low level TEX

paragraphs

Contents

1 Introduction 1
2 Properties 1
3 Wraping up 3
4 Shapes 4
5 Modes 11
6 Normalization 11

1 Introduction

This manual is mostly discussing a few wrappers around low level TEX features. Its writing is triggered by an update to the MetaFun and LuaMetaFun manuals where we mess a bit with shapes. It gave a good reason to also cover some more paragraph related topics but it might take a while to complete. Remind me if you feel that takes too much time.

2 Properties

A paragraph is just a collection of lines that result from one input line that got broken. This process of breaking into lines is influenced by quite some parameters. In traditional TEX and also in LuaMetaTEX by default the values that are in effect when the end of the paragraph is met are used. So, when you change them in a group and then ends the paragraph after the group, the values you've set in the group are not used.

However, in LuaMetaTEX we can optionally store them with the paragraph. When that happens the values current at the start are frozen. You can still overload them but that has to be done explicitly then. The advantage is that grouping no longer interferes with the line break algorithm. The magic primitive is `\snapshotpar`.

```
\hsize
\leftskip
\rightskip
\hangindent
\hangafter
\parindent
\parfillleftskip
\parfillrightskip
\adjustspacing
\protrudechars
```

```
\pretolerance
\tolerance
\emergencystretch
\looseness
\lastlinefit
\linepenalty
\interlinepenalty
\clubpenalty
\widowpenalty
\displaywidowpenalty
\brokenpenalty
\adjdemerits
\doublehyphendemerits
\finalhyphendemerits
\parshape
\interlinepenalties
\clubpenalties
\widowpenalties
\displaywidowpenalties
\baselineskip
\lineskip
\lineskiplimit
\adjustspacingstep
\adjustspacingshrink
\adjustspacingstretch
\hyphenationmode
```

There are more paragraph related parameters than in for instance pdfTEX and LuaTEX and these are (to be) explained in the LuaMetaTEX manual. You can imagine that keeping this around with the paragraph adds some extra overhead to the machinery but most users won't notice that because is is compensated by gains elsewhere.

In LMTX taking these snapshots is turned on by default and because it thereby fundamentally influences the par builder, users can run into compatibility issues but in practice there has been no complaints (and this feature has been in use quite a while before this document was written). One reason for users not noticing is that one of the big benefits is probably handled by tricks mentioned on the mailing list. Imagine that you have this:

```
{\bf watch out:} here is some text
```

In this small example the result will be as expected. But what if something magic with the start of a paragraph is done? Like this:

```
\placefigure[left]{A cow!}{\externalfigure[cow.pdf]}

{\bf watch out:} here is some text ... of course much more is needed to
    get a flow around the figure!
```

The figure will hang at the left side of the paragraph but it is put there when the text starts and that happens inside the bold group. It means that the properties we set in order to get the shape around the figure are lost as soon as we're at '`here is some text`' and definitely is wrong when the paragraph ends and the par builder has to use them to get the shape right. We get text overlapping the figure. A trick to overcome this is:

```
\dontleavehmode {\bf watch out:} here is some text ... of course much
    more is needed to get a flow around the figure!
```

where the first macro makes sure we already start a paragraph before the group is entered (using a `\strut` also works). It's not nice and I bet users have been bitten by this and by now know the tricks. But, with snapshots such fuzzy hacks are not needed any more! The same is true with this:

```
{\leftskip 1em some text \par}
```

where we had to explicitly end the paragraph inside the group in order to retain the skip. I suppose that users normally use the high level environments so they never had to worry about this. It's also why users probably won't notice that this new mechanism has been active for a while. Actually, when you now change a parameter inside the paragraph will not be applied (unless you prefix it with `\frozen`) but no one did that anyway.

todo: freeze categories, overloading, turning on and off, etc

3 Wraping up

In ConTEXt LMTX we have a mechanism to exercise macros (or content) before a paragraph ends. This is implemented using the `\wrapuppar` primitive. The to be wrapped up material is bound to the current paragraph which in order to get this done has to be started when this primitive is used.

Although the high level interface has been around for a while it still needs a bit more

testing (read: use cases are needed). In the few cases where we already use it application can be different because again it relates to snapshots. This because in the past we had to use tricks that also influenced the user interface of some macros (which made them less natural as one would expect). So the question is: where do we apply it in old mechanisms and where not.

todo: accumulation, interference, where applied, limitations

4 Shapes

In ConTEXt we don't use `\parshape` a lot. It is used in for instance side floats but even then not in all cases. It's more meant for special applications. This means that in MkII and MkIV we don't have some high level interface. However, when MetaFun got upgraded to LuaMetaFun, and the manual also needed an update, one of the examples in that manual that used shapes also got done differently (read: nicer). And that triggered the arrival of a new high level shape mechanism.

One important property of the `\parshape` mechanism is that it works per paragraph. You define a shape in terms of a left margin and width of a line. The shape has a fixed number of such pairs and when there is more content, the last one is used for the rest of the lines. When the paragraph is finished, the shape is forgotten.

Not discussed here is a variant that will end up in LuaMetaTEX that works with the progression, i.e. takes the height of the content so far into account. This is somewhat tricky because for that to work vertical skips need to be frozen, which is no real big deal but has to be done careful in the code.

The high level interface is a follow up on the example in the MetaFun manual and uses shapes that carry over to the next paragraph. In addition we can cycle over a shape. In this interface shapes are defined using keyword. Here are some examples:

```
\startparagraphshape[test]
    left 1mm right 1mm
    left 5mm right 5mm
\stopparagraphshape
```

This shape has only two entries so the first line will have a 1mm margin while later lines will get 5mm margins. This translates into a `\parshape` like:

```
\parshape 2
    1mm \dimexpr\hsize-1mm\relax
    5mm \dimexpr\hsize-5mm\relax
```

Watch the number 2: it tells how many specification lines follow. As you see, we need to calculate the width.

```
\startparagraphshape[test]
    left 1mm right 1mm
    left 5mm right 5mm
    repeat
\stopparagraphshape
```

This variant will alternate between 1mm and 5mm margins. The repeating feature is translated as follows. Maybe at some point I will introduce a few more options.

```
\parshape 2 options 1
    1mm \dimexpr\hsize-1mm\relax
    5mm \dimexpr\hsize-5mm\relax
```

A shape can have some repetition, and we can save keystrokes by copying the last entry. The resulting `\parshape` becomes rather long.

```
\startparagraphshape[test]
    left 1mm right 1mm
    left 2mm right 2mm
    left 3mm right 3mm
    copy 8
    left 4mm right 4mm
    left 5mm right 5mm
    left 5mm hsize 10cm
\stopparagraphshape
```

Also watch the `hsize` keyword: we don't calculate the hsize from the `left` and `right` values but explicitly set it.

```
\startparagraphshape[test]
    left 1mm right 1mm
    right 3mm
    left 5mm right 5mm
    repeat
\stopparagraphshape
```

When a `right` keywords comes first the `left` is assumed to be zero. In the examples that follow we will use a couple of definitions:

```
\startparagraphshape[test]
    both 1mm both 2mm both 3mm both 4mm both 5mm both 6mm
    both 7mm both 6mm both 5mm both 4mm both 3mm both 2mm
\stopparagraphshape

\startparagraphshape[test-repeat]
    both 1mm both 2mm both 3mm both 4mm both 5mm both 6mm
    both 7mm both 6mm both 5mm both 4mm both 3mm both 2mm
    repeat
\stopparagraphshape
```

The last one could also be defines as:

```
\startparagraphshape[test-repeat]
    \rawparagraphshape{test} repeat
\stopparagraphshape
```

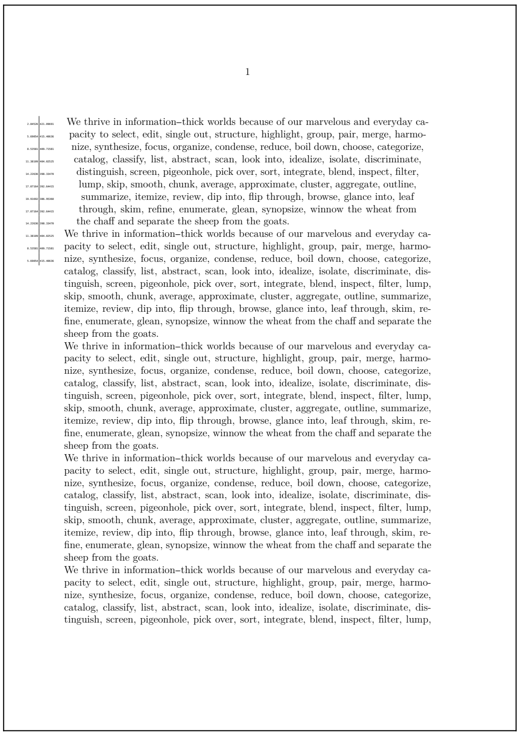
In the previous code we already introduced the `repeat` option. This will make the shape repeat at the engine level when the shape runs out of specified lines. In the application of a shape definition we can specify a `method` to be used and that determine if the next paragraph will start where we left off and discard afterwards (`shift`) or that we move the discarded lines up front so that we never run out of lines (`cycle`). It sounds complicated but just keep in mind that `repeat` is part of the `\parshape` and act within a paragraph while `shift` and `cycle` are applied when a new paragraph is started.

In figure 1 you see the following applied:

```
\startshapedparagraph[list=test]
    \dorecurse{8}{\showparagraphshape\samplefile{tufte} \par}
\stopshapedparagraph

\startshapedparagraph[list=test-repeat]
    \dorecurse{8}{\showparagraphshape\samplefile{tufte} \par}
\stopshapedparagraph
```

In figure 2 we use this instead:

```
\startshapedparagraph[list=test,method=shift]
    \dorecurse{8}{\showparagraphshape\samplefile{tufte} \par}
\stopshapedparagraph

\startshapedparagraph[list=test-repeat]
    \dorecurse{8}{\showparagraphshape\samplefile{tufte} \par}
```



1

We thrive in information–thick worlds because of our marvelous and everyday capacity to select, edit, single out, structure, highlight, group, pair, merge, harmonize, synthesize, focus, organize, condense, reduce, boil down, choose, categorize, catalog, classify, list, abstract, scan, look into, idealize, isolate, discriminate, distinguish, screen, pigeonhole, pick over, sort, integrate, blend, inspect, filter, lump, skip, smooth, chunk, average, approximate, cluster, aggregate, outline, summarize, itemize, review, dip into, flip through, browse, glance into, leaf through, skim, refine, enumerate, glean, synopsize, winnow the wheat from the chaff and separate the sheep from the goats.

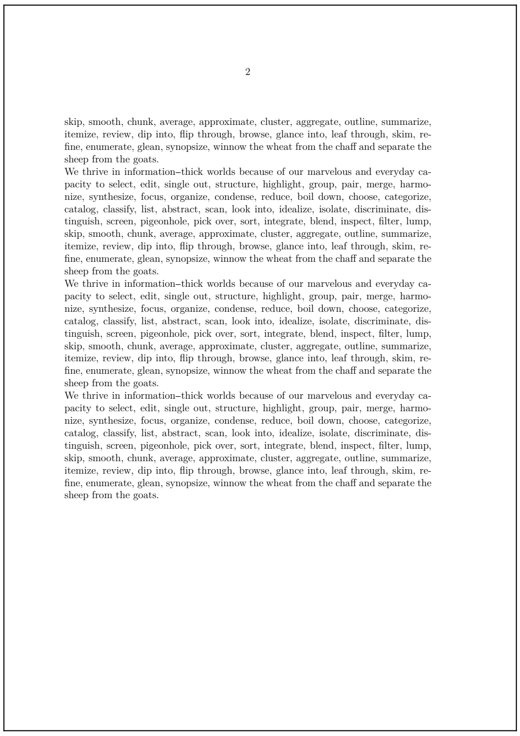
We thrive in information–thick worlds because of our marvelous and everyday capacity to select, edit, single out, structure, highlight, group, pair, merge, harmonize, synthesize, focus, organize, condense, reduce, boil down, choose, categorize, catalog, classify, list, abstract, scan, look into, idealize, isolate, discriminate, distinguish, screen, pigeonhole, pick over, sort, integrate, blend, inspect, filter, lump, skip, smooth, chunk, average, approximate, cluster, aggregate, outline, summarize, itemize, review, dip into, flip through, browse, glance into, leaf through, skim, refine, enumerate, glean, synopsize, winnow the wheat from the chaff and separate the sheep from the goats.

We thrive in information–thick worlds because of our marvelous and everyday capacity to select, edit, single out, structure, highlight, group, pair, merge, harmonize, synthesize, focus, organize, condense, reduce, boil down, choose, categorize, catalog, classify, list, abstract, scan, look into, idealize, isolate, discriminate, distinguish, screen, pigeonhole, pick over, sort, integrate, blend, inspect, filter, lump, skip, smooth, chunk, average, approximate, cluster, aggregate, outline, summarize, itemize, review, dip into, flip through, browse, glance into, leaf through, skim, refine, enumerate, glean, synopsize, winnow the wheat from the chaff and separate the sheep from the goats.

We thrive in information–thick worlds because of our marvelous and everyday capacity to select, edit, single out, structure, highlight, group, pair, merge, harmonize, synthesize, focus, organize, condense, reduce, boil down, choose, categorize, catalog, classify, list, abstract, scan, look into, idealize, isolate, discriminate, distinguish, screen, pigeonhole, pick over, sort, integrate, blend, inspect, filter, lump, skip, smooth, chunk, average, approximate, cluster, aggregate, outline, summarize, itemize, review, dip into, flip through, browse, glance into, leaf through, skim, refine, enumerate, glean, synopsize, winnow the wheat from the chaff and separate the sheep from the goats.

We thrive in information–thick worlds because of our marvelous and everyday capacity to select, edit, single out, structure, highlight, group, pair, merge, harmonize, synthesize, focus, organize, condense, reduce, boil down, choose, categorize, catalog, classify, list, abstract, scan, look into, idealize, isolate, discriminate, distinguish, screen, pigeonhole, pick over, sort, integrate, blend, inspect, filter, lump,

discard, finite shape, page 1



2

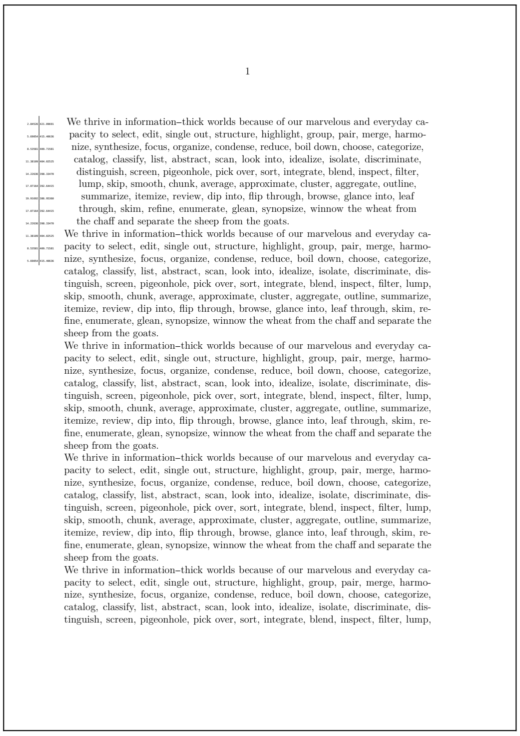
skip, smooth, chunk, average, approximate, cluster, aggregate, outline, summarize, itemize, review, dip into, flip through, browse, glance into, leaf through, skim, refine, enumerate, glean, synopsize, winnow the wheat from the chaff and separate the sheep from the goats.

We thrive in information–thick worlds because of our marvelous and everyday capacity to select, edit, single out, structure, highlight, group, pair, merge, harmonize, synthesize, focus, organize, condense, reduce, boil down, choose, categorize, catalog, classify, list, abstract, scan, look into, idealize, isolate, discriminate, distinguish, screen, pigeonhole, pick over, sort, integrate, blend, inspect, filter, lump, skip, smooth, chunk, average, approximate, cluster, aggregate, outline, summarize, itemize, review, dip into, flip through, browse, glance into, leaf through, skim, refine, enumerate, glean, synopsize, winnow the wheat from the chaff and separate the sheep from the goats.

We thrive in information–thick worlds because of our marvelous and everyday capacity to select, edit, single out, structure, highlight, group, pair, merge, harmonize, synthesize, focus, organize, condense, reduce, boil down, choose, categorize, catalog, classify, list, abstract, scan, look into, idealize, isolate, discriminate, distinguish, screen, pigeonhole, pick over, sort, integrate, blend, inspect, filter, lump, skip, smooth, chunk, average, approximate, cluster, aggregate, outline, summarize, itemize, review, dip into, flip through, browse, glance into, leaf through, skim, refine, enumerate, glean, synopsize, winnow the wheat from the chaff and separate the sheep from the goats.

We thrive in information–thick worlds because of our marvelous and everyday capacity to select, edit, single out, structure, highlight, group, pair, merge, harmonize, synthesize, focus, organize, condense, reduce, boil down, choose, categorize, catalog, classify, list, abstract, scan, look into, idealize, isolate, discriminate, distinguish, screen, pigeonhole, pick over, sort, integrate, blend, inspect, filter, lump, skip, smooth, chunk, average, approximate, cluster, aggregate, outline, summarize, itemize, review, dip into, flip through, browse, glance into, leaf through, skim, refine, enumerate, glean, synopsize, winnow the wheat from the chaff and separate the sheep from the goats.

discard, finite shape, page 2



1

We thrive in information–thick worlds because of our marvelous and everyday capacity to select, edit, single out, structure, highlight, group, pair, merge, harmonize, synthesize, focus, organize, condense, reduce, boil down, choose, categorize, catalog, classify, list, abstract, scan, look into, idealize, isolate, discriminate, distinguish, screen, pigeonhole, pick over, sort, integrate, blend, inspect, filter, lump, skip, smooth, chunk, average, approximate, cluster, aggregate, outline, summarize, itemize, review, dip into, flip through, browse, glance into, leaf through, skim, refine, enumerate, glean, synopsize, winnow the wheat from the chaff and separate the sheep from the goats.

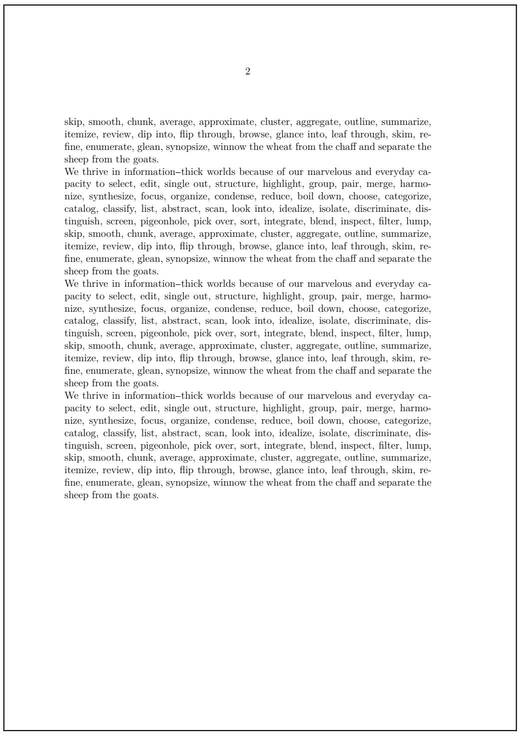
We thrive in information–thick worlds because of our marvelous and everyday capacity to select, edit, single out, structure, highlight, group, pair, merge, harmonize, synthesize, focus, organize, condense, reduce, boil down, choose, categorize, catalog, classify, list, abstract, scan, look into, idealize, isolate, discriminate, distinguish, screen, pigeonhole, pick over, sort, integrate, blend, inspect, filter, lump, skip, smooth, chunk, average, approximate, cluster, aggregate, outline, summarize, itemize, review, dip into, flip through, browse, glance into, leaf through, skim, refine, enumerate, glean, synopsize, winnow the wheat from the chaff and separate the sheep from the goats.

We thrive in information–thick worlds because of our marvelous and everyday capacity to select, edit, single out, structure, highlight, group, pair, merge, harmonize, synthesize, focus, organize, condense, reduce, boil down, choose, categorize, catalog, classify, list, abstract, scan, look into, idealize, isolate, discriminate, distinguish, screen, pigeonhole, pick over, sort, integrate, blend, inspect, filter, lump, skip, smooth, chunk, average, approximate, cluster, aggregate, outline, summarize, itemize, review, dip into, flip through, browse, glance into, leaf through, skim, refine, enumerate, glean, synopsize, winnow the wheat from the chaff and separate the sheep from the goats.

We thrive in information–thick worlds because of our marvelous and everyday capacity to select, edit, single out, structure, highlight, group, pair, merge, harmonize, synthesize, focus, organize, condense, reduce, boil down, choose, categorize, catalog, classify, list, abstract, scan, look into, idealize, isolate, discriminate, distinguish, screen, pigeonhole, pick over, sort, integrate, blend, inspect, filter, lump, skip, smooth, chunk, average, approximate, cluster, aggregate, outline, summarize, itemize, review, dip into, flip through, browse, glance into, leaf through, skim, refine, enumerate, glean, synopsize, winnow the wheat from the chaff and separate the sheep from the goats.

We thrive in information–thick worlds because of our marvelous and everyday capacity to select, edit, single out, structure, highlight, group, pair, merge, harmonize, synthesize, focus, organize, condense, reduce, boil down, choose, categorize, catalog, classify, list, abstract, scan, look into, idealize, isolate, discriminate, distinguish, screen, pigeonhole, pick over, sort, integrate, blend, inspect, filter, lump,

discard, repeat in shape, page 1



2

skip, smooth, chunk, average, approximate, cluster, aggregate, outline, summarize, itemize, review, dip into, flip through, browse, glance into, leaf through, skim, refine, enumerate, glean, synopsize, winnow the wheat from the chaff and separate the sheep from the goats.

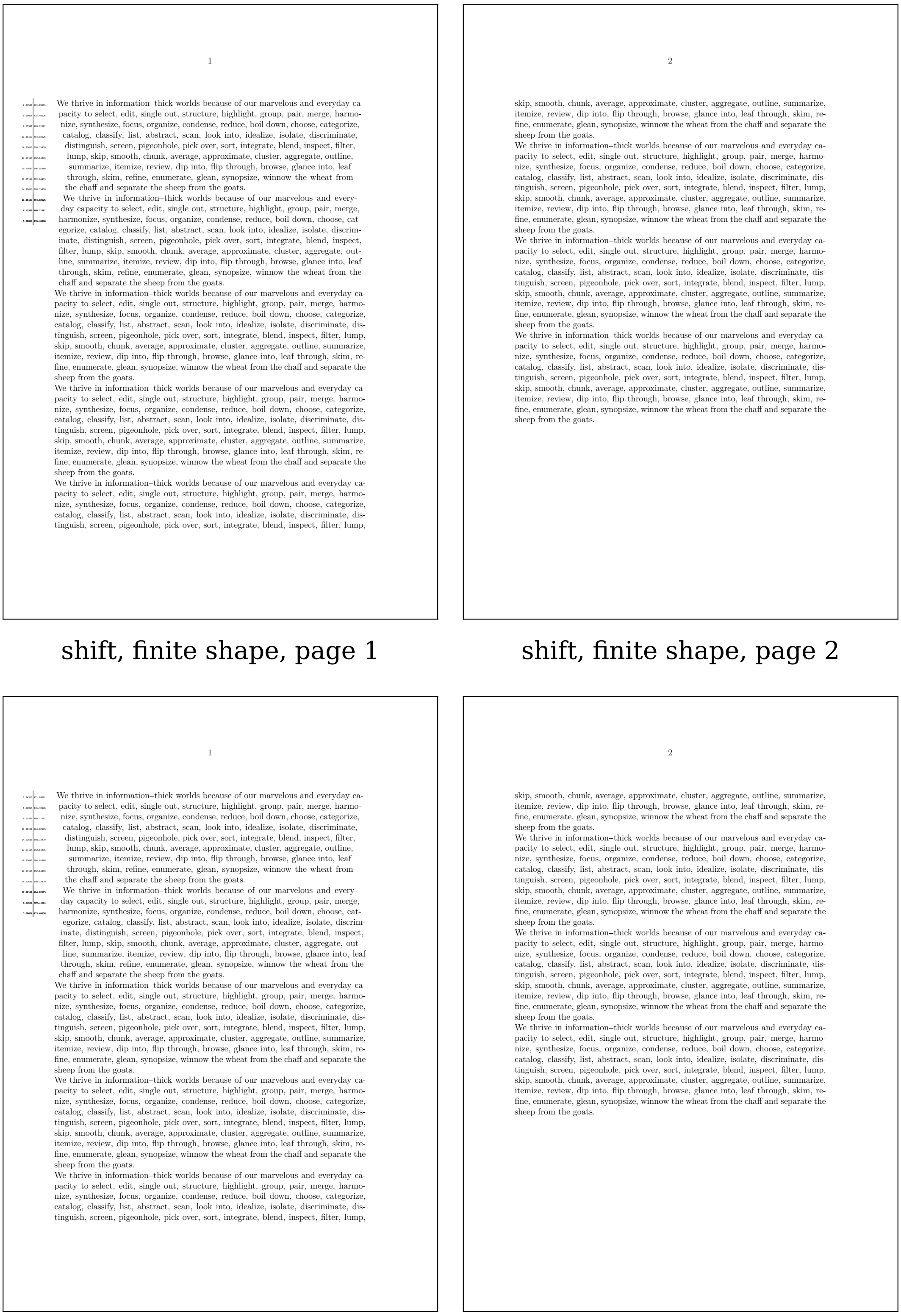
We thrive in information–thick worlds because of our marvelous and everyday capacity to select, edit, single out, structure, highlight, group, pair, merge, harmonize, synthesize, focus, organize, condense, reduce, boil down, choose, categorize, catalog, classify, list, abstract, scan, look into, idealize, isolate, discriminate, distinguish, screen, pigeonhole, pick over, sort, integrate, blend, inspect, filter, lump, skip, smooth, chunk, average, approximate, cluster, aggregate, outline, summarize, itemize, review, dip into, flip through, browse, glance into, leaf through, skim, refine, enumerate, glean, synopsize, winnow the wheat from the chaff and separate the sheep from the goats.

We thrive in information–thick worlds because of our marvelous and everyday capacity to select, edit, single out, structure, highlight, group, pair, merge, harmonize, synthesize, focus, organize, condense, reduce, boil down, choose, categorize, catalog, classify, list, abstract, scan, look into, idealize, isolate, discriminate, distinguish, screen, pigeonhole, pick over, sort, integrate, blend, inspect, filter, lump, skip, smooth, chunk, average, approximate, cluster, aggregate, outline, summarize, itemize, review, dip into, flip through, browse, glance into, leaf through, skim, refine, enumerate, glean, synopsize, winnow the wheat from the chaff and separate the sheep from the goats.

We thrive in information–thick worlds because of our marvelous and everyday capacity to select, edit, single out, structure, highlight, group, pair, merge, harmonize, synthesize, focus, organize, condense, reduce, boil down, choose, categorize, catalog, classify, list, abstract, scan, look into, idealize, isolate, discriminate, distinguish, screen, pigeonhole, pick over, sort, integrate, blend, inspect, filter, lump, skip, smooth, chunk, average, approximate, cluster, aggregate, outline, summarize, itemize, review, dip into, flip through, browse, glance into, leaf through, skim, refine, enumerate, glean, synopsize, winnow the wheat from the chaff and separate the sheep from the goats.

discard, repeat in shape, page 2

Figure 1 Discarded shaping



shift, finite shape, page 1

shift, finite shape, page 2

shift, repeat in shape, page 1

shift, repeat in shape, page 2

Figure 2 Shifted shaping

```
\stopshapedparagraph
```

Finally, in figure 3 we use:

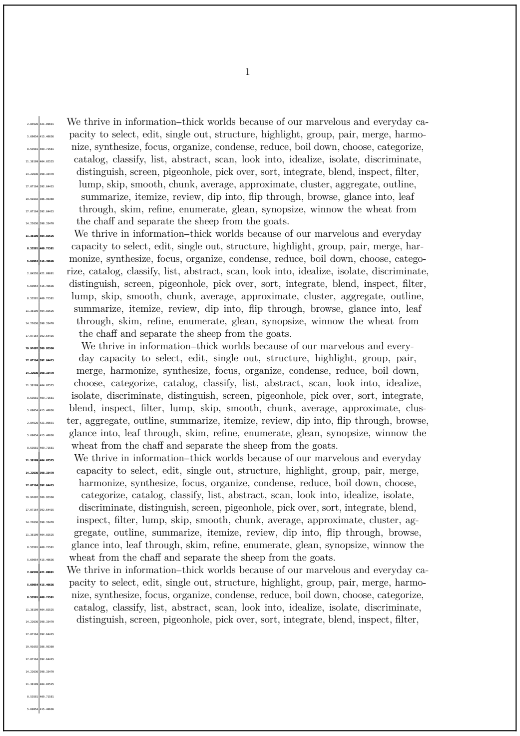
```
\startshapedparagraph[list=test,method=cycle]
    \dorecurse{8}{\showparagraphshape\samplefile{tufte} \par}
\stopshapedparagraph

\startshapedparagraph[list=test-repeat]
    \dorecurse{8}{\showparagraphshape\samplefile{tufte} \par}
\stopshapedparagraph
```

These examples are probably too small to see the details but you can run them yourself or zoom in on the details. In the margin we show the values used. Here is a simple example of (non) poetry. There are other environments that can be used instead but this makes a good example anyway.

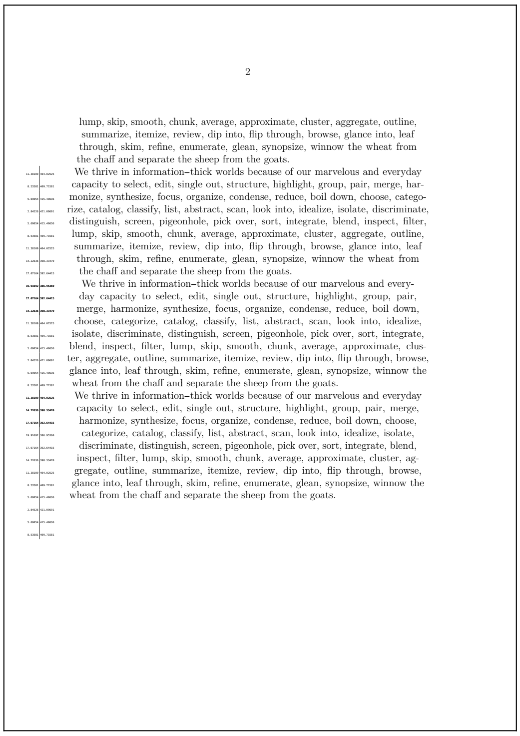
```
\startparagraphshape[test]
    left 0em right 0em
    left 1em right 0em
    repeat
\stopparagraphshape

\startshapedparagraph[list=test,method=cycle]
    verse line 1.1\crlf verse line 2.1\crlf
    verse line 3.1\crlf verse line 4.1\par
    verse line 1.2\crlf verse line 2.2\crlf
    verse line 3.2\crlf verse line 4.2\crlf
    verse line 5.2\crlf verse line 6.2\par
\stopshapedparagraph
```

verse line 1.1
  verse line 2.1
verse line 3.1
  verse line 4.1

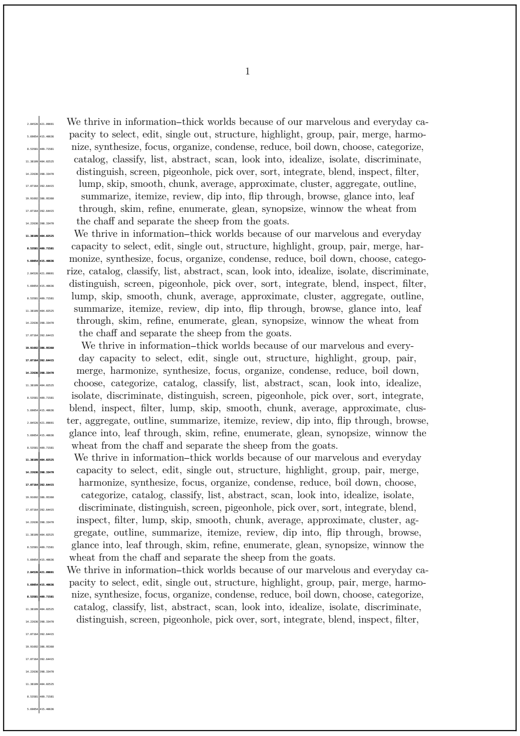
verse line 1.2
  verse line 2.2
verse line 3.2
  verse line 4.2
verse line 5.2
  verse line 6.2



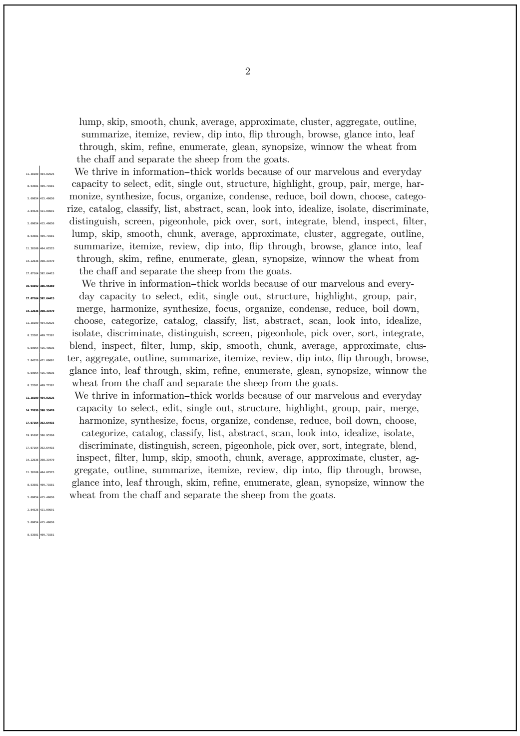
cycle, finite shape, page 1



cycle, finite shape, page 2



cycle, repeat in shape, page 1



cycle, repeat in shape, page 2

Figure 3 Cycled shaping

todo: move the new (still in `meta-imp-txt.mkxl`*) code into the core and integrate it in* `\startshapedparagraph` *as method* `mp` *in which case the list is a list of graphics.*

```
\startshapedparagraph[list={test 1,test 2,test 3,test 4},method=mp]
    .....
\stopshapedparagraph
```

So methods then become kind of plugins.

A mechanism like this is often never completely automatic in the sense that you need to keep an eye on the results. Depending on user demands more features can be added. With weird shapes you might want to set up the alignment to be `tolerant` and have some `stretch`.

5 Modes

todo: some of the side effects of so called modes

6 Normalization

todo: users don't need to bother about this but it might be interesting anyway