1 Important information

Reading: de Berg, M., O. Cheong, M. van Kreveld, M. Overmars, *Computational Geometry: Algorithms and Applications* (Springer, 2008). Pages 1-14.

EDGE (explain, demonstrate, guide, enable) is a training method that is taught in the Boy Scouts of America.

2 Explain

Convex hull: A subset $\mathscr{S}$ of a plane is called *convex* if and only if, for any pair of points $p, q \in \mathscr{S}$, the line segment, $\overline{pq}$ is completely contained in $\mathscr{S}$. The *convex hull* $\mathscr{CH}(\mathscr{S})$ of a set $\mathscr{S}$ is the smallest convex set that contains $\mathscr{S}$.

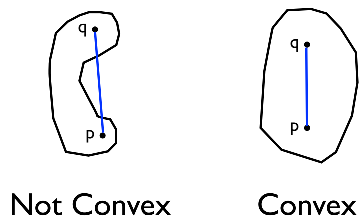

Figure 1: Cartoon depiction of a convex hull.

3 Demonstrate

3.1 Nails-in-Board demo

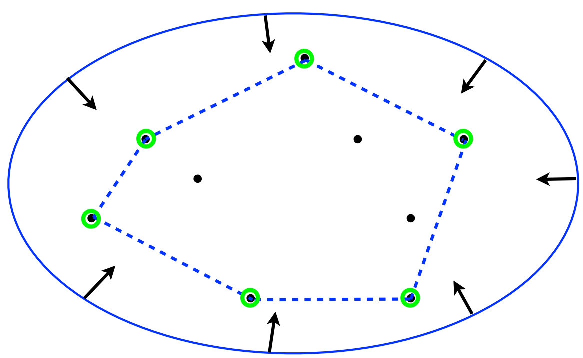

Figure 2: Cartoon depiction of a rubber band analogy.

4 Guide

4.1 Graham's scan $\mathscr{O}(n \log n)$

Algorithm CONVEXHULL(P)
Input: A set P of points in the plane
Output: A list containing the vertices of $\mathscr{CH}(\mathscr{P})$ in clockwise order

1. Sort the points by $x-$coordinate, resulting in a sequence $p_1, \ldots, p_n$.
2. Put the points p_1 and p_2 in a list $\mathscr{L}_{upper}$, with p_1 as the first point.
3. **for** $ii \leftarrow 3$ **to** n
4. **do** Append p_i to $\mathscr{L}_{upper}$.
5. **while** $\mathscr{L}_{upper}$ contains more than 2 points **and** the last 3 points do not make a right turn.
6. **do** Delete the middle of the last three points from $\mathscr{L}_{upper}$.
7. Put the points p_n and p_{n-1} in a list $\mathscr{L}_{lower}$, with p_n as the first point.
8. **for** $ii \leftarrow n-2$ **downto** 1
9. **do** Append p_i to $\mathscr{L}_{lower}$.
10. **while** $\mathscr{L}_{lower}$ contains more than 2 points **and** the last 3 points do not make a right turn.
11. **do** Delete the middle of the last 3 points from $\mathscr{L}_{lower}$.
12. Remove the first and the last point from $\mathscr{L}_{lower}$ to avoid duplication of the points where the upper and lower hull meet.
13. Append $\mathscr{L}_{lower}$ to $\mathscr{L}_{upper}$, and call the resulting list $\mathscr{L}$.
14. **return** $\mathscr{L}$.

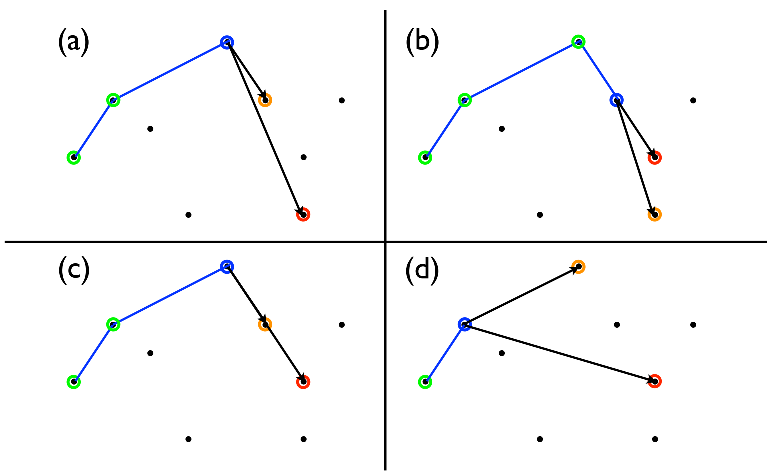

Figure 3: Cartoon depiction of a Graham's scan. (a) The sequence is satisfied with the addition of the red point. (b) The sequence is denied with the addition of the red point. (c) Removing the middle point (orange) from (b), the sequence is again denied, as the orange point and the red point are collinear with the blue point. (d) Removing the middle point from (c), the sequence is confirmed.

5 Enable

5.1 MATLAB function

The following MATLAB function uses Graham's scan.

```
%% Graham's scan

% 14. \textbf{return} L
function L = convex_hull(P)

% 1. Sort the points by x-coordinate, resulting in a sequence
% p1,...,pn.
Ps = sortrows(P,2);      % sort points by y-value
Ps = sortrows(Ps,1);     % sort points by x-value

% X. Remove duplicate points
for c = length(Ps):-1:2
    if isequal(Ps(c,:),Ps(c-1,:)), Ps(c,:) = []; end
end

% 2. Put the points p1 and p2 in a list L_upper, with
% p1 as the first point.
Lu = Ps(1:2,:);

% 3. \textbf{for} ii <- 3 \textbf{to} n
for ii = 3:length(Ps)
    % 4. \textbf{do} Append p_i to L_upper.
    Lu = [Lu;Ps(ii,:)];
    % 5.1 \textbf{while} L_upper contains more than 2 points
    while length(Lu) > 2
        p1 = [Lu(end-2,:),0];
        p2 = [Lu(end-1,:),0];
        p3 = [Lu(end,:),0];
        % 5.2 \textbf{while} the last 3 points in L_upper do
        % not make a right turn.
        c = cross(p3-p1,p2-p1);
        % 6. \textbf{do} Delete the middle of the last three points from
        % L_upper.
        if c(3) ≤ 0
            Lu(end-1,:) = [];
        else
            break
        end
    end
end

% 7. Put the points pn and p(n-1) in a list L_lower,
% with pn as the first point.
Ll = [Ps(end,:);Ps(end-1,:)];

% 8. \textbf{for} ii <- n-2 \textbf{downto} 1
for ii = 3:length(Ps)
    % 9. \textbf{do} Append p_i to L_lower.
    Ll = [Ll;Ps(end-ii+1,:)];    % add next point to set
    % 10.1 \textbf{while} L_lower contains more than 2 points
    while length(Ll) > 2
        p3 = [Ll(end,:),0];
        p2 = [Ll(end-1,:),0];
        p1 = [Ll(end-2,:),0];
        % 10.2 \textbf{while} the last 3 points in L_lower do
        % not make a right turn.
        c = cross(p3-p1,p2-p1);
        % 11. \textbf{do} Delete the middle of the last 3 points from
        % L_lower.
        if c(3) ≤ 0
            Ll(end-1,:) = [];
        else
            break
        end
    end
end

% 12. Remove the first and the last point from L_lower to
```

```
% avoid duplication of the points where the upper and lower hull meet.
Ll(1,:) = []; Ll(end,:) = [];

% 13. Append 𝓛lower to 𝓛upper, and call the
% resulting list 𝓛.
L = [Lu;Ll];
```

5.2 Practical uses

See the reading for a discussion on the usage of a convex hull in computer graphics, robotics, geographic information systems (GIS), CAD/CAM, and pattern recognition. Delaunay triangulation uses a convex hull to create discrete meshes of objects for use in direct numerical simulations. Image processing uses a convex hull for a variety of reasons. One particular example is the fitting of an ellipse to an object that is anticipated to be somewhat elliptical.

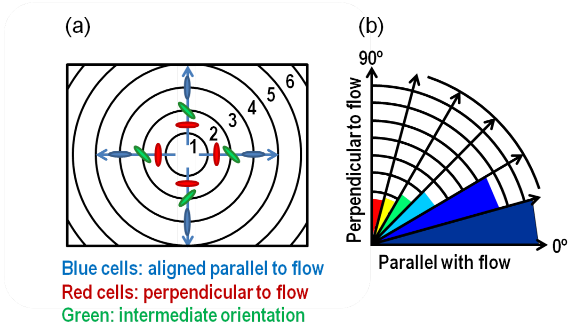

Figure 4: Cartoon of cell alignment and corresponding statistical rosette plot [1].

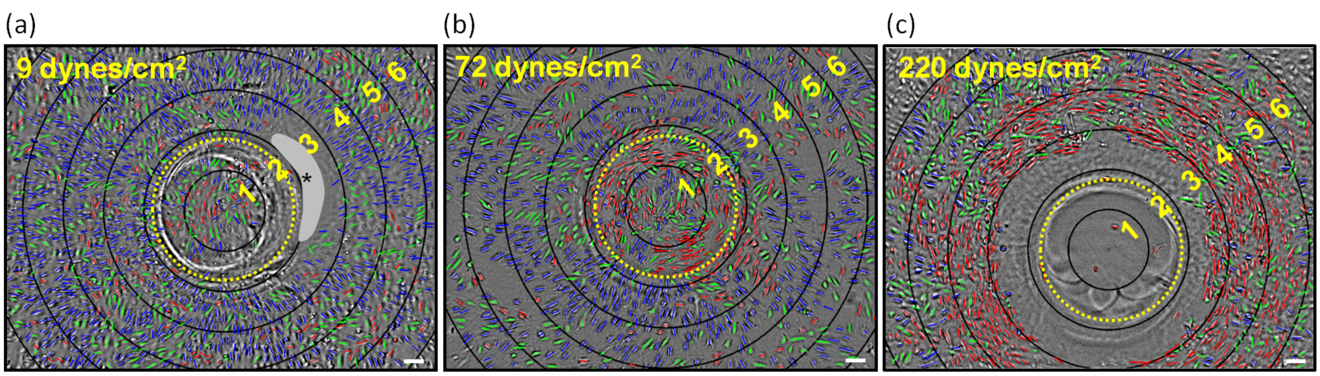

Figure 5: Endothelial cell response to impinging jet flows of varying magnitude after 21 hours [1].

Literature Cited

[1] M. Ostrowski, N. Huang, T. Walker, T. Verwijlen, C. Poplawski, A. Khoo, J. Cooke, G. Fuller, and A. Dunn. Microvascular Endothelial Cells Migrate Upstream and Align Against the Shear Stress Field Created by Impinging Flow. Biophysical Journal, 106(2):366 – 374, 2014.