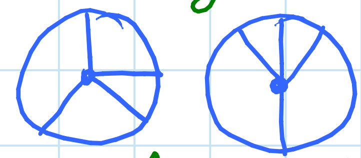
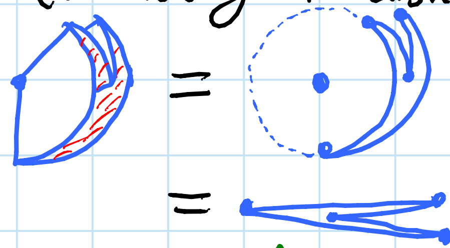
| 6.849 | Lecture 3 | Sept. 15, 2010 |
|---|---|---|

<u>Single-vertex crease pattern</u> (without loss of generality)
= disk of paper,
creases emanate from center



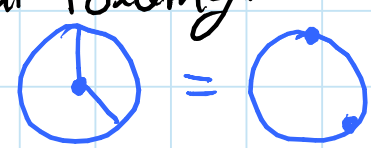
<u>Idea:</u> capture local foldability around a vertex
= circular sequence of angles $\theta_1, \theta_2, \ldots, \theta_n$
- normally, $\theta_1 + \theta_2 + \cdots + \theta_n = 360°$
- allow other sums, especially $\leq 360°$ (convex cone)
  in particular for induction

<u>Flat folding</u> = folding of 1D circle (boundary of disk) on the circle
= folding of 1D circle onto line

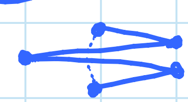


(assuming convex cone & at least one fold
$\Rightarrow$ can't reach all the way around circle)

<u>Differences</u> from 1D (segment) flat folding:
- not all crease patterns flat foldable:



- alternating M/V can fail:



- equilateral $\not\Rightarrow$ all mountain-valley patterns possible
  e.g. all valleys

- mingling $\not\Rightarrow$ existence of crimp:
  could have $\cdots$ ( ] ( ] ( ] $\cdots$ circularly

## Characterization of flat-foldable single-vertex crease pat.:

[Kawasaki 1989; Justin 1989; Hull 1994]

$\theta_1, \theta_2, \dots, \theta_n$ is flat-foldable convex cone

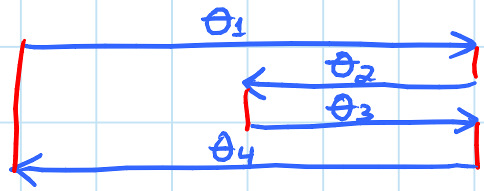
$\iff \theta_1 + \theta_3 + \dots + \theta_{n-1} = \theta_2 + \theta_4 + \dots + \theta_n$ (& $n$ even)

$= 180°$ for flat paper

**Proof:**

($\Rightarrow$) – angles $\theta_i$ measure travel on circle/line

– creases switch direction of travel




$\Rightarrow$ $n$ must be even (cycle of switches)

& total motion $= \pm(\theta_1 - \theta_2 + \theta_3 - \theta_4 + \dots + \theta_{n-1} - \theta_n)$

– total motion $= 0$ to end where we started

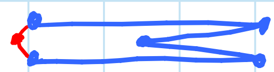
(assuming convex cone & at least one fold — else $\equiv 0 \pmod{360°}$)

$\Rightarrow$ alternating sum of angles $= 0$

($\Leftarrow$) – cut at an "extreme" crease (e.g., leftmost)

$\Rightarrow$ 1D segment crease pattern

– fold flat e.g. accordion



– two ends corresponding to cut crease are aligned because total motion $= 0$ & can join because extreme $\square$

**Nonconvex cone:** $\theta_1 - \theta_2 + \dots = 0$ or $\pm 360°$ [Demaine & O'Rourke 2007]

# Flat-foldable single-vertex mountain-valley patterns

Count: # mountains − # valleys = ±2 in convex cone [Maekawa; Justin 1986]

Proof: measure total turn angle = 180° − interior angle
(>0 for convex, <0 for reflex vertices)

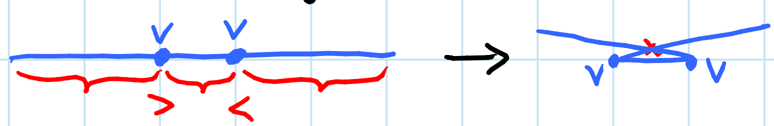
- mountain turns +180°, valley turns −180°
- small turn caused by circle, but cancels out assuming convex cone ⇒ can't reach around
- no crossing ⇒ total turn angle = ±360°

⇒ 180°·# mountains − 180°·# valleys = ±360°

⇒ # mountains − # valleys = ±2. □

Nonconvex cones: if $\theta_1 - \theta_2 + \cdots = \pm 360°$, #M − #V = 0

Generic case: strict local minimum angle is surrounded by one mountain & one valley




[Kawasaki 1989; Justin 1984]

⇒ can immediately crimp any such angle
- preserves flat foldability as before for 1D segments:

Remaining case: equal angles

# Characterization of flat-foldable single-vertex mountain-valley pattern:

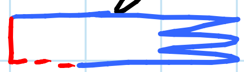
[Hull 2001 & 2003; Demaine & O'Rourke 2007]

Local counts: Among k equal angles surrounded by strictly larger angles (e.g. globally smallest angle),

$$\#\text{mountains} - \#\text{valleys} = \begin{cases} 0 & \text{if } k \text{ is odd} \\ \pm 1 & \text{if } k \text{ is even} \end{cases}$$

Proof: build cone from k equal angles & larger angles

- if k even then extend one larger angle to match the other



- if k odd then add new angle of $\Sigma$ larger angles − equal angle

$\Rightarrow$ flat folding of cone with same M-V assign. & $\theta_1 - \theta_2 + \cdots + \theta_{n-1} + \theta_n = 0$

- Maekawa's Theorem $\Rightarrow$ # mountains − # valleys = ±2 (Cone might be nonconvex but still $\theta_1 - \theta_2 + \cdots = 0$)
- if k even then one new crease $\Rightarrow$ 180° turn + new crease $\Rightarrow$ #M − #V = ±1
- if k odd then two new creases (same dir.) $\Rightarrow$ 0° turn + 2 new creases $\Rightarrow$ #M − #V = 0. □

$\Rightarrow$ there is at least one crimp among these creases

- applies unless all angles are equal
  - $\Rightarrow$ crimp exists by #mountains − #valleys = ±2 (or 0)
  - unless just 2 creases $\Rightarrow$ same direction (or opposite direction if $\theta_1 - \theta_2 = \pm 360°$)
- linear-time algorithm (maintain crimps)

Combinatorics of single-vertex mountain-valley patterns:

linear-time algorithm to count [Hull 2003]

- smallest in generic case $\Rightarrow \underset{\text{choices per crimp}}{2}^{\overbrace{n/2}^{\text{number of crimps}}}$
- largest in equal-angle case $\Rightarrow \underset{\#M-\#V=+2 \text{ or } -2}{2}\underbrace{\binom{n}{n/2-1}}_{\text{Ms \& Vs}}$

OPEN: polynomial-time characterization for $k$-vertex crease pattern, $k$ small?

- $n^{f(k)}$? $f(k)\, n^{O(1)}$?

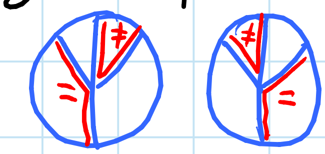
(we will see in Lecture 5 that the general problem is NP-hard)

## Local foldability: [Bern & Hayes 1996]

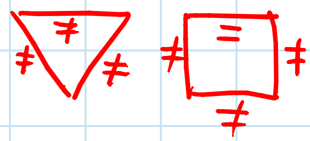
linear-time algorithm finds a consistent mountain-valley assignment (if possible) such that each vertex locally folds flat

Proof: all possible mountain-valley assignments of a single vertex generated by crimps

- crimped pair forced unequal
- final pair forced equal




- cycles can have parity issue:




- pairing unique in generic case
- if equal angle next to crimped angle then can interchange order of crimps
- merge if interchange decreases # paths/cycles of =/≠ constraints
- merges only fix parity problems (lemma)
- cook until done □

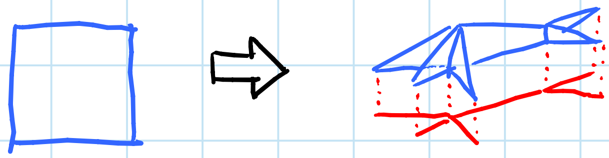
PROJECT: implement local foldability test converting crease pattern → M-V pattern

OPEN: minimum number of added creases to make given crease pattern [locally] flat foldable

- with or without mountain-valley assignment
- always possible via disk-packing fold & cut

Tree method: [Lang 1994–2003; Lang, Demaine, Demaine 2004–]

algorithm to find folding of smallest square into "uniaxial" origami base whose projection is a desired metric tree




But: – optimization is difficult: exponential time, NP-hard [Demaine, Fekete, Lang – OSME 2010]
↳ Lecture 5

but good heuristics

– non-self-intersection is only conjectured (we're working on it)

More next lecture!