

Algorithm Applications

Computer & Information Sciences

W. H. Bell

Overview

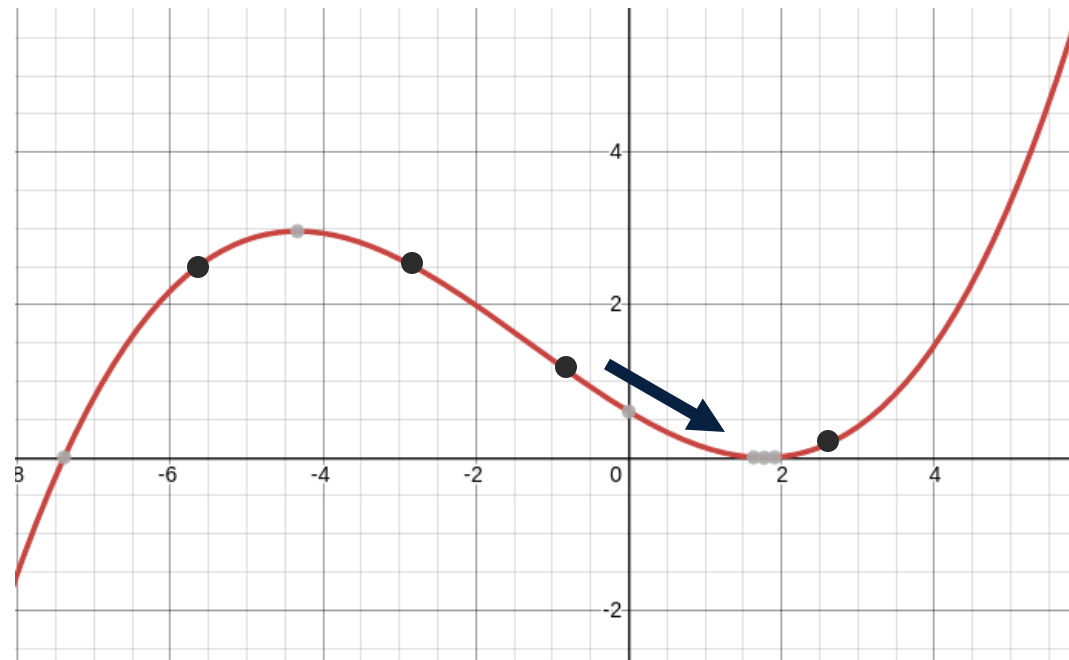
- Solving difficult problems.
- Numerical minimisation.
- Simulating systems.
- Data analysis.
- Summary.

Solve difficult problems

- Finding solutions to complicated functions.
- Numeric minimisation or maximisation.
- Many different variables.
- Numeric integration.
- Simulating systems and predicting outcomes.
- Probability distributions.
- Analysing large amounts of data.
- Parallel, distributed processing.

Numeric minimisation

- Calculate points at random and then walk down the distribution.



Simulating systems

- Simulate probabilistic systems with random numbers.
- Systems that have uncertainty or are random by their nature.

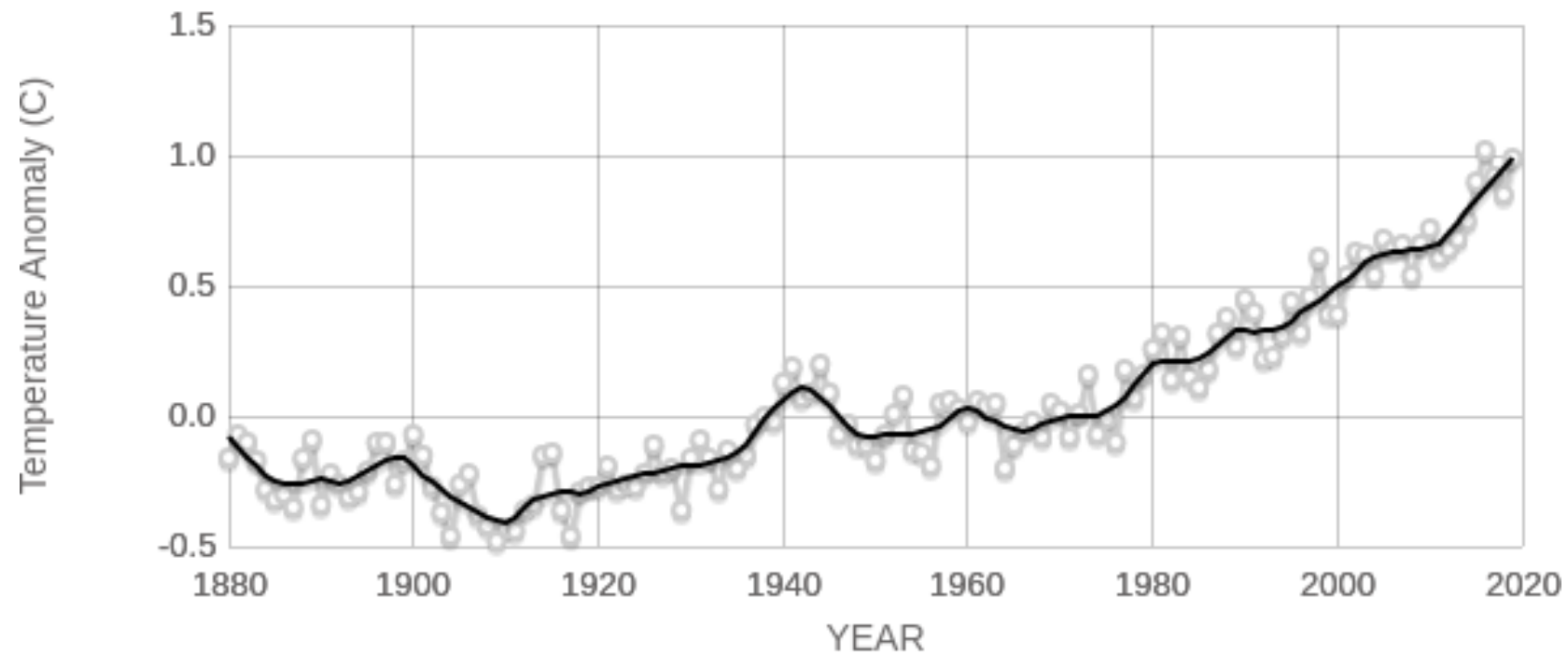
```
import random
numberOfRolls = 10000
average = 0.
for i in range(numberOfRolls):
    twoDice = random.randint(1,6)
    twoDice += random.randint(1,6)
    average += float(twoDice)/numberOfRolls
print("Average = " + str(average))
```

Output

```
Average = 7.0053000000000212
```

Data analysis

- Smoothing, signal extraction, numeric inversion.



Source: climate.nasa.gov

Summary

- Solve problems by writing computer programs.
- Many of these problems would be difficult or impossible to solve without a computer.
- Demonstrate the usefulness of software.
- Watch the live coding demonstrations.