Aero-Works

New Mexico
Supercomputing Challenge
Final Report
April 5, 2006

Team: 71
Navajo Preparatory School

Team Members:
Myron Benally
Lane Thomas

Teachers:
Mavis Yazzie

Project Mentor:

Executive Summary

In our project we wanted to find which one of the three car designs had a more aerodynamic design. Aerodynamics of a car is very important. With the shape of each car comes different variables; greater amount of drag and if a car has more drag the more work it has to do to move forward. If a car had a lot of drag and it was harder for it to cut through the air we would all be paying more to help power our vehicles to perform this difficult task—drag.

To compare the three different car designs we used two methods and a computer program and two different mathematical equations. The computer model would help us measure how far air particles have to travel to get around our car designs. Them our first equation gave us a factor number that said the higher the number the less aerodynamic the design was but the lower the number the aerodynamic the design is. The third equation compares each car design's drag.

` Through those two methods we were able to determine which of the three car designs was the most aerodynamic.

Introduction

Aerodynamic is and object designed with rounded edges so as to reduce wind drag and thereby increase fuel efficiency. Used especially of motor vehicles. Today, some cars are more aerodynamic than others. Some are big square boxes others are shaped like knives. Box shaped cars seem to have a greater amount of drag because it needs more power to move through the air. But a car shaped like a knife uses less power to makes its way through the air.

Methods

To begin our process we had to research aerodynamics. We found that the more the car has a stream line design the more efficient it makes it way through the air. We also found that if the car is more of a box, the more work it has to do.

We also found that coefficient of drag plays a major role in car designs. When each car is in motion it seems to be dragging a parachute behind it, the parachute being drag. If the car has a slick design it has a smaller parachute but the more box shape that bigger the parachute.

The second part of our model was to design three different car designs to compare. Then we had to test each design by running them down a ramp. We also had to find a mathematical model. We found two different math models. The first gave us a factor number. The higher the factor numbers the less aerodynamic. We also had to acquire a computer program. The computer program that we used help measure; how long does it take for air particle to move over a car design.

Project Description

The purpose of our project is to find a more aerodynamic design using three different designs. We used two math models, a computer program and actual modeling.

In our project we have three different cars named: a, b, and c. All of the designs are different. One of them being box shaped and the others looking like cars. These are the three designs that we compared.

Math Model

We used two different math models.

Factor number for each car design

CD- Coefficient of Drag

FA- Frontal Area

Car #1: A - CD (0.34) x FA (8.25 sq in) = **2.8050 Factor Number**

Car #2: B - CD (0.30) x FA (6.75 sq in) = **2.0250 Factor Number**

Car #3: C - CD (0.57) x FA (7.5 sq in) = **4.2750 Factor Number**

Stokes Law

Given this Eqautions:

6π : Constant Numbers

η : $1.84 * 10^{-5}$

a : Area of Sphere

v : Velocity

Area of Car A

= 60.84 inches ^{2}

Area of Car A = Area of Sphere

$60.84 = 4\pi r^2$

$r = (60.84 / 4\pi)^{½}$

$r = 6.913$

Area of Sphere = $4\pi r^2$

$= 4\pi (6.913)^2$

$= 600.54$

$F_A = 6\pi (1.84 * 10^{-5}) (600.54) v$

Area of Car B

= 60.028 inches ^{2}

Area of Car A = Area of Sphere

$60.028 = 4\pi r^2$

$r = (60.028 / 4\pi)^{½}$

$r = 6.866$

Area of Sphere = $4\pi r^2$

$= 4\pi (6.866)^2$

$= 592.40$

$F_A = 6\pi (1.84 * 10^{-5}) (592.40) v$

Area of Car C

= 81 inches ^{2}

Area of Car A = Area of Sphere

$81 = 4\pi r^2$

$r = (81 / 4\pi)^{½}$

$r = 7.976$

Area of Sphere = $4\pi r^2$

$$= 4 \pi (7.976)^2$$
$$= 799.43$$

Computer Model

The computer model we used was *Starlogo.* IN the program we drew three different cars. In the model we had noe side where the air particles could leave the screen and one where they started. In the computer model the air particles move across the design and measure the number that escaped over time.

Results

Through the different testings we found that Car B had the overall best aerodynamic design.

Conclusion

In our intractable model car B had the best time going down the ramp. It average time on the first day was 23.8 seconds and on the second day it had a time of 25.06In our factor number equation car B. also had the lowest number which was 2.0250. Then in the Drag comparison model car B. also was the best overall. And in our computer program 85.33 air particles escaped in two minutes which was the best overall.

Acknowledgment

When we started this project we thought we would never finish it. But with the help of several people it finally was done. We would like to thank Mavis Yazzie, Bill Hall, Amanda Martin, and Betty Strong. These are the one that made this project come through, thank you.

Star Logo Code

Code for Computer

StarLogo, a programmable modeling environment designed to help you model and explore the workings of decentralized systems, such as bird flocks, traffic jams, and market economies.

Turtle Procedures

```
turtles-own [energy panic-level stomped]

to placeUnique
   if egress >= 0 and (count-turtles-here = 1) [stop]
   setxy wall-random-x wall-random-y
   placeUnique
end

to wall-random-x
  output (random wallwidth) + wallminx
end

to wall-random-y
  output (random wallheight) + wallminy
end

to exit
   set escaped escaped + 1
   die
end

;; moves turtles towards the safe area.
;; if it runs into another turtle it tries to move sideways
to move-to-safety
   ;; try to move downhill if not blocked
   let [:dir pick [90 -90]]                          ;; direction to turn
   rt pick [0 90 180 270]                            ;; initial direction
   repeat 4 [
     if (((egress-towards 0 1) >= 0) and
         ((egress-towards 0 1) < egress) ; and
             ;((count-turtles-towards 0 1) = 0)
          )
            [
       fd panic-level * 1 / egress
       if count-turtles-here > 1 [set stomped stomped + 1]
       if stomped > death-stomps [die]
       ifelse egress = 0 [ exit ] [ stop ]
     ]
     rt :dir
   ]
   ;; try to move sideways if not blocked
   repeat 4 [
     if (((egress-towards 0 1) >= 0) and
             ((count-turtles-towards 0 1) = 0)) [
       fd 1
```

```
      stop
    ]
    rt :dir
  ]
end
```

Observers Procedures

```
patches-own [egress]
globals [escaped timesteps
         level
         randomscan
         wallminx wallmaxx wallminy wallmaxy wallwidth wallheight]

to no-gradient
  ask-patches [if pc >= 100 [setpc black]]
  ask-patches [set egress 1]
  ask-patches-with [pc = gray] [ setegress -2 ]
end

to setup-plan
  ;; clear all
  ca
  ;; initialize the plan
  set randomscan false
  ;; draw the plan
  if plan = 1 [ draw-refuge-1 draw-walls-1 ]
  if plan = 2 [ draw-refuge-1 draw-walls-2 ]
  if plan = 3 [ draw-refuge-1 draw-walls-3 ]
  if plan = 4 [ draw-refuge-1 draw-walls-4 ]
  if plan = 5 [ draw-refuge-1 draw-walls-5 ]
  if plan = 6 [ draw-refuge-1 draw-walls-6 ]
  if plan = 7 [ draw-refuge-1 draw-walls-7 ]
  if plan = 8 [ draw-refuge-2 draw-walls-8 ]
  if plan = 9 [ draw-refuge-2 draw-walls-9 ]
  if plan = 10 [ draw-refuge-2 draw-walls-10 ]
  if plan = 11 [ draw-refuge-2 draw-walls-11 ]
  if plan = 12 [ draw-refuge-2 draw-walls-12 ]
end

to setup-field
  ;; initialize escape field
  ask-patches [setegress -1]
  ask-patches-with [pc = green] [setegress 0 ]
  ask-patches-with [pc = gray] [ setegress -2 ]
  ;; build the escape field
  ;; start from safety zone and grow
```

```
  ;; loop terminate by counting cells at new level
  setlevel 0
  loop [
    ask-patches-with [egress = level] [
      sprout [
        repeat 4 [
          rt 90
          if (egress-towards 0 1) = -1 [
            setegress-towards 0 1 (level + 1)
          ]
        ]
        die
      ]
    ]
    wait-until [count-turtles = 0]
    set level (level + 1)
    ask-patches-with [egress > 0] [ scale-pc blue egress 1 level ]
    if (count-patches-with [egress = level]) = 0 [ stop ]
  ]
end

to setup-turtles
  ;; erase the turtles
  ct
  ;; initialize the monitor variables
  setescaped 0
  settimesteps 0
  ;; setup the turtles
  create-and-do number-of-people [
    setxy wall-random-x wall-random-y
    ;; place turtle uniquely!!
    ;; handle the case that a turtle lands on a wall
    placeUnique
    setenergy random 100
    set stomped 0
    set panic-level energy / 20
    ;; set turtle energy to a random value within a range
    scale-color red energy 0 100
    ;; show the turtles energy
    seth pick [0 90 180 270]
  ]
end

to go
  if count-turtles = 0 [ stopbutton2 ]
  set timesteps timesteps + 1
```

```
  ifelse randomscan [
    repeat count-turtles [
      ask-turtle pick list-of-turtles [
        move-to-safety
        scale-color red energy 0 100
      ]
    ]
  ] [
    ask-turtles [
      move-to-safety
      scale-color red energy 0 100
    ]
  ]
end

to draw-refuge-1
  ;; wall around entire screen
  create-turtles-and-do 1 [
    setxy -61 -56
    setcolor green
    seth 0
    pendown
    repeat 2 [ fd 112 rt 90 fd 122 rt 90 ]
    die
  ]
  wait-until [count-turtles = 0]
end

to draw-refuge-2
  ;; wall down right edge
  create-turtles-and-do 1 [
    setxy (screen-half-width - 1) screen-half-height
    setcolor green
    seth 180
    pendown
    fd screen-height
    setxy screen-half-width screen-half-height
    setcolor gray
    seth 180
    pendown
    fd screen-height
    die
  ]
  wait-until [count-turtles = 0]
end
```

```
to draw-walls-1
  ;; box with one patch doorway
  create-turtles-and-do 1 [
    setxy -19 1
    setcolor gray
    seth 0
    pendown
    fd 18 rt 90
    repeat 3 [ fd 38 rt 90 ]
    fd 18
    die
  ]
  draw-walls-finish-1
end

to draw-walls-finish-1
  wait-until [count-turtles = 0]
  set wallminx -19
  set wallmaxx 38 - 19
  set wallminy 19 - 38
  set wallmaxy 19
  set wallwidth 38
  set wallheight 38
end

to draw-walls-2
  ;; box with two patch doorway
  create-turtles-and-do 1 [
    setxy -19 2
    setcolor gray
    seth 0
    pendown
    fd 17 rt 90
    repeat 3 [ fd 38 rt 90 ]
    fd 18
    die
  ]
  draw-walls-finish-1
end

to draw-walls-3
  ;; box with 3 patch doorway
  create-turtles-and-do 1 [
    setxy -19 2
    setcolor gray
    seth 0
```

```
    pendown
    fd 17 rt 90
    repeat 3 [ fd 38 rt 90 ]
    fd 17
    die
  ]
  draw-walls-finish-1
end

to draw-walls-4
  ;; box with 3 patch doorway on 4 walls
  create-turtles-and-do 1 [
    setxy -19 2
    setcolor gray
    seth 0
    repeat 4 [
      pd fd 17 rt 90 fd 17 pu fd 4
    ]
    die
  ]
  draw-walls-finish-1
end

to draw-walls-5
  ;; big box
  create-turtles-and-do 1 [
    setxy -55 1
    setcolor gray
    seth 0
    repeat 2 [
      pd fd 49 rt 90 fd 110 rt 90 fd 49 pu fd 3
    ]
    die
  ]
  draw-walls-finish-5
end

to draw-walls-finish-5
  wait-until [count-turtles = 0]
  set wallminx -55
  set wallmaxx 55
  set wallminy -50
  set wallmaxy 50
  set wallwidth 111
  set wallheight 101
end
```

```
to draw-walls-6
  ;; big box with rooms inside
  create-turtles-and-do 1 [
    setxy -55 3
    setcolor gray
    seth 0
    repeat 2 [
      pd fd 47 rt 90 fd 110 rt 90 fd 47 pu fd 7
    ]
    repeat 2 [
      fd 5 pd rt 90
      fd 16 pu fd 5 pd fd 16 lt 90 fd 42 bk 42 rt 90
      fd 16 pu fd 5 pd fd 15 lt 90 fd 42 bk 42 rt 90
      fd 16 pu fd 5 pd fd 16 lt 90 fd 42 bk 42 rt 90
      pu rt 90 fd 12
    ]
    die
  ]
  draw-walls-finish-5
end

to draw-walls-7
  ;; big box with labyrinthe inside
  create-turtles-and-do 1 [
    setxy -55 -50
    setcolor gray
    seth 0
    let [:v 100]
    let [:h 110]
    pd
    repeat 16 [
      fd :v rt 90 fd :h rt 90
      set :v (:v - 5)
      set :h (:h - 5)
    ]
    die
  ]
  draw-walls-finish-5
end

to draw-walls-finish-8
  wait-until [count-turtles = 0]
  set wallminx (0 - screen-width / 2)
  set wallmaxx wallminx + screen-width / 8
  set wallminy (0 - screen-height / 2)
```

```
  set wallmaxy screen-height / 2
  set wallwidth screen-width / 8
  set wallheight screen-height
end

to draw-walls-8
    create-turtles-and-do 1 [
     ;;setxy (0 - screen-width / 4)  (0 - screen-height / 2 + 1)
     setcolor gray
     seth 0
     pd
     fd 2.5 * screen-width / 12
     rt 90 fd 2 * screen-width / 4
     rt 90 fd 2.5 * screen-width / 12
     rt 90 fd 2 * screen-width / 4
    die
   ]
   draw-walls-finish-8
end

to draw-walls-finish-9
  wait-until [count-turtles = 0]
  set wallminx (0 - screen-width / 2)
  set wallmaxx wallminx + screen-width / 8
  set wallminy (0 - screen-height / 2)
  set wallmaxy screen-height / 2
  set wallwidth screen-width / 8
  set wallheight screen-height
end

to draw-walls-9
   create-turtles-and-do 1 [
     setxy (0 - screen-width / 4)  (0 - screen-height / 2 + 1)
     setcolor gray
     seth 0
       fd 20
     pd
     fd 2.5 * screen-width / 12
     rt 90 fd 6 * screen-width / 12
     rt 90 fd 2.5 * screen-width / 12
     rt 90 fd 6 * screen-width / 12
    die
   ]
   draw-walls-finish-9
end
```

```
to draw-walls-finish-10
  wait-until [count-turtles = 0]
  set wallminx (0 - screen-width / 2)
  set wallmaxx wallminx + screen-width / 8
  set wallminy (0 - screen-height / 2)
  set wallmaxy screen-height / 2
  set wallwidth screen-width / 8
  set wallheight screen-height
end

to draw-walls-10
   create-turtles-and-do 1 [
     setxy (0 - screen-width / 4)  (0 - screen-height / 2 + 1)
     setcolor gray
     seth 0
       fd 20
     pd
     fd 2.5 * screen-width / 12
     rt 90 fd 6 * screen-width / 12
     rt 90 fd 2.5 * screen-width / 12
     rt 90 fd 6 * screen-width / 12
    die
   ]
   draw-walls-finish-10
end

to draw-walls-finish-11
  wait-until [count-turtles = 0]
  set wallminx (0 - screen-width / 2)
  set wallmaxx wallminx + screen-width / 8
  set wallminy (0 - screen-height / 2)
  set wallmaxy screen-height / 2
  set wallwidth screen-width / 8
  set wallheight screen-height
end

to draw-walls-11
   create-turtles-and-do 1 [
     setxy (0 - screen-width / 4)  (0 - screen-height / 2 + 1)
     setcolor gray
     seth 0
       fd 20
     pd
     rt 8 fd .85 * screen-width / 12
     rt 44 fd .4375 * screen-width / 12
     rt 27 fd 3.375 * screen-width / 12
```

```
    lt 47 fd .8125 * screen-width / 12
    rt 39 fd .5 * screen-width / 12
    rt 33 fd .4375 * screen-width / 12
    rt 52 fd .75 * screen-width / 12
    lt 19 fd .75 * screen-width / 12
    rt 56 fd 1.3 * screen-width / 12
    rt 77 fd 5.55 * screen-width / 12
   die
  ]
  draw-walls-finish-11
end

to draw-walls-finish-12
  wait-until [count-turtles = 0]
  set wallminx (0 - screen-width / 2)
  set wallmaxx wallminx + screen-width / 8
  set wallminy (0 - screen-height / 2)
  set wallmaxy screen-height / 2
  set wallwidth screen-width / 8
  set wallheight screen-height
end

to draw-walls-12
  create-turtles-and-do 1 [
    setxy (0 - screen-width / 4)  (0 - screen-height / 2 + 1)
    setcolor gray
    seth 0
      fd 20
    pd
    rt 25 fd 1.25 * screen-width / 12
    rt 53 fd 1.6875 * screen-width / 12
    lt 24 fd 1.5 * screen-width / 12
    rt 55 fd 1.5625 * screen-width / 12
    rt 33 fd 1.125 * screen-width / 12
    rt 45 fd 1 * screen-width / 12
    rt 83 fd 5.45 * screen-width / 12
   die
  ]
  draw-walls-finish-12
end
```

Works Cited

Skinner, Ray. Mechanics. Blaisdell Publishing Company, Waltham, Massachusetts © 1969.

http://www.nas.nasa.gov/About/Education/Racecar/glossary.html

http://www.madsci.org/posts/archives/nov99/942685064.Eg.r.html

http://www.encyclopedia.com/html/a1/aerodyna.asp

http://www.grc.nasa.gov/WWW/K-12/airplane/presar.html

http://www.aeromech.usyd.edu.au/aero/fprops/cvanalysis/node38.html

http://www.grc.nasa.gov/WWW/K-12/airplane/drageq.html