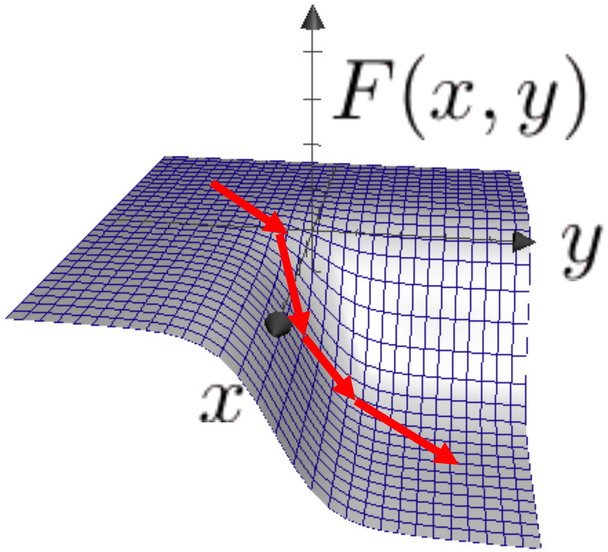
Lecture 35
Synthesis with Optimization

Sankha Narayan Guria
with slides from
Armando Solar-Lezama
in turn with slides from
Swarat Chaudhuri

Numerical Optimization

Goal: Find a (hopefully global) minima of a function

- Sample the function on a small set of values
- Based on the result sample in a new set of locations



- Gradient descent
- Newton iteration

…

Numerical Optimization for Synthesis

Many synthesis problems are best framed as optimization problems

- Especially true for problems involving hybrid systems or probabilities

The Parameter Synthesis Problem

```
tOff := ??;    tOn := ??;  h := ??
forever{
   temp := readTemp();
   if (isOn() && temp > tOff)
        switchHeaterOff();
  elseif ( !isOn() && temp < tOn)
       switchHeaterOn(h);
}
```

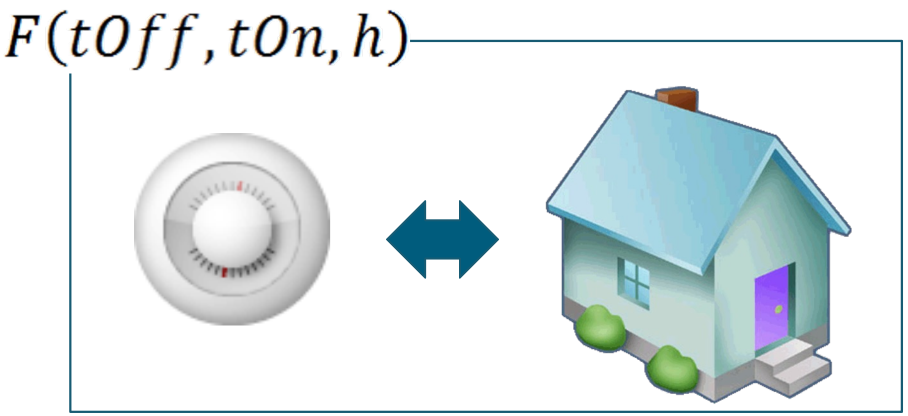
Warming:

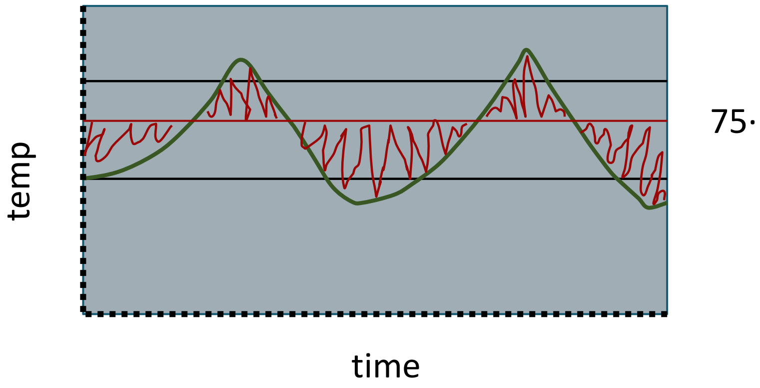
$$\frac{d}{dt} temp = -k \cdot temp + h$$

Cooling:

$$\frac{d}{dt} temp = -k \cdot temp$$

The Parameter Synthesis Problem



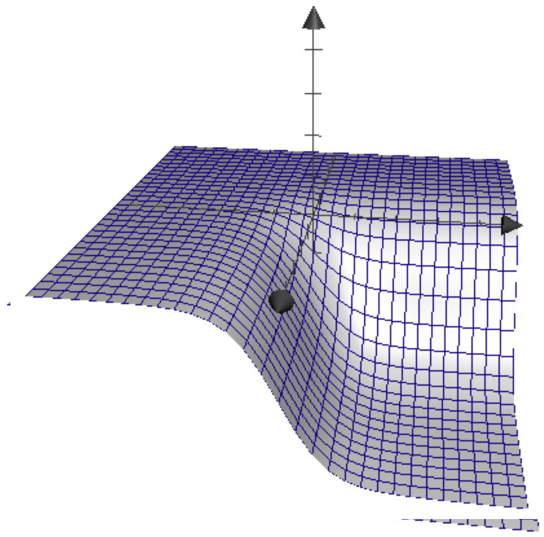


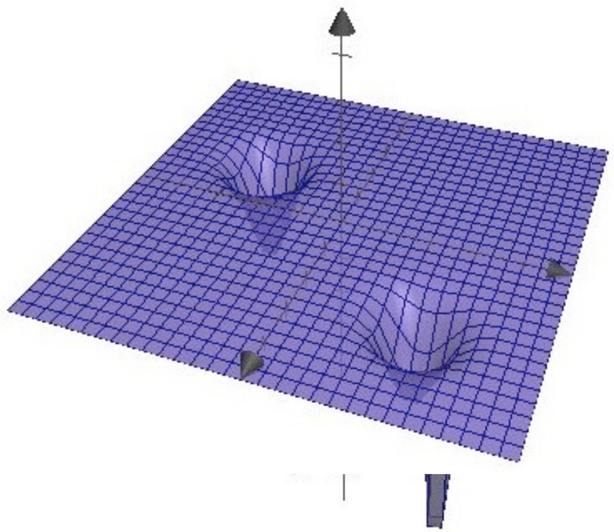
$$F(tOff, tOn, h) = [\, t_0, t_1, \dots, t_k]$$

$$Err(tOff, tOn, h) := \sum (t_i - 75)^2$$

Can we find values of (tOff, tOn, h) to minimize Err?

The challenge of optimizing programs





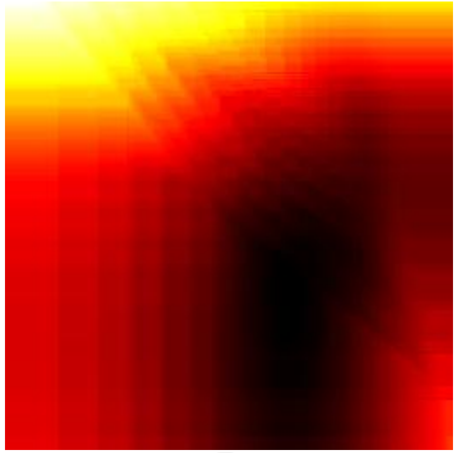
Plateaus and discontinuities thwart search

Branches introduce discontinuities

For the thermostat example

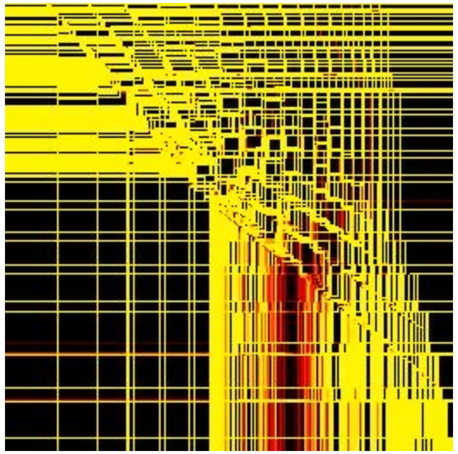
$Err(tOn, tOff, 150)$

$tOff$



tOn

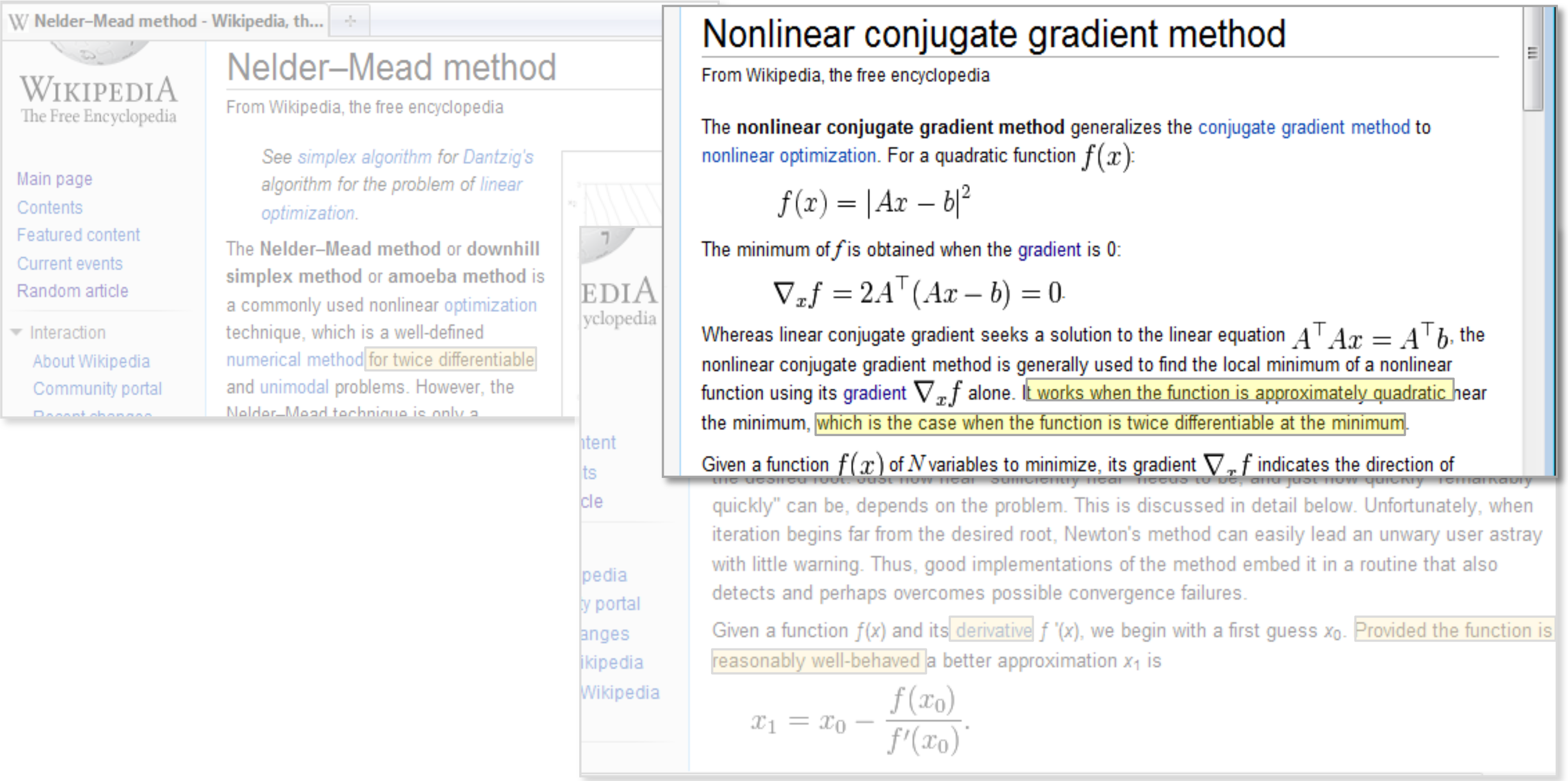
$\left|\vec{\nabla} Err(tOn, tOff, 150)\right|$



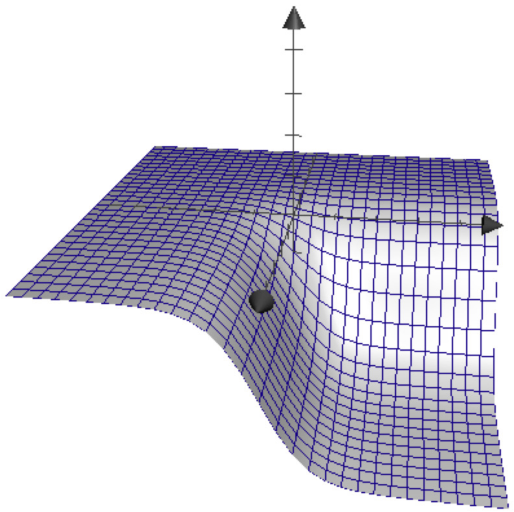
tOn

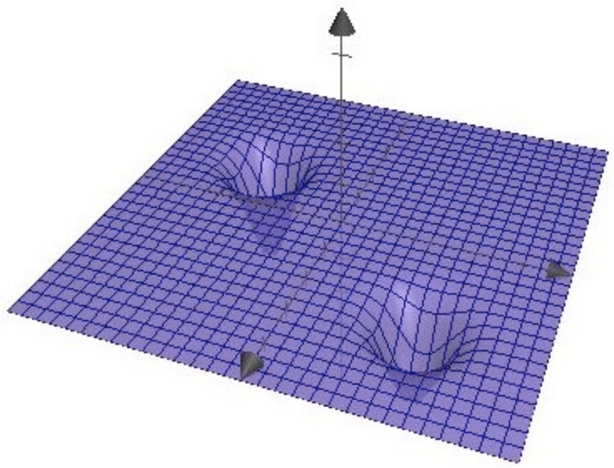
Minimization by numerical Search

Discontinuities affect even the best methods



Key Idea: Smooth the Function

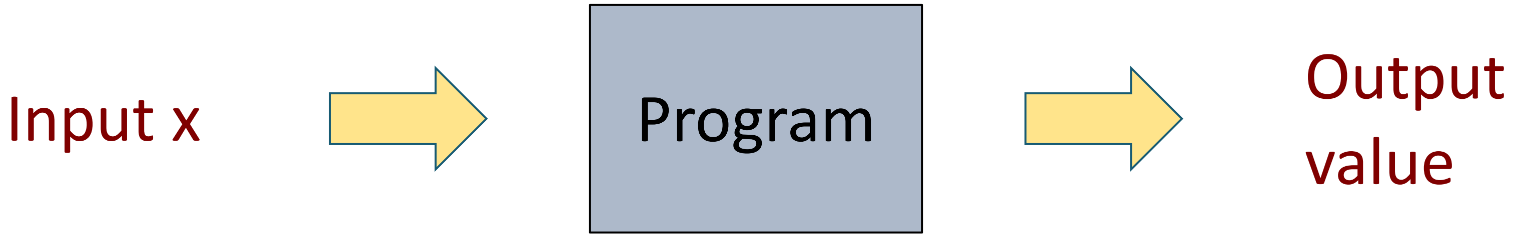




Approximate the problem by a smooth function

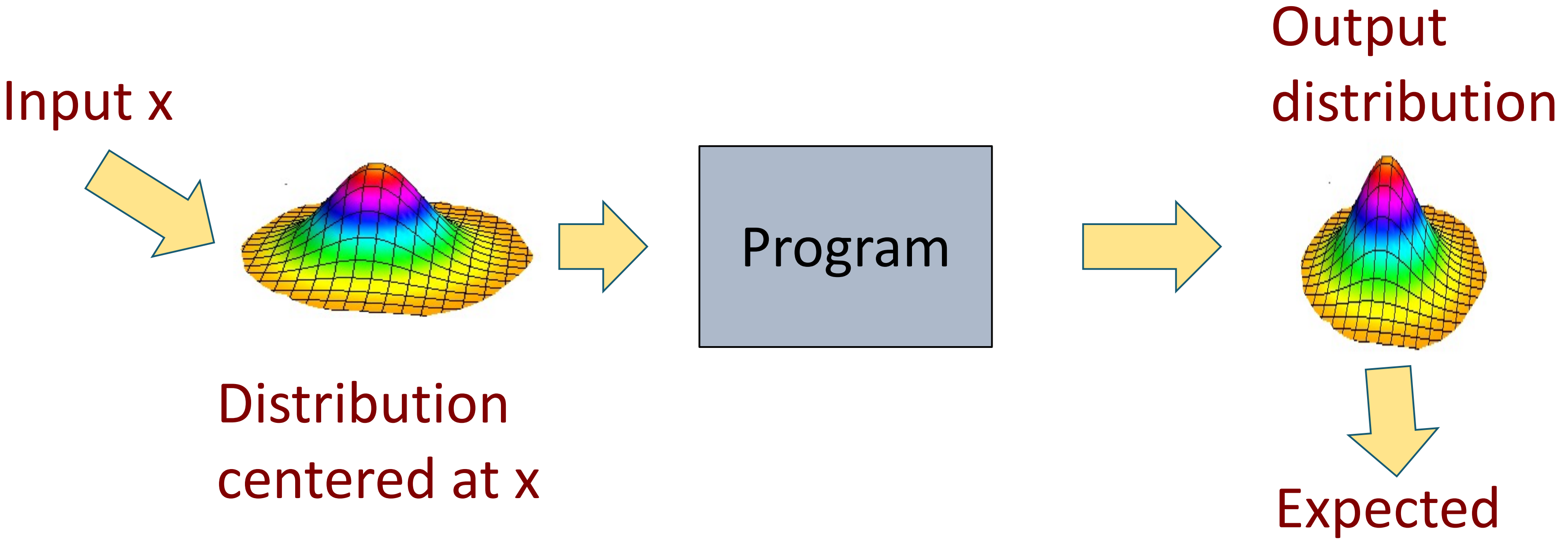
Example: Gaussian Smoothing

Execute on a distribution instead of a value



Example: Gaussian Smoothing

Execute on a distribution instead of a value

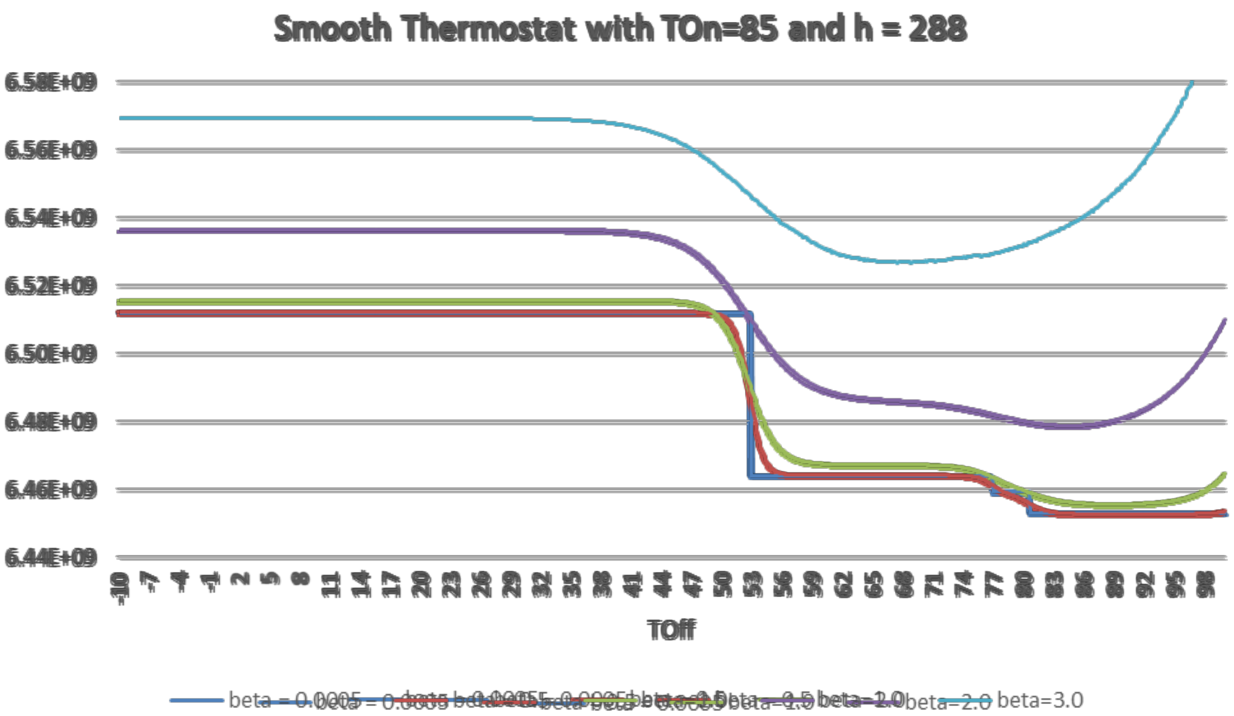


Expected value smooths discontinuities

Parameterized Smoothing

Parameterized by a value β

- Corresponds to std dev when smoothing with Gaussian noise

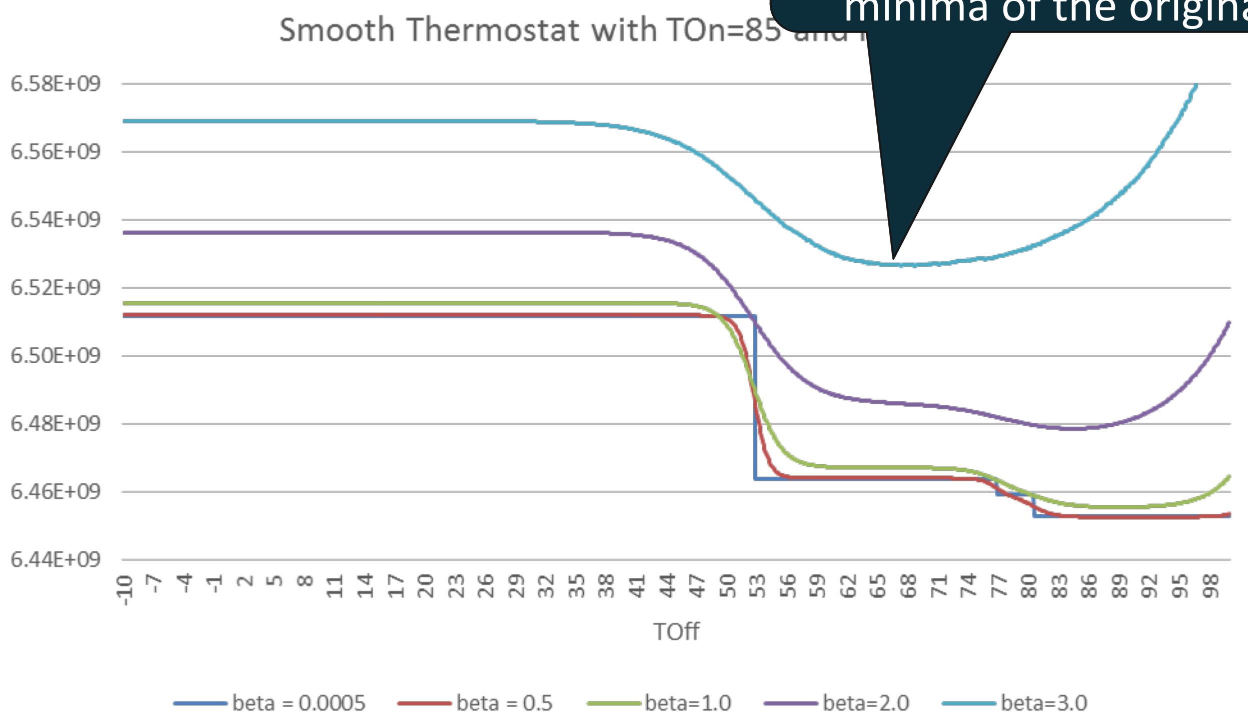


Parameterized Smoothing

$F_{smooth}(p, \beta_i)$ parameterized by a value β

- Corresponds to std dev when smoothing with Gaussian noise

Minima of smooth function is different from minima of the original



Smoothed Numerical Search

$p_0 = random\ starting\ point$

$\beta_0 = initial\ beta$

$While\ \beta_i > \epsilon$

$\quad p_{i+1} = optimize\ F_{smooth}(p, \beta_i) starting\ at\ p_i$

$\quad \beta_{i+1} = \beta_i * q$

$return\ p_i$

$0 \leq q < 1$

Smoothed Numerical Search

Very general procedure

Steps:

- Frame your problem as an optimization problem
- Define a parameterized smoothing procedure
- Run the algorithm

Smoothing doesn't have to be perfect

- But it should approximate the desired function in the limit as $\beta \to 0$

Euler:
SNS for programs

SNS in Euler

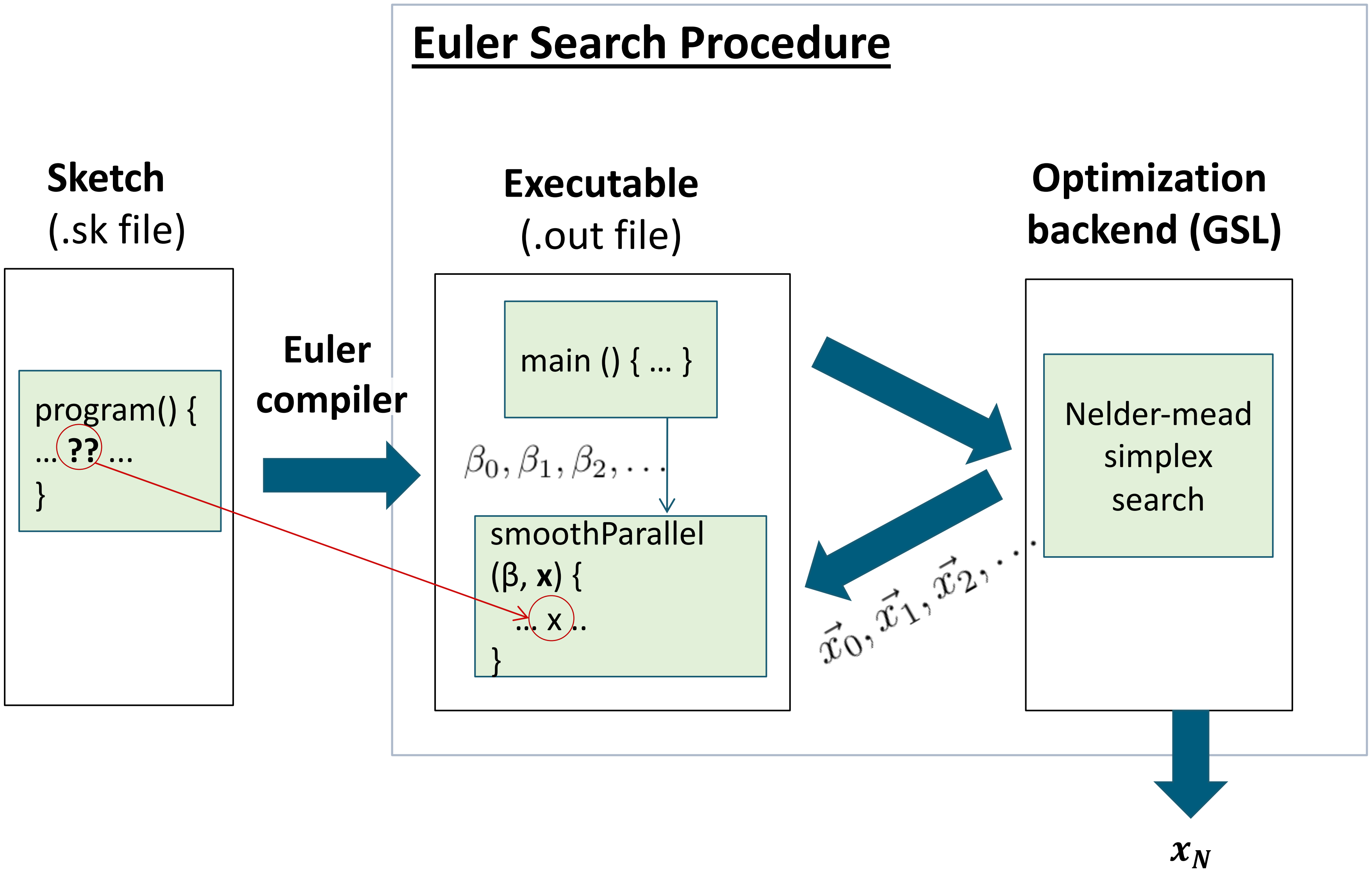
Use Symbolic Gaussian Smoothing

- β is simply the standard deviation

Quantifier alternation is handled through fixed test harnesses

- not ideal,
- but works well in practice

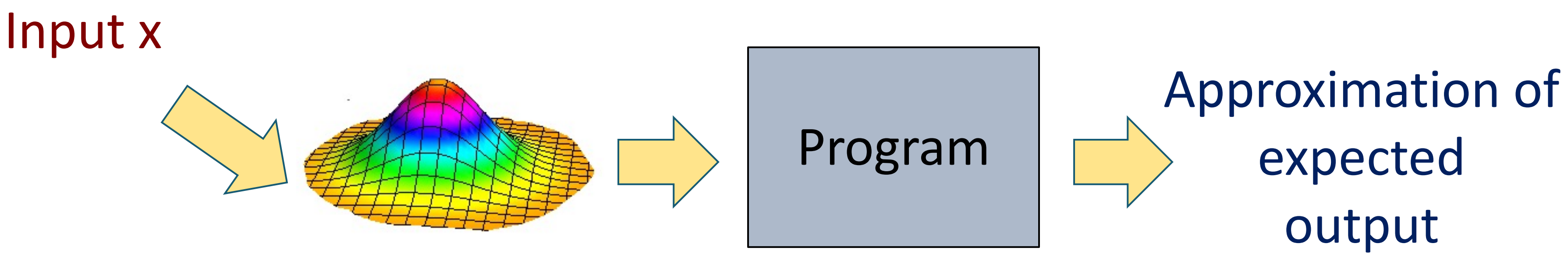
Smoothed Numerical Search in Euler



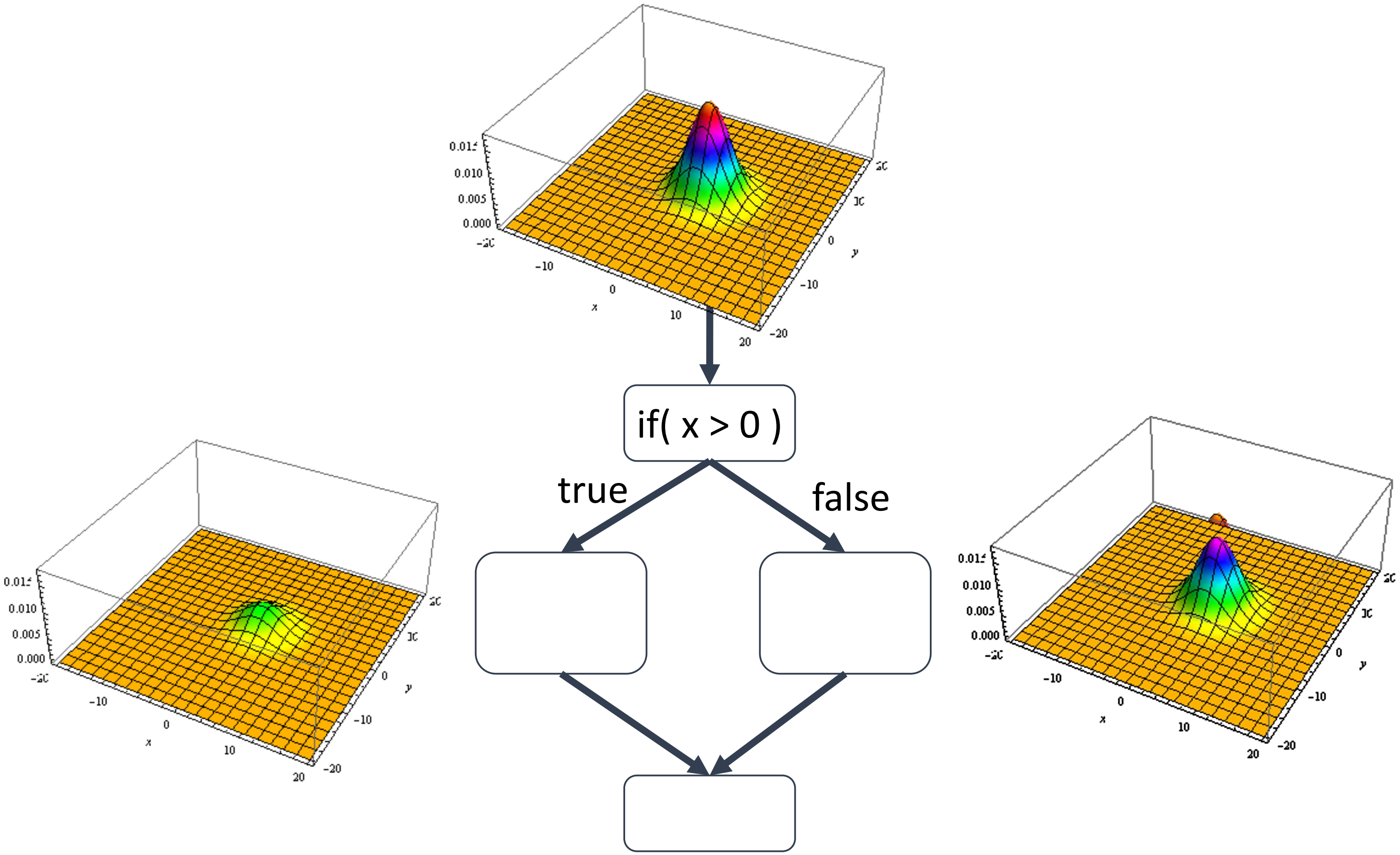
Smooth Interpretation

How do we execute a program on a distribution of inputs?

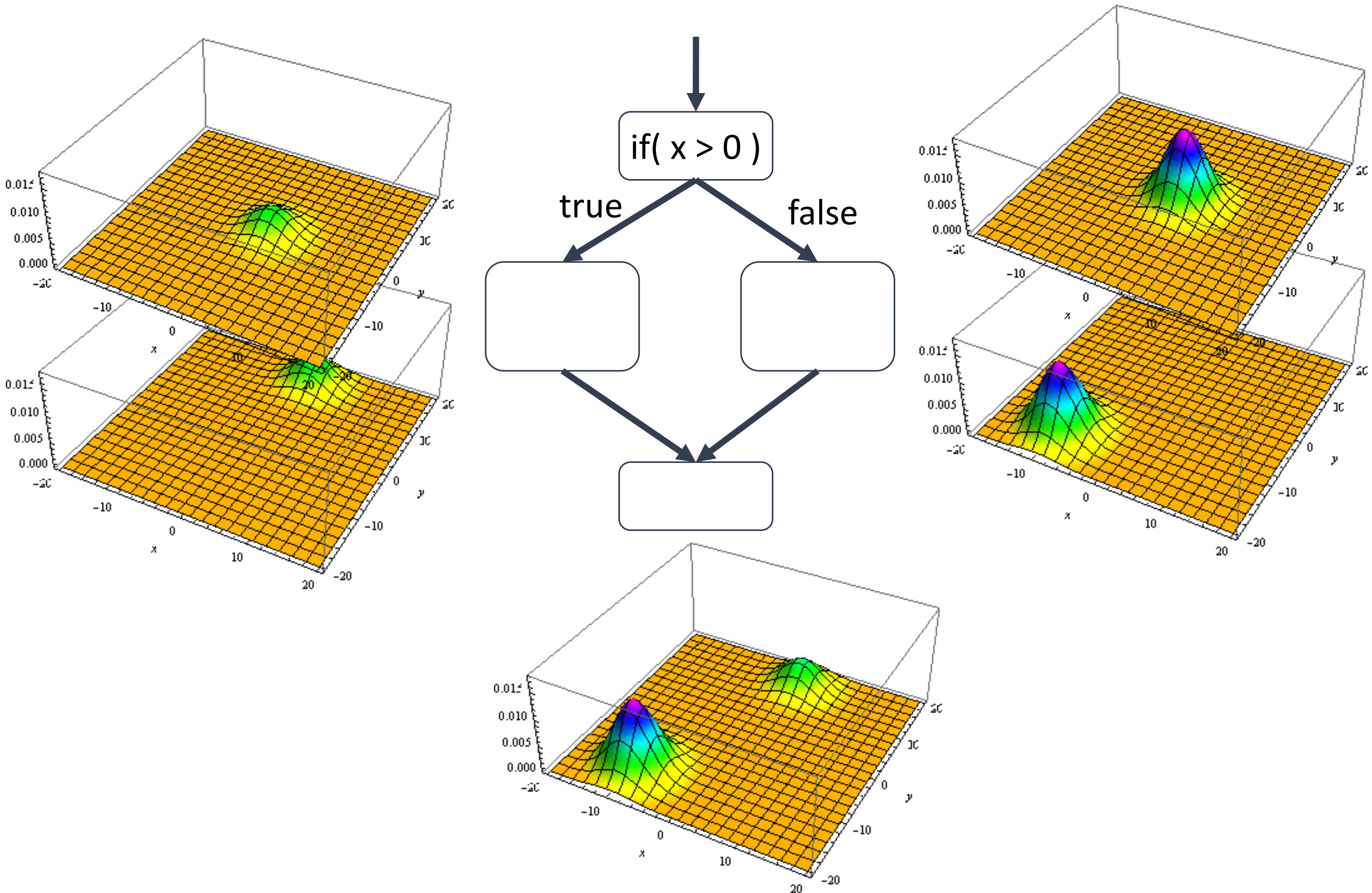
- Not by sampling
  - we would miss essential features of the search landscape.
- Symbolic execution:
  - propagate symbolic representations of distributions.



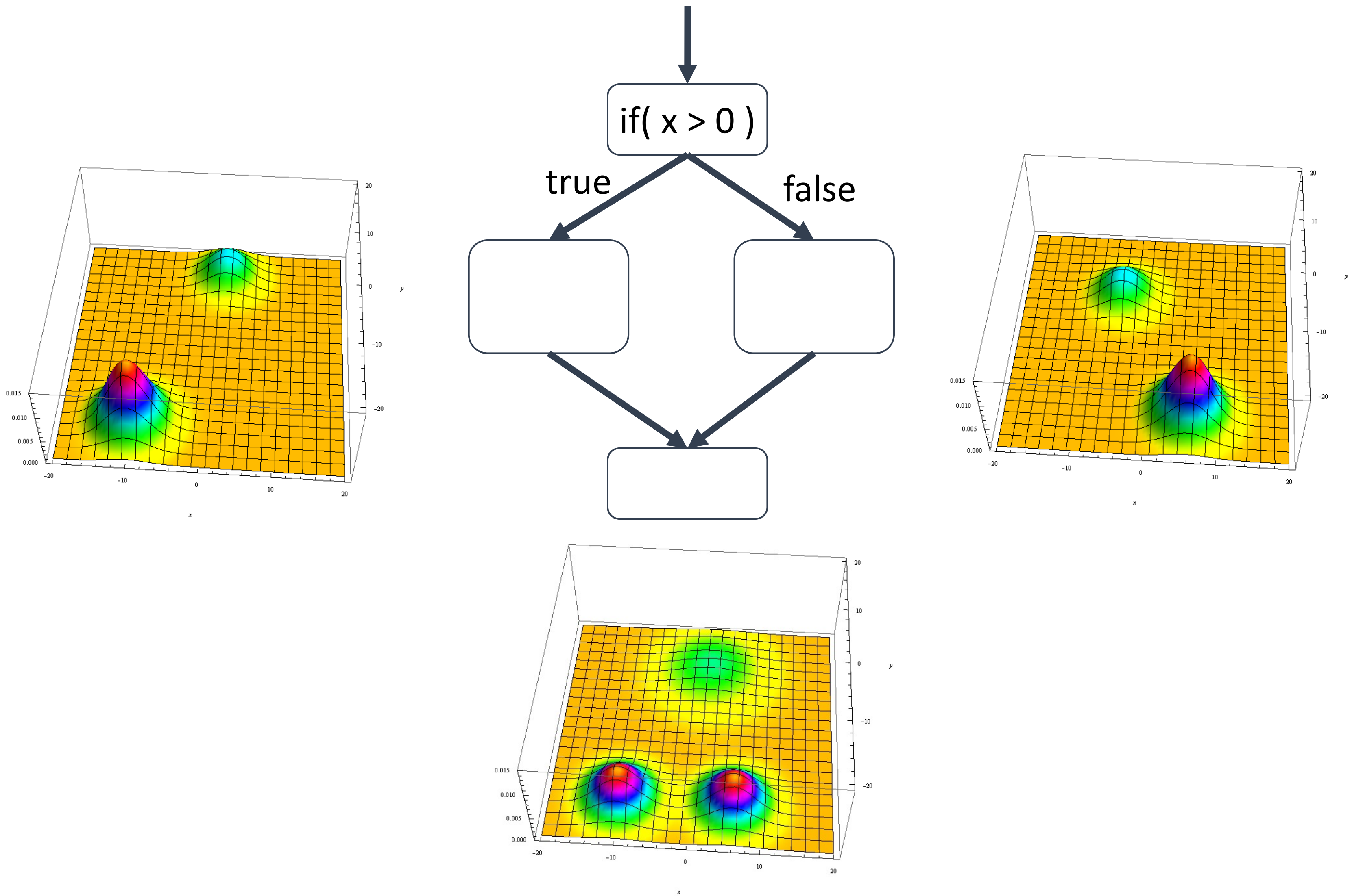
Smooth Interpretation: Branch



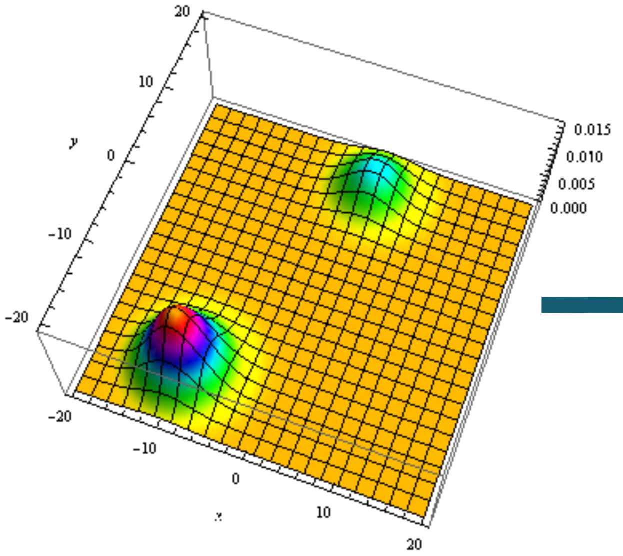
Smooth Interpretation: Join



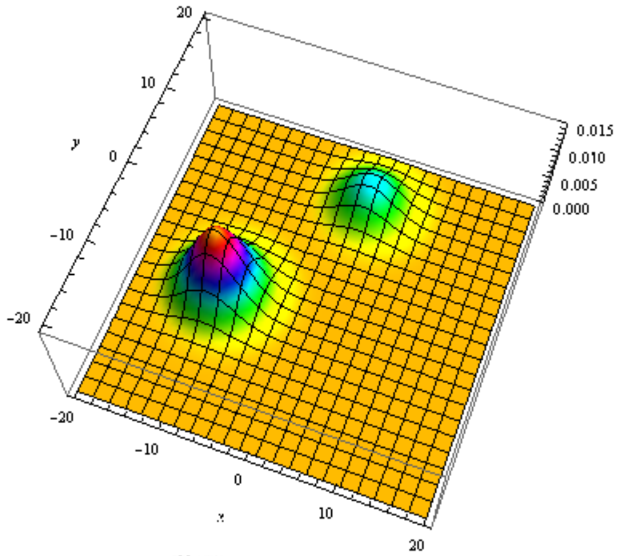
Smooth Interpretation: Join

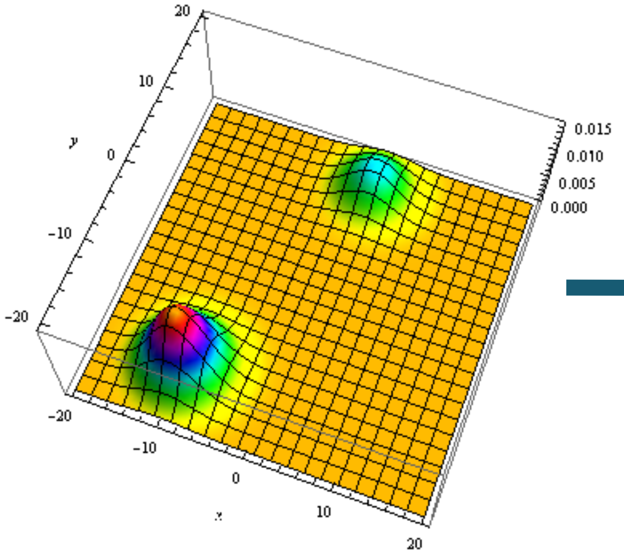


Smooth Interpretation: Assignment

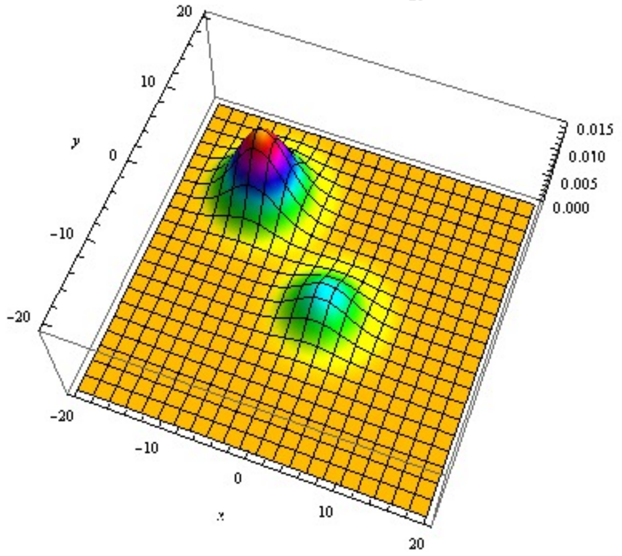


`x = x - y;`





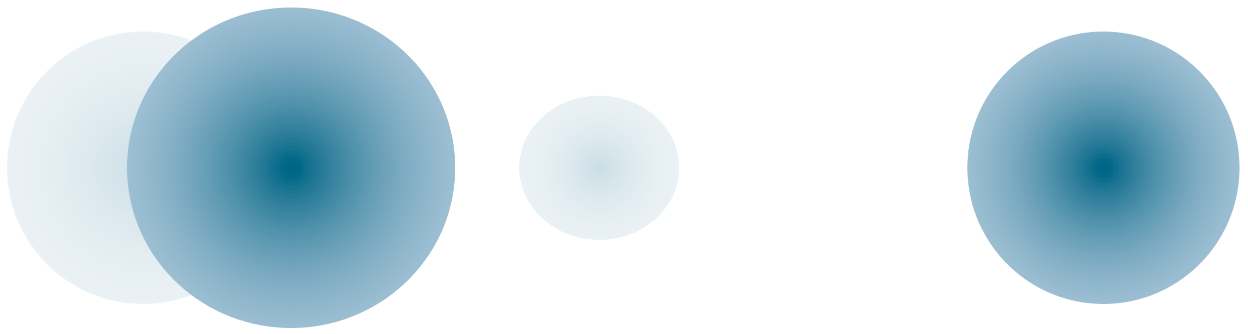
`x = -y;`



Approximations

Trick Question:

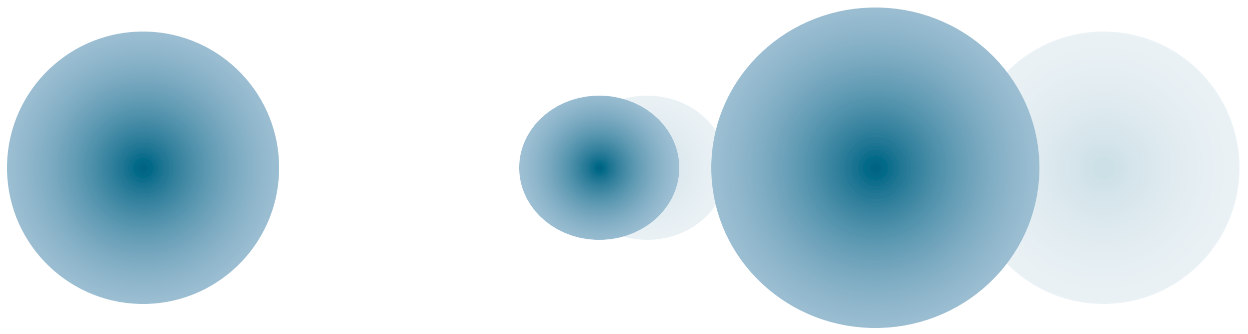
- Which of the previous approximations is problematic?



Approximations

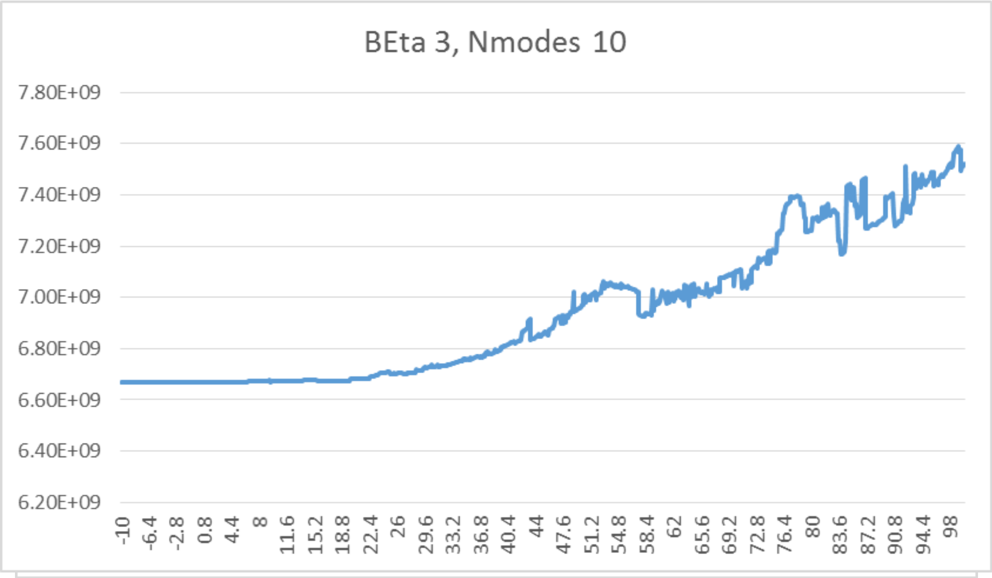
Trick Question:

- Which of the previous approximations is problematic?



Merging can introduce discontinuities!

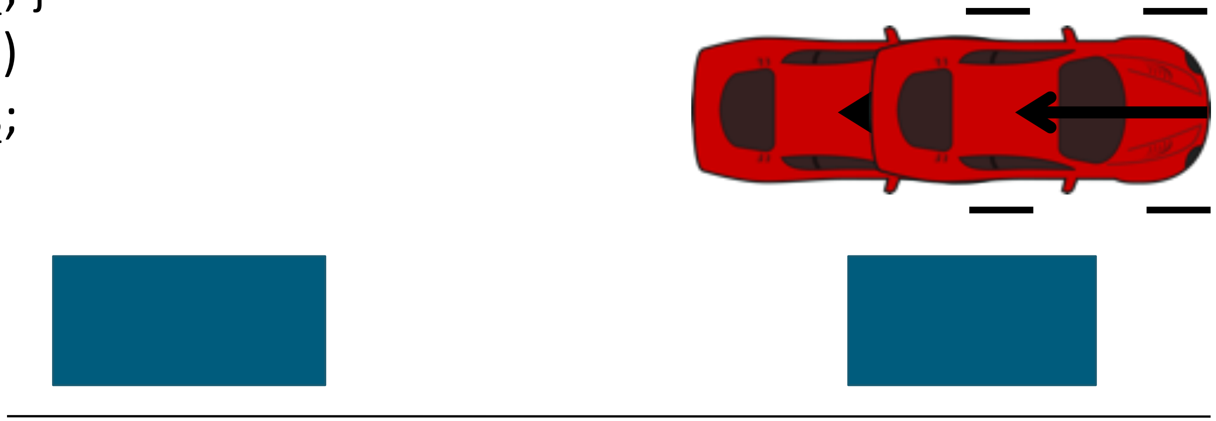
Excessive merging can cause problems



Examples

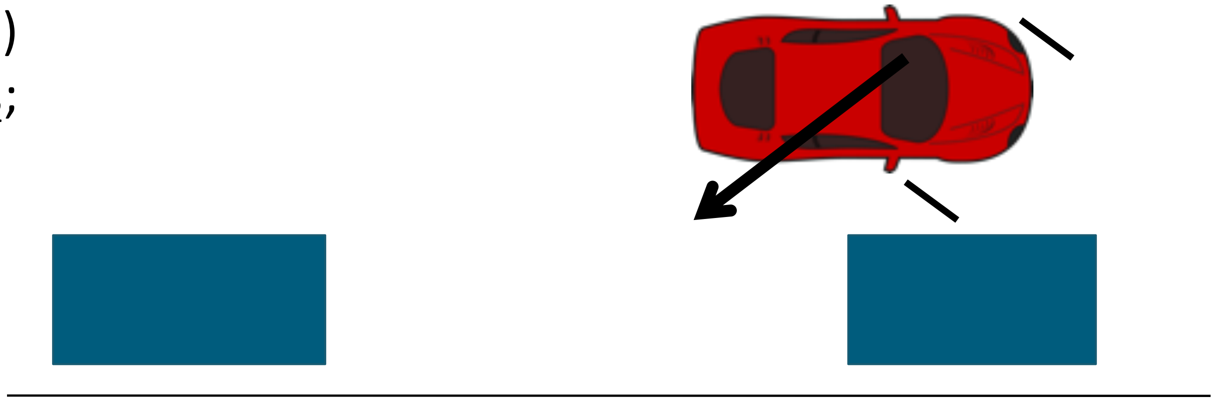
Parameter synthesis

```
t1 := ??;    t2 := ??;  A1 := ??;
A2 := ??;
repeat  (every ΔT)  {
    if (stage = BACK && time > t1)
        stage = INTURN;
    if (stage = INTURN) {
        if (time > t2) then stage = OUTTURN;
        angle = angle – A1; }
    if (stage = OUTTURN)
        angle = angle + A2;
  moveCar();
}
```



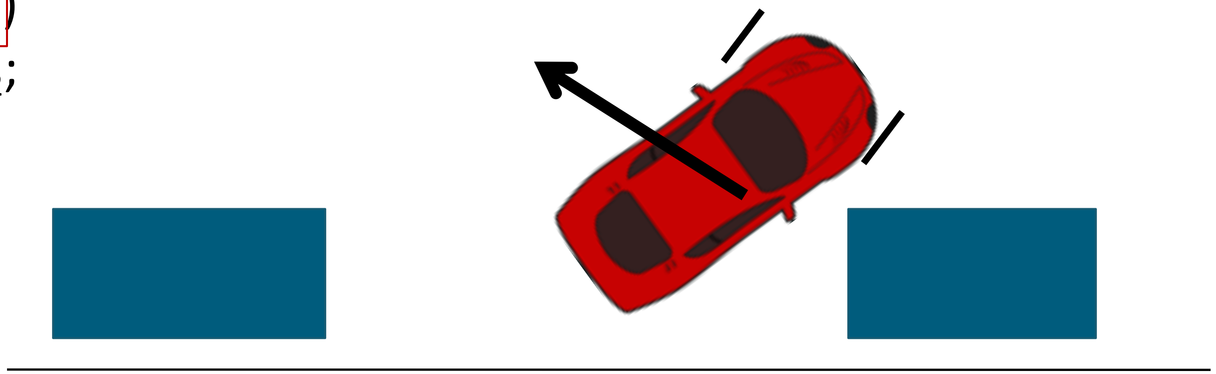
Parameter synthesis

```
t1 := ??;     t2 := ??;  A1 := ??;
A2 := ??;
repeat  (every ΔT)   {
    if (stage = BACK && time > t1)
         stage = INTURN;
    if (stage = INTURN) {
        if (time > t2) then stage = OUTTURN;
        angle = angle – A1; }
    if (stage = OUTTURN)
        angle = angle + A2;
  moveCar();
}
```



Parameter synthesis

```
t1 := ??;    t2 := ??;  A1 := ??;
A2 := ??;
repeat  (every ΔT)  {
    if (stage = BACK && time > t1)
         stage = INTURN;
    if (stage = INTURN) {
        if (time > t2) then stage = OUTTURN;
        angle = angle – A1; }
    if (stage = OUTTURN)
        angle = angle + A2;
  moveCar();
}
```

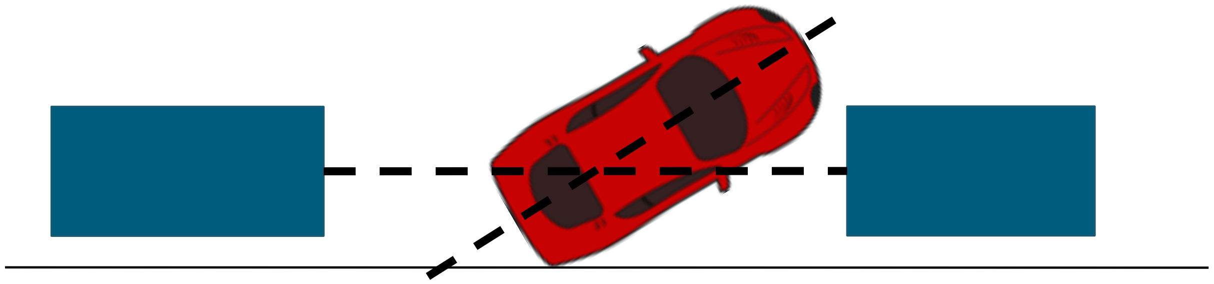


Application : Parameter synthesis

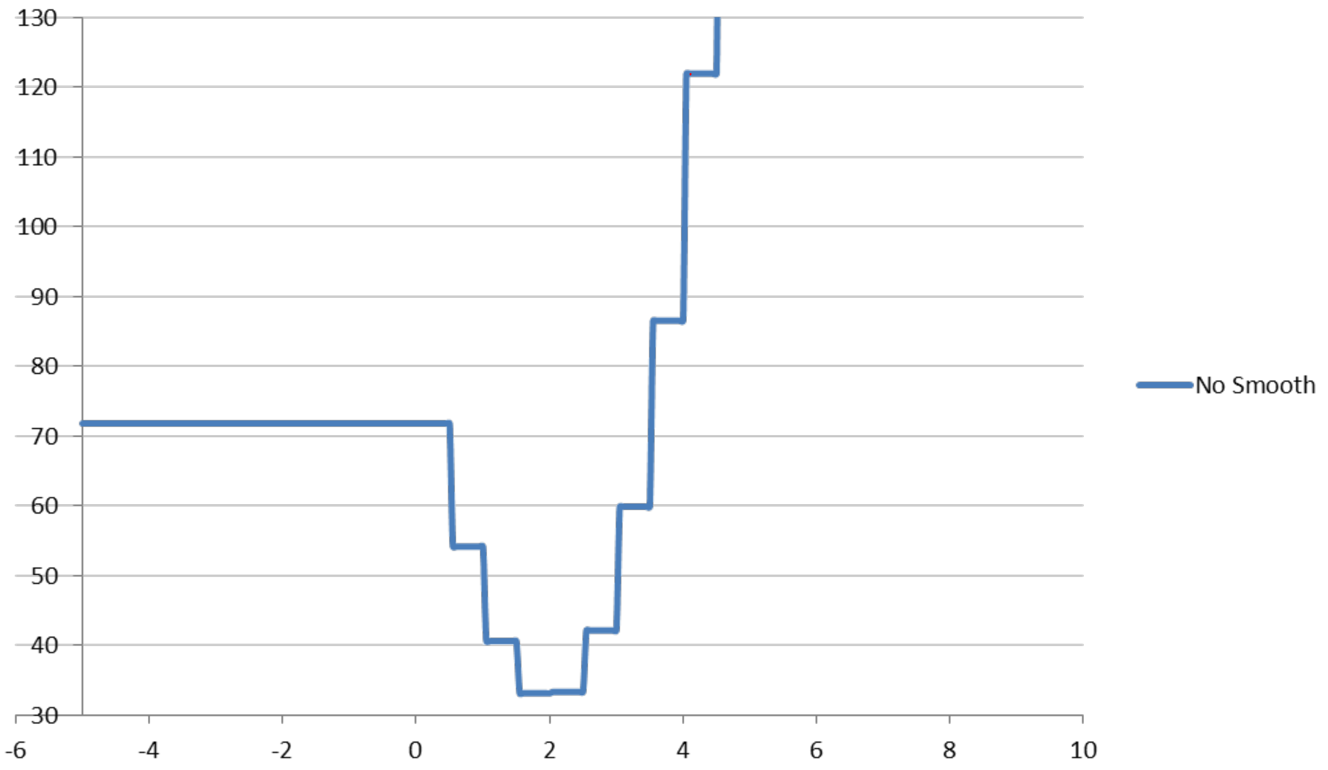
```
t1 := ??;    t2 := ??;  A1 := ??;
A2 := ??;
repeat  (every ΔT)   {
    if (stage = BACK && time > t1)
         stage = INTURN;
    if (stage = INTURN) {
        if (time > t2) then stage = OUTTURN;
        angle = angle – A1; }
    if (stage = OUTTURN)
        angle = angle + A2;
   moveCar();
}
```

Error = distance between ideal and actual final position

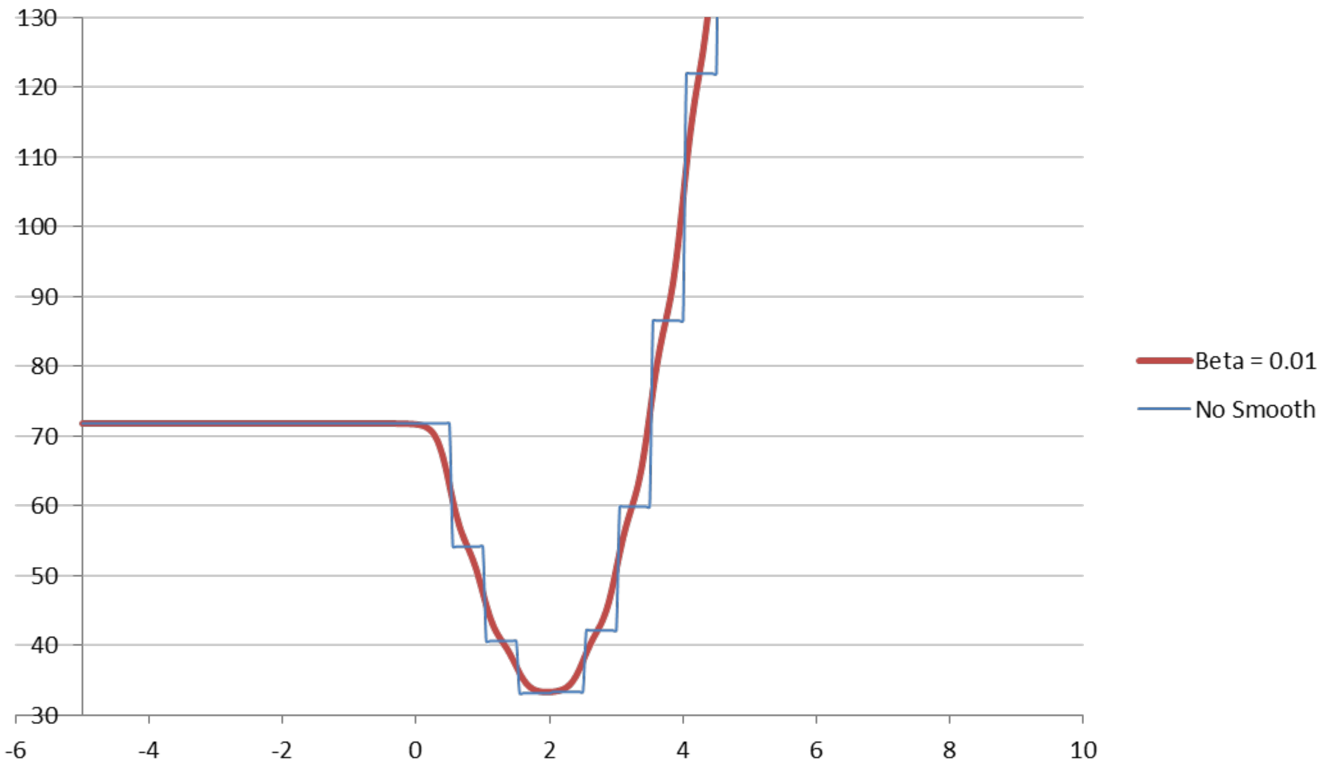
Goal: minimize Error.



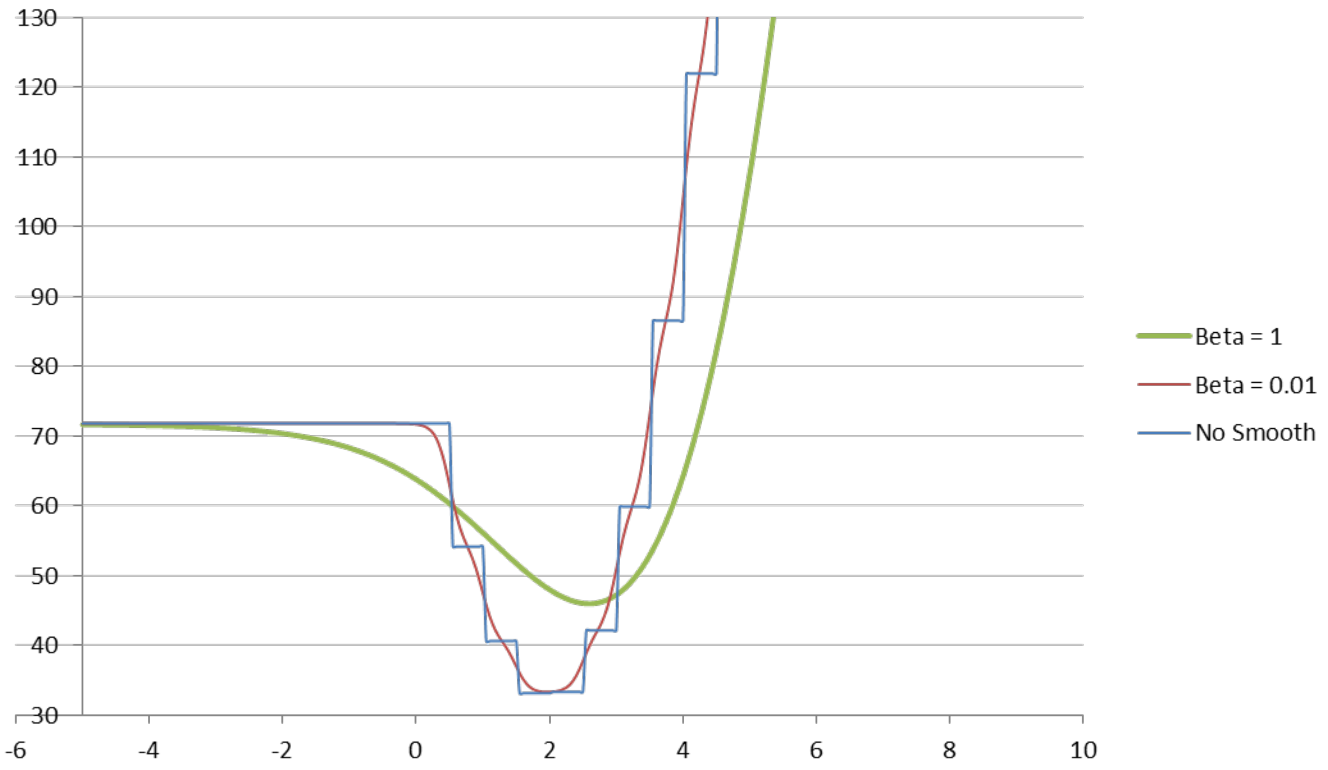
Slice of Error function WRT T2



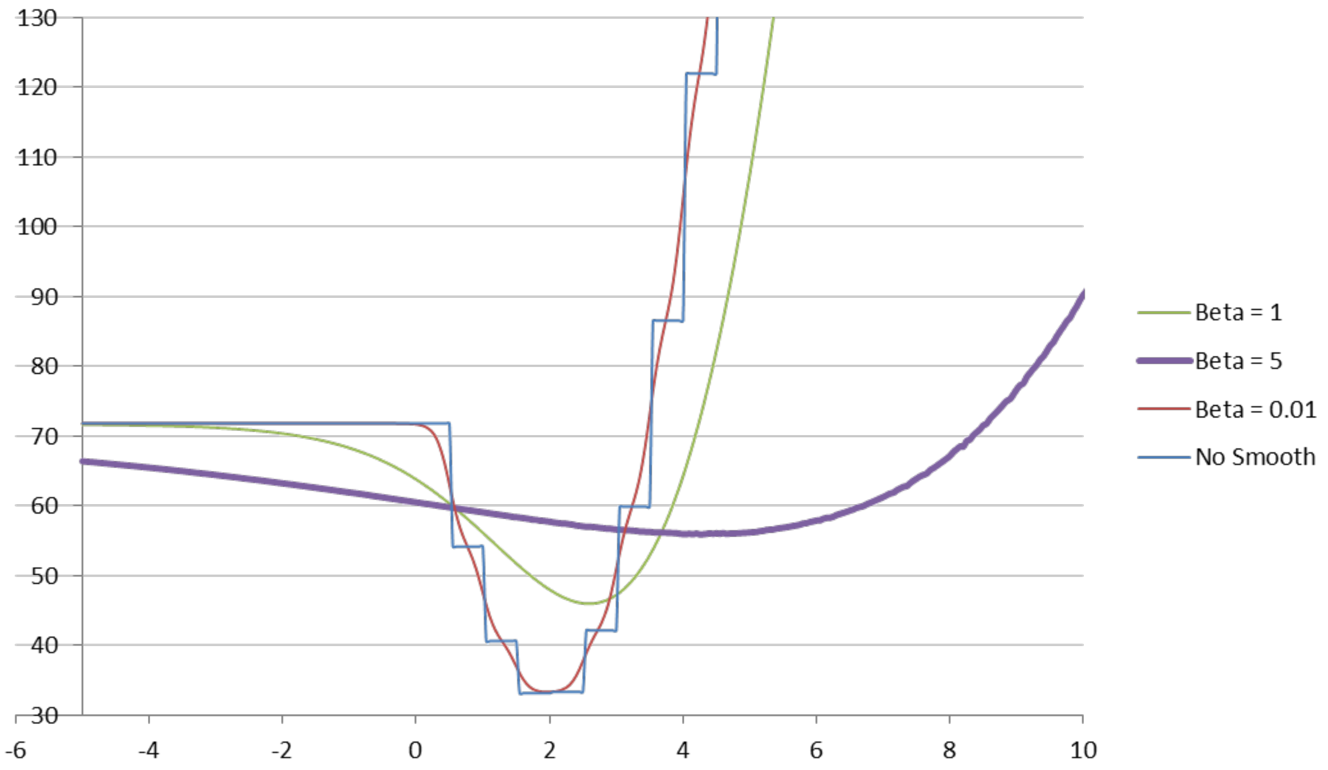
Slice of Error function WRT T2



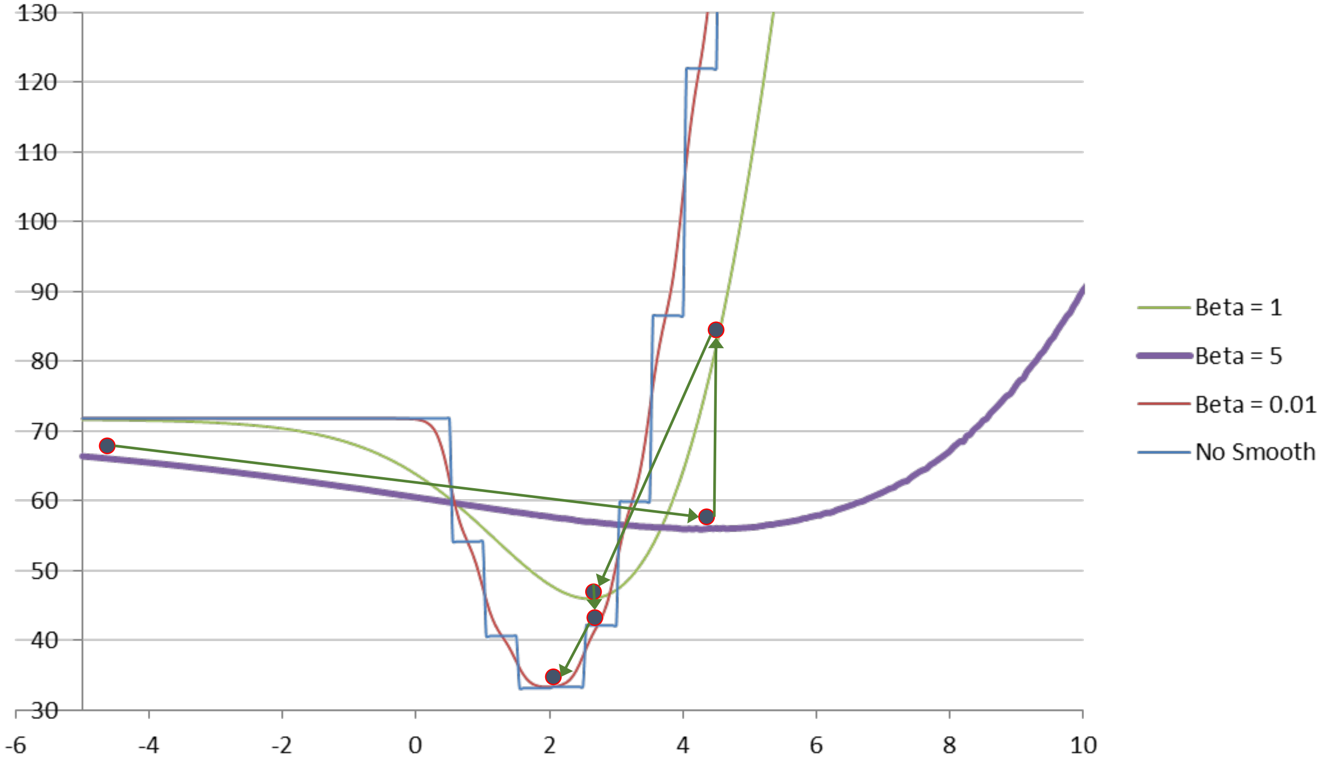
Slice of Error function WRT T2



Slice of Error function WRT T2



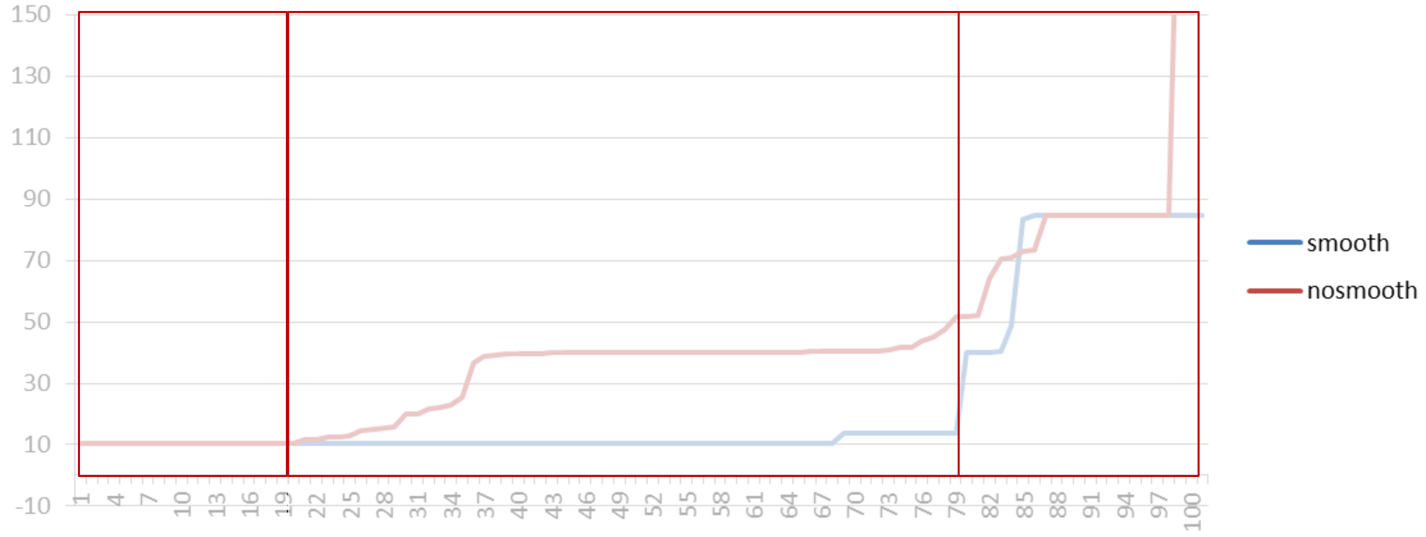
Numerical Search with Smoothing



Errors from different starting points

Both are quite good

Both are quite bad



Result of smoothing is much better