CS5540: Computational Techniques for Analyzing Clinical Data

Prof. Ramin Zabih (CS)

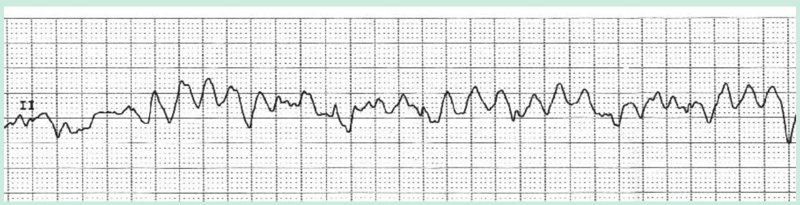
Prof. Ashish Raj (Radiology)

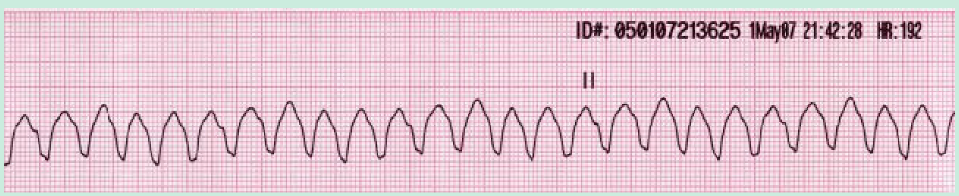
Administrivia

- We’re going to try to end class by 2:25
  - Like the registrar believes
- Sign up online!

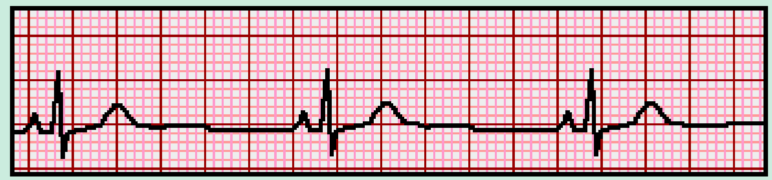
Preview of Project #1

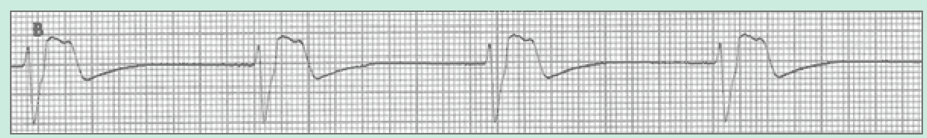
Shockable





Non shockable





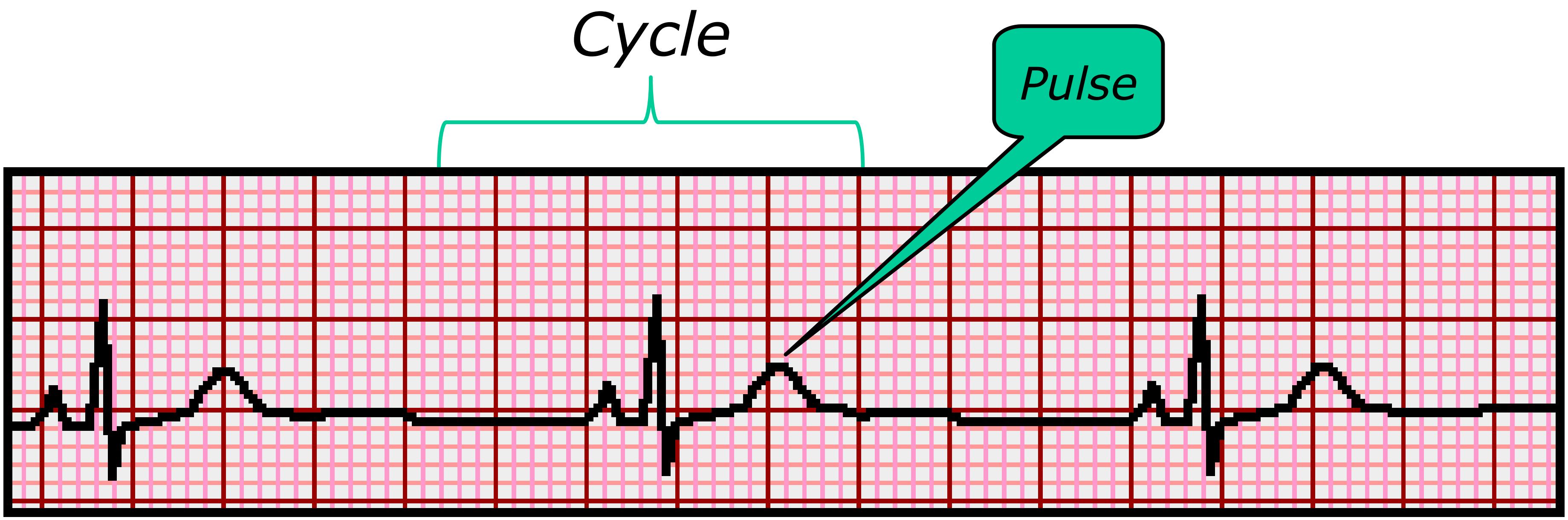
Algorithms

- Dynamic programming
- Graph algorithms, esp. min cut
- Fitting via least squares & its variants
- Gradient descent, conjugate gradient, PCG
- k-NN, SVM classification

Today's topics

- General methods for analyzing 1D data
  - Like an ECG
  - General (as opposed to model-based) techniques are usually more successful
- Today we will look at finding the cycles and the pulses

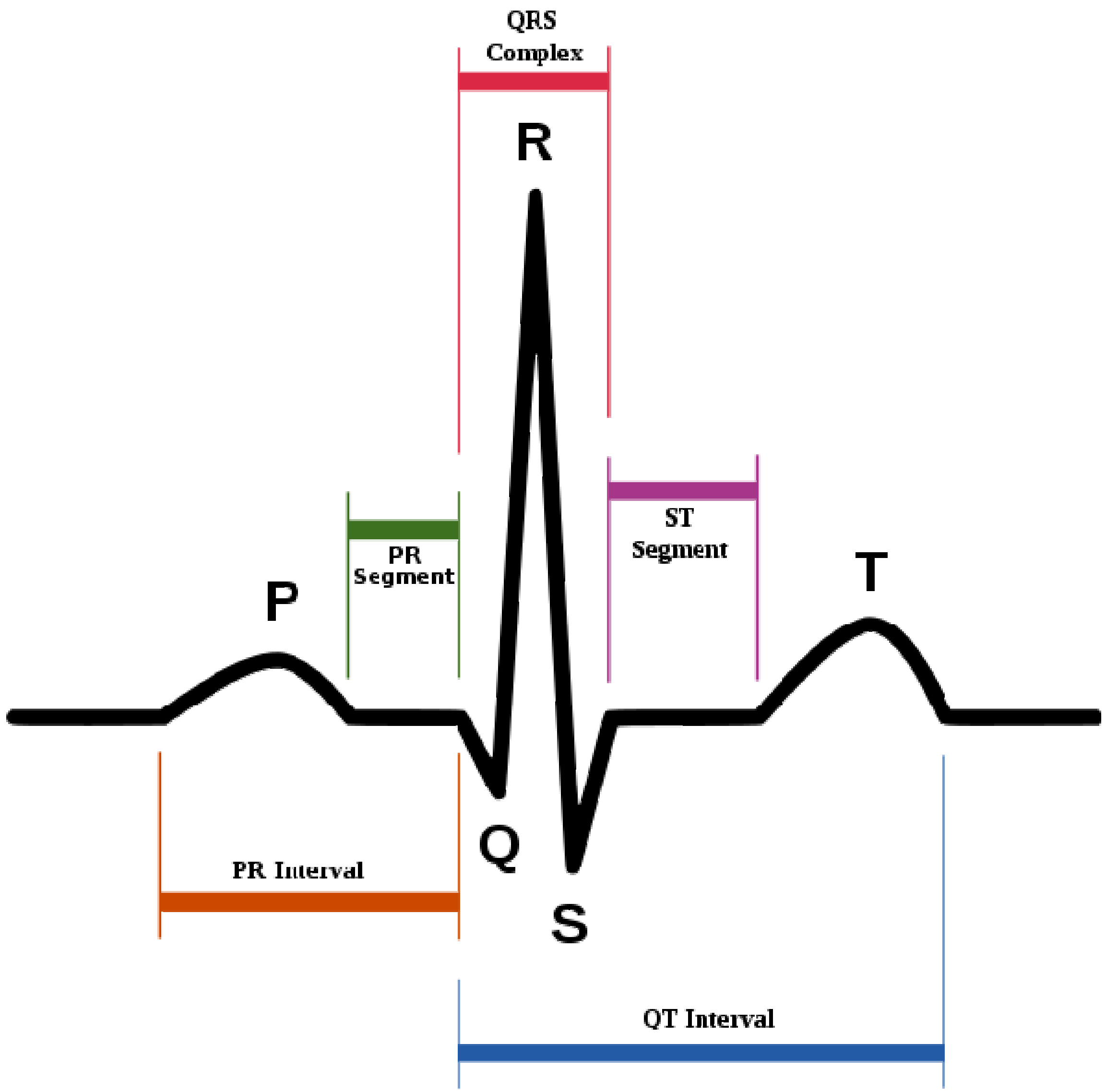
Cycles and pulses



www.uptodate.com

Note: cycle is standard terminology, but pulse is not. The ECG literature calls these "waves" but we want to be more general than just ECG.

Ideal pulses: P, Q, R, S, T

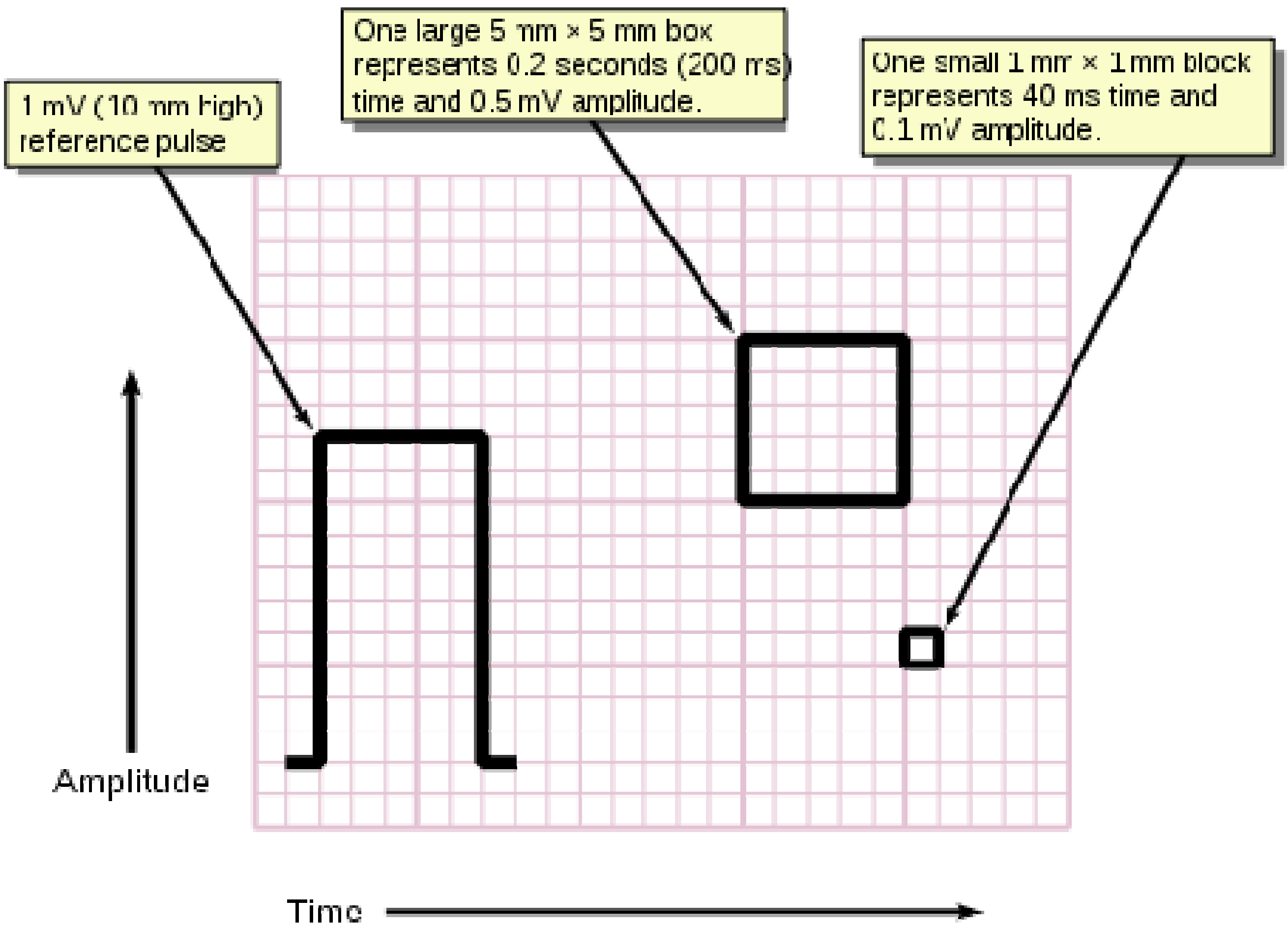


Why a general method??

Ideas?

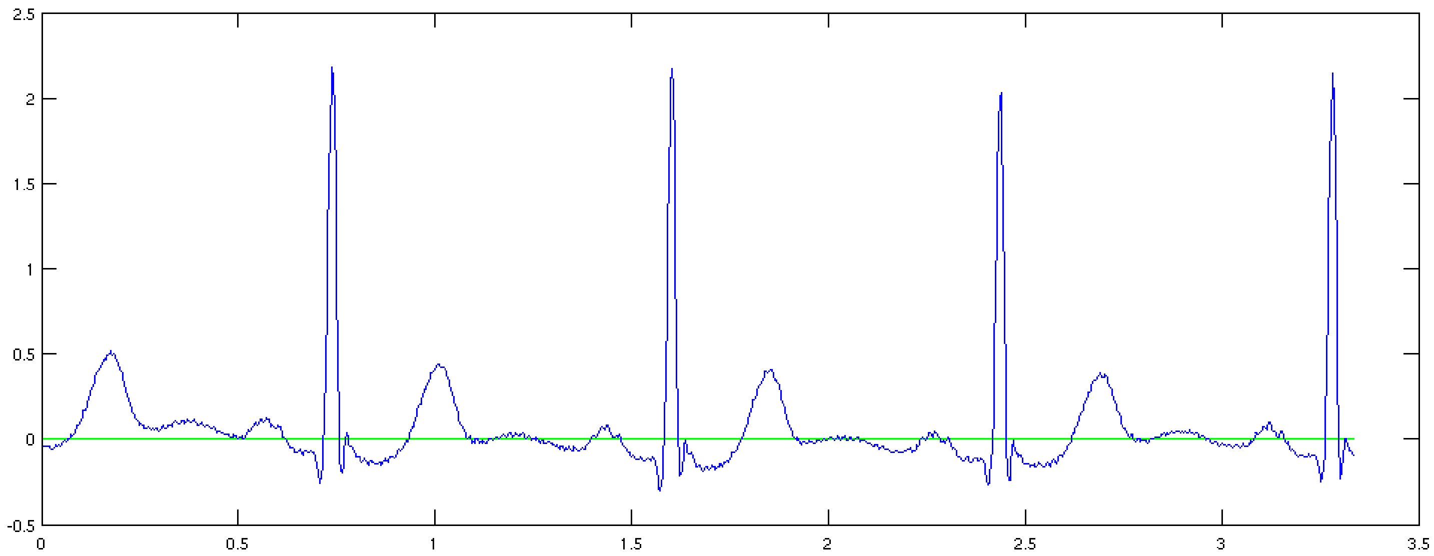
- Let’s try to find the places where the ECG goes above or below its baseline
- Almost all medical measurements are in **physical** units, which is unusual
  - In a picture, someone’s smile might be 100 pixels wide
  - How big is a pixel??

ECG graph paper

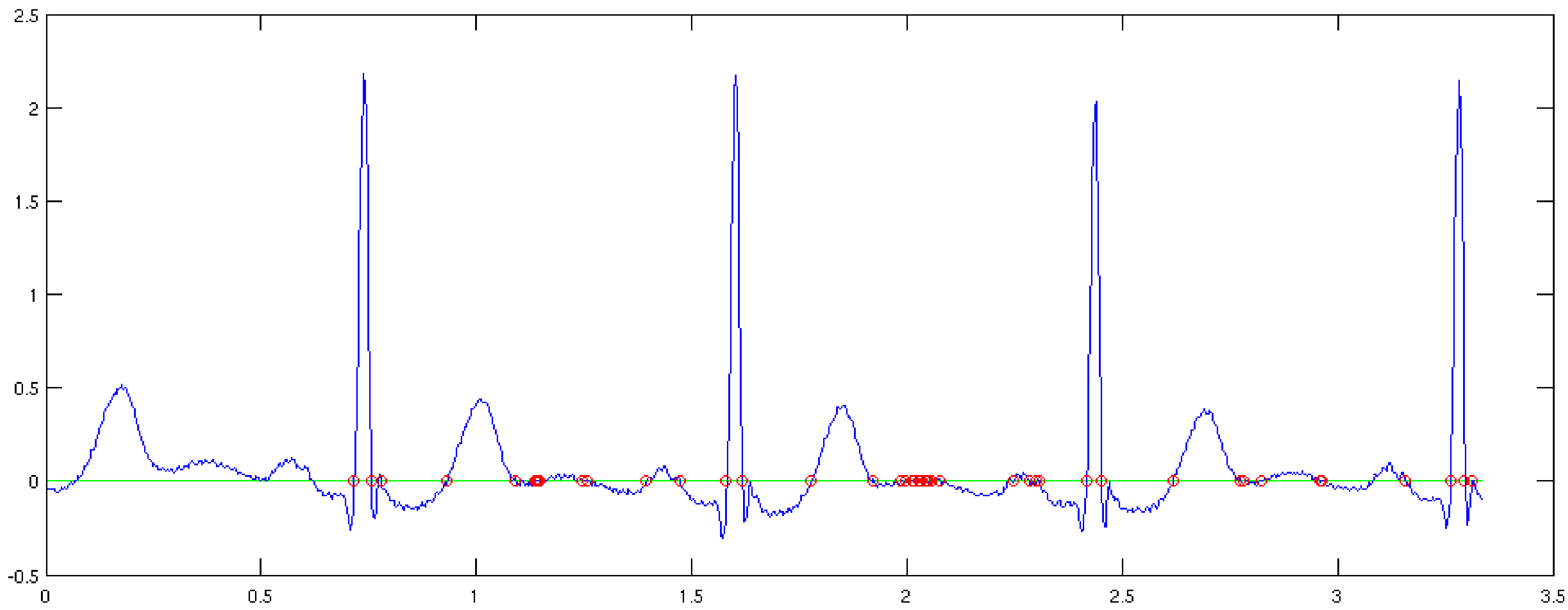


Strategy

- Find the long-term trend (0 for ECG) and detect when the ECG crosses this value
  - This looks too simple.
  - It is too simple.

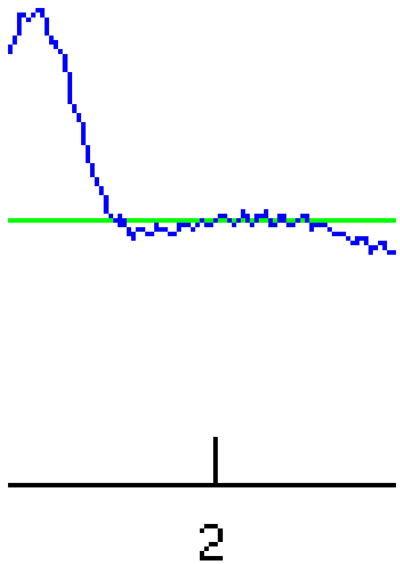


The problem



What is going wrong?

- If we look closely the data has a lot of small "wiggles" in it



  - Usually called "noise"
    - There are technical uses of the term, but in practice it means small unmodeled variations
  - Why do you see this in ECG data?

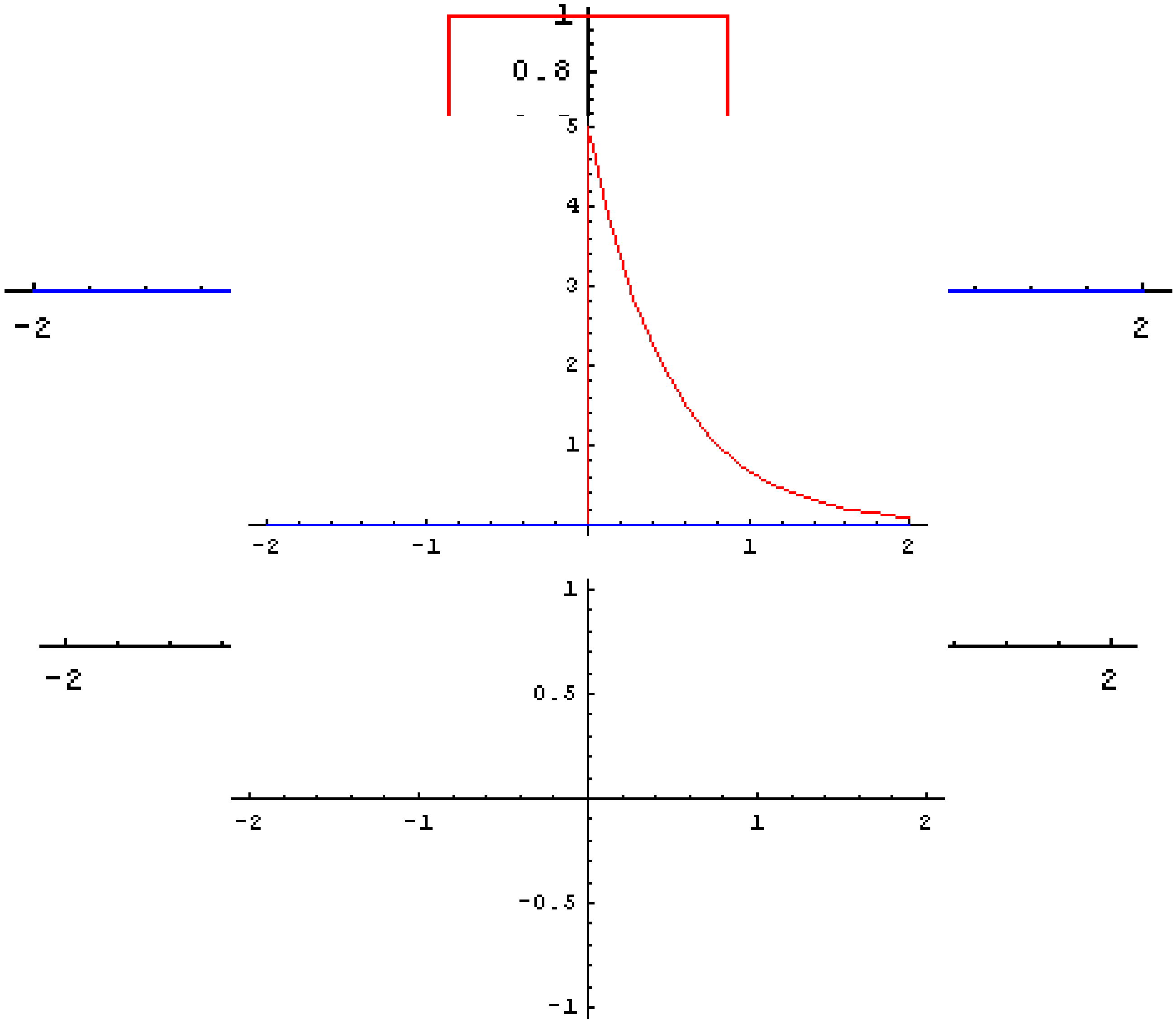
Idea: local averaging

- The obvious way to solve this problem, which turns out to be remarkably powerful
- Summary: replace every data point by the average with its two neighbors

  ```
  out[t] = 1/3 in[t-1] + 1/3 in[t] + 1/3 in[t+1]
  ```

    - Example: [… 10 11 9 13…] => [… 10 11 …]
  - We can write this as a stencil [1/3 1/3 1/3] which is applied to the sequence
    - At every point, multiply each neighbor by the stencil's value
  - Wider average: [1/5 1/5 1/5 1/5 1/5]

Local averaging in action



Convolution

- This idea (generalized to arbitrary stencils) is called convolution
  - Procedurally, it is totally trivial
- Remarkably useful and deep
- Useful both for smoothing but also for <u>finding </u>things of known shape
  - Such as, e.g., a (normal) ECG pulse
  - Not a panacea, but a vital tool

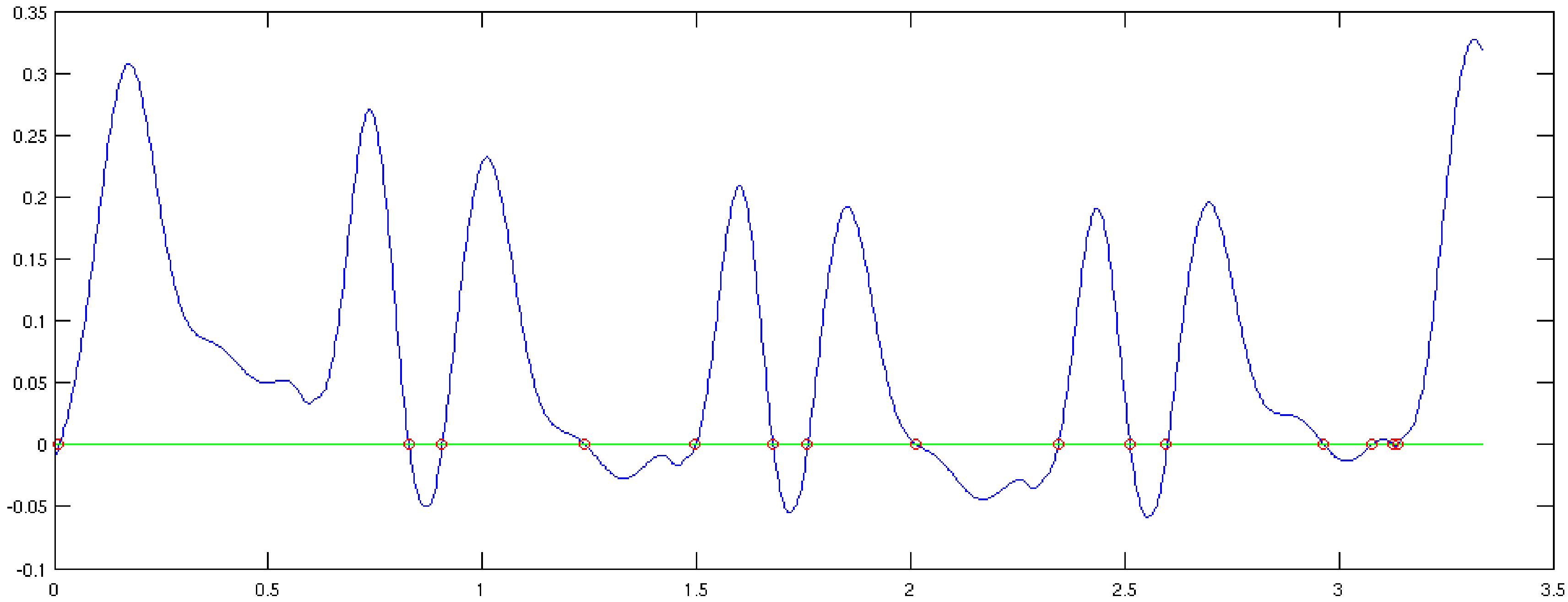
Convolution for smoothing

- The standard way to remove noise is by convolution with a Gaussian (“bell curve”)
  - Colloquially called a “low pass filter”
- Why a Gaussian?
  - If you really care, the central limit theorem plus a few other important properties
  - It also generally works pretty well

Parameters

- How wide an average do you need?
- This is not remotely obvious, and there aren't any great ideas about it
- More annoyingly, it makes a huge impact on practical performance
  - Too little smoothing means too many zero crossings (like original data)
  - Too much smoothing means whole ECG it completely blurred out
    - Can only see cycles, not peaks!

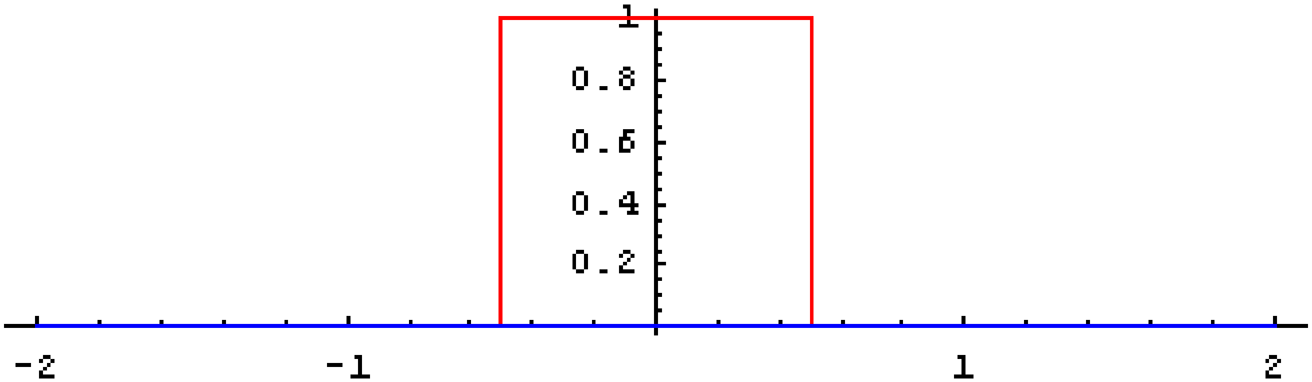
Smoothing parameter effects

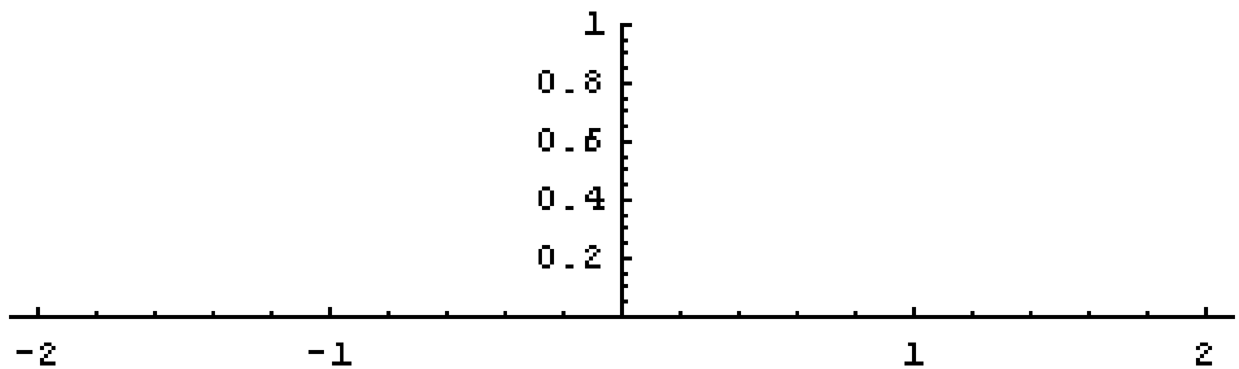


Matched filters

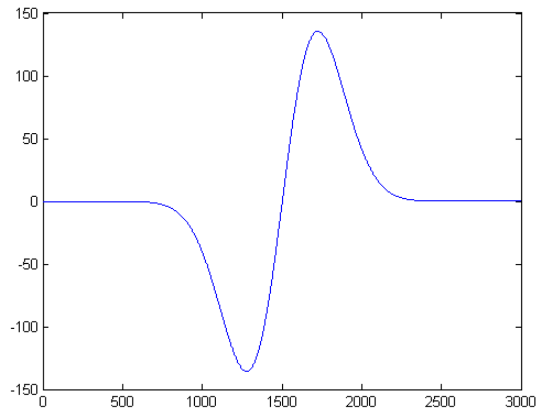
- Convolution can be used to find pulses
  - This is actually closely related to smoothing
  - Ashish's talk on Friday mentioned a few applications of this to images
- How do we find a known pulse in an ECG? Convolve the ECG with our template!
  - E.g. to find something in the ECG that looks like [1 6 -10] we convolve with [1 6 -10]
- Question: what sense does this make?
  - Anecdotally it worked for finding boxes

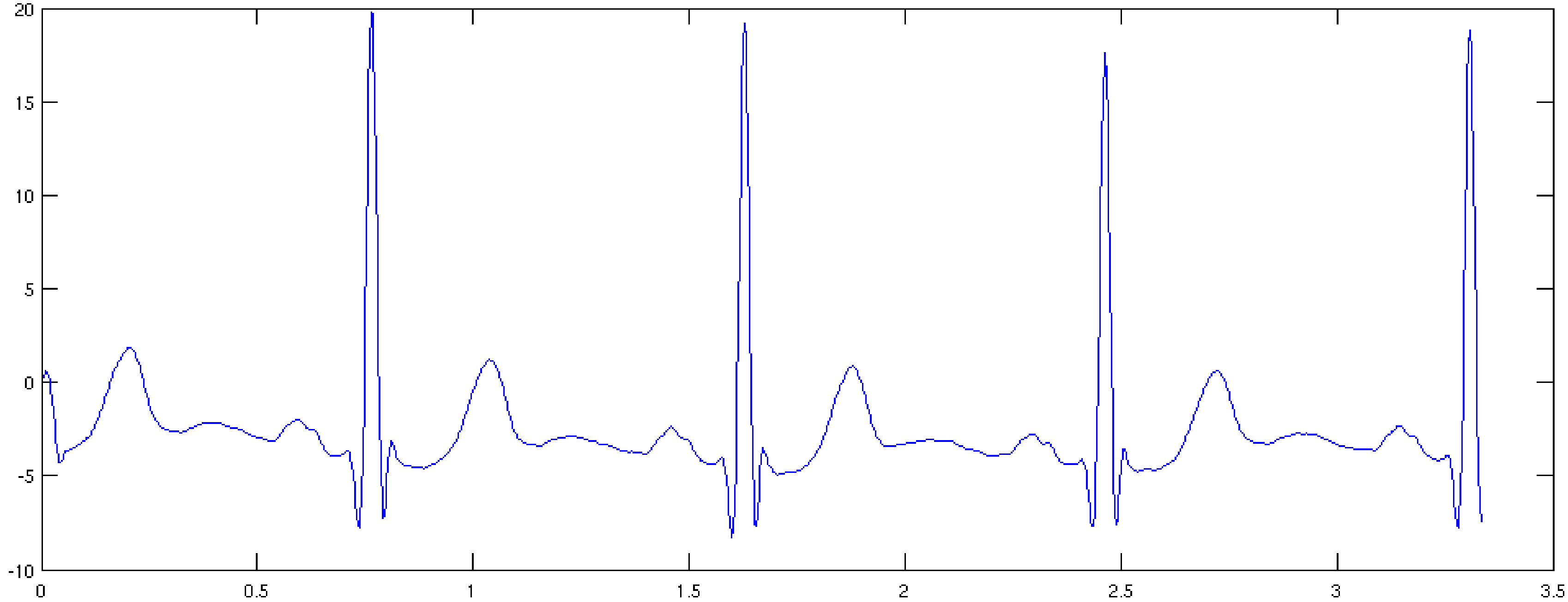
Box finding example





Pulse finding example



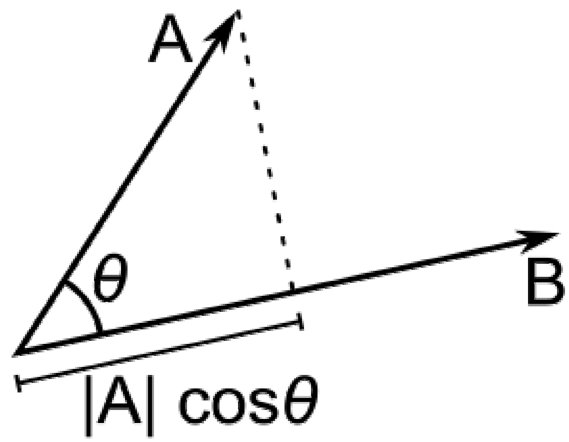


Why does this work?

- Some nice optimality properties, but the way I described it, the algorithm fails
- Idea: the [1 6 -10] template gives biggest response when signal is [… 1 6 -10 …]
  - Value is 137 at this point
- But is this actually correct?
  - You actually need both the template and the ECG to have a zero mean and unit energy (sum of squares)
    - Easily accomplished: subtract -1, then divide by 137, get 1/137 * [2 7 -9]

Geometric intuition

- Taking the dot product of two vectors
  - Recall [a b c]@[e f g] = ae + bf + cg
  - Basically the projection of a vector on another



- The normalized vector with the biggest projection on x is, of course: x!