# L30: Discussion

## Numerical Differentiation; Numerical Integration; Project

Lucas A. J. Bastien

E7 Spring 2017, University of California at Berkeley

April 7, 2017

Version: release

# Announcements

**Lab 11 is due on April 14 at 12 pm (noon)**

**Lab 11 is significantly shorter than most previous labs**

Use the opportunity to:

- Get a lot of points on lab 11!
- Work on your project!

**Today:**

- Numerical differentiation and integration:
  Examples and practice questions
- Project discussion, tips, and recommendations

**Next week:**

- Ordinary differential equations (Chapter 19)

# Getting help on lab assignments and on the project

**Be proactive about getting help! We are here to help you!**

- **Lab sections:** arrive in your Monday or Tuesday lab section, having looked at the lab, and with questions ready

# Getting help on lab assignments and on the project

**Be proactive about getting help! We are here to help you!**

- **Lab sections:** arrive in your Monday or Tuesday lab section, having looked at the lab, and with questions ready
- **My office hours:** I can help you understand the theoretical concepts and the methods used; I can help you work some of the math out on paper

# Getting help on lab assignments and on the project

**Be proactive about getting help! We are here to help you!**

- **Lab sections:** arrive in your Monday or Tuesday lab section, having looked at the lab, and with questions ready
- **My office hours:** I can help you understand the theoretical concepts and the methods used; I can help you work some of the math out on paper
- **bCourses Discussions**:
  - I answer your questions in the corresponding Frequently Asked Questions (FAQ) Page
  - The more questions you ask, the more answers you are likely to get

# Getting help on lab assignments and on the project

**Be proactive about getting help! We are here to help you!**

- **Lab sections:** arrive in your Monday or Tuesday lab section, having looked at the lab, and with questions ready
- **My office hours:** I can help you understand the theoretical concepts and the methods used; I can help you work some of the math out on paper
- **bCourses Discussions**:
  - I answer your questions in the corresponding Frequently Asked Questions (FAQ) Page
  - The more questions you ask, the more answers you are likely to get
- **bCourses Frequently Asked Questions (FAQ) Pages:** spend a minute to consult them once or twice a week, you might find useful information there

# Getting help on lab assignments and on the project

**Be proactive about getting help! We are here to help you!**

- **Lab sections:** arrive in your Monday or Tuesday lab section, having looked at the lab, and with questions ready
- **My office hours:** I can help you understand the theoretical concepts and the methods used; I can help you work some of the math out on paper
- **bCourses Discussions**:
  - I answer your questions in the corresponding Frequently Asked Questions (FAQ) Page
  - The more questions you ask, the more answers you are likely to get
- **bCourses Frequently Asked Questions (FAQ) Pages:** spend a minute to consult them once or twice a week, you might find useful information there
- **Other bCourses Pages**

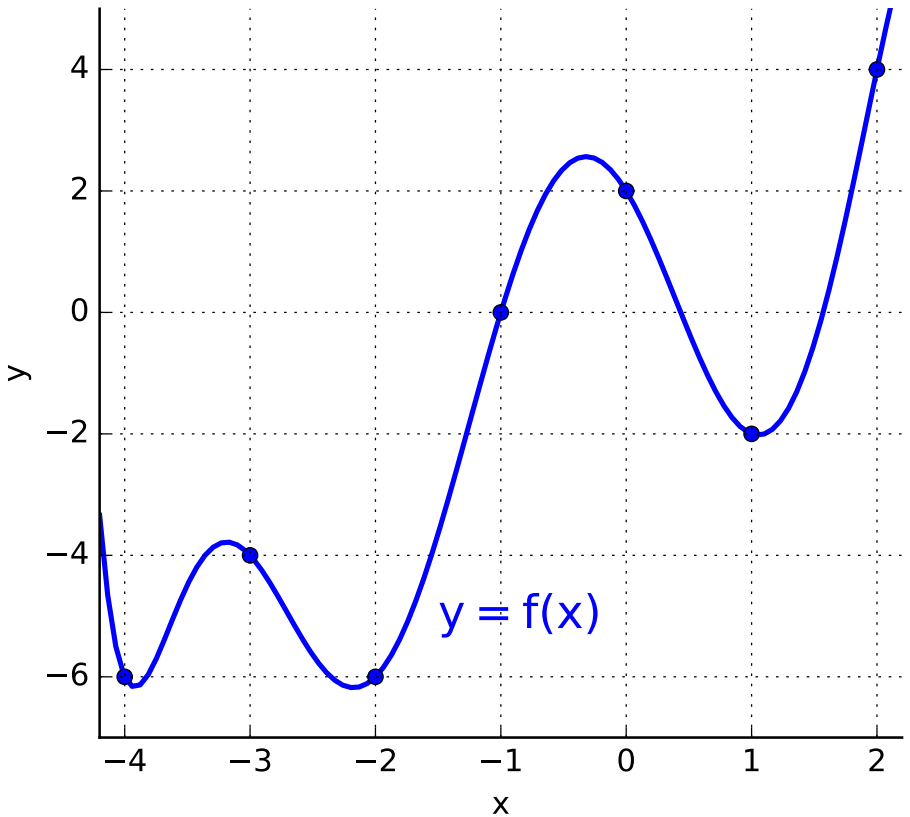
# Getting help on lab assignments and on the project

**Be proactive about getting help! We are here to help you!**

- **Lab sections:** arrive in your Monday or Tuesday lab section, having looked at the lab, and with questions ready
- **My office hours:** I can help you understand the theoretical concepts and the methods used; I can help you work some of the math out on paper
- **bCourses Discussions**:
  - I answer your questions in the corresponding Frequently Asked Questions (FAQ) Page
  - The more questions you ask, the more answers you are likely to get
- **bCourses Frequently Asked Questions (FAQ) Pages:** spend a minute to consult them once or twice a week, you might find useful information there
- **Other bCourses Pages**
- **Friday morning lab drop-in hours**

# Numerical differentiation: practice question

What is the value of $f'(-1)$ as estimated by the backward finite-difference approximation with spacing $\Delta x = 2$?




**(A)** 1

**(B)** 2

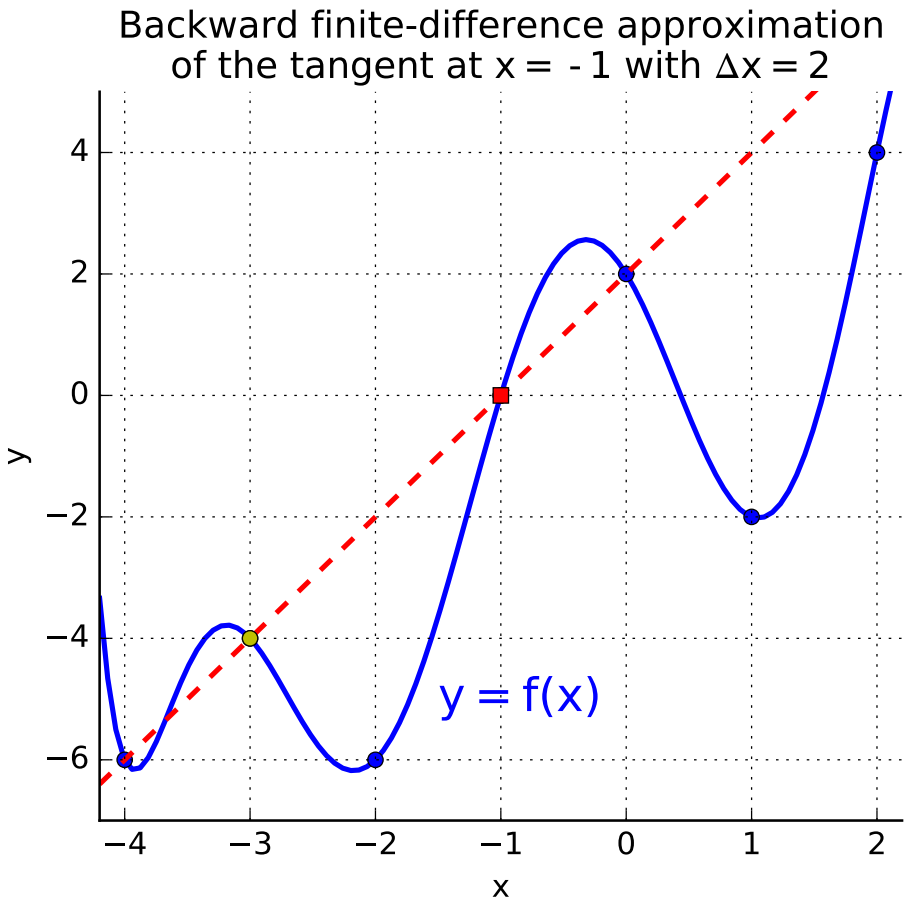
**(C)** 6

**(D)** -1

**(E)** 3

# Numerical differentiation: practice question

What is the value of $f'(-1)$ as estimated by the backward finite-difference approximation with spacing $\Delta x = 2$?




(A) 1

(B) 2

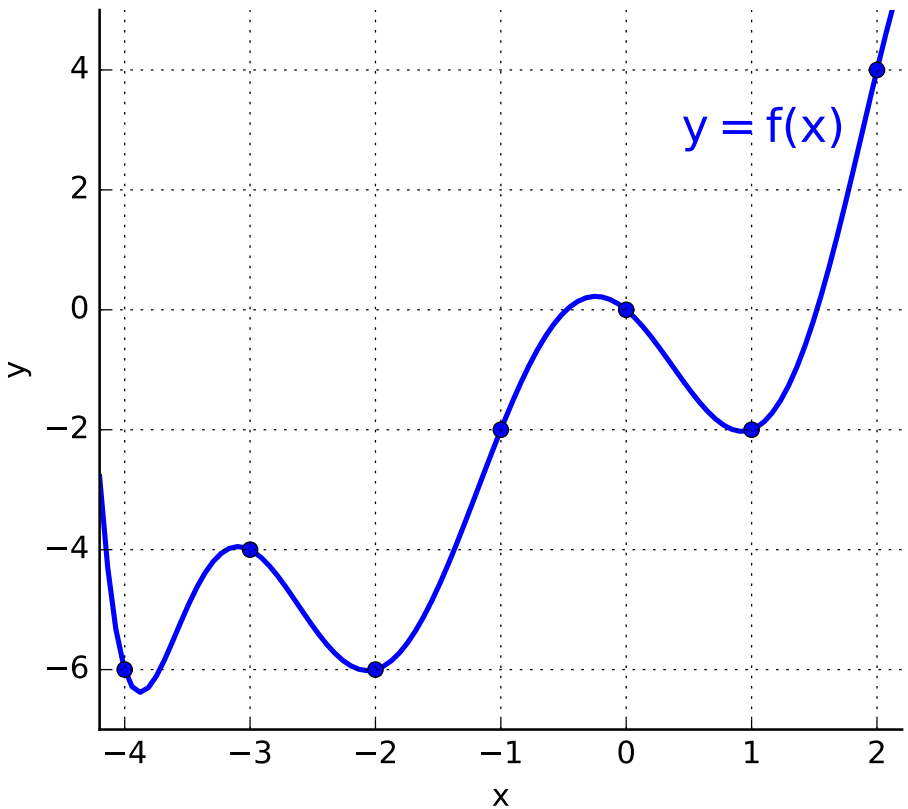
(C) 6

(D) -1

(E) 3

# Numerical integration: practice question

What is the value of $\int_{-4}^{0} f(x)dx$ as approximated by the midpoint formula with spacing (*i.e.* width of sub-intervals) $\Delta x = 2$?




**(A)** -12

**(B)** 12

**(C)** -10

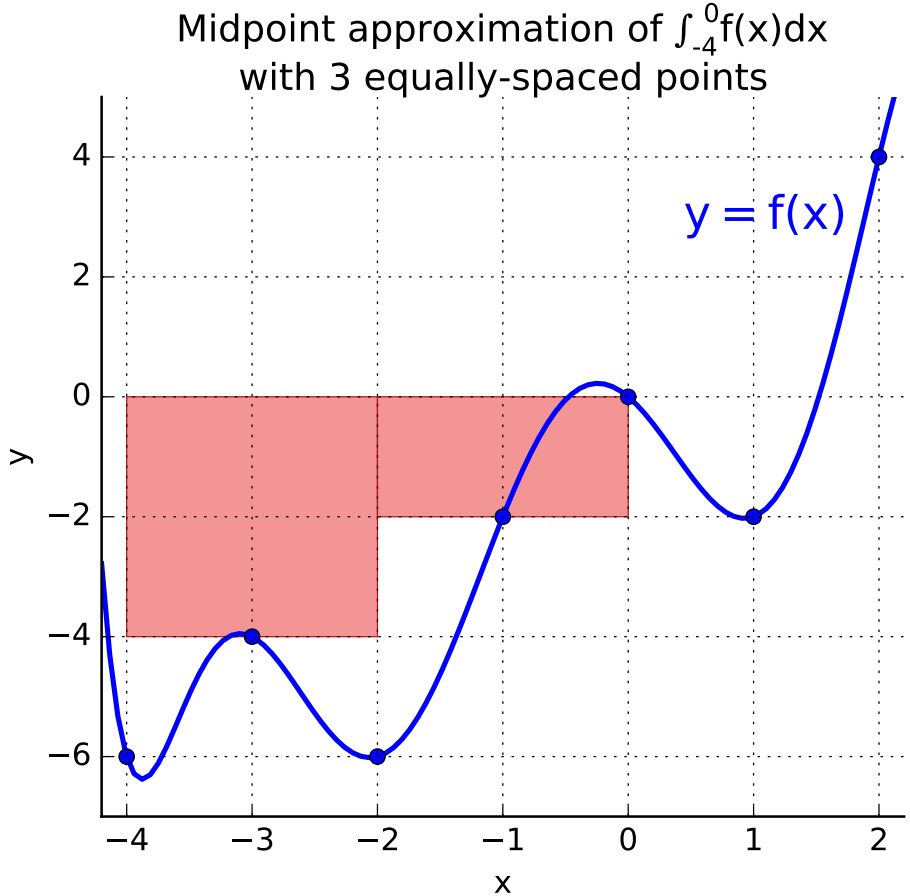
**(D)** 10

**(E)** -6

**(F)** 6

# Numerical integration: practice question

What is the value of $\int_{-4}^{0} f(x)dx$ as approximated by the midpoint formula with spacing (*i.e.* width of sub-intervals) $\Delta x = 2$?




**(A)** -12

**(B)** 12

**(C)** -10

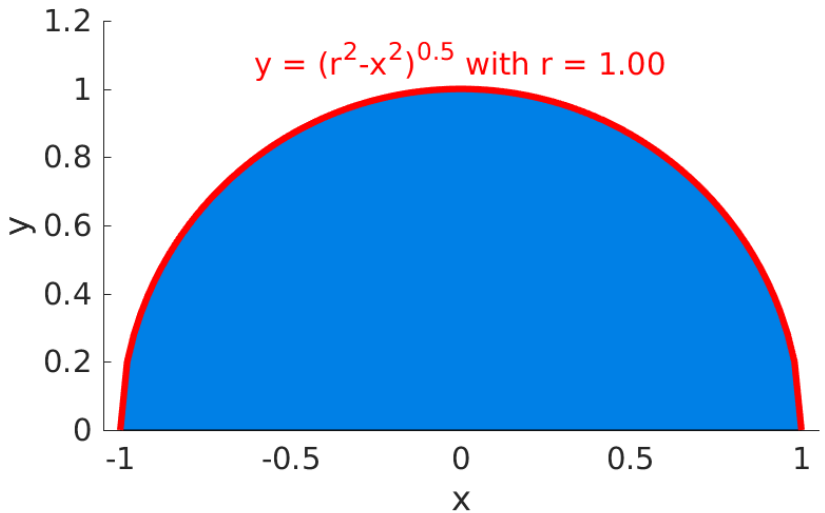
**(D)** 10

**(E)** -6

**(F)** 6

# Estimating $\pi$ using numerical integration

$$\int_{-r}^{r} \sqrt{r^2 - x^2}dx = \text{Area of half-circle of radius r} = \frac{\pi r^2}{2}$$

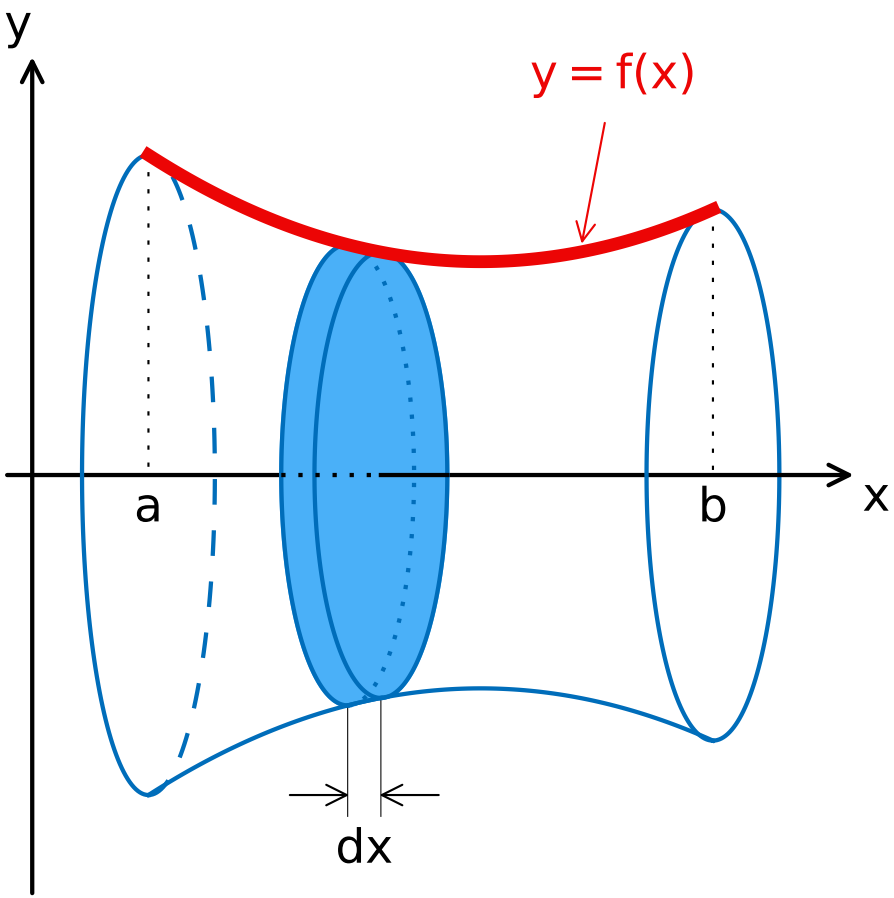



$$\pi = \frac{2}{r^2} \int_{-r}^{r} \sqrt{r^2 - x^2}dx$$

We can use numerical integration to estimate the value of the integral (see script: `estimate_pie_numerical_integration.m`)

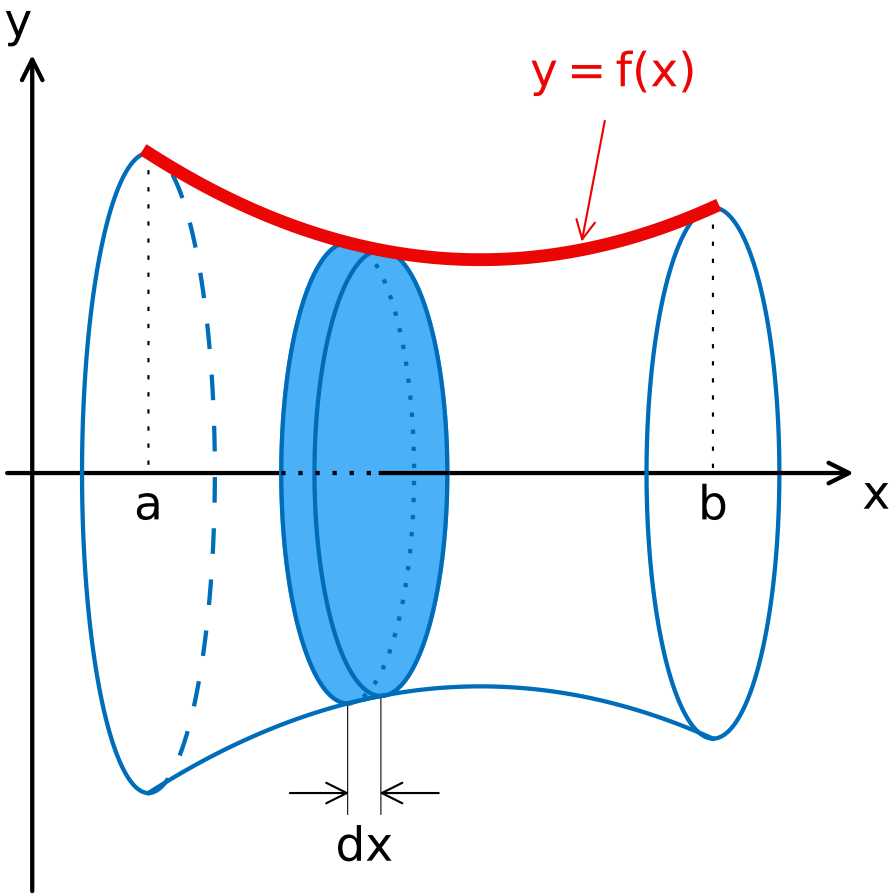
# Volume of solids of revolution using numerical integration

**Question 3 of lab 11:** using numerical integration, estimate the volume of solids of revolution given their "profile" *i.e.* the line of equation $y = f(x)$ that is being rotated to form the solid



# Volume of solids of revolution using numerical integration

**Question 3 of lab 11:** using numerical integration, estimate the volume of solids of revolution given their "profile" *i.e.* the line of equation $y = f(x)$ that is being rotated to form the solid

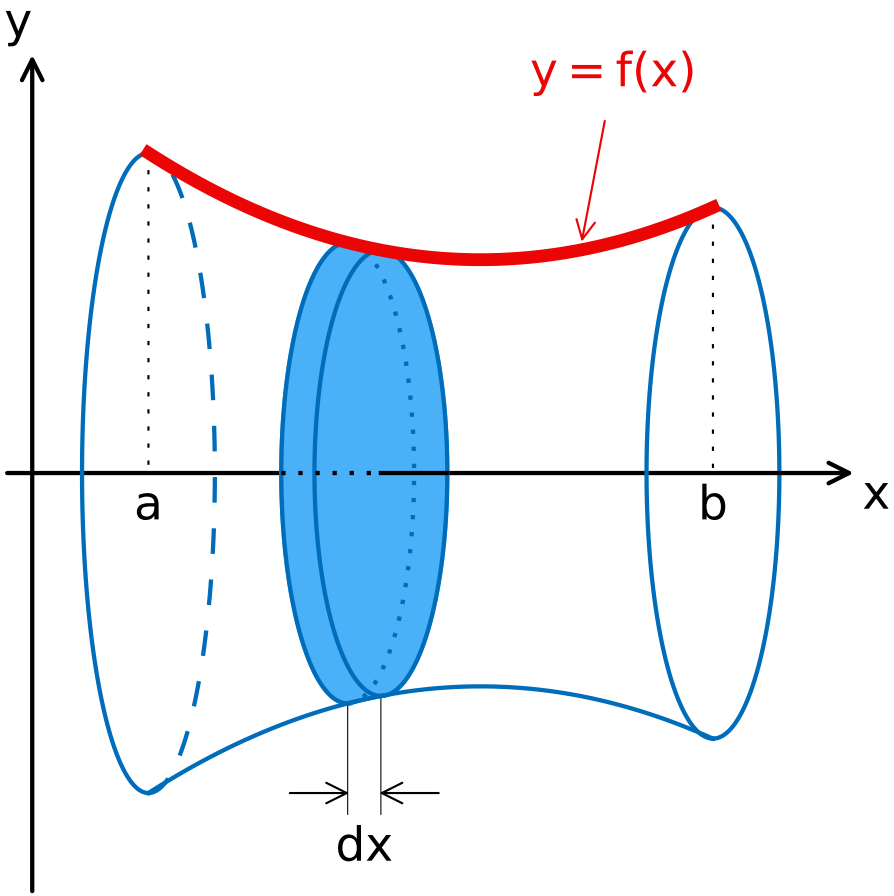



Volume of the slice of infinitesimally small width $dx$:

$$\pi f(x)^2 dx$$

# Volume of solids of revolution using numerical integration

**Question 3 of lab 11:** using numerical integration, estimate the volume of solids of revolution given their "profile" *i.e.* the line of equation $y = f(x)$ that is being rotated to form the solid




Volume of the slice of infinitesimally small width $dx$:

$$\pi f(x)^2 dx$$

Volume $V$ of the solid:

$$V = \sum \text{volume of each slice}$$
$$= \int_a^b \pi f(x)^2 dx$$

# Project: grading

# Project: grading

- **Beta test (15 points): due Friday April $14^{\text{th}}$ at 12 pm (noon)**
  - Your function will be graded on three maps. Each map will have only one scrap, no impassable areas, no slow-down areas, no ghosts, and wrap-around moves will not be necessary

# Project: grading

- **Beta test (15 points): due Friday April $14^{\text{th}}$ at 12 pm (noon)**
  - Your function will be graded on three maps. Each map will have only one scrap, no impassable areas, no slow-down areas, no ghosts, and wrap-around moves will not be necessary

- **Final code (70 points): due Friday April $28^{\text{th}}$ at 11:59 pm**
  - Graded on multiple maps, which vary in difficulty

# Project: grading

- **Beta test (15 points): due Friday April $14^{\text{th}}$ at 12 pm (noon)**
  - Your function will be graded on three maps. Each map will have only one scrap, no impassable areas, no slow-down areas, no ghosts, and wrap-around moves will not be necessary
- **Final code (70 points): due Friday April $28^{\text{th}}$ at 11:59 pm**
  - Graded on multiple maps, which vary in difficulty
- **Final write-up (15 points): due Friday April $28^{\text{th}}$ at 11:59 pm**
  - 1 to 2 pages (no more), PDF format
  - Describe the approach and algorithms that you used in your code
  - Describe in which situations your code works well, and in which situations it does not
  - You can talk about algorithms you tried or wanted to try, but did not end up using in your final code
  - Use sections, headings, and figures as appropriate to support the text

# Project: tips and recommendations

**It is most likely better to have a function that performs simple tasks well but which cannot handle complex tasks, rather than to have a function that tries to do everything but does not succeed at any of it**

# Project: tips and recommendations

**It is most likely better to have a function that performs simple tasks well but which cannot handle complex tasks, rather than to have a function that tries to do everything but does not succeed at any of it**

Main advice:

- Start with basic functionality, don't try to address every aspect of the problem at once

# Project: tips and recommendations

**It is most likely better to have a function that performs simple tasks well but which cannot handle complex tasks, rather than to have a function that tries to do everything but does not succeed at any of it**

Main advice:

- Start with basic functionality, don't try to address every aspect of the problem at once. For example:
  1. Make your spaceship pick up the nearest scrap on maps with no obstacles and no slow-down areas, and without using wrap-around moves

# Project: tips and recommendations

**It is most likely better to have a function that performs simple tasks well but which cannot handle complex tasks, rather than to have a function that tries to do everything but does not succeed at any of it**

Main advice:

- Start with basic functionality, don't try to address every aspect of the problem at once. For example:
  1. Make your spaceship pick up the nearest scrap on maps with no obstacles and no slow-down areas, and without using wrap-around moves
  2. Start thinking about how to go around obstacles, detect which scraps are unreachable, etc.

# Project: tips and recommendations

**It is most likely better to have a function that performs simple tasks well but which cannot handle complex tasks, rather than to have a function that tries to do everything but does not succeed at any of it**

Main advice:

- Start with basic functionality, don't try to address every aspect of the problem at once. For example:
  1. Make your spaceship pick up the nearest scrap on maps with no obstacles and no slow-down areas, and without using wrap-around moves
  2. Start thinking about how to go around obstacles, detect which scraps are unreachable, etc.
  3. Start thinking about how to use wrap-around moves

# Project: tips and recommendations

**It is most likely better to have a function that performs simple tasks well but which cannot handle complex tasks, rather than to have a function that tries to do everything but does not succeed at any of it**

Main advice:

- Start with basic functionality, don't try to address every aspect of the problem at once. For example:
  1. Make your spaceship pick up the nearest scrap on maps with no obstacles and no slow-down areas, and without using wrap-around moves
  2. Start thinking about how to go around obstacles, detect which scraps are unreachable, etc.
  3. Start thinking about how to use wrap-around moves
  4. Start thinking about the order in which to pick up scraps

# Project: tips and recommendations

**It is most likely better to have a function that performs simple tasks well but which cannot handle complex tasks, rather than to have a function that tries to do everything but does not succeed at any of it**

Main advice:

- Start with basic functionality, don't try to address every aspect of the problem at once. For example:
  1. Make your spaceship pick up the nearest scrap on maps with no obstacles and no slow-down areas, and without using wrap-around moves
  2. Start thinking about how to go around obstacles, detect which scraps are unreachable, etc.
  3. Start thinking about how to use wrap-around moves
  4. Start thinking about the order in which to pick up scraps
- Divide your code in sub-functions

# Project: tips and recommendations

**It is most likely better to have a function that performs simple tasks well but which cannot handle complex tasks, rather than to have a function that tries to do everything but does not succeed at any of it**

Main advice:

- Start with basic functionality, don't try to address every aspect of the problem at once. For example:
  1. Make your spaceship pick up the nearest scrap on maps with no obstacles and no slow-down areas, and without using wrap-around moves
  2. Start thinking about how to go around obstacles, detect which scraps are unreachable, etc.
  3. Start thinking about how to use wrap-around moves
  4. Start thinking about the order in which to pick up scraps
- Divide your code in sub-functions
- Test your code at each step of the process

# Project: create your own maps

For example, see the script `create_my_map.m`, that was used to create the following map: