden.

Dehierarchisierung:

$$u_m := \frac{1}{2}(u_l + u_r) + v_m$$

Die Dehierarchisierung ist ebenfalls als top-down- oder bottom-up-Operation realisierbar. Bei einer Transformation „am Platz“ (Überschreibung) muß jetzt allerdings top-down vorgegangen werden.

Algebra:

Seien $u_1, u_2 \in X_0(\bar{\Omega})$, $u_{1,n}^{(\infty)}$ und $u_{2,n}^{(\infty)}$ deren Interpolanten in $V_n^{(\infty)}$ in hierarchischer Darstellung. Gesucht sind die Lösungen der oben angeführten arithmetischen Operationen Summe, Differenz, skalare Multiplikation, Produkt und Quotient, und zwar jeweils wiederum in hierarchischer Darstellung.

Die Addition ist einfach. Hier müssen nur die hierarchischen Koeffizienten addiert werden (also komponentenweise Addition):

$$\begin{aligned}
v_{1+2,m} &= u_{1+2,m} - \frac{1}{2}(u_{1+2,l} + u_{1+2,r}) \\
&= u_{1,m} + u_{2,m} - \frac{1}{2}(u_{1,l} + u_{2,l} + u_{1,r} + u_{2,r}) \\
&= u_{1,m} - \frac{1}{2}(u_{1,l} + u_{1,r}) + u_{2,m} - \frac{1}{2}(u_{2,l} + u_{2,r}) \\
&= v_{1,m} + v_{2,m}
\end{aligned}$$

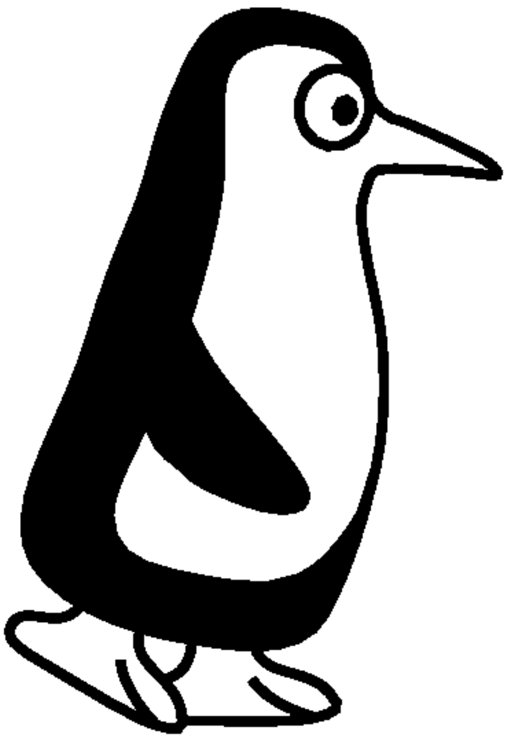
Die Subtraktion erfolgt analog!

Die Multiplikation ist komplizierter, da hier das Produkt $v_{1,m} \cdot v_{2,m}$ der hierarchischen Überschüsse offensichtlich nicht funktioniert. Als einfache Abhilfe bietet sich an, die Faktoren $u_{1,n}^{(\infty)}$ und $u_{2,n}^{(\infty)}$ zu dehierarchisieren, dann das Produkt zu bilden und das Ergebnis wieder zu hierarchisieren. Hierbei ist aber zu beachten, daß das Resultat nur in den Gitterpunkten mit dem Produkt von $u_{1,n}^{(\infty)}$ und $u_{2,n}^{(\infty)}$ übereinstimmt, da dieses nicht mehr in $V_n^{(\infty)}$ liegt. Schließlich ergibt das Produkt zweier stückweise linearer Funktionen eine stückweise quadratische Funktion!

Bemerkung:

Zur Parameterversorgung der rekursiven Routinen:

Bei einer Realisierung als Geflecht empfiehlt es sich, die Verweise auf die hierarchischen Nachbarn als Aufrufparameter mit zu übergeben. Bei Hashs ist dies nicht erforderlich, da man ja (über Level und Nummer) Direktzugriff auf die Nachbarn hat.



Leider war der Autor zu XXXX, um hier noch weiter zu schreiben.

Kapitel 4

Partielle Differentialgleichungen, Finite Elemente

4.1 Diskretisierung partieller Differentialgleichungen

Anstelle von der Diskretisierung partieller Differentialgleichungen reden wir lieber von der Diskretisierung eines Randwert- oder eines Anfangsrandwertproblems, bestehend aus einer oder mehreren partiellen Differentialgleichungen sowie Rand- und Anfangsbedingungen. Letztere sind i. d. R. dazu da, die Eindeutigkeit der Lösung sicherzustellen.

Eine partielle Differentialgleichung (PDE) bestimmt eine Funktion u, die von mehreren Variablen abhängt, wobei partielle Ableitungen nach mehreren Variablen auftreten. Typischerweise handelt es sich bei den Variablen um Raumkoordinaten x, y bzw. x, y, z sowie um die Zeit t.

Ein einfaches Beispiel einer (rein ortsabhängigen) PDE ist die allgemeine lineare partielle Differentialgleichung 2. Ordnung in d Variablen:

$$\sum_{i,j=1}^{d} a_{ij}(\underline{x}) u_{x_i x_j}(\underline{x}) \ + \ \sum_{i=1}^{d} a_i(\underline{x}) u_{x_i}(\underline{x}) \ + \ a(\underline{x}) u(\underline{x}) \ = \ f(\underline{x}) \ .$$

Hier unterscheidet man drei Typen:

„elliptisch in x“:	die (o. B. d. A. symmetrische) Matrix $A(\underline{x}) = \big(a_{ij}(\underline{x})\big)_{i,j}$ ist positiv definit oder negativ definit
„hyperbolisch in x“:	A hat einen positiven und $n-1$ negative Eigenwerte oder umgekehrt
„parabolisch in x“:	ein Eigenwert von A ist Null, $n-1$ Eigenwerte haben gleiches Vorzeichen, Rang $(A(\underline{x}), c(\underline{x})) = n$ mit $c(\underline{x}) = (a_i(\underline{x}))_i$

Dementsprechend wird eine PDE elliptisch/parabolisch/hyperbolisch in Ω genannt, wenn obige Beziehungen für alle $x \in \Omega$ gelten. Zum besseren Verständnis der Begriffsbildung erinnere man sich an Quadriken (Ellipsoide, Hyperboloide, Paraboloide etc.).

Einige Beispiele:

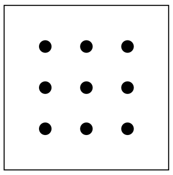
elliptisch	Potentialgleichung/Laplace-Gleichung	$\Delta u = u_{xx} + u_{yy} = 0$
hyperbolisch	Wellengleichung	$u_{xx} - u_{yy} = 0$
parabolisch	Wärmeleitungsgleichung	$u_{xx} - u_t = 0$

Die drei Typen haben stark unterschiedliche Eigenschaften und erfordern daher auch unterschiedliche Techniken zu ihrer numerischen Behandlung. Für uns ist hier vor allem der elliptische Typ von Interesse.

Diskretisierungsansätze:

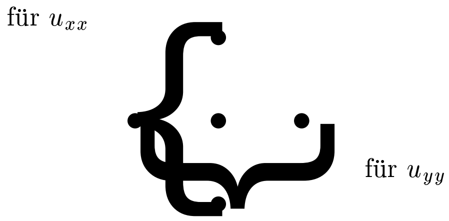
(i) Finite Differenzen:
approximiere den Differentialquotienten (Ableitung) durch einen Differenzenquotient

$$
\begin{aligned}
\frac{\partial u}{\partial x}(\xi) &\doteq \frac{u(\xi+h)-u(\xi)}{h}, \frac{u(\xi+h)-u(\xi-h)}{2h}, \frac{u(\xi)-u(\xi-h)}{h} \\
\frac{\partial^2 u}{\partial x^2}(\xi) &\doteq \frac{u(\xi+h)-2u(\xi)+u(\xi-h)}{h^2} \\
&\quad \ldots
\end{aligned}
$$



Man stellt die Differenzengleichung in jedem Gitterpunkt auf und erhält so ein System von M linearen Gleichungen in M Unbekannten. Dirichlet-Randwerte (u am Rand gegeben) wandern in die rechte Seite des Gleichungssystems.

Ein zweidimensionales Beispiel: Zu lösen sei $\Delta u = 0$ auf einem $N \times N$ Gitter, es gibt also $M := (N-2)^2$ innere Gitterpunkte. Für die Matrix A gilt $A \in \mathbb{R}^{M \times M}$, für die rechte Seite b gilt $b \in \mathbb{R}^M$ und für den Lösungsvektor x ebenfalls $x \in \mathbb{R}^M$. Bei passender Anordnung der Gitterpunkte hat A Bandstruktur (in jeder diskreten Gleichung sind nur 5 Elemente involviert):



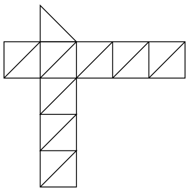
Sinnvollerweise sollte gelten, daß ein feineres Gitter (ein größeres N bzw. eine kleinere Maschenweite h) zu einer besseren Näherungslösung führt.

(ii) Finite-Elemente:
siehe Abschnitt 4.2

(iii) Finite Volumen:
Die interessierenden physikalischen Zusammenhänge können i. d. R. auf Erhaltungsgesetze reduziert werden (Massenerhaltung, Impulserhaltung etc.). Bei Finite-Volumen-Verfahren wird eine solche Erhaltung für jedes einzelne Teilvolumen, in die das Grundgebiet zerlegt ist, gefordert. Kurz: Der Massenfluß in eine Zelle hinein muß gleich dem Massenfluß aus einer Zelle heraus sein. Die Wortwahl deutet es schon an: Finite-Volumen-Verfahren sind vor allem in der numerischen Strömungsmechanik verbreitet.

4.2 Finite Elemente Verfahren

Sehweise des Ingenieurs: zerlege ein kompliziertes Gebilde in einzelne Teilgebiete endlicher Größe (d. h. nicht infinitesimal klein); löse hierauf geeignet formulierte Probleme und baue aus diesen „Elementlösungen“ die Gesamtlösung;



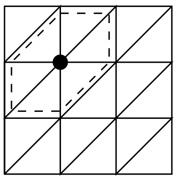
Kran → FEM kommt aus Statik/Elastizitätstheorie/Mechanik

Sehweise des Mathematikers: Zerlegung wie oben; gehe über zur „schwachen Form“ der Differentialgleichung (Variationsansatz), definiere auf den Elementen Ansatz- und Testfunktionen; Prinzip der gewichteten Residuen (Galerkin, Ritz-Galerkin) führt auf ein lineares Gleichungssystem;

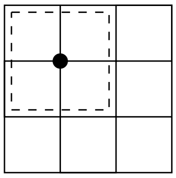
Bearbeitungsschritte:

1) Gittergenerierung:

Zerlegung des Gebiets in einzelne „Elemente“; die Eckpunkte, an denen verschiedene Elemente zusammentreffen, bilden ein Gitter Ω_n



bzw.



Jedem Gitterpunkt x_k ist eine Ansatzfunktion zugeordnet, beispielsweise eine Pagodenfunktion im zweidimensionalen Fall. Diese verschwindet auf allen außer den angrenzenden finiten Elementen (endlicher Träger). Alle Ansatzfunktionen zusammen spannen einen linearen Raum V_n auf:

$$V_n \;:=\; \operatorname{span}\{\varphi_k : x_k \in \Omega_n\} \;,$$

in der Tat bilden die φ_k eine Basis von V_n. In V_n suchen wir eine Approximation der Lösung unserer partiellen Differentialgleichung bzw. unseres Randwertproblems.

2) Schwache Form der partiellen Differentialgleichung:

Wir schwächen $Lu = f$ auf Ω ab zu

$$\int_\Omega Lu \cdot \psi_l \; d\Omega \;=\; \int_\Omega f \cdot \psi_l \; d\Omega$$

für bestimmte Testfunktionen ψ_l (Methode der gewichteten Residuen, Galerkin-Ansatz). Die Testfunktionen ψ_l kann man auf verschiedene Art wählen. Sind die Testfunktionen mit den Ansatzfunktionen identisch, spricht man von Ritz-Galerkin-Verfahren, andernfalls von Petrov-Galerkin-Verfahren. Im ersten Fall bedeutet die schwache Form also

$$\int_\Omega Lu \cdot \varphi \; d\Omega \;=\; \int_\Omega f \cdot \varphi \; d\Omega \qquad \forall \varphi \in V_n \;.$$

Aufgrund der Linearität reicht es, wenn man dies für alle Basisfunktionen φ_k fordert. Mit der Bilinearform

$$a(u,v) \ := \ \int_\Omega Lu \cdot v \ d\Omega$$

und der Linearform

$$b(v) \ := \ \int_\Omega f \cdot v \ d\Omega$$

für $u,v \in V_n$ erhält man schließlich folgendes System linearer Gleichungen:

$$a(u,\varphi_k) \ = \ b(\varphi_k) \qquad \forall \ \varphi_k \in V_n .$$

3) Diskrete Approximation:

Jetzt bedenken wir, daß wir unsere Approximation u_n in V_n suchen, also als Linearkombination der Basisfunktionen φ_l:

$$u_n = \sum_l \alpha_l \cdot \varphi_l \ .$$

Diesen Ansatz für u_n setzen wir jetzt in das aus der schwachen Form abgeleitete Gleichungssystem ein und erhalten

$$a(u,\varphi_k) \ = \ a\left(\sum_l \alpha_l \cdot \varphi_l \ , \ \varphi_k\right) \ = \ \sum_l \alpha_l \cdot a(\varphi_l,\varphi_k) \ = \ b(\varphi_k) \qquad \forall \varphi_k \in V_n \ .$$

Alle $a(\varphi_l,\varphi_k)$ und $b(\varphi_k)$ hängen nur vom gegebenen Problem, nicht jedoch von der aktuellen Näherungslösung ab. Folglich müssen sie nur einmal (z. B. zu Beginn) berechnet werden.

Somit haben wir ein lineares Gleichungssystem $Ax = b$ in den Unbekannten α_i:

$$\begin{aligned} A &= \left(a(\varphi_i,\varphi_j)\right)_{i,j} \ , \\ b &= \left(b(\varphi_i)\right)_i + \ < \text{Einfluß der Randbedingungen} > \ , \\ x &= \left(\alpha_i\right)_i \ . \end{aligned}$$

Die Matrix A wird hierbei Steifigkeitsmatrix genannt.

4) Lösung des linearen Gleichungssystems:

Die Lösung x des linearen Gleichungssystems $Ax = b$ stellt gerade die Finite-Elemente-Approximation in V_n zur Lösung unseres Ausgangsproblems dar. Somit ist die Steifigkeitsmatrix mit ihrer Gestalt und ihren Eigenschaften für den Lösungsprozeß von entscheidender Bedeutung. Optimal ist natürlich eine Diagonalmatrix (d. h., die Basis $\{\varphi_k\}$ ist a-orthogonal), dann ist das Lösen von $Ax = b$ trivial. Einige abschließende Bemerkungen zu A:

- Oft ist A symmetrisch positiv definit.
- Bei Verwendung von Stützpunktbasen ist A dünn besetzt und hat Bandstruktur. Denn für alle Basisfunktionen φ_i und φ_j, deren Träger sich nicht überlappen, gilt

  $$a_{i,j} = a(\varphi_i,\varphi_j) \ = \ \int_\Omega L\varphi_i \cdot \varphi_j \ d\Omega \ = \ 0 \ .$$

  Hier wird der Term „finite“ Elemente klar.
- Bei hierarchischen Basen überlappen wesentlich mehr Elemente. Der Träger der hierarchisch höchsten Basisfunktion etwa (aus $W_{(1,\ldots,1)}$) überlappt mit allen Elementen. Man bekommt einen fill-in, die Matrix ist aber nach wie vor dünn besetzt (die Elemente aus jedem Teilraum $W_{\underline{l}}$ sind paarweise disjunkt). Durchschnittlich ergeben sich pro Zeile $O(n)$ Nicht-Null-Einträge, wenn n die Auflösung des Raumes angibt ($h = 2^{-n}$).

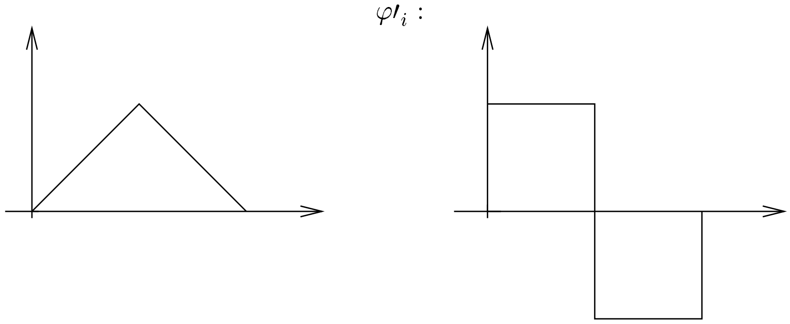
- Das Ziel muß eine Basis sein, die zu möglichst einfach bzw. schnell zu lösenden linearen Gleichungssystemen führt. Im Idealfall ist A diagonal, zumindest aber sollte A so beschaffen sein, daß die gängigen Iterationsverfahren (siehe Abschnitt 4.3) schnell konvergieren.

Wir betrachten als einfaches Beispiel die eindimensionale Poisson-Gleichung $-\Delta u = f$. Die erste Greensche Formel liefert

$$a_{i,j} \;=\; \int\limits_{\Omega} \nabla\varphi_i \cdot \nabla\varphi_j \; d\Omega \; .$$

Bei Verwendung der stückweise linearen hierarchischen Basis ergibt sich folgendes Bild:

φ_i : $\qquad$ $\varphi\prime_i$:



Offensichtlich gilt $a_{i,j} \neq 0 \;\Leftrightarrow\; i = j$. Somit haben wir einen der seltenen Fälle, in denen es gelingt, eine diagonale Steifigkeitsmatrix zu bekommen!

<u>Konvergenzaussagen:</u>

Von zentraler Bedeutung ist die Eigenschaft der Best-Approximation (Céa-Lemma):

Sei $u_n^{FE} \in V_n$ die Finite-Elemente-Approximation zur Lösung u des Randwertproblems mit symmetrischem und positiv definitem a bzw. A. Dann gilt in der Norm $\|u\|_a := \sqrt{a(u,u)}$ (im Poisson-Fall gerade unsere Energie-Norm):

$$\|u - u_n^{FE}\|_a \;\leq\; \inf_{u_n \in V_n} \|u - u_n\|_a \; .$$

Die Finite-Elemente-Lösung ist also Best-Approximation von u in V_n.

Die Abschätzung des Interpolationsfehlers in der Energienorm nach Lemma 6 ($\|\cdot\|_E \equiv \|\cdot\|_a$ für Poisson/Laplace) ist obere Schranke für den Fehler der Finite-Elemente-Lösung u_n^{FE}, da der Interpolant $u_n^{(\infty)} \in V_n$ zulässiger Kandidat bei der Suche nach dem Infinum ist. Daraus folgt, daß Konvergenzaussagen in der Finite-Elemente-Welt bzgl. der (problemabhängigen) $\|\cdot\|_a$ -Norm i. a. leicht, bzgl. der (problemunabhängigen) L_∞- bzw. L_2-Norm dagegen oft ungleich schwerer sind.

<u>Zentrale Aufgabe in der (mathematischen) Finite-Elemente-Welt:</u>

Finde endlichdimensionale Approximationsräume V_n mit Basen $\{\varphi_k\}$, so daß erstens die jeweiligen Näherungslösungen u_n die Lösung u gut (und mit wachsendem n immer besser) approximieren und zweitens die u_n leicht (d. h. billig) zu berechnen sind.

Bisher haben wir nur die (zugegebenermaßen verbesserungsfähigen) Räume $V_n^{(\infty)}$ kennengelernt, in Kapitel 5 kommen neue!

4.3 Schnelle Lösung der Gleichungssysteme

Wir betrachten ein lineares Gleichungssystem $Ax = b$ mit einer sehr großen, dünn besetzten Matrix A (i. a. symmetrisch positiv definit), die zu groß für eine direkte Lösung (etwa mittels Gauß-Elimination) sei. Deshalb müssen iterative Verfahren eingesetzt werden. Mit wachsender Genauigkeit der Diskretisierung n steigt auch die Anzahl der Unbekannten und somit die Dimension der Matrix A. Gesucht sind Verfahren, deren Konvergenzgeschwindigkeit nicht von n abhängt.

Als Beispiel betrachten wir den Raum $V_n^{(\infty)}$. Im d-dimensionalen Fall gilt bekanntlich $|V_n^{(\infty)}| = (2^n - 1)^d =: N$. Für die Matrix A gilt folglich $A \in \mathbb{R}^{N \times N}$. Sei $\rho \in]0,1[$ der Faktor, um den der Fehler in jedem Iterationsschritt eines geeigneten iterativen Verfahrens zur Lösung von $Ax = b$ reduziert wird:

$$\|x - x^{(it+1)}\| \;<\; \rho \cdot \|x - x^{(it)}\| \; .$$

Nach it Schritten haben wir also eine Reduktion um ρ^{it}. Ziel muß es sein, Verfahren zu konstruieren, bei denen ρ nicht von N und damit nicht von n abhängt.

$$x^{(0)}(\text{Start}) \;\longrightarrow \ldots \longrightarrow x^{(it)} \longrightarrow x^{(it+1)} \longrightarrow \ldots \longrightarrow x^{(\infty)} \;\equiv\; x \;\equiv u_n^{FE}$$

4.3.1 Klassische Iterationsverfahren

<u>Jacobi- und Gauß-Seidel-Iteration</u>

- Jacobi: for k = 0,1,...

  for i = 1,...,N: $y_i \;:=\; \frac{1}{a_{ii}} \cdot \left(b_i - \sum_{j=1}^{N} a_{ij} x_j \right)$

  for i = 1,...,N: $x_i \;:=\; x_i + y_i$

  d. h. in jedem Iterationsschritt:
  Korrigiere x_i so, daß die i-te Gleichung erfüllt ist. Führe die Korrekturen aber erst am Ende des Schritts aus.

- Gauß-Seidel: for k = 0,1,...

  for i = 1,...,N: $y_i \;:=\; \frac{1}{a_{ii}} \cdot \left(b_i - \sum_{j=1}^{N} a_{ij} x_j \right)$

  $x_i \;:=\; x_i + y_i$

  jetzt: Führe jede Korrektur sofort aus!

Manchmal erweist sich eine Dämpfung bzw. eine Überrelaxation als vorteilhaft:

$$x_i \;:=\; x_i + \alpha_i \cdot y_i \; , \qquad \alpha_i \;\in]0,2[\; .$$

<u>Konvergenz:</u>

- Jacobi: hinreichende Bedingung: A und $(2 \cdot \operatorname{diag}(A) - A)$ sind positiv definit;
- Gauß-Seidel: A ist positiv definit;

Bei beiden Verfahren haben wir das Problem, daß ρ von der Diskretisierungstiefe n abhängt. Bei der Poisson-Gleichung ergibt sich beispielsweise ($V_n^{(\infty)}$, finite Elemente mit stückweise linearen Stützpunktbasen)

$$\rho \;=\; O(1 - h^2) \;=\; O(1 - \frac{1}{4^n}) \; .$$

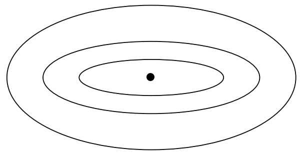
Das ist fatal: Je feiner wir diskretisieren, je größer also unsere Gleichungssysteme werden, desto langsamer konvergiert das Lösungsverfahren!

4.3.2 cg

Ausgangspunkt:

Ausgangspunkt ist hier die Erkenntnis, daß unter bestimmten Bedingungen die Lösung des linearen Gleichungssystems $Ax = b$ gleichbedeutend mit einem quadratischen Optimierungsproblem ist:

$$Ax = b \quad \overset{A \text{ positiv definit}}{\Longleftrightarrow} \quad F(x) = \min_y F(y), \quad F(y) = \frac{1}{2} y^T A y - b^T y \ .$$



Höhenlinien von $F(y)$, Minimum „in der Mitte“

steepest descent:

Die Methode des steilsten Abstiegs sucht immer in Richtung des negativen Gradienten. Die Iteration braucht u. U. beliebig lange, bis sie konvergiert.

cg (conjugate gradients):

Beim Verfahren der konjugierten Gradienten sucht man in „konjugierten Richtungen“: $Ap \perp q$. Die wichtige Konsequenz ist, daß man das bisher Erreichte nicht wieder zerstören kann, da bei Optimierung in q-Richtung die vorige Optimierung in p-Richtung „erhalten“ bleibt. Somit ist das cg-Verfahren bei einer $N \times N$-Matrix nach N Schritten im Optimum. Theoretisch handelt es sich also um einen direkten Löser, praktisch ist N zu groß, und cg wird als iteratives Verfahren eingesetzt.

Konvergenz:

Wir betrachten dasselbe Beispiel wie zuvor (Poisson, $V_n^{(\infty)}$, finite Elemente mit stückweise linearen Stützpunktbasen). Es ergibt sich

$$\rho \ \approx \ 1 - \frac{1}{\sqrt{\mathrm{cond}(A)}} \ \approx \ 1 - h = 1 - \frac{1}{2^n} \ .$$

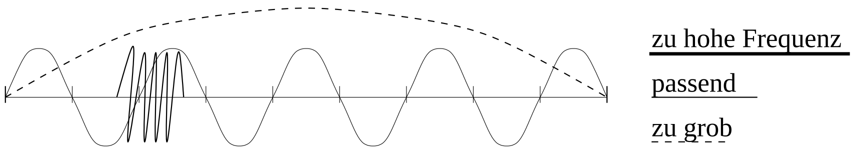
Die Kondition einer Matrix A ist dabei definiert als Verhältnis von betragsgrößtem zu betragskleinstem positiven Eigenwert:

$$\mathrm{cond}(A) \ := \ \frac{\lambda_{\max}(A)}{\lambda_{\min}(A)} \ .$$

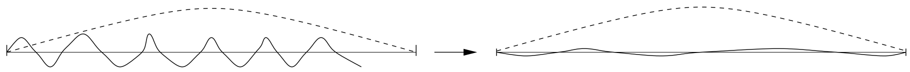
Wir stehen folglich wieder vor dem Problem, daß sich bei wachsender Feinheit der Diskretisierung, also bei steigender Auflösung n, das Konvergenzverhalten drastisch verschlechtert. Bereits bei Tiefe $n = 10$ wird der Fehler pro Iterationsschritt nur um einen Faktor $\rho \approx 1 - \frac{1}{2^n} \approx 0.999$ reduziert.

4.3.3 Abhilfen

- Beim cg-Verfahren ist Vorkonditionierung der Stein der Weisen: Anstelle von $Ax = b$ lösen wir $WAx = Wb$ mit einem geeigneten Vorkonditionierer W. Ideal ist natürlich $W = A^{-1}$. Da das aber viel zu teuer ist, sucht man nach einfach zu berechnenden W mit großer Wirkung (etwa $\text{cond}(WA) = O(1)$).
- Bei den Verfahren aus Abschnitt 4.3.1 haben Mehrgittermethoden den Durchbruch gebracht. Um das Prinzip zu verstehen, schauen wir die Fouriertransformation des Fehlers an. Diese zeigt Anteile hoher und Anteile niedriger Frequenzen. Für jedes Gitter gilt demzufolge: Es gibt für dieses Gitter zu feine Fehleranteile, passende und zu grobe.

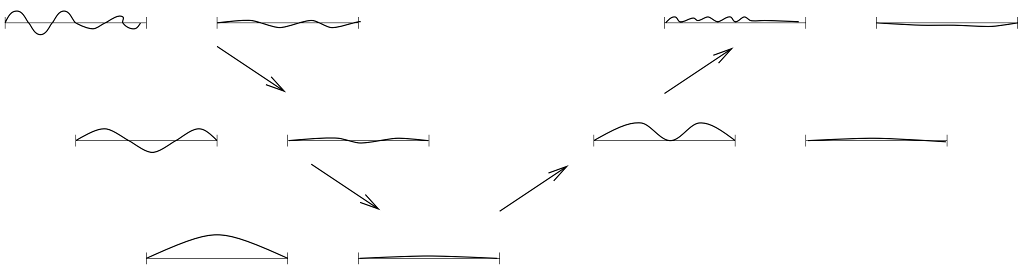


Die zu feinen kann man auf diesem Gitter sowieso nicht darstellen. Für die zu groben ist das Gitter unnötig fein – ein gröberes (und damit weniger Rechenaufwand produzierendes) Gitter wäre ausreichend. Zudem glätten Jacobi und Gauß-Seidel den Fehler:



Die groben Fehleranteile werden also kaum reduziert. Dies ist der Grund für das immer schlechtere Konvergenzverhalten bei wachsendem n.

Deshalb verwendet man Folgen von Gittern der Maschenweiten h, $2h$, $4h$, $8h$. Auf jedem von diesen Gittern glättet man die „passenden“ Fehleranteile heraus (das geht sehr schnell – einige wenige Iterationsschritte reichen).



"Mehrgitterverfahren"

Mit solchen „V-Zyklen“ als Iterationsschritte ist $\rho = 1 - c$, $c = \text{const.}$ erreichbar! Die Anzahl von Iterationsschritten hängt somit nicht mehr von der Diskretisierungsfeinheit ab.

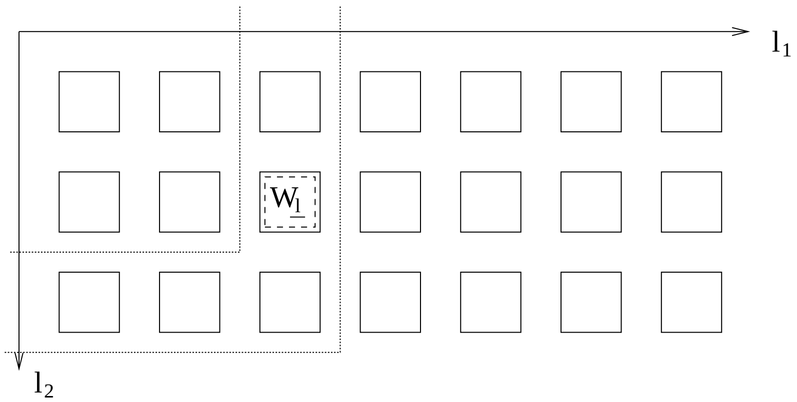
Mehrgitterverfahren und nach dem Mehrgitterprinzip konstruierte Multilevel-Vorkonditionierer sind state-of-the-art!

Kapitel 5

Dünne Gitter

5.1 Herleitung der Dünngitterräume

Nach dem Intermezzo über partielle Differentialgleichungen und finite Elemente gehen wir wieder zurück zur hierarchischen Teilraumzerlegung von Kapitel 3:



In Abschnitt 3.3 hatten wir dort folgendes gezeigt:

Lemma 1: $|W_{\underline{l}}| \;=\; 2^{|\underline{l}-\underline{1}|_1}$

Lemma 2: $||\phi_{\underline{l},\underline{i}}||_{\{\infty,2,E\}} \;=\; \ldots$

Lemma 3: $v_{\underline{l},\underline{i}} \;=\; \int \ldots$

Lemma 4: $|v_{\underline{l},\underline{i}}| \;\leq\; \ldots$

Lemma 5:
$$\begin{aligned} ||u_{\underline{l}}||_\infty &= O(2^{-2|\underline{l}|_1}) \\ ||u_{\underline{l}}||_2 &= O(2^{-2|\underline{l}|_1}) \\ ||u_{\underline{l}}||_E &= O\left(2^{-2|\underline{l}|_1} \cdot \left(\sum_{j=1}^{d} 4^{l_j}\right)^{1/2}\right) \end{aligned}$$

Bisher wurden als (endlichdimensionale) Approximationsräume ausschließlich die Räume

$$V_n^{(\infty)} = \bigoplus_{\underline{l}_\infty \leq n} W_{\underline{l}}$$

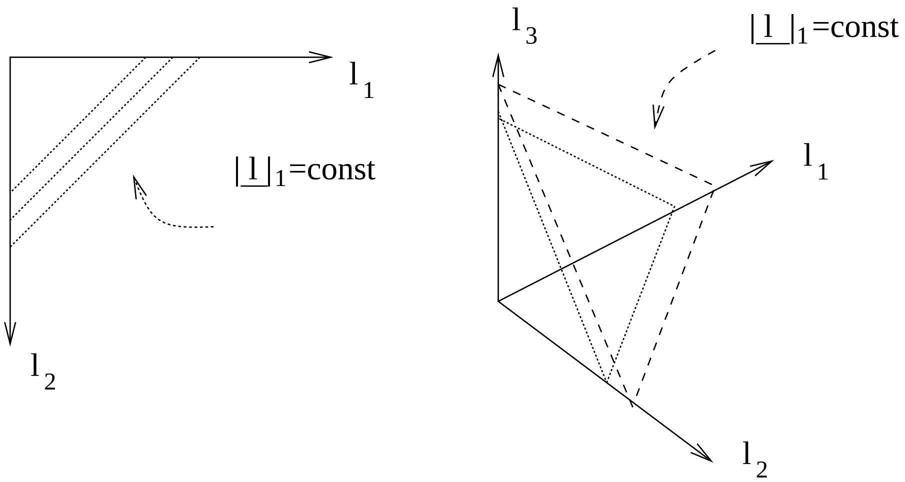
betrachtet. Die zugehörigen Gitter sind voll, d. h. äquidistant mit Maschenweite 2^{-n} in jeder Koordinatenrichtung. Für eine Bewertung dieser Räume betrachten wir ihre Kosten, d. h. die Anzahl der involvierten Freiheitsgrade bzw. Gitterpunkte, sowie ihren Nutzen, also die Genauigkeit der Interpolation bzw. den Interpolationsfehler. Hierbei gilt für die Kosten

$$|V_n^{(\infty)}| = (2^n - 1)^d = O(2^{n \cdot d}) = O(h^{-d}),$$

und über den Nutzen gibt Lemma 6 Auskunft:

$$\begin{aligned}
\|u - u_n^{(\infty)}\|_\infty &= O(4^{-n}) &= O(h^2)\,, \\
\|u - u_n^{(\infty)}\|_2 &= O(4^{-n}) &= O(h^2)\,, \\
\|u - u_n^{(\infty)}\|_E &= O(2^{-n}) &= O(h)\,.
\end{aligned}$$

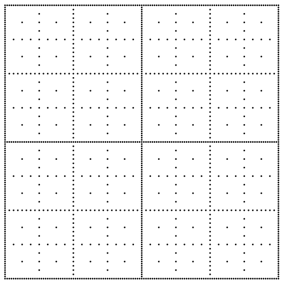
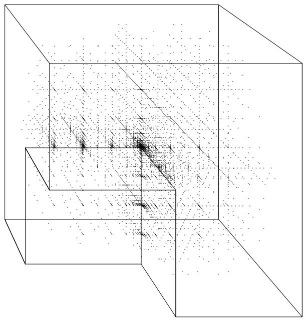
Jetzt machen wir uns auf die Suche nach Approximationsräumen V_n mit besserem Kosten-Nutzen-Verhältnis. Dazu bieten Lemma 1 und Lemma 5 einen Ansatzpunkt. Sowohl die Kosten (Punktezahl) als auch der Nutzen (Beitrag zum Interpolanten) eines Teilraums $W_{\underline{l}}$ hängen ab von $|\underline{l}|_1 = l_1 + \ldots + l_d$, wenn wir den Nutzen in der L_∞- bzw. in der L_2-Norm messen. Deshalb erscheint es als besser, dreieckige (simpliziale) Fenster aus dem Teilraumschema auszuschneiden und als Approximationsräume zu verwenden:



Dies führt auf die Approximationsräume $V_n^{(1)}$,

$$V_n^{(1)} := \bigoplus_{|\underline{l}|_1 \leq n+d-1} W_{\underline{l}}\,.$$

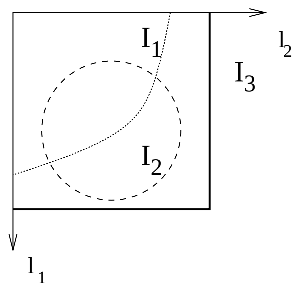
Die den Räumen $V_n^{(1)}$ entsprechenden Gitter heißen (L_∞- bzw. L_2-) dünne Gitter und stehen im Zentrum des Interesses in diesem Kapitel.

Unsere zunächst heuristische Betrachtung wollen wir jetzt durch die formale Herleitung eines Optimalitätskriteriums auf sichere Beine stellen. Dazu bieten sich zwei Wege an, ein kontinuierlicher mit Methoden der Analysis und ein diskreter mit Methoden der kombinatorischen Optimierung.

Kontinuierliche Optimierung:

- Der Multiindex $\underline{l} \in \mathbb{N}^d$ wird zunächst zu einem rellen Multiindex $\underline{l} \in \mathbb{R}_+^d$ verallgemeinert. Anschließend definieren wir für jedes solche $\underline{l} \in \mathbb{R}_+^d$ die lokale Kostenfunktion $c(\underline{l})$,
$$c(\underline{l}) \;:=\; 2^{|\underline{l}-\underline{1}|_1},$$
sowie die lokale Nutzenfunktion $p(\underline{l})$, die sich analog aus einer Verallgemeinerung der in Lemma 5 angegebenen oberen Schranken für $||u_{\underline{l}}||^2$ ergibt.
- Unter einem (zugegebenermaßen verallgemeinerten) Gitter I versteht man jetzt eine Menge $I \subset \mathbb{R}_+^d$ bzw. $I \subset [0, N]^d =: I_{\max}$ mit hinreichend großem N.
- Die Suche nach einem optimalen (verallgemeinerten) Gitter I_{opt} läßt sich formulieren als Optimierungsproblem unter einer Nebenbedingung:
$$\max_{C(I)=\text{const.}} P(I)$$
bzw.
$$\min_{P(I)=\text{const.}} C(I)\ ,$$
wobei die globale Kostenfunktion $C(I)$ sowie die globale Nutzenfunktion $P(I)$ definiert sind als
$$C(I) \;:=\; \int_I c(\underline{l})\ d\underline{l}\ ,$$
$$P(I) \;:=\; \int_I c(\underline{l})\ d\underline{l}\ ,$$
d. h. als Summation bzw. Integration der jeweiligen lokalen Größen. Der Begriff lokal bezeichnet in diesem Zusammenhang (in Gedanken schnell wieder zurück zum natürlichen Multiindex) Teilraumeigenschaften, global dagegen meint Eigenschaften des gesamten Gitters.
- Ausgehend von einem beliebigen $I \subset I_{max}$ und einer hinreichend glatten Abbildung $\tau : \mathbb{R}_+^d \rightarrow \mathbb{R}_+^d$ mit $\tau(\underline{l}) = 0$ für $\underline{l}$ mit $l_j = 0$ definieren wir nun eine Störung $\varphi_{\varepsilon,\tau}$ des Gitters:
$$\varphi_{\varepsilon,\tau} : I \;\rightarrow\; I_{\varepsilon,\tau} \subset I_{\max}\ , \qquad \varepsilon \in \mathbb{R}\ ,$$
$$\varphi_{\varepsilon,\tau}(\underline{l}) \;:=\; \underline{l} \;+\; \varepsilon \cdot \tau(\underline{l})\ .$$



Für die globalen Kosten des gestörten Gitters $I_{\varepsilon,\tau}$ gilt:

$$C(I_{\varepsilon,\tau}) \;=\; \int\limits_{I_{\varepsilon,\tau}} c(\underline{k})\; d\underline{k} \;=\; \int\limits_{I} c(\underline{l}+\varepsilon\cdot\tau(\underline{l}))\cdot|\det D\varphi_{\varepsilon,\tau}|\; d\underline{l}\;.$$

Taylorentwicklung von $c(\underline{l}+\varepsilon\cdot\tau(\underline{l}))$ um $\varepsilon = 0$ liefert

$$c(\underline{l}+\varepsilon\cdot\tau(\underline{l})) \;=\; c(\underline{l})+\varepsilon\cdot\nabla c(\underline{l})\cdot\tau(\underline{l})+O(\varepsilon^2)\;.$$

Man beachte, daß $\nabla c(\underline{l})\cdot\tau(\underline{l})$ hier das Skalarprodukt zweier Vektoren bezeichnet. Des weiteren kann man für die Funktionaldeterminante zeigen

$$|\det D\varphi_{\varepsilon,\tau}| \;=\; 1+\varepsilon\cdot\mathrm{div}\tau+O(\varepsilon^2)\;.$$

Somit folgt mit dem Satz von Gauß,

$$\int\limits_{I} \mathrm{div}F\; d\underline{l} \;=\; \int\limits_{\partial I} F\; d\vec{S}, \qquad \text{hier} \quad F = c(\underline{l})\cdot\tau(\underline{l})\;,$$

sowie wegen $I \subset I_{\max}$ endlich

$$\begin{aligned} C(I_{\varepsilon,\tau}) \;&=\; \int\limits_{I} \Big(\big(\,c(\underline{l})+\varepsilon\cdot\nabla c(\underline{l})\cdot\tau(\underline{l})\big)\,(1+\varepsilon\cdot\mathrm{div}\tau)+O(\varepsilon^2)\Big)\;\; d\underline{l} \\ &=\; \int\limits_{I} c(\underline{l})\; d\underline{l} \;+\; \varepsilon\cdot\int\limits_{I} \nabla c(\underline{l})\cdot\tau(\underline{l})\; d\underline{l} \;+\; \varepsilon\cdot\int\limits_{I} c(\underline{l})\cdot\mathrm{div}\tau(\underline{l})\; d\underline{l} \;+\; O(\varepsilon^2) \\ &=\; C(I) \;+\; \varepsilon\cdot\int\limits_{\partial I} c(\underline{l})\cdot\tau(\underline{l})\; d\vec{S} \;+\; O(\varepsilon^2)\;. \end{aligned}$$

Betrachtet man nun die Ableitung nach ε, so gilt

$$\frac{C(I_{\varepsilon,\tau})-C(I)}{\varepsilon} \;\stackrel{\varepsilon\to 0}{\longrightarrow}\; \int\limits_{\partial I} c(\underline{l})\cdot\tau(\underline{l})\; d\vec{S}$$

und analog

$$\frac{P(I_{\varepsilon,\tau})-P(I)}{\varepsilon} \;\stackrel{\varepsilon\to 0}{\longrightarrow}\; \int\limits_{\partial I} p(\underline{l})\cdot\tau(\underline{l})\; d\vec{S}\;.$$

- Im optimalen I_{opt} folgt mit dem Lagrange-Prinzip für die Optimierung unter Nebenbedingungen

$$\lambda\cdot\int\limits_{\partial I_{\mathrm{opt}}} c(\underline{l})\cdot\tau(\underline{l})\; d\vec{S} \;=\; \int\limits_{\partial I_{\mathrm{opt}}} p(\underline{l})\cdot\tau(\underline{l})\; d\vec{S}$$

und somit wegen $\tau \equiv 0$ auf $\partial\mathbb{R}_+^d$

$$\lambda \cdot \int\limits_{\partial I_{\mathrm{opt}} \setminus \partial R_+^d} c(\underline{l}) \cdot \tau(\underline{l}) \; d\vec{S} \;=\; \int\limits_{\partial I_{\mathrm{opt}} \setminus \partial R_+^d} p(\underline{l}) \cdot \tau(\underline{l}) \; d\vec{S} \; .$$

Da dies für alle geeigneten glatten τ gilt, folgt schließlich

$$\lambda \cdot c(\underline{l}) \;=\; p(\underline{l}) \qquad \text{auf} \quad \partial I_{\mathrm{opt}} \; .$$

Die entscheidende Konsequenz hieraus ist, daß auf dem Rand von in unserem Sinne optimalen Gittern I

$$\frac{c(\underline{l})}{p(\underline{l})} \;=\; \mathrm{const}\,.$$

gelten muß. Somit ist der globale Optimierungsprozeß reduziert auf das Studium der lokalen Quotienten $\frac{c(\underline{l})}{p(\underline{l})}$, also der lokalen Kosten-Nutzen-Verhältnisse der einzelnen Teilräume $W_{\underline{l}}$, wenn wir wieder zu $\underline{l} \in \mathbb{N}^d$ zurückgehen.

Diskrete Optimierung:

- Jetzt bleiben wir im Diskreten, d. h. bei $\underline{l} \in \mathbb{N}^d$. Als lokale Kostenfunktion definieren wir

  $$c(\underline{l}) \;:=\; |W_{\underline{l}}| \in \mathbb{N}\,,$$

  als lokale Nutzenfunktion

  $$p(\underline{l}) \;:=\; \gamma \cdot b(\underline{l})$$

  mit $\|u_{\underline{l}}\|^2 \;\leq\; b(\underline{l})$ und γ so, daß $\gamma \cdot b(\underline{l}) \in \mathbb{N}$.

- Die Suche nach einem optimalen Gitter $I \subset \left\{\underline{l} : \underline{l} \in \mathbb{N}^d\right\}$ bzw. $I \subset \{\underline{l} : |\underline{l}|_\infty \leq N\} =: I_{\max}$ mit hinreichend großem N führt wie zuvor auf ein beschränktes Optimierungsproblem.
  Als globale Kostenfunktion definieren wir

  $$C(I) \;:=\; \sum_{\underline{l} \in I} c(\underline{l}) \;=\; \sum_{\underline{l} \in I_{\max}} x(\underline{l}) \cdot c(\underline{l})$$

  mit

  $$x(\underline{l}) \;:=\; \begin{cases} 0 & : \; \underline{l} \notin I \\ 1 & : \; \underline{l} \in I \end{cases} \; .$$

  Dementsprechend erhalten wir die globale Nutzenfunktion über den Beitrag zum Interpolationsfehler:

  $$\begin{aligned} \left\| u - \sum_{\underline{l} \in I} u_{\underline{l}} \right\|^2 \;&\leq\; \sum_{\underline{l} \in I_{\max} \setminus I} \|u_{\underline{l}}\|^2 \\ &\leq\; \sum_{\underline{l} \in I_{\max} \setminus I} \gamma \cdot b(\underline{l}) \\ &=\; \sum_{\underline{l} \in I_{\max}} (1 - x(\underline{l})) \cdot \gamma \cdot b(\underline{l}) \; . \end{aligned}$$

Hierbei haben wir die Beiträge aus den Teilräumen außerhalb $I_{\max}$ vernachlässigt. Da wir $I_{\max}$ aber beliebig groß wählen können, sollte dies kein Problem sein.

Somit erhalten wir als Optimierungsproblem

$$\min_{I \subset I_{\max}} \sum_{\underline{l} \in I_{\max}} (1 - x(\underline{l})) \cdot \gamma \cdot b(\underline{l}) \qquad \text{unter} \qquad \sum_{\underline{l} \in I_{\max}} x(\underline{l}) \;=\; \text{const}\,.$$

bzw.

$$\max_{I \subset I_{\max}} \sum_{\underline{l} \in I_{\max}} x(\underline{l}) \cdot \gamma \cdot b(\underline{l}) \qquad \text{unter} \qquad \sum_{\underline{l} \in I_{\max}} x(\underline{l}) \;=\; \text{const}\,.$$

bzw.

$$\max_x a^T x \qquad \text{unter} \qquad c^T x = \text{const.} =: W$$

mit $a_i \in \mathbb{N}$, $c_i \in \mathbb{N}$, $x_i \in \{0,1\}$ und vorgegebenen Kosten $W \in \mathbb{N}$. Ein solches Problem ist unter dem Namen Rucksack-Problem bzw. Knapsack Problem in der kombinatorischen Optimierung bekannt. (Was soll ein Wanderer in seinem Rucksack mit vorgegebener Kapazität mitnehmen?)

- Es ist bekannt, daß die Lösung dieses binären Rucksack-Problems sehr aufwendig ist (NP-vollständig). Leichter tut man sich dagegen mit dem rationalen Rucksack-Problem, bei dem – bei ansonsten gleicher Aufgabenstellung – $x_i \in [0,1] \cap \mathbb{Q}$ erlaubt ist. (Man darf also auch eine halbe Tafel Schokolade einpacken.) Hier existiert ein ganz einfacher Lösungsalgorithmus:

(1) sortiere so, daß $\frac{a_1}{c_1} \geq \frac{a_2}{c_2} \ldots \geq \frac{a_M}{c_M}$, $M := (2^N - 1)^d$

(2) $r := \max\left\{ j : \sum\limits_{i=1}^{j} c_i \leq W \right\}$

(3)
$$\left.\begin{array}{l} x_1 := \ldots := x_r := 1 \\[2mm] x_{r+1} := \left(W - \sum\limits_{i=1}^{r} c_i \right) / c_{r+1} \\[2mm] x_{r+2} := \ldots := x_M := 0 \end{array}\right\} \quad \text{ist optimal!}$$

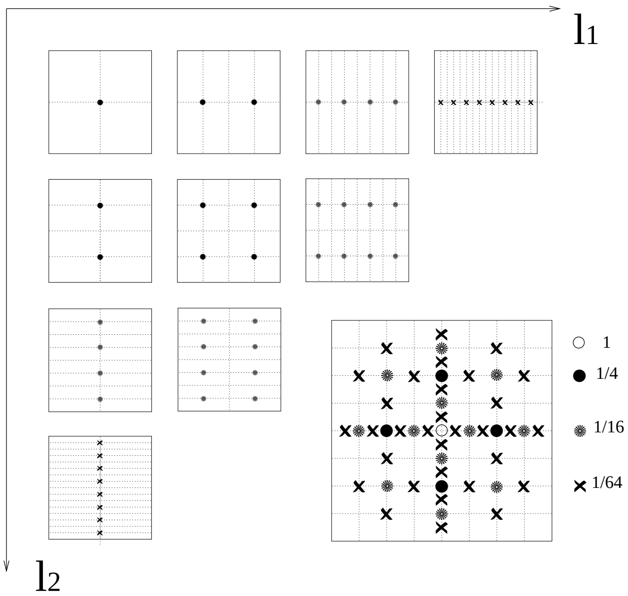
Diese rationale Lösung hat i. a. beliebig wenig mit der binären Lösung zu tun. Da aber nur ein x_i evtl. nicht ganzzahlig ist und wir die vogegebenen Kosten W ja beliebig festlegen können, kann $x_{r+1} \in \mathbb{N}$ und somit $x \in \{0,1\}^M$ gewährleistet werden!

Damit ist das globale Optimierungsproblem wie zuvor reduziert auf das Studium der lokalen Terme $\frac{a_i}{c_i}$, d. h. der lokalen Kosten-Nutzen-Verhältnisse aller Teilräume $W_{\underline{l}}$.

Optimale Gitter:

Sowohl die kontinuierliche als auch die diskrete Betrachtungsweise definieren optimale Gitter über die Größenordnung der Quotienten $p(\underline{l})/c(\underline{l})$. Nimmt man die L_2- oder die L_∞-Norm zur Messung von $p(\underline{l})$, dann gilt nach dem eingangs Gesagten, daß dieser Quotient (größenordnungsmäßig) konstant ist für konstantes $|\underline{l}|_1$. Damit ist gezeigt, daß die zuvor definierten Dünngitterräume $V_n^{(1)}$ bzgl. der L_2- und der L_∞-Norm optimal im Sinne des Kosten-Nutzen-Verhältnisses sind. Dies bestätigt unsere heuristischen Überlegungen.

Führt man den Optimierungsprozeß für die Energienorm durch, so erhält man wegen des Summenterms $\sum\limits_{j=1}^{d} 4^{l_j}$ in der oberen Schranke für $\|u_{\underline{l}}\|_E^2$ gemäß Lemma 5 andere optimale Gitter, da



der Quotient $p(\underline{l})/c(\underline{l})$ jetzt nicht mehr für konstantes $|\underline{l}|_1$ von konstanter Größenordnung ist. Nun gilt

$$\begin{aligned} p(\underline{l}) &= \frac{1}{4\cdot 12^{d-1}} \cdot 2^{-4|\underline{l}|_1} \cdot \left(\sum_{j=1}^{d} 4^{l_j}\right) \cdot |u|_\infty^2 \,, \\ c(\underline{l}) &= 2^{|\underline{l}-\underline{1}|_1}, \\ \gamma(\underline{l}) &:= \frac{p(\underline{l})}{c(\underline{l})} = \frac{3}{6^d} \cdot 2^{-5|\underline{l}|_1} \cdot \left(\sum_{j=1}^{d} 4^{l_j}\right) \cdot |u|_\infty^2 \,. \end{aligned}$$

Jetzt suchen wir die „Grenzlinie“ $\partial I_{\mathrm{opt}}$, längs der $\gamma(\underline{l})$ konstant ist. Den Wert der Konstanten gewinnen wir (beliebig) als $\gamma(\bar{\underline{l}})$ mit $\bar{\underline{l}} = (1, \ldots, 1, n)$:

$$\gamma(\bar{\underline{l}}) = \frac{3}{6^d} \cdot 2^{-5(n+d-1)} \cdot (4^n + 4(d-1)) \cdot |u|_\infty^2 \,.$$

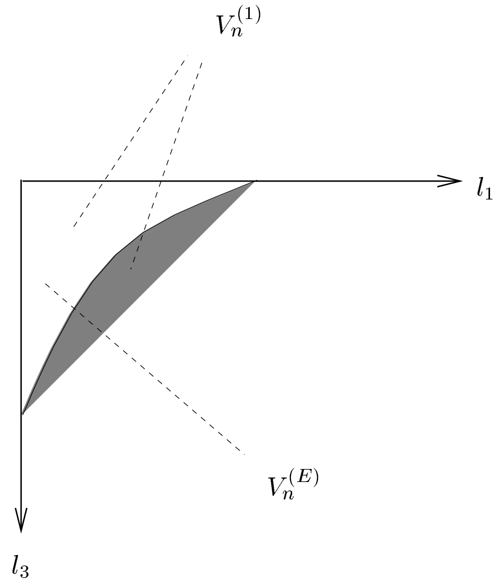
D. h., wir fixieren die Endpunkte von $\partial I_{\mathrm{opt}}$ und suchen dessen Verlauf dazwischen. Zum optimalen Gitter sollen dann alle Teilräume $W_{\underline{l}}$ mit besserem Kosten-Nutzen-Verhältnis als $W_{\bar{\underline{l}}}$ zählen, also

$$\begin{aligned} \underline{l} \in I_{\mathrm{opt}} \quad &\Leftrightarrow \quad \gamma(\underline{l}) \;\geq\; \gamma(\bar{\underline{l}}) \\ &\Leftrightarrow \quad 2^{-5|\underline{l}|_1} \cdot \left(\sum_{j=1}^{d} 4^{l_j}\right) \;\geq\; 2^{-5(n+d-1)} \cdot (4^n + 4(d-1)) \\ &\Leftrightarrow \quad \underbrace{|\underline{l}|_1 - \frac{1}{5}\log_2\left(\sum_{j=1}^{d} 4^{l_j}\right) \;\leq\; n + d - 1 - \frac{1}{5}\log_2(4^n + 4d - 4)}_{(\star)} \,. \end{aligned}$$

Dies definiert das optimale Energiegitter, das Energie-optimale dünne Gitter:

$$V_n^{(E)} := \bigoplus_{\underline{l}:(\star)} W_{\underline{l}}.$$

Man kann sich $V_n^{(E)}$ als ein „angeknabbertes“ dünnes Gitter $V_n^{(1)}$ vorstellen:



Die Kunst beim „Anknabbern“ bzw. bei der Optimierung ist dabei, so viele Teilräume $W_{\underline{l}}$ zu entfernen, daß man auf spürbar weniger Gitterpunkte kommt und trotzdem die Approximationsordnung nicht (signifikant) verschlechtert. Zur näheren Untersuchung von $V_n^{(1)}$ und $V_n^{(E)}$ studieren wir im folgenden deren Approximationseigenschaften.

5.2 Approximationstheorie

Wir interessieren uns für Kosten und Nutzen der Approximationsräume $V_n^{(1)}$ und $V_n^{(E)}$. Zunächst befassen wir uns mit der Anzahl der Gitterpunkte im L_∞- bzw. L_2-dünnen Gitter, also im Gitter zu $V_n^{(1)}$.

Lemma 7:

Für die Anzahl der Punkte bzw. Basisfunktionen in $V_n^{(1)}$ gilt

$$\begin{aligned}
|V_n^{(1)}| &= \sum_{i=0}^{n-1} 2^i \cdot \binom{d-1+i}{d-1} \\
&= (-1)^d \; + \; 2^n \cdot \sum_{i=0}^{d-1} \binom{n+d-1}{i} \cdot (-2)^{d-1-i} \\
&= 2^n \cdot \left(\frac{n^{d-1}}{(d-1)!} \; + \; O(n^{d-2}) \right) \\
&= O(2^n \cdot n^{d-1}) \; .
\end{aligned}$$

Beweis:

Die erste Darstellung folgt unmittelbar aus der Definition von $V_n^{(1)}$ sowie Lemma 1:

$$\begin{aligned}
|V_n^{(1)}| \; = \; \left| \bigoplus_{|\underline{l}|_1 \le n+d-1} W_{\underline{l}} \right| \; = \sum_{|\underline{l}|_1 \le n+d-1} 2^{|\underline{l}-\underline{1}|_1} &= \sum_{i=d}^{n+d-1} \sum_{|\underline{l}|_1 = i} 2^{|\underline{l}-\underline{1}|_1} \\
&= \sum_{i=d}^{n+d-1} 2^{i-d} \sum_{|\underline{l}|_1 = i} 1 \\
&= \sum_{i=d}^{n+d-1} 2^{i-d} \cdot \binom{i-1}{d-1} \\
&= \sum_{i=0}^{n-1} 2^i \cdot \binom{d-1+i}{d-1},
\end{aligned}$$

da es $\binom{i-1}{d-1}$ Möglichkeiten gibt, aus d natürlichen Zahlen die Summe i zu bilden. Nun wenden wir wieder einen Standard-Trick an:

$$\begin{aligned}
\sum_{i=0}^{n-1} 2^i \cdot \binom{d-1+i}{d-1} &= \frac{1}{(d-1)!} \cdot \sum_{i=0}^{n-1} (i+d-1) \dots (i+1) \cdot x^i \Big|_{x=2} \\
&= \frac{1}{(d-1)!} \cdot \sum_{i=0}^{n-1} (x^{i+d-1})^{(d-1)} \Big|_{x=2} \\
&= \frac{1}{(d-1)!} \cdot \left(x^{d-1} \cdot \frac{1-x^n}{1-x} \right)^{(d-1)} \Bigg|_{x=2} \\
&= \frac{1}{(d-1)!} \cdot \sum_{i=0}^{d-1} \binom{d-1}{i} \cdot (x^{d-1} - x^{n+d-1})^{(i)} \cdot \left(\frac{1}{1-x} \right)^{(d-1-i)} \Bigg|_{x=2} \\
&= \dots \\
&= (-1)^d \; + \; 2^n \cdot \sum_{i=0}^{d-1} \binom{n+d-1}{i} \cdot (-2)^{d-1-i} \; .
\end{aligned}$$

Die Ordnungsaussagen sind offensichtlich. □

Wegen $h = 2^{-n}$ gilt

$$2^n \cdot n^{d-1} \;=\; \frac{1}{h} \cdot |\log h|^{d-1},$$

d. h., im Vergleich zu $V_n^{(\infty)}$ mit $|V_n^{(\infty)}| = O(2^{nd}) = O(h^{-d})$ erhält man ein deutlich gebremstes Wachstum der Gitterpunktzahl.

Schauen wir uns den Unterschied einmal anhand einiger konkreter Zahlenbeispiele an:

	$d=2$		$d=3$		$d=4$	
	$n=10$	$n=20$	$n=10$	$n=20$	$n=10$	$n=20$
$\lvert V_n^{(\infty)}\rvert$	$1.05 \cdot 10^6$	$1.1 \cdot 10^{12}$	$1.07 \cdot 10^9$	$1.15 \cdot 10^{18}$	$1.10 \cdot 10^{12}$	$1.21 \cdot 10^{24}$
$\lvert V_n^{(1)}\rvert$	9217	$1.99 \cdot 10^7$	47103	$2.0 \cdot 10^8$	$1.78 \cdot 10^5$	$1.41 \cdot 10^9$

Damit ist klar: $V_n^{(1)}$ ist signifikant billiger als $V_n^{(\infty)}$. Die Gretchenfrage lautet nun, mit welchem Verlust an Genauigkeit wir uns das erkaufen.

Satz 1:

Für den Interpolationsfehler $u - u_n^{(1)}$ des Interpolanten $u_n^{(1)} \in V_n^{(1)}$ zu $u \in X_0(\Omega)$ bzgl. L_2-, L_∞- und Energienorm gilt:

$$\begin{aligned}
\|u - u_n^{(1)}\|_\infty &\le \frac{2 \cdot |u|_\infty}{8^d} \cdot 2^{-2d} \cdot A(d,n)\,, \\
\|u - u_n^{(1)}\|_2 &\le \frac{2 \cdot |u|_2}{12^d} \cdot 2^{-2d} \cdot A(d,n)\,, \\
\|u - u_n^{(1)}\|_E &\le \frac{d \cdot |u|_\infty}{2 \cdot 3^{(d-1)/2} \cdot 4^{d-1}} \cdot 2^{-n}
\end{aligned}$$

mit $A(d,n) = \sum_{k=0}^{d-1} \binom{n+d-1}{k} = O(n^{d-1})$.
D. h.,

$$\begin{aligned}
\|u - u_n^{(1)}\|_\infty &= O(2^{-2n} \cdot n^{d-1}) \;=\; O(h^2 \cdot |\log h|^{d-1})\,, \\
\|u - u_n^{(1)}\|_2 &= O(2^{-2n} \cdot n^{d-1}) \;=\; O(h^2 \cdot |\log h|^{d-1})\,, \\
\|u - u_n^{(1)}\|_E &= O(2^{-n}) \;=\; O(h)\;.
\end{aligned}$$

Die Aussagen von Satz 1 sind zu vergleichen mit den Ergebnissen aus Lemma 6 für $V_n^{(\infty)}$. Demnach verschlechtert sich die Ordnung des L_2- und L_∞-Fehlers um einen Faktor $|\log h|^{d-1}$, die Ordnung des Energiefehlers bleibt (trotz deutlich kleinerer Punktezahl) gleich.

Beweis:

Der Beweis geht mittels

$$\|u - u_n^{(1)}\|_\infty \;\le \sum_{|\underline{l}|_1 > n+d-1} \|u_{\underline{l}}\|_\infty \;\;\ldots$$

und ist etwas länglich, weshalb wir ihn hier auslassen. Trotzdem: $\square$

Bemerkung:

Der Approximationsraum $V_n^{(1)}$ ist – bzgl. L_2- und L_∞-Norm – optimal hinsichtlich der eingesetzten Ressourcen: mehr Genauigkeit und weniger Fehler geht nicht! Schauen wir uns auch das anhand einer Tabelle an. Aber Vorsicht: Alle Zahlen geben nur die Größenordnung an!

		$d=2$		$d=3$		$d=4$	
		$n=10$	$n=20$	$n=10$	$n=20$	$n=10$	$n=20$
$V_n^{(\infty)}$:	2^{-2n}	$\approx 10^{-6}$	$\approx 10^{-12}$	10^{-6}	10^{-12}	10^{-6}	10^{-12}
$V_n^{(1)}$:	$2^{-2n} \cdot n^{d-1}$	$\approx 10^{-5}$	$\approx 2 \cdot 10^{-11}$	$\approx 10^{-4}$	$3.6 \cdot 10^{-10}$	10^{-3}	$\approx 7 \cdot 10^{-9}$
	Verlust	10	20	100	400	1000	8000
	Gewinn bei # Punkten	114	55000	22716	$5.75 \cdot 10^9$	$6.18 \cdot 10^6$	$8.58 \cdot 10^{14}$

Jetzt wenden wir uns $V_n^{(E)}$ zu. Im vorigen Abschnitt war es über den Optimierungsprozeß definiert worden:

$$V_n^{(E)} = \bigoplus_{\underline{l}:(\star)} W_{\underline{l}}$$

mit $(\star)$:

$$|\underline{l}|_1 - \frac{1}{5}\log_2\left(\sum_{j=1}^{d} 4^{l_j}\right) \le n + d - 1 - \frac{1}{5}\log_2(4^n + 4d - 4)\ .$$

Lemma 8:

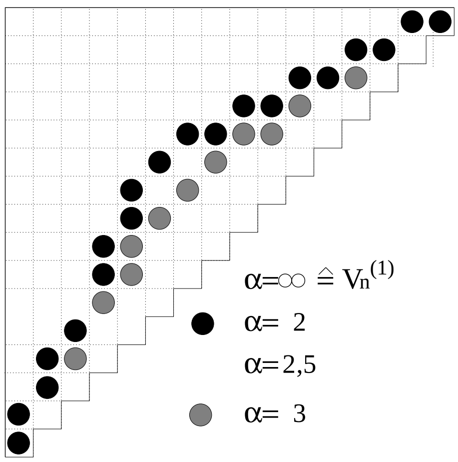
Für die Anzahl der Punkte bzw. Basisfunktionen in $V_n^{(E)}$ gilt

$$|V_n^{(E)}| = O(2^n), \qquad |V_n^{(E)}| \le 2^n \cdot \frac{d}{2} \cdot (2+\sqrt{2})^d\ .$$

Beweis:

Die Vorgehensweise ist schon bekannt: Addiere die Dimension aller an $V_n^{(E)}$ beteiligten Teilräume $W_{\underline{l}}$. Dazu schließen wir $V_n^{(E)}$ geeignet ein. Wir betrachten die Approximationsräume $V_{n,\alpha}$:

$$V_{n,\alpha} := \bigoplus_{i=0}^{n-1} \bigoplus_{\substack{|\underline{l}|_1 = n+d-1-i \\ |\underline{l}|_\infty \ge n-\alpha i}} W_{\underline{l}}\ .$$



Man kann zeigen, daß für hinreichend großes n gilt:

$$V_{n,2.5} \subseteq V_n^{(E)} \subseteq V_{n,3} \ .$$

Nun werden die Behauptungen des Lemmas für $V_{n,3}$ gezeigt, woraus dann alles folgt. □

Bemerkung:

Mit Lemma 8 haben wir eines unserer großen Ziele erreicht: Die Dimension des Approxmationsraumes $V_n^{(E)}$ hängt bezüglich der n-Asymptotik (d. h. festes d und $n \longrightarrow \infty$) nicht von der Dimensionalität d ab:

$$|V_n^{(E)}| = O(2^n) \ .$$

Möglich (und Realität!) sind natürlich von d abhängige Faktoren, etwa $3^d \cdot 2^n$, aber die wachsen im Gegensatz zu n^{d-1} aus Lemma 7 bei $V_n^{(1)}$ nicht in n, sind also n-asymptotisch irrelevant.

Und der Fehler? Beim Übergang von $V_n^{(\infty)}$ zu $V_n^{(1)}$ hatten wir uns in der L_2- und L_∞-Norm einen log-Term eingehandelt, in der Energienorm ordnungstechnisch aber nichts verloren. Da $V_n^{(E)}$ ja nur bzgl. der Energienorm optimal ist, beschränken wir uns auf den Fehler in der Energienorm.

Satz 2:

Für den Interpolationsfehler $u - u_n^{(E)}$ des Interpolanten $u_n^{(E)} \in V_n^{(E)}$ zu $u \in X_0(\Omega)$ bzgl. der Energienorm gilt:

$$\|u - u_n^{(E)}\|_E \leq \frac{d \cdot |u|_\infty}{3^{(d-1)/2} \cdot 4^{d-1}} \cdot \left(\frac{1}{2} + \left(\frac{5}{2}\right)^{d-1} \right) \cdot 2^{-n} = O(h) \ .$$

Bemerkung:

(1) In der L_2- und L_∞-Norm können wir das Resultat von $V_n^{(1)}$, also $2^{-2n} \cdot n^{d-1} = h^2 \cdot |\log h|$, nicht halten: Schließlich ist $V_n^{(1)}$ ja bezüglich $\|.\|_2$ und $\|.\|_\infty$ optimal!

(2) In der Energienorm bleiben wir ordnungstechnisch bei $O(2^{-n}) = O(h)$. Unsere Optimierung hat also gewaltig etwas gebracht:

- In der Punktzahl geht es 'runter von $O(2^n \cdot n^{d-1})$ auf $O(2^n)$.
- Im Fehler bleibt's bei $O(2^{-n})$.

(3) Derartige Ordnungsresultate sind immer mit Vorsicht zu genießen, weil es asymptotische Aussagen sind: $100^{d-1} \cdot 2^n$ bleibt bis $n = 100$ (d. h. bis $h = 10^{-30}$) größer als $n^{d-1} \cdot 2^n$, obwohl es ordnungstechnisch besser ist. Wir sind hier aber auf der sicheren Seite, weil erstens $V_n^{(E)} \subseteq V_n^{(1)}$ und damit $|V_n^{(E)}| \leq |V_n^{(1)}| \ \forall n$ gilt und weil wir zweitens in $V_n^{(E)}$ ja im Optimum sind.

(4) Von Riesenschranken sollte man sich nicht abschrecken lassen. Oft stehen sie nur da, weil man zu blöde ist, bessere zu finden!

Somit kommen wir zum Beweis von Satz 2.

Beweis:

Es gilt

$$\|u - u_n^{(E)}\|_E \;\leq\; \|u - u_n^{(1)}\|_E \;+\; \|u_n^{(1)} - u_n^{(E)}\|_E \,,$$

wobei der erste Term nach Satz 1 von der Ordnung $O(h)$ ist. Bleibt der zweite Summand, also der Unterschied zwischen $V_n^{(1)}$ und $V_n^{(E)}$.
Wegen $V_{n,2.5} \subseteq V_n^{(E)}$ gilt

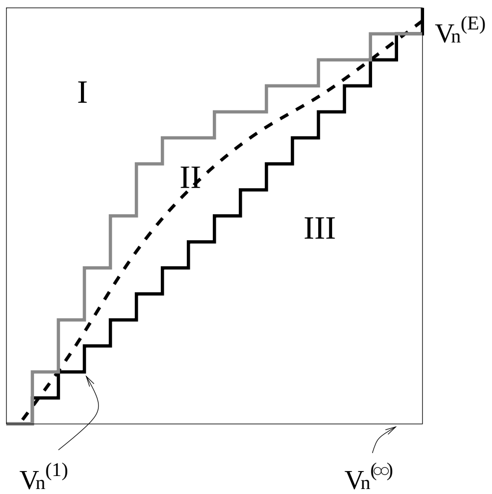
$$\begin{aligned}
\|u_n^{(1)} - u_n^{(E)}\|_E \;&\leq\; \sum_{W_{\underline{l}} \subseteq V_n^{(1)} \setminus V_n^{(E)}} \|u_{\underline{l}}\|_E \\
&\leq\; \sum_{W_{\underline{l}} \subseteq V_n^{(1)} \setminus V_{n,2.5}} \|u_{\underline{l}}\|_E \\
&\leq\; \ldots \\
&\leq\; \frac{d \cdot |u|_\infty}{3^{(d-1)/2}} \cdot \left(\frac{5}{8}\right)^{d-1} \cdot 2^{-n} \,.
\end{aligned}$$

Damit und mit Satz 1 folgt die Behauptung □

Bemerkung:

(1) $V_{n,2}$ ist die „Schallmauer“: Ab hier, d. h. für $\alpha \leq 2$, wird der Interpolationsfehler schlechter als $O(h)$.

(2) Die andere Grenze ist $V_{n,\infty} = V_n^{(1)}$, der Standard-Dünngitterraum: Ab hier tritt der log-Faktor n^{d-1} in der Raumdimension auf.



I) $\|u - u_n^{(E)}\|_E$ schlechter als $O(h)$
$\#$ Punkte $= O(\frac{1}{h})$ und besser

II) $\|u - u_n^{(E)}\|_E = O(h)$
$\#$ Punkte $= O(\frac{1}{h})$

III) $\|u - u_n^{(E)}\|_E = O(h)$
$\#$ Punkte schlechter als $O(\frac{1}{h})$

Komplexitätsabschätzungen wie $|V_n^{(1)}| \leq \ldots$ bzw. $|V_n^{(1)}| = O(...)$ sind eine klassische Problemstellung in der Informatik. Diese Aufgabe kann man auch über Rekursionsformeln angehen.

5.3 Rekursionsformeln und Komplexität

Rekursionsformeln:

Bislang haben wir dünne Gitter über explizite Formeln definiert. Jetzt wollen wir auf ihren rekursiven Charakter eingehen, um so weitere Aussagen über ihre Komplexität bzw. ihr asymptotisches Verhalten zu gewinnen. Zu diesem Zweck leiten wir eine Rekursionsformel für $|V_n^{(1)}|$ her. Nach Lemma 7 gilt

$$|V_n^{(1)}| \;=\; \sum_{i=0}^{n-1} 2^i \cdot \binom{d-1+i}{d-1} \;.$$

Hier treten zwei Parameter auf: n (Auflösung) und d (Dimensionalität). Mit der Definition

$$c_{n,d} \;:=\; |V_n^{(1)}|$$

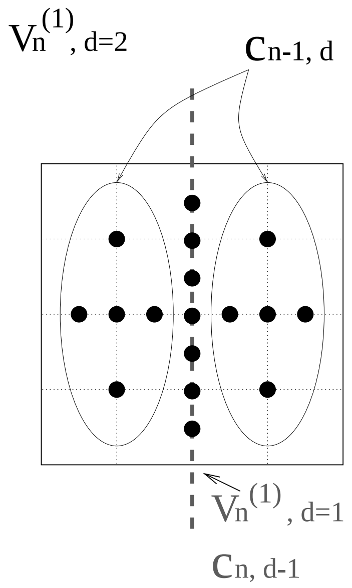
gilt

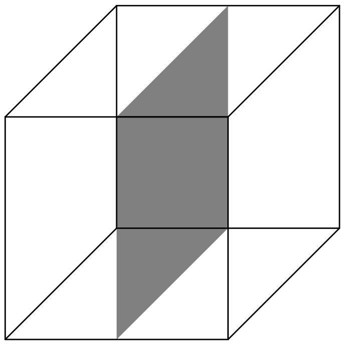
$$\begin{aligned} c_{n,d} &= \sum_{i=0}^{n-1} 2^i \cdot \binom{d-1+i}{d-1} \\ &= 1 + \sum_{i=1}^{n-1} 2^i \cdot \left(\binom{d-2+i}{d-1} + \binom{d-2+i}{d-2} \right) \\ &= \sum_{i=0}^{n-1} 2^i \cdot \binom{d-2+i}{d-2} \;+\; \sum_{i=1}^{n-1} 2^i \cdot \binom{d-2+i}{d-1} \\ &= c_{n,d-1} \;+\; 2 \cdot \sum_{i=0}^{n-2} 2^i \cdot \binom{d-1+i}{d-1} \,, \end{aligned}$$

d. h.

$$c_{n,d} \;=\; c_{n,d-1} + 2 \cdot c_{n-1,d} \;.$$

Das d-dimensionale dünne Gitter der Tiefe n besteht folglich aus einem $d-1$-dimensionalen der Tiefe n (Separator) und zwei d-dimensionalen der Tiefe $n-1$.





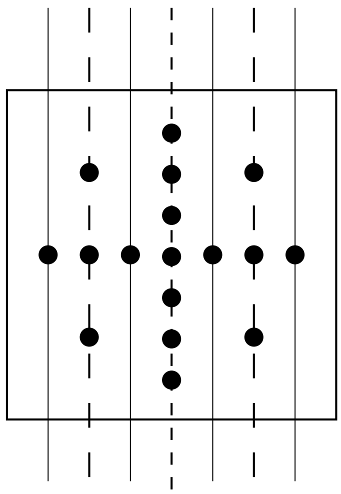
Diese Zerlegung können wir rekursiv fortführen, indem auch die beiden Hälften ($c_{n-1,d}$) zerlegt werden und so fort:

$$\begin{aligned} c_{n,d} &= c_{n,d-1} + 2 \cdot c_{n-1,d} \\ &= c_{n,d-1} + 2 \cdot (c_{n-1,d-1} + 2 \cdot c_{n-2,d}) \\ &= c_{n,d-1} + 2 \cdot c_{n-1,d-1} + 4 \cdot c_{n-2,d} \\ &= \dots \end{aligned}$$

und damit (beachte $c_{1,d} = c_{1,d-1} = 1$)

$$c_{n,d} = \sum_{i=0}^{n-1} 2^i \cdot c_{n-i,d-1} .$$

D. h., jedes dünne Gitter $V_n^{(1)}$ der Tiefe n und der Dimensionalität d zerfällt vollständig in dünne Gitter $V_k^{(1)}$, $k = 0, \dots, n$, der Dimensionalität $d-1$.



Die Formulierung via Rekursionsformeln gestattet eine weitergehende Untersuchung der Komplexität dünner Gitter. Neben der (bisher studierten) n-Asymptotik wollen wir jetzt auch die d-Asymptotik (d. h., was passiert bei wachsendem d und festem n?) betrachten.

Für Rekursions- bzw. Rekurrenzformeln gibt es zahlreiche Techniken. Die meisten gestatten aber nur die Untersuchung einparametriger Formeln, also Rekurrenzen der Art $a_n = a_{n-1} + 4\, a_{n-2}$.

Wir wollen hier nur kurz ein paar interessante Eigenschaften der $c_{n,d}$ ohne Beweis auflisten. Als Startwerte hat man

$$\begin{aligned} c_{1,d} &= 1 \qquad \forall d \in \mathbb{N} , \\ c_{n,1} &= 2^n - 1 \qquad \forall n \in \mathbb{N} . \end{aligned}$$

(i) $c_{n,d} \in \mathbb{N} \qquad \forall n, d \in \mathbb{N}$

(ii) $c_{n,d}$ streng monoton wachsend in d für $n > 1$
($c_{n,d} - c_{n,d-1} = 2 \cdot c_{n-1,d} \geq 2 \quad \surd$)

(iii) $c_{n,d}$ streng monoton wachsend in $n \quad \forall d \in \mathbb{N}$
($c_{n,d} - c_{n-1,d} = c_{n-1,d} + c_{n,d-1} \geq 2 \quad \surd$)

(iv) $c_{n,d} \longrightarrow \infty$ für $n \longrightarrow \infty$ sowie für $d \longrightarrow \infty$, falls $n > 1$

(v) $\frac{c_{n,d+1}}{c_{n,d}} \longrightarrow 1 \qquad$ für $\quad d \longrightarrow \infty \qquad (\star)$

$\frac{c_{n,d+1}}{c_{n,d}} \longrightarrow \infty \qquad$ für $\quad n \longrightarrow \infty$

($\star$) bedeutet: Bei fester Auflösung n wirkt sich ein größeres d im Grenzfall $d \longrightarrow \infty$ nicht aus. Das ist nicht nur eine Ordnungsaussage, sondern eine harte!

(vi) $\frac{c_{n+1,d}}{c_{n,d}} \longrightarrow 2 \qquad$ für $\quad n \longrightarrow \infty \qquad (\star)$

$\frac{c_{n+1,d}}{c_{n,d}} \longrightarrow \infty \qquad$ für $\quad d \longrightarrow \infty$

($\star$) bedeutet: Bei fester Dimensionalität d verdoppelt sich im Grenzfall $n \longrightarrow \infty$ bei halber Maschenweite die Punktezahl (d. h., der log-Term wirkt sich nicht aus!)

ε-Komplexität:

Ein Standardbegriff in der Komplexitätstheorie numerischer Algorithmen ist die ε-Komplexität. Sie gibt den Aufwand an, der zur Erzielung einer Näherungslösung der vorgegebenen Genauigkeit ε erforderlich ist. Bei der Interpolation ist dabei „Aufwand" immer gleichbedeutend mit der Anzahl der Gitterpunkte. Im folgenden steht „Genaugkeit" für den Fehler in der Energienorm, und das Gleichheitszeichen „=" ist stets größenordnungsmäßig zu lesen:

(i) V_n^{∞} :

$$\begin{aligned} \varepsilon &= h \\ N &= h^{-d} \\ \Rightarrow \underline{\underline{N_{(\varepsilon)}}} &\underline{\underline{= \varepsilon^{-d}}} \end{aligned}$$

Um eine Genauigkeit von sechs Stellen (also 10^{-6}) zu erreichen, sind im 1D-Fall größenordnungsmäßig also 10^6 Speicherplätze erforderlich, in 2D bereits 10^{12} und in 3D gar 10^{18}.

(ii) $V_n^{(1)}$:

$$\begin{aligned} \varepsilon &= h \\ N &= h^{-1}|\log h|^{d-1} \\ \Rightarrow \underline{\underline{N_{(\varepsilon)}}} &\underline{\underline{= \varepsilon^{-1} \cdot |\log \varepsilon|^{d-1}}} \end{aligned}$$

Um eine Genauigkeit von sechs Stellen zu erreichen, sind jetzt im 1D-Fall größenordnungsmäßig 10^6 Speicherplätze erforderlich, in 2D $2 \cdot 10^7$ und in 3D $4 \cdot 10^8$.

(iii) $V_n^{(E)}$:

$$\begin{aligned} \varepsilon &= h \\ N &= h^{-1} \\ \Rightarrow \underline{\underline{N_{(\varepsilon)}}} &\underline{\underline{= \varepsilon^{-1}}} \end{aligned}$$

Um eine Genauigkeit von sechs Stellen zu erreichen, sind jetzt in 1D, 2D und 3D größenordnungsmäßig 10^6 Speicherplätze erforderlich.

In (i) wird der „Fluch der Dimensionalität" offenkundig: Der Aufwand wächst exponentiell in d. Bei (ii) ist es schon besser, (iii) ist optimal: Doppelte Genauigkeit bedeutet doppelten Aufwand, d interessiert nicht. So etwas kennt man beispielsweise aus der numerischen Quadratur von den Monte-Carlo-Verfahren.

5.4 Numerische Quadratur

5.4.1 Allgemeines

Es gibt zwei Möglichkeiten bzw. Zeitpunkte, Gitter zu optimieren:

- ⋆ a priori: Eine Gitteroptimierung vor einer konkreten Berechnung kann nur problemklassenabhängig erfolgen, nicht jedoch problemabhängig. Ein derartiges Vorgehen ist strukturell, es optimiert bzw. definiert die Gitterstruktur vor der eigentlichen Rechnung. Dies entspricht unserer Vorgehensweise in den Abschnitten 5.1 bis 5.3.
- ⋆ a posteriori: Verbessert man die Gittergestalt dagegen während einer Berechnung (insofern ist der Begriff „a posteriori“ etwas irreführend), weil etwa die Notwendigkeit weiterer Gitterpunkte in einem bestimmten Teilgebiet erkannt wird, dann handelt es sich um ein problemabhängiges Vorgehen. Dies haben wir bei der adaptiven Gitterverfeinerung.

Optimal ist ein Zusammenspiel von beidem. Es empfiehlt sich, möglichst viel Struktur bereits vor Beginn einer Rechnung festzulegen und nur den problemabhängigen Rest dann zur Laufzeit zu erledigen. Bei der Diskretisierung partieller Differentialgleichungen wurden strukturelle Ansätze bisher kaum beachtet. Anders sieht es in den Bereichen Interpolation/Approximation/Quadratur aus, wo die a priori Gitteroptimierung eine lange Tradition hat. Deshalb wollen wir nochmals einen Abstecher in die numerische Quadratur unternehmen.

<u>Prinzipien numerischer Quadratur:</u>

Zu berechnen sei das Integral einer (in zunächst irgendeinem Sinne) integrierbaren Funktion f über der (meßbaren) Punktmenge B:

$$If \;:=\; \int_B f(x)\;dx\;.$$

Wir wollen das Funktional If, also das Integral numerisch berechnen. Gesucht ist somit eine Näherung Qf mit

$$|Qf - If| \;\leq\; \varepsilon$$

mit einer vorgegebenen Genauigkeit ε. Zur Verfügung steht dazu eine endliche Anzahl diskreter Samples (Auswertungen) $f(x_i)$, $i = 1, \ldots, n$. Eine Näherung wird dann üblicherweise definiert als

$$Q_n f \;:=\; \sum_{i=1}^{n} c_i \cdot f(x_i)$$

mit Gewichten c_i. Wegen

$$\begin{aligned}
\left|Q_n\tilde{f} - Q_n f\right| \;=\; \left|Q_n(\tilde{f} - f)\right| \;&\leq\; \sum_{i=1}^{n} |c_i| \cdot \left|\tilde{f}(x_i) - f(x_i)\right| \\
&\leq\; \underbrace{\left(\sum_{i=1}^{n} |c_i|\right)}_{(\star)} \cdot \|\tilde{f} - f\|_\infty
\end{aligned}$$

fordert man i. a. $c_i \geq 0$ und $\sum\limits_{i=1}^{n} c_i = \mathrm{vol}(B)$. Dann ist das numerische Quadraturproblem bzgl. Schwankungen im Integranden f gut konditioniert. Als Konditionszahl $(\star)$ erhält man nämlich

$$\sum_{i=1}^{n} |c_i| \;=\; \sum_{i=1}^{n} c_i \;=\; \mathrm{vol}(B)\;.$$

Außerdem ist damit sichergestellt, daß konstante Funktionen c korrekt integriert werden, was eine naheliegende Forderung ist:

$$Qc \;=\; \sum_{i=1}^{n} c_i \cdot c \;=\; c \cdot \mathrm{vol}(B) \;=\; \int_B c\ dx \;=\; Ic\ .$$

Bei Gewichten mit wechselnden Vorzeichen ist dagegen das Szenario $\sum_{i=1}^{n} c_i = \mathrm{vol}(B)$, $\sum_{i=1}^{n} |c_i| \longrightarrow \infty$ für wachsendes n möglich. Dann wird die Kondition katastrophal schlecht, obwohl man doch mit $n \longrightarrow \infty$ immer genauer approximieren möchte!

5.4.2 Univariate Integrationsformeln

Univariate Integrationsformeln kennen auch im Mehrdimensionalen nur eine Maschenweite h. Verfeinert man das Gitter in einer Richtung, dann auch in allen anderen. Multivariate Integrationsformeln gestatten dagegen die separate Behandlung der einzelnen Koordinatenrichtungen. Wir betrachten hier den eindimensionalen Fall.

Konstruktionsprinzipien univariater Quadraturformeln:

i) Riemannsche Summen:
Hier ist die Konvergenz $Q_n f \to If$ für $n \to \infty$ per Definition gesichert (genau so ist Integrierbarkeit im Riemannschen Sinn definiert):

$$B = [a, b], \qquad x_i = a + i \cdot h\,, \qquad i = 0\,, \ldots, n\,, \qquad h = \frac{b-a}{n}\,,$$
$$R_n f \;:=\; \sum_{i=1}^{n} h \cdot f(\xi_i)\,, \qquad \xi_i \in [x_{i-1}, x_i]\ .$$

Die Quadraturformel $R_n f$ ist sehr robust, schließlich konvergiert sie für beliebiges Riemann-integrierbares f. Allerdings läßt die Konvergenzordnung zu wünschen übrig. Für eine Genauigkeit ε sind $O(\frac{1}{\varepsilon})$ Gitterpunkte erforderlich.

ii) Approximation:
Hier wird der Integrand f durch ein leicht zu integrierendes $\tilde{f}$ ersetzt, das anschließend exakt integriert wird. Einige Beispiele:

- Bernsteinpolynome:

$$B_n(f)(x) \;:=\; \sum_{i=0}^{n} \binom{n}{i} \cdot f\left(\frac{i}{n}\right) \cdot x^i \cdot (1-x)^{n-i} \quad \text{auf } [0,1]$$
$$\to B_n f \;:=\; \int_0^1 B_n(f)(x)\ dx \;=\; \frac{1}{n+1} \cdot \sum_{i=0}^{n} f\left(\frac{i}{n}\right)$$

- Taylorpolynome:

$$T_n(f)(x) \;:=\; \sum_{i=0}^{n} f_{(0)}^{(i)} \cdot \frac{x^i}{i!} \quad \text{auf } [-1,1]$$
$$\to T_n f \;:=\; \int_{-1}^{1} T_n(f)(x)\ dx \;=\; \ldots$$

- interpolierende Polynome:

  Wähle $P_{n-1}(f) \in \mathbb{P}_{n-1}$ so, daß $P_{n-1}(f)(x_i) = f(x_i)$ $(i = 1, \ldots, n)$ gilt.

$$\begin{aligned} \to P_{n-1}f \ := \int_a^b P_{n-1}(f)(x)\,dx \ &= \int_a^b \sum_{i=1}^n L_{n-1,i}(x) \cdot f(x_i)\,dx \\ &= \sum_{i=1}^n \underbrace{\int_a^b L_{n-1,i}(x)\,dx}_{c_i} \cdot f(x_i) \end{aligned}$$

  Die Polynome $L_{n-1,i}$ sind dabei die Lagrange-Polynome vom Grad $n-1$ zu den Stützstellen $x_1, \ldots, x_n$: $L_{n-1,i}(x_j) = \delta_{ij}$ (Kronecker-Symbol). Polynome p vom Grad kleiner oder gleich $n-1$ werden durch diesen Interpolationsprozeß natürlich reproduziert. Für sie gilt also $P_{n-1}(p) = p$ und folglich $P_{n-1}p = Ip$ (exakte Integration).

  Interpolierende Polynome stellen den Standardfall univariater Quadraturformeln dar. Aus einer kurzen Betrachtung der Qualität der Approximation,

$$|Q_n f - If| \ = \ |IP_{n-1}(f) - If| \ \leq \ (b-a) \cdot \|P_{n-1}(f) - f\|_\infty \ ,$$

  lassen sich zwei mögliche Strategien zur Erhöhung der Genauigkeit ableiten:

  1. $n \uparrow$ (Erhöhung des Polynomgrads):
     Hier spricht man von einfachen Formeln, weil stets nur ein Interpolationspolynom betrachtet (und integriert) wird.
  2. $b-a \downarrow$ (Unterteilung des Integrationsgebiets):
     Hier spricht man von zusammengesetzten Formeln, weil der Polynomgrad konstant gehalten wird, dafür aber das Ausgangsintervall in immer kleinere Subintervalle zerlegt wird, für die Interpolation und Integration dann separat durchgeführt werden.

Eine weitere Möglichkeit der Erhöhung der Genauigkeit bietet die Extrapolation nach Richardson und Romberg, die bereits in Kapitel 2 kurz angesprochen wurde und hier nicht weiter diskutiert werden soll.

Beispiele einfacher Integrationsformeln via Interpolation:

Stützstellen	Klasse
äquidistant $\begin{cases} \text{mit Randpunkten} \\ \text{ohne Randpunkte} \end{cases}$	$\left.\begin{array}{c} \text{geschlossene} \\ \text{offene} \end{array}\right\}$ Newton-Cotes-Formeln
Nullstellen von $\begin{cases} \text{Legendre-Polynomen } P_n \\ \text{Laguerre-Polynomen } L_n \\ \text{Hermite-Polynomen } H_n \\ \text{Jacobi-Polynomen } P_n^{\alpha,\beta} \end{cases}$	Gauß - $\left\{\begin{array}{c} \text{Legendre} \\ \text{Laguerre} \\ \text{Hermite} \\ \text{Jacobi} \end{array}\right\}$ - Formeln
$\left.\begin{array}{c} \text{Nullstellen} \\ \text{Extrema} \end{array}\right\}$ von Čebyšev-Polynomen	$\left.\begin{array}{c} \text{klassische} \\ \text{praktische} \end{array}\right\}$ Clenshaw-Curtis-Formeln

Bei allen Formeln stellen sich die folgenden spannenden Fragen:

- Ist $c_i \geq 0$ gewährleistet?
- Wie groß ist der **Genauigkeitsgrad** (GG; der Polynomgrad, bis zu dem exakt integriert wird)?

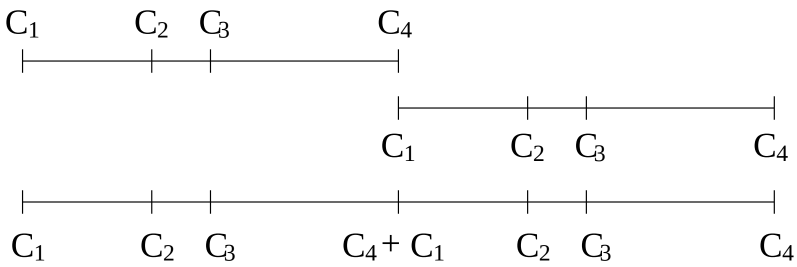
Die naheliegende äquidistante Vorgehensweise führt also auf die sog. Newton-Cotes-Formeln. Im geschlossenen Fall erhalten wir für $n = 2$ gerade die Trapez- und für $n = 3$ die Faßregel. Leider stellen sich ab $n = 9$ jedoch **negative Gewichte** c_i ein, so daß der äquidistante Ansatz nur für kleine n taugt. Außerdem ergibt sich ein magerer Genauigkeitsgrad. Es gilt für Polynome vom Grad $2n - 1$ bzw. $2n$

$$GG_{2n-1} \;=\; GG_{2n} \;=\; 2n - 1\ .$$

Deshalb werden weit häufiger die Formeln nach **Gauß-Legendre** verwendet (die klassische Gauß-Quadratur aus der Numerik-Vorlesung). Hier werden nicht nur die Gewichte dazu verwendet, einen möglichst hohen Genauigkeitsgrad zu erhalten, sondern auch die Abszissen der Stützpunkte. Da man damit bei n Stützstellen $2n - 1$ Freiheitsgrade hat, ist dies auch eine obere Schranke für den Genauigkeitsgrad, und es kann gezeigt werden, daß die Gauß-Legendre-Formeln diesen auch realisieren. Außerdem hat man stets nicht-negative Gewichte, ja es gilt sogar, daß die Gauß-Legendre-Formeln als Riemann-Summen interpretiert werden können, wodurch die Konvergenz $Q_n f \rightarrow If$ sichergestellt ist. Dies ist der klassische Fall einer **a priori Gitteroptimierung**: Ohne Kenntnis des Integranden f werden die Gitterpunkte so gewählt, daß sie für die jeweiligen Ziele (hoher Genauigkeitsgrad, keine negativen Gewichte) optimal sind.

<u>Zusammengesetzte Formeln</u>:

Das Prinzip ist einfach: Teile das Integrationsgebiet $[a, b]$ in k Teilintervalle gleicher Länge und wende dort jeweils die Quadraturformel Q_n an:



Aus der Trapezregel wird so die **Trapezsumme** ($h = (b - a)/k$)

$$h \cdot \left(\frac{1}{2} f(a) + 1 \cdot f(a+h) + \ldots + \frac{1}{2} f(b) \right)\ .$$

Ein solches Vorgehen ist i. a. sinnvoller für eine schnelle Konvergenz, als in einer einfachen Formel den Polynomgrad zu erhöhen! Außerdem sind zusammengesetzte Formeln prädestiniert für Extrapolation (man erinnere sich an die Euler-MacLaurinsche Summenformel).

5.4.3 Multivariate Integrationsformeln

Multivariate Integrationsformeln basieren zunächst auf den gleichen Konstruktionsprinzipien wie die univariaten Formeln. Die Integration im Mehrdimensionalen ist jedoch aus verschiedenen Gründen substantiell schwieriger (das gilt schon für die Analysis und erst recht für die Numerik: die Gebiete sind nicht affin äquivalent, die Definition orthogonaler Polynome ist komplizierter, ...). Außerdem lauert wieder der „Fluch der Dimensionalität“: Die Punktezahl droht, exponentiell zu wachsen. Da hochdimensionale Integrale aber an der Tagesordnung sind (Erwartungswerte usw.), muß man sich etwas einfallen lassen. Wir wollen uns im folgenden kurz mit drei verschiedenen Ansätzen befassen.

Zahlentheoretischer Ansatz:

- Eine Folge (x_i) heißt gleichverteilt auf $C = [0,1]^d$ genau dann, wenn für alle Riemann-integrierbaren Funktionen f gilt:

$$\lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} f(x_i) \;=\; \int_C f(x)\; dx \;.$$

  Machen wir uns den Sinn des Begriffs „gleichverteilt“ klar: Die charakteristische Funktion φ_E eines Subwürfels E von C ist natürlich Riemann-integrierbar. Somit gilt im gleichverteilten Fall

$$\lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} \varphi_E(x_i) \;=\; \int_C \varphi_E(x)\; dx \;=\; \mathrm{vol}(E) \;.$$

  Andererseits gibt die Summe aber gerade die Anzahl der x_i aus $x_1, \ldots, x_n$ an, die in E liegen:

$$\frac{1}{n} \cdot \sum_{i=1}^{n} \varphi_E(x_i) \;=\; \frac{1}{n} \cdot \text{ Anzahl\_Treffer\_in\_E}(x_1, \ldots, x_n) \;.$$

  Das heißt: Falls die Folge (x_i) gleichverteilt ist, dann konvergiert $Q_n f := \frac{1}{n} \sum_{i=1}^{n} f(x_i)$, also die einfache Mittelung (die denkbar einfachste Wahl der Gewichte c_i), gegen das Integral If nach Definition. Weil hier eine große Nähe zu statistischen Verfahren besteht (siehe den nachfolgenden Abschnitt), spricht man auch von Quasi-Monte-Carlo-Methoden. Der Unterschied besteht darin, daß wir hier gleichverteilte Folgen nicht über einen Zufallsprozeß, sondern deterministisch bestimmen wollen.

  Ein einfaches Beispiel einer eindimensionalen gleichverteilten Folge liefert folgende Definition:

$$x_i \;:=\; i \cdot \theta - \lfloor i \cdot \theta \rfloor, \quad \theta \in (\mathbb{R} \backslash \mathbb{Q})_+ \;.$$

  Eine d-dimensionale gleichverteilte Folge erhält man über d Zahlen $\theta_1, \ldots, \theta_d$, falls diese über $\mathbb{Q}$ linear unabhängig gewählt werden.

- Wir führen die Diskrepanz D einer endlichen Folge als Maß für ihr Abweichen von der Gleichverteilung ein. Sei $\mathcal{M} \neq \emptyset$ eine Menge meßbarer Teilmengen von $C = [0,1]^d$:

$$D_n^{\mathcal{M}}(x_1, \ldots, x_n) := \sup_{E \in \mathcal{M}} \left| \frac{\text{Anzahl\_Treffer\_in\_E}}{n} - \int_C \varphi_E(x)\; dx \right| \;.$$

  Von praktischer Bedeutung sind vor allem

$$\begin{aligned} D_n \;&:=\; D_n^{\mathcal{C}} \quad \text{mit} \quad \mathcal{C} = \{[a_1, b_1) \times \ldots \times [a_d, b_d) \subseteq C\} \;, \\ D_n^\star \;&:=\; D_n^{\mathcal{C}_0} \quad \text{mit} \quad \mathcal{C}_0 = \{[0, b_1) \times \ldots \times [0, b_d) \subseteq C\} \;. \end{aligned}$$

Es gilt

$$(x_i) \quad \text{gleichverteilt} \quad \Leftrightarrow \quad \lim_{n\to\infty} D_n = 0$$

und

$$D_n^\star \;\leq\; D_n \;\leq\; 2^d \cdot D_n^\star \;.$$

- Von fundamentaler Bedeutung ist nun die Koksma-Hlawka-Ungleichung. Sie besagt folgendes: Für jede Quadratformel $Q_n f := \frac{1}{n} \cdot \sum_{i=1}^{n} f(x_i)$ für $If = \int_C f(x)\,dx$ gilt:

  $$|Q_n f - If| \;\leq\; Vf \cdot D_n^\star(x_1, \ldots, x_n) \;.$$

  Hierbei ist $V(f)$ die sogenannte Variation von f (ein Maß für dessen Glattheit). Die Abschätzung ist scharf:

  $$\forall\,(x_i), \forall \varepsilon: \quad \exists f \text{ mit } V(f) = 1, f \in \mathcal{C}^\infty(C), \quad |Q_n f - If| > D^\star - \varepsilon \;.$$

  Wichtig ist die Quintessenz dieser Abschätzung:

  $$\text{Integrationsfehler} \;=\; \underbrace{\text{(Eigenschaft von } f\text{)}}_{\text{Variation}} \times \underbrace{\text{(Eigenschaft des Gitters)}}_{\text{Diskrepanz}}$$

  Die Eigenschaften von f (problemabhängig) sowie von (x_i), also des Gitters (strukturell), sind somit völlig entkoppelt! Daraus resultiert nun die optimale Vorgehensweise: Wähle Gitter als Folgen minimaler Diskrepanz, denn das ist das Optimum dessen, was a priori getan werden kann. Ist Die Glattheit von f lokal stark unterschiedlich, so bringt später auch adaptive Verfeinerung etwas (Isolation von Regionen großer Variation; dort mehr Punkte einsetzen).

  Bisher ist die Diskrepanz für endliche Folgen eingeführt. Beim Übergang von n nach $n+1$ kann die Suche nach Folgen minimaler Diskrepanz zu einer ganz neuen Folge führen. Dies will man jedoch oft nicht, insb. dann nicht, wenn der Aufwand der Funktionsauswertung $f(x_i)$ sehr hoch ist. In solchen Fällen möchte man das bisher Berechnete erhalten, also nur zu den schon gewählten Punkten x_1 bis x_n einen neuen hinzufügen. Damit sind auch unendliche Folgen geringer Diskrepanz von Interesse. Diese liefern für alle oder zumindest unendlich viele endlichen Ausschnitte geringe Diskrepanzwerte.

- Jetzt schauen wir uns ein paar Beispiele an. Zunächst die (eindimensionalen) van-der-Corput-Folgen:
  Entwickle $i \in \mathbb{N}_0$ in seine Darstellung zur Basis $b \in \mathbb{N},\ b \geq 2$:

  $$i \;=\; \sum_{j=0}^{J(i)} d_j(i) \cdot b^j \;.$$

  Die Ziffern schreiben wir explizit auf und spiegeln sie an der Null:

  $$i \;=\; d_j \ldots d_2 d_1 d_0 \;\mapsto\; 0.d_0 d_1 \ldots d_j \;=:\; \varphi_b(i) \;.$$

  Die Definition $x_i := \varphi_b(i-1)$ für $i = 1, \ldots$ liefert eine van-der-Corput-Folge. Es gilt für alle $n \in \mathbb{N}$

  $$D_n^\star \;=\; D_n \;=\; O\left(\frac{\log n}{n}\right),$$

  und dies ist für eine unendliche eindimensionale Folge optimal! Womit wir bei der Frage nach der minimalen Diskrepanz wären. Im eindimensionalen Fall haben wir das optimale Resultat für unendliche Folgen gerade gesehen. Bei endlichen Folgen erhält man ein etwas besseres Ergebnis (hier kann man ja für jedes n alle Punkte neu bestimmen), nämlich $O(n^{-1})$.

Für $d = 2$ gibt es noch bewiesene Aussagen, danach ist alles Vermutung. Für endliche d-dimensionale Folgen nimmt man

$$D_n^\star(x_1, \dots x_n) \;\geq\; c(d) \cdot \frac{(\log n)^{d-1}}{n}$$

an, im unendlichen Fall vermutet man

$$D_n^\star(x_1, \dots x_n) \;\geq\; c'(d) \cdot \frac{(\log n)^{d}}{n}$$

für ∞ viele n jeder unendlichen Folge.

Achtung: Man erkenne die Dünngitterformeln wieder (mit $h = n^{-1}$)!

Nun zu einem höherdimensionalen Beispiel, den Halton-Folgen. Zu d paarweise teilerfremden Basen $b_1, \dots, b_d$ erzeugt man jeweils die van-der-Corput-Folge und definiert

$$x_i \;:=\; (\varphi_{b_1}(i-1), \dots, \varphi_{b_d}(i-1)) \;, \qquad i = 1, \dots$$

Offenkundig sind Halton-Folgen unendliche Folgen. Für ihre Diskrepanz gilt

$$D_n^\star \;\leq\; c(b_1, \dots b_d) \cdot \frac{(\log n)^d}{n} \;+\; O\left(\frac{(\log n)^{d-1}}{n}\right) \;,$$

was optimal ist – vorausgesetzt, unsere obige Vermutung stimmt.

Ein Beispiel für endliche Folgen (wohl) minimaler Diskrepanz stellen die Hammersley-Folgen dar. Auch hier bilden paarweise teilerfremde b_j und daraus generierte van-der-Corput-Folgen den Ausgangspunkt:

$$x_i \;:=\; \left(\frac{i-1}{n}, \varphi_{b_1}(i-1), \dots, \varphi_{b_{d-1}}(i-1)\right) \;, \qquad i = 1, \dots, n \;.$$

Für die Diskrepanz der Hammersley-Folgen gilt

$$D_n^\star \;\leq\; c'(b_1, \dots b_{d-1}) \cdot \frac{(\log n)^{d-1}}{n} \;+\; O\left(\frac{(\log n)^{d-2}}{n}\right) \;.$$

Wenn obige Vermutung zutrifft, ist auch dies ein optimales Resultat.

Möglichst kleine konstante Faktoren $c(b_1, \dots b_d)$ bzw. $c'(b_1, \dots b_{d-1})$ erhält man in beiden Fällen, wenn man die ersten Primzahlen als b_j wählt. In der Praxis sind Hammersley-Folgen den Halton-Folgen oft deutlich überlegen.

Pseudorandom-/Monte-Carlo-Methoden:

- Im zahlentheoretischen Ansatz haben wir die Gitterpunkte (x_i) deterministisch erzeugt. Jetzt wollen wir sie als Auswertung einer (im statistischen Sinne) gleichverteilten Zufallsvariablen gewinnen. Dazu schreiben wir zunächst unser Integral If geeignet um:

$$\begin{aligned} If \;&=\; \int_B f(x)\,dx \;=\; \mathrm{vol}(B) \cdot \int_{\mathbb{R}^n} \underbrace{\mathrm{vol}(B)^{-1} \cdot \varphi_B(x)}_{\text{Wahrscheinlichkeitsdichte}} \cdot f(x)\,dx \\ &=\; \mathrm{vol}(B) \cdot \underbrace{\mu(f)}_{\text{Erwartungswert}} \;. \end{aligned}$$

Damit ist unser numerisches Quadraturproblem nichts anderes als die Approximation des Erwartungswerts durch den Mittelwert:

$$\mu(f) \;\approx\; Q_n f \;=\; \frac{1}{n} \cdot \sum_{i=1}^{n} f(x_i) \;.$$

Aufgrund des Gesetzes der großen Zahlen ist die Konvergenz gesichert:

$$Q_n f \longrightarrow \mu(f) \quad \text{für} \quad n \longrightarrow \infty \text{ mit Wahrscheinlichkeit } 1 \,.$$

- Was die Qualität obiger Approximation angeht, so liefert die Standardabweichung

  $$\sigma(Q_n f) := \sqrt{\mu(Q_n f - \mu(f))^2}$$

  einen Schätzer für $|Q_n f - \mu(f)|$. Wegen

  $$\sigma(Q_n f) \;=\; \frac{\sigma(f)}{\sqrt{n}}$$

  gilt folglich

  $$|Q_n f - \mu(f)| \;=\; O\left(\sigma(f) \cdot \frac{1}{\sqrt{n}}\right) \;.$$

  Dies ist das Pendant zur Ungleichung von Koksma-Hlawka: Der Fehler zerfällt in eine gitterabhängige Komponente ($n^{-1/2}$ ist so etwas wie die Diskrepanz von Zufallsfolgen) und in eine problemabhängige Komponente ($\sigma(f)$).

- Hauptvorteil der Monte-Carlo-Verfahren ist die Tatsache, daß die Genauigkeit und der Aufwand von der Dimensionalität d nicht abhängen. Es hat sich also ausgeflucht. Jedoch darf man nicht vergessen, daß hier nur wahrscheinlichkeitstheoretisch argumentiert werden kann („mit Wahrscheinlichkeit 1“ etc.). Ferner werden Zufallsfolgen benötigt. Schließlich ist die Genauigkeit $n^{-1/2}$ relativ schlecht. Als Ansatzpunkte zur Verbesserung bieten sich an:

  - ⋆ n erhöhen (teuer!)
  - ⋆ $\sigma(f)$ reduzieren durch varianzvermindernde Techniken:
    z. B. $If = I\tilde{f} + I(f - \tilde{f})$ mit einer Approximation $\tilde{f}$, für die $I\tilde{f}$ billig zu bestimmen ist und die $|I\tilde{f} - If| \le \varepsilon$ erfüllt; dann gilt

    $$\sigma(f - \tilde{f}) \;\le\; \varepsilon^2 \;.$$

Smolyak -Quadratur:

Hierbei handelt es sich um einen 1963 für hochdimensionale Quadraturprobleme konzipierten Tensorproduktansatz. Ziel aktueller Untersuchungen ist die Berechnung der Diskrepanz. Das Problem ist hierbei die rekursive Definition der Gewichte in den Formeln.

Gegeben sei eine eindimensionale Quadraturformel $Q_n f$:

$$Q_n f \;:=\; \sum_{i=0}^{\nu_n} \omega_i^n \cdot f(x_i^n) \;.$$

Ein Standard-Tensorproduktansatz führt auf

$$\begin{aligned} U_n^{(d)} f \;&:=\; \left(Q_n \otimes U_n^{(d-1)}\right) f \\ &=\; \left(\sum_{i=0}^{n} (Q_i - Q_{i-1}) \otimes U_n^{(d-1)}\right) f \;, \end{aligned}$$

wobei ($Q_{-1} \equiv 0$). Smolyak variiert diesen zu

$$Q_n^{(d)} f \ := \ \left(\sum_{i=0}^{n} (Q_i - Q_{i-1}) \otimes Q_{n-i}^{(d-1)} \right) f \ .$$

Man sieht: Wie bei dünnen Gittern wird hier nicht in jeder Richtung dieselbe Auflösung gewählt. Vielmehr ist die „Summe der Auflösungen" konstant gleich n (vgl. $|\underline{l}|_1 = n + d - 1$ bei dünnen Gittern).

Als eindimensionale Formel Q_n wählt man etwa zusammengesetzte Mittelpunktsformeln oder Trapezformeln, also beispielsweise:

- $Q_0 f \ = \ f(0.5)$;
- wende Q_0 auf p^n Intervalle $[i/p^n, (i+1)/p^n]$ an, $i = 0, \ldots, p^n - 1$;

Dies führt auf dünngitterähnliche Strukturen!

Vgl. die entsprechenden Dünngitterformeln:

$$\begin{aligned} V_n^{(d,\infty)} \ &= \ \sum_{|\underline{l}|_\infty \leq n} W_{\underline{l}}^{(d)} \ = \ V_n^{(1,\infty)} \otimes V_n^{(d-1,\infty)} \ = \ \sum_{l=1}^{n} W_l^{(1)} \otimes V_n^{(d-1,\infty)} \ , \\ V_n^{(d,1)} \ &= \ \sum_{|\underline{l}|_\infty \leq n+d-1} W_{\underline{l}}^{(d)} \ = \ \sum_{l=1}^{n} W_l^{(1)} \otimes V_{n+d-1-l}^{(d-1,1)} \ . \end{aligned}$$

5.5 Partielle Differentialgleichungen

Wir entwerfen einen unidirektionalen Dünngitteralgorithmus für die Laplace-Gleichung auf dem Einheitsquadrat. Die Verallgemeinerung auf allgemeine lineare elliptische Differentialgleichungen zweiter Ordnung und kompliziertere Gebiete ist möglich und schon realisiert, sprengt aber den Rahmen der Vorlesung. Wie bisher verwenden wir einen Finite-Elemente-Ansatz mit stückweise linearen hierarchischen Basen. Auch hier gilt, daß dies keine konzeptuelle Einschränkung darstellt: Es wurden bereits hierarchische Polynombasen beliebigen Grades implementiert.

<u>Wesentliche Eigenschaften/Entwurfsparadigmen/Design Patterns</u>:

- unidirektional:
  Algorithmische Action findet immer nur in einer Koordinatenrichtung statt.
- dimensionsrekursiv:
  Der unidirektionale Ansatz ermöglicht die rekursive Rückführung des d-dimensionalen Falls auf den $(d-1)$-dimensionalen. Somit wird d zu einem Eingabeparameter des Programms. Dasselbe Programm kann also beliebig dimensionale Probleme behandeln.
- modular:
  In der Dünngitterdatenstruktur wird nur die Matrix-Vektor-Multiplikation Ax realisiert. Dies ist dann Ausgangspunkt für verschiedene Iterationsverfahren (Richardson, Jacobi, cg, Mehrgitter), die jedoch völlig unabhängig von der Dünngitterei programmiert werden können (alle stützen sich algorithmisch i. w. auf Skalarprodukt und Matrix-Vektor-Produkt ab).
- effizient:
  Bei N Unbekannten, d.h. $A \in \mathbb{R}^{\mathbb{N} \times \mathbb{N}}$, erfordert die Berechnung von $A \cdot x$ $O(N)$ arithmetische Operationen und $O(N)$ Speicherplätze. Dies schließt übrigens die explizite a priori

Berechnung und Speicherung von A aus, da die hierarchische Basis ja zu einem fill-in der Steifigkeitsmatrix A führt (durchschnittlich O(Baumtiefe) Nicht-Nullen pro Zeile). Die explizite Ermittlung von A ist aber für eine effiziente Berechnung der $\sum\limits_j a_{ij}x_j$ für alle i gar nicht erforderlich!

Gestalt der Einträge a_{ij}:

Wir starten mit der (Tensorprodukt-) Gestalt einer d-dimensionalen hierarchischen Basisfunktion φ_i:

$$\varphi_i(x_1,\dots,x_d) \;=\; \prod_{l=1}^{d} \varphi_{i,l}(x_l)\ ,$$

wobei $\varphi_{i,l}(x_l)$ gerade eine eindimensionale hierarchische Basisfunktion darstellt (im linearen Fall also unsere bekannte Hutfunktion). Wir wissen aus Kapitel 4, daß für den Laplace-Operator $\triangle$ die Matrixeinträge a_{ij} folgende Gestalt haben:

$$a_{ij} \;=\; \int\limits_{\Omega} \nabla\varphi_i(x_1,\dots,x_d) \underbrace{\bullet}_{\text{Skalarprodukt}} \nabla\varphi_j(x_1,\dots,x_d)\ d\Omega\ .$$

Setzen wir hier die Produktdarstellung von φ_i und φ_j ein, so folgt

$$\begin{aligned}
\nabla\varphi_i \bullet \nabla\varphi_j &= \sum_{k=1}^{d} \frac{\partial\varphi_i}{\partial x_k}\cdot\frac{\partial\varphi_j}{\partial x_k} \\
&= \sum_{k=1}^{d} \left(\frac{\partial}{\partial x_k}\prod_{l=1}^{d}\varphi_{i,l}(x_l)\right)\cdot\left(\frac{\partial}{\partial x_k}\prod_{l=1}^{d}\varphi_{j,l}(x_l)\right) \\
&= \sum_{k=1}^{d} \left(\frac{\partial}{\partial x_k}\varphi_{i,k}(x_k)\cdot\frac{\partial}{\partial x_k}\varphi_{j,k}(x_k)\cdot\prod_{l\neq k}\left(\varphi_{i,l}(x_l)\cdot\varphi_{j,l}(x_l)\right)\right)\ .
\end{aligned}$$

Für $a_{i,j}$ gilt somit

$$\begin{aligned}
a_{ij} &= \int\limits_{\mathrm{supp}(\varphi_i)\cap\mathrm{supp}(\varphi_j)} \nabla\varphi_i\bullet\nabla\varphi_j\ d\Omega \\
&= \int\limits_{\mathrm{supp}(\varphi_i)\cap\mathrm{supp}(\varphi_j)} \sum_{k=1}^{d}\left(\frac{\partial}{\partial x_k}\varphi_{i,k}(x_k)\cdot\frac{\partial}{\partial x_k}\varphi_{j,k}(x_k)\cdot\prod_{l\neq k}\varphi_{i,l}(x_l)\cdot\varphi_{j,l}(x_l)\right) d\Omega \\
&= \sum_{k=1}^{d}\ \int\limits_{\mathrm{supp}(\varphi_i)\cap\mathrm{supp}(\varphi_j)} \left(\frac{\partial}{\partial x_k}\varphi_{i,k}(x_k)\cdot\frac{\partial}{\partial x_k}\varphi_{j,k}(x_k)\cdot\prod_{l\neq k}\varphi_{i,l}(x_l)\cdot\varphi_{j,l}(x_l)\right) d\Omega \\
&= \sum_{k=1}^{d}\left(\ \int\limits_{\mathrm{supp}(\varphi_{i,k}\cdot\varphi_{j,k})} \frac{\partial}{\partial x_k}\varphi_{i,k}(x_k)\cdot\frac{\partial}{\partial x_k}\varphi_{j,k}(x_k)\ dx_k\cdot\prod_{l\neq k}\ \int\limits_{\mathrm{supp}(\varphi_{i,l}\cdot\varphi_{j,l})} \varphi_{i,l}(x_l)\cdot\varphi_{j,l}(x_l)\ dx_l\right) \\
&=: \sum_{k=1}^{d}\prod_{l=1}^{d} a_{ij}(k,l),
\end{aligned}$$

wobei $a_{ij}(k,l)$ entsprechend definiert sei.

Folglich haben wir d Summanden, von denen jeder ein Produkt von d eindimensionalen Integralen ist, und zwar

- ein Integral über ein Produkt zweier erster Ableitungen eindimensionaler Basisfunktionen,

$$\int_{\mathrm{supp}(\varphi_{i,k}\cdot\varphi_{j,k})} \frac{\partial}{\partial x_k}\varphi_{i,k}(x_k) \cdot \frac{\partial}{\partial x_k}\varphi_{j,k}(x_k)\ dx_k\ ,$$

  sowie

- $d-1$ Integrale über ein Produkt zweier eindimensionaler Basisfunktionen,

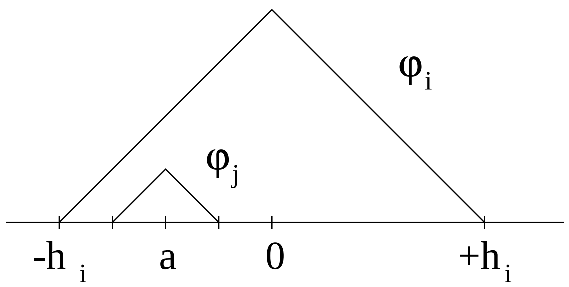
$$\int_{\mathrm{supp}(\varphi_{i,l}\cdot\varphi_{j,l})} \varphi_{i,l}(x_l) \cdot \varphi_{j,l}(x_l)\ dx_l\ .$$

Damit ist die Grundlage für eine unidirektionale Vorgehensweise gelegt: Alle zu realisierenden Teile des Algorithmus sind inhärent eindimensional! Zwei Typen von Integralen sind also zu berechnen:

$$\left.\begin{array}{l} \bullet\ \int \varphi_i' \cdot \varphi_j'\ dx \\ \bullet\ \int \varphi_i \cdot \varphi_j\ dx \end{array}\right\} \text{mit 1D-Basisfunktionen } \varphi_i,\ \varphi_j.$$

Da wir ja aber nicht jedes a_{ij} explizit und separat berechnen wollen, sondern gleich die akkumulierten Summen $\sum_j a_{ij}x_j$, erschlagen wir für festes i alle j auf einmal. Natürlich müssen nur die j berücksichtigt werden, für die die Träger von φ_i und φ_j überlappen (sonst gilt $a_{ij}=0$). Hierbei müssen drei Fälle unterschieden werden:

Fall 1: j hierarchisch niedriger als i, $\mathrm{supp}(\varphi_j) \subset \mathrm{supp}(\varphi_i)$



$$\varphi_i(x) = \begin{cases} \frac{x+h_i}{h_i}, & -h_i \le x \le 0, \\ \frac{h_i-x}{h_i}, & 0 \le x \le h_i, \end{cases}$$

$$\varphi_i'(x) = \begin{cases} \frac{1}{h_i}, & -h_i \le x \le 0, \\ \frac{-1}{h_i}, & 0 \le x \le h_i, \end{cases}$$

$$\varphi_j(x) = \begin{cases} \frac{x-(a-h_j)}{h_j}, & a-h_j \le x \le a, \\ \frac{a+h_j-x}{h_j}, & a \le x \le a+h_j, \end{cases}$$

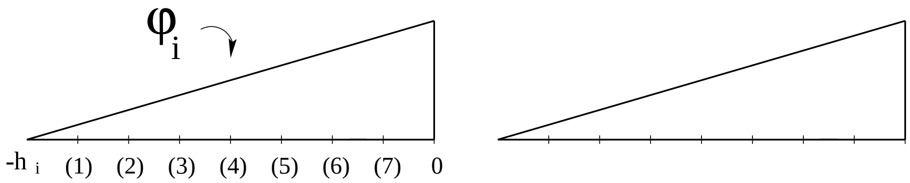
$$\varphi_j'(x) = \begin{cases} \frac{1}{h_j}, & a-h_j \le x \le a, \\ \frac{-1}{h_j}, & a \le x \le a+h_j. \end{cases}$$

Hier ergibt sich für unsere beiden Integrale

$$\int\limits_{a-h_j}^{a+h_j} \varphi_i' \cdot \varphi_j' \, dx \;=\; \int\limits_{a-h_j}^{a} \frac{1}{h_i} \cdot \frac{1}{h_j} \, dx \;+\; \int\limits_{a}^{a+h_j} \frac{-1}{h_i} \cdot \frac{1}{h_j} \, dx \;=\; 0\,,$$

$$\int\limits_{a-h_j}^{a+h_j} \varphi_i \cdot \varphi_j \, dx \;=\; \ldots \;=\; \frac{h_j(a+h_i)}{h_i}\,.$$

Diese Werte schauen wir uns genauer an:



(1) $a = -\frac{7}{8} \cdot h_i \;,\; h_j = \frac{1}{8} \;\Rightarrow\; \frac{1}{64} h_i$

(2) $a = -\frac{3}{4} \cdot h_i \;,\; h_j = \frac{1}{4} \;\Rightarrow\; \frac{1}{16} h_i$

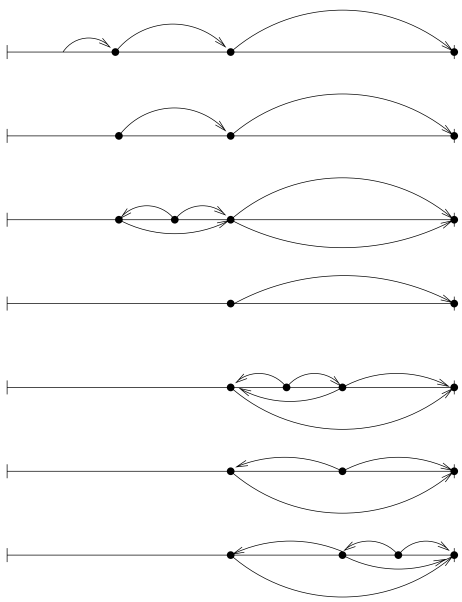
(3) $a = -\frac{5}{8} \cdot h_i \;,\; h_j = \frac{1}{8} \;\Rightarrow\; \frac{3}{64} h_i$

(4) $a = -\frac{1}{2} \cdot h_i \;,\; h_j = \frac{1}{2} \;\Rightarrow\; \frac{1}{4} h_i$

(5) $a = -\frac{3}{8} \cdot h_i \;,\; h_j = \frac{1}{8} \;\Rightarrow\; \frac{5}{64} h_i$

(6) $a = -\frac{1}{4} \cdot h_i \;,\; h_j = \frac{1}{4} \;\Rightarrow\; \frac{3}{16} h_i$

(7) $a = -\frac{1}{8} \cdot h_i \;,\; h_j = \frac{1}{8} \;\Rightarrow\; \frac{7}{64} h_i$



Wenn jeder Knoten j an seine hierarchischen Nachbarn (Randpunkte des Trägers von φ_j) das bisher von unten Akkumulierte mit dem Gewicht $\frac{1}{2}$ versieht und weiterreicht bzw. „wirft“ (ganz unten, d. h. in den Blättern: h_j), dann kommt in jedem Knoten die richtige Gewichtung an!

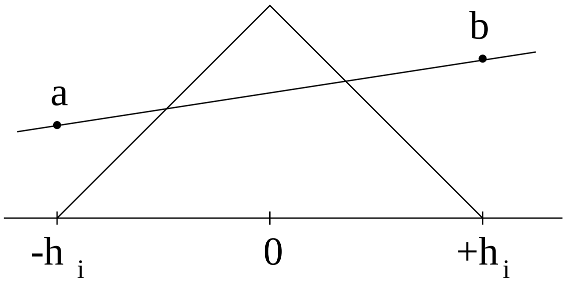
Fall 2: $j = i$

Hier gilt für die beiden Integrale:

$$\int_{-h_i}^{+h_i} \left(\varphi_i'(x)\right)^2 \, dx = \int_{-h_i}^{+h_i} \frac{1}{h_i^2} \, dx = \frac{2}{h_i} \, ,$$

$$\int_{-h_i}^{+h_i} \varphi_i^2(x) \, dx = \ldots = \frac{2}{3} \cdot h_i \, .$$

Fall 3: i hierarchisch höher als j



Alles hierarchisch Höhere zusammengenommen führt zu einer auf $[-h_i, h_i]$ linearen Funktion $F(x)$ mit Funktionswerten $F(-h_i) = a$ und $F(h_i) = b$, also

$$F(x) := \frac{a(h_i - x) + b(h_i + x)}{2h_i} \, ,$$
$$F'(x) = \frac{b-a}{2h_i} \, .$$

Damit folgt für die Integrale:

$$\int_{-h_i}^{+h_i} F' \cdot \varphi_i' \, dx = \int_{-h_i}^{0} \frac{b-a}{2h_i} \cdot \frac{1}{h_i} \, dx + \int_{0}^{+h_i} \frac{b-a}{h_i} \cdot \frac{-1}{h_i} \, dx = 0 \, ,$$

$$\int_{-h_i}^{+h_i} F \cdot \varphi_j \, dx = \ldots = \frac{a+b}{2} \cdot h_i \, .$$

Bemerkung:

Anhand von Fall 1 und 3 sieht man, daß nicht die a_{ij} bzw. die Integrale $\int \varphi_i \cdot \varphi_j \, dx$ sowie $\int \varphi_i' \cdot \varphi_j' \, dx$ einzeln berechnet werden, sondern vielmehr werden direkt Teilsummen von $\sum a_{ij} x_j$ berechnet:

$$\sum_j a_{ij} x_j \; = \sum_{j \text{ hierarchisch niedriger } i} a_{ij} x_j \; + \; a_{ii} x_i \; + \sum_{j \text{ hierarchisch höher } i} a_{ij} x_j \; .$$

Dies alles kann in einem vollständigen Baumdurchlauf pro Koordinatenrichtung erfolgen (einmal top-down für Fall 3, einmal bottom-up für Fall 1).

Wir machen uns die Bedeutung der Summe $\sum a_{ij} x_j$ klar:

$$\begin{aligned}
\sum_j a_{ij} x_j &= \sum_j \left(\int_\Omega \nabla \varphi_i \bullet \nabla \varphi_j \, d\Omega \right) x_j \\
&= \int_\Omega \nabla \left(\sum_j x_j \cdot \varphi_j \right) \bullet \nabla \varphi_i \, d\Omega \\
&= \int_\Omega \nabla v^{akt} \bullet \nabla \varphi_i \, d\Omega \; .
\end{aligned}$$

Hierbei ist $v^{akt} := \sum\limits_j x_j \cdot \varphi_j$ die aktuelle Näherungslösung. Die Summe zerfällt in einen hierarchisch höheren („down“), einen hierarchisch tieferen („up“) und einen lokalen (φ_i) Anteil.

Zusammenfassung:

k-ter Summand, d. h. Ableitung nach x_k	j niedriger i	j = i	j höher i
Summation/Werfen/Berechnung der Integrale in Richtung k (d. h. $\int \varphi_i' \cdot \varphi_j'$)	0	$\frac{2}{h_i}$	0
Summation/Werfen/Berechnung der Integrale in Richtung $l \neq k$ (d. h. $\int \varphi_i \cdot \varphi_j$)	$\frac{h_j(a+h_i)}{h_i}$	$\frac{2}{3} h_i$	$\frac{a+b}{2} h_i$

h_i , h_j: h-Werte in Summationsrichtung k bzw. l!

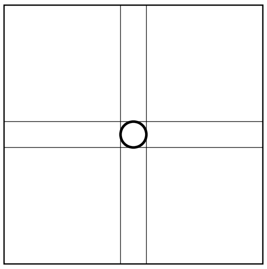
Wegen $a_{ij} \; = \; \sum\limits_{k=1}^{d} \prod\limits_{l=1}^{d} a_{ij}(k,l)$ gilt (die $a_{ij}(k,l)$ sind die Tabelleneinträge von soeben):

(1) Falls in allen d Koordinatenrichtungen i hierarchisch höher oder hierarchisch niedriger als j ist, gilt $a_{ij} = 0$ (ein Faktor verschwindet in jedem Summand).

(2) Für jede Richtung k mit $x_{i,k} \; = \; x_{j,k}$ kann sich ein von Null verschiedener Summand ergeben.

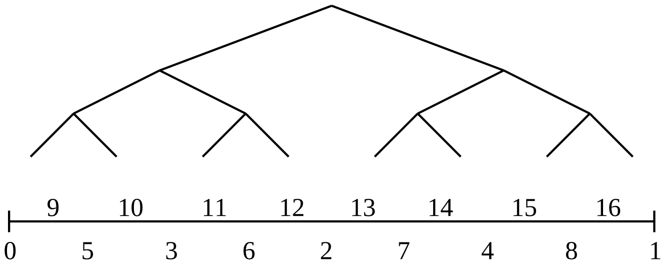
Beispiel 2 D:

nur vom Kreuz kommen Beiträge!



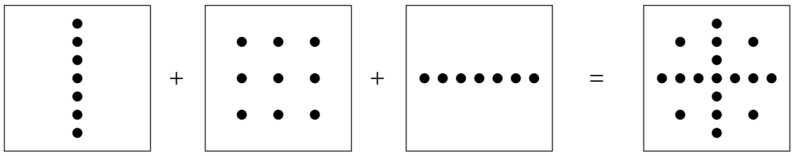
Vorbesetzungen im Perl-Programm (2D):

$$\forall \;\; t1 \;=\; 0 \;\ldots\; t$$
$$\forall \;\; i1 \;=\; 2 \;\ldots\; 2\hat{\;}t1$$
$$\forall \;\; i2 \;=\; 2 \;\ldots\; 2\hat{\;}(t-t1)$$
$$x[1] \;=\; i1$$
$$x[2] \;=\; i2$$



Beispiel t = 4 $\qquad (\; t \;\hat{=}\; n+d-1 \;,$ d.h. $t \;=\; 4 \;\hat{=}\; n \;=\; 3 \;)$

t1 = 0	—	
t1 = 1	i1 = 2	i2 = 2...8
t1 = 2	i1 = 2...4	i2 = 2...4
t1 = 3	i1 = 2...8	i2 = 2
t1 = 4	—	



Lösung des Gleichungssystems im Perl-Programm:

$$\begin{array}{lll} Ax \; = \; b & & (\; x \mathrel{\hat{=}} v \;) \\ \to \text{Richardson-Iteration:} & & x^{(i+1)} \; := \; x^{(i)} \; - \; \Theta \cdot (Ax^{(i)} - b) \end{array}$$

Algorithmus:

$$v^{(0)} \; := \; \begin{cases} v_{Rand} & \text{am Rand} \qquad \text{(Dirichlet)} \\ 0 & \text{im Inneren} \qquad \text{(bilinearer Interpolant des Randes)} \end{cases}$$

(v hierarchische Koeffizienten/Überschüsse)

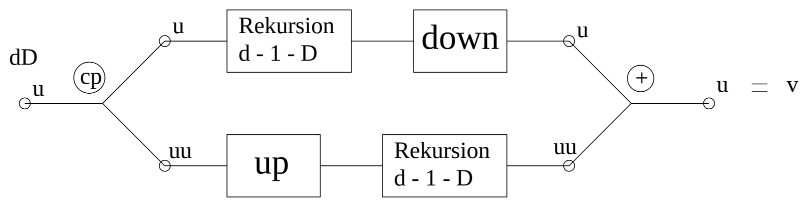
$$r^{(0)} \; := \; \begin{cases} A \cdot v^{(0)} & \text{im Inneren} \\ 0 & \text{am Rand} \end{cases}$$

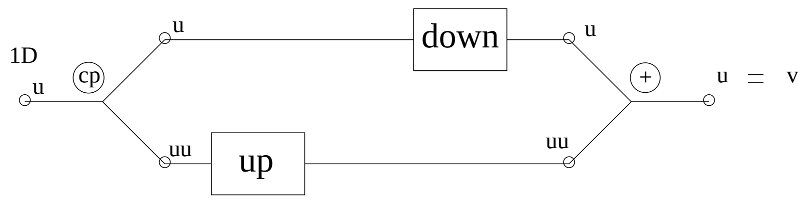
Korrekturen nur im Inneren $\quad (\; \delta v^{(i)}|_{Rand} \; \equiv \; \forall \, i \;) \; !$

$$\begin{aligned} v^{(i+1)} \; &:= \; v^{(i)} \; + \; \delta v^{(i)} \\ &= \; v^{(i)} \; - \; \Theta \cdot r^{(i)} \\ r^{(i)} \; &= \; A \cdot v^{(i)} \\ &= \; A \left(v^{(i-1)} \; + \; \delta v^{(i-1)} \right) \\ &= \; r^{(i-1)} \; + \; A \cdot \delta v^{(i-1)} \\ &= \; r^{(i-1)} \; + \; \delta r^{(i-1)} \end{aligned}$$

$\to$ alles wird inkrementell berechnet: $\quad \delta v^{(i)} \; , \, \delta r^{(i)} \; = \; A \cdot \delta v^{(i)}$

Rekursionsschema:

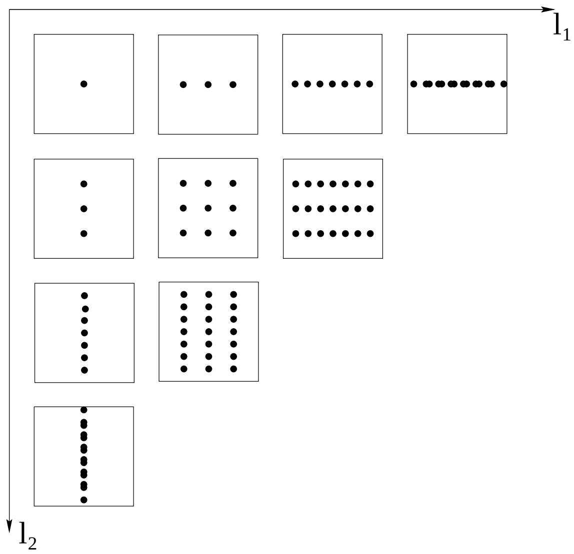




d.h. abgesehen von der Kopie erfolgt die Berechnung am Platz:
am Anfang enthält die Variable den Vektor x, am Ende das Produkt $A \cdot x$

5.6 Kombinationstechnik

Wir haben in diesem Kapitel gesehen, daß die Dünngitterräume $V_n^{(1)}$ und $V_n^{(E)}$ eine deutliche Effizienzsteigerung mit sich bringen und in bezug auf die Kosten-Nutzen-Relation sogar optimal sind. Ärgerlich ist nur, daß vieles neu zu programmieren ist und daß die Algorithmen z. T. wesentlich komplizierter werden (vgl. etwa Abschnitt 5.5). Es stellt sich somit die Frage, ob man nicht billiger zu Dünngitterlösungen und damit auch zu Dünngittereffizienz kommt. Dazu gehen wir zunächst zum Teilraumschema der $V_{\underline{l}}$:



Mit $\Omega_{\underline{l}}$ bezeichnen wir das regelmäßige Gitter der Maschenweite $\underline{h} = 2^{-\underline{l}} = (2^{-l_1}, \ldots 2^{-l_d})$, $u_{\underline{l}} \in V_{\underline{l}}$ sei eine auf $\Omega_{\underline{l}}$ berechnete Näherungslösung einer gegebenen PDE $Lu = f$. Ferner sei $\chi_{\underline{l}}$ die charakteristische Funktion von $\Omega_{\underline{l}}$ (d. h. 1 in allen Gitterpunkten von $\Omega_{\underline{l}}$ und 0 sonst). Im folgenden beschränken wir uns der Einfachheit halber auf den zweidimensionalen Fall. Für

$$\chi_n^{(c)}(\underline{x}) := \sum_{|\underline{l}|_1 = n+1} \chi_{\underline{l}}(\underline{x}) - \sum_{|\underline{l}|_1 = n} \chi_{\underline{l}}(\underline{x})$$

gilt $\chi_n^{(c)}(\underline{x}) = 1$ für alle Gitterpunkte des L_∞- bzw. L_2-dünnen Gitters der Tiefe n. Dies führt uns auf folgende Definition:

$$u_n^{(c)}(\underline{x}) := \sum_{|\underline{l}|_1 = n+1} u_{\underline{l}}(\underline{x}) - \sum_{|\underline{l}|_1 = n} u_{\underline{l}}(\underline{x}) .$$

$u_n^{(l)}(\underline{x})$ liegt in $V_n^{(1)}$ und heißt *Kombinationslösung.* Obiger extrapolationsähnliche Ansatz, der aus einer Menge von (groben) Vollgitterlösungen durch Linearkombination eine Dünngitterlösung erzeugt, wird *Kombinationstechnik* genannt.

Im Dreidimensionalen gilt

$$u_n^{(c)}(\underline{x}) = \sum_{|\underline{l}|_1 = n+2} u_{\underline{l}}(\underline{x}) - 2 \cdot \sum_{|\underline{l}|_1 = n+1} u_{\underline{l}}(\underline{x}) + \sum_{|\underline{l}|_1 = n} u_{\underline{l}}(\underline{x})$$

Gegenüber der unmittelbaren Diskretisierung auf dünnen Gittern bietet die Kombinationstechnik vor allem zwei wesentliche Vorteile:

(1) Zur Erreichung der Summanden $u_{\underline{l}}(\underline{x})$ können bereits verfügbare Standardcodes eingesetzt werden. $u_{\underline{l}}(\underline{x})$ ist ja Nährung zur Lösung u von $Lu = f$ auf dem (vollen) Gitter $\Omega_{\underline{l}}$, das jedoch mit $O(2^{-n})$ Punkten sehr grob ist. Alle $u_{\underline{l}}(\underline{x})$ sind also billig und mit Standardprogrammen zu berechnen. Damit kann man dünne Gitter auch schnell an komplizierten Problemen erproben (z. B. Turbulenzsimulation in der numerischen Strömungsmechanik).

(2) Die Lösungen $u_{\underline{l}}(\underline{x})$ können parallel berechnet werden. Da sich die einzelnen $\Omega_{\underline{l}}$ hinsichtlich ihrer Größe nicht stark unterscheiden (auf jeder Diagonallinie bzw. -fläche im Teilraumschema ist die Punktzahl von derselben Größenordnung, beim Übergang von einer Diagonallinie bzw. -fläche zur nächsten ergibt sich ein Faktor 2), ist auch die Lastbalancierung einfach.

Man kann zeigen, daß unter bestimmten Glattheitsvoraussetzungen die Kombinationslösung die typische Dünngittergenauigkeit hat $(h^2 |\log h|^{d-1}$ bzgl. L_2- und L_∞-Norm).

Vergleich:

	Dünngitterdiskretisierung (FE)	Kombinationstechnik
Kosten	$O(2^n \cdot n^{d-1})$	$O(2^n \cdot n^{d-1})$
Interpolation		
L_2	$O(4^{-n} \cdot n^{d-1})$	$O(4^{-n} \cdot n^{d-1})$
L_∞		
Energie	$O(2^{-n})$	?
PDE-Lösung		
L_2	$O(4^{-n} \cdot n^{d-1})$???	$O(4^{-n} \cdot n^{d-1})$
L_∞		
Energie	$O(2^{-n})$	?