Inside SPICE

Laurence W. Nagel
Omega Enterprises Consulting
Kensington, CA 94708

nVidia Corporation
Santa Clara, California
May 19, 2011

SPICE (Simulation Program with Integrated Circuit Emphasis)

- First Released in 1971 and announced in 1973 at the Sixteenth Midwest Symposium on Circuit Theory
- Rapidly adopted by universities and industry in the 1970's
- SPICE 2G6 became the de facto industry standard in the 1980's
- Now ubiquitous in the electronics industry

SPICE (Simulation Program with Integrated Circuit Emphasis)

- DC operating point analysis, small-signal AC analysis and transient analysis in one package
- Built-in models for diodes and bipolar transistors
- Modified Newton-Raphson iteration with heuristics that worked well with bipolar circuits
- Implicit integration techniques reduced problems with the widely spread time constants of an IC
- Utilized sparse matrix techniques, so it could run circuits with hundreds of nodes

The SPICE Era

- 1973 - SPICE1
- 1981 - SPICE2
- 1981 - HSPICE
- 1984 - PSPICE
- 1984 - Eldo
- 1986 - SPECTRE
- 1989 - SPICE3

The SPICE Era

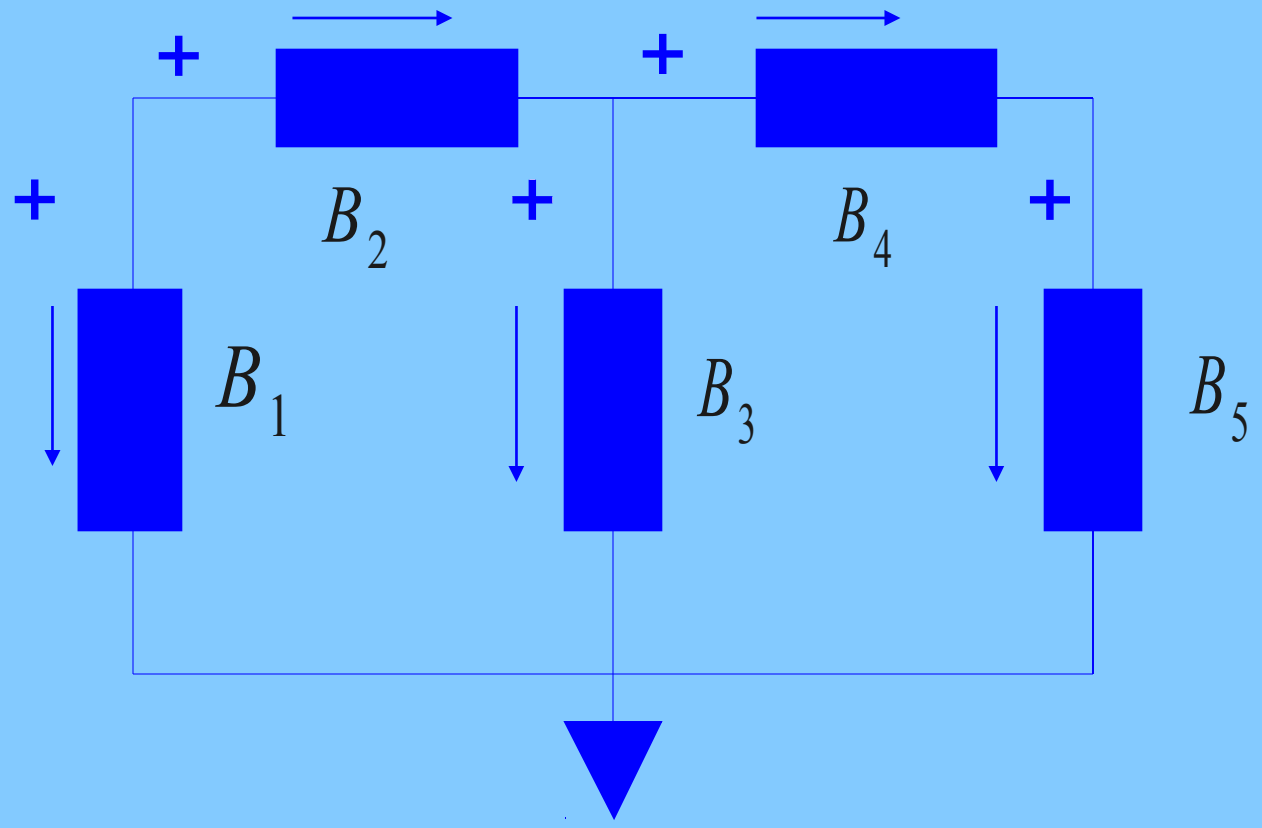
- SPICE applications were
  - Analog circuits (small)
  - Critical paths in digital circuits
  - Memories
- SPICE algorithms were tuned to go faster but not work smarter
- Model development driven by technology evolution and digital circuits

Circuit Simulation Algorithms – Then and Now

- Circuit formulation techniques
  - Construct a set of integro-differential equations describing the circuit
- Numerical integration techniques
  - Solve the time-domain problem by a transient analysis of sequential timesteps
- Nonlinear equation techniques
  - Solve the set of nonlinear equations at each timestep by an iterative sequence of linear equations
- Linear equation techniques
  - Solve the linear equations at each nonlinear iteration

Circuit Formulation

Represent circuit as an interconnection of branches



Circuit Formulation Constraints

- Branch constituent relations (BCR)
  - Determines the branch current or the branch voltage as a function of circuit variables and environmental variables (time, temperature, etc.)
- Kirchkoff's Voltage Law (KVL)
  - The sum of voltages around each loop must equal zero
- Kirchkoff's Current Law (KCL)
  - The sum of currents at each node must equal zero

Branch Constituent Relations

Current-defined branches

$$I_b = f(V, I, t, T) \qquad I_b = \frac{dQ}{dt}(V, I, t, T)$$

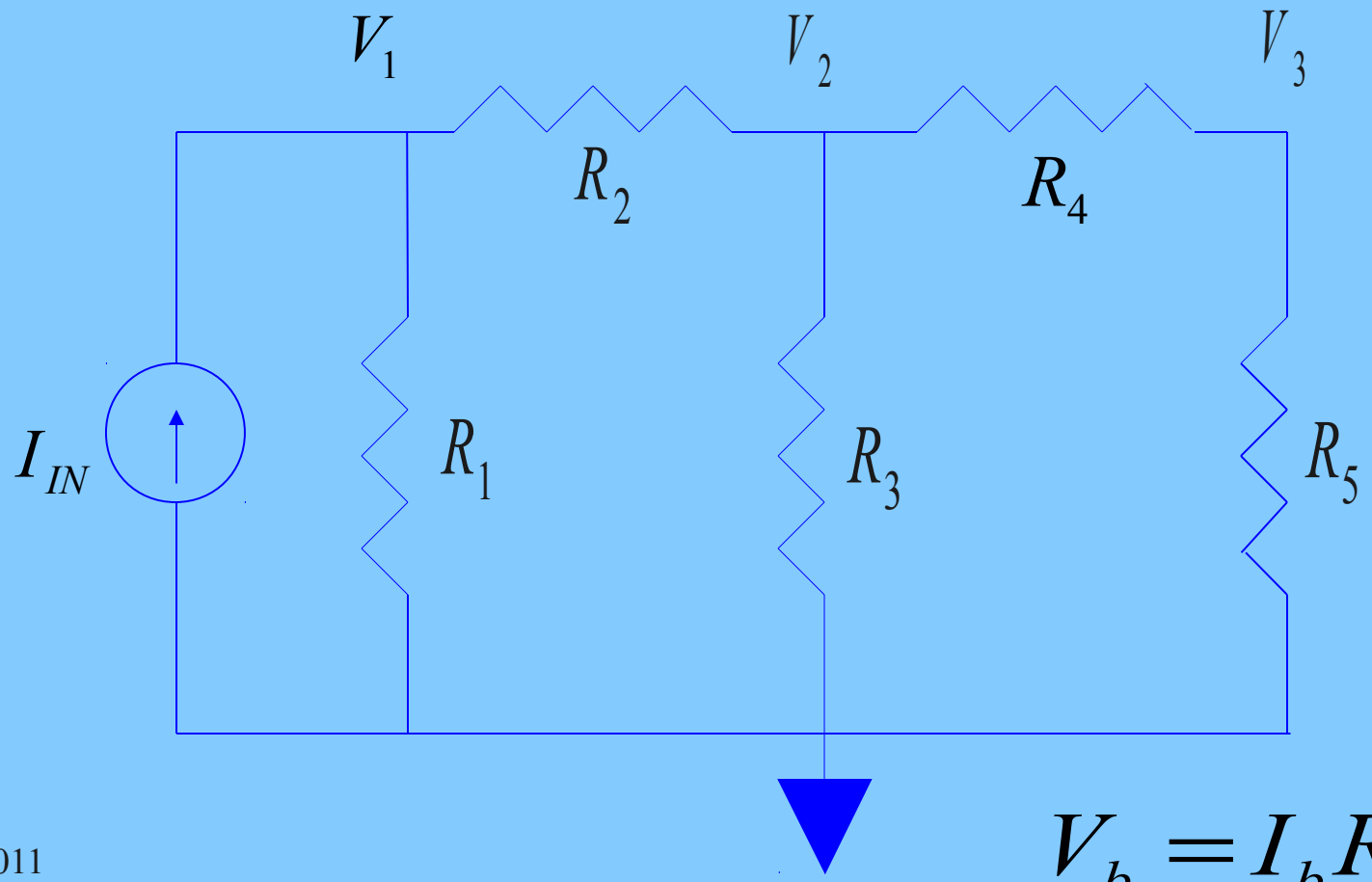
Voltage-defined branches

$$V_b = f(V, I, t, T) \qquad V_b = \frac{d\phi}{dt}(V, I, t, T)$$

Nodal Analysis

- Works only if circuit contains only current-defined branches
- Choose *n-1* node voltages as the unknown vector, automatically satisfying KVL
- Write *n-1* KCL equations for the *n-1* node voltages

DC Nodal Analysis Example



$$V_b = I_b R$$

Kirchkoff's Current Laws

$$\frac{V_1}{R_1} + \frac{(V_1 - V_2)}{R_2} = I_{IN}$$

$$\frac{(V_2 - V_1)}{R_2} + \frac{V_2}{R_3} + \frac{(V_2 - V_1)}{R_4} = 0$$

$$\frac{(V_3 - V_2)}{R_4} + \frac{V_3}{R_5} = 0$$

The DC Nodal Equations

$$\begin{bmatrix} \frac{1}{R_1}+\frac{1}{R_2} & -\frac{1}{R_2} & 0 \\ -\frac{1}{R_2} & \frac{1}{R_2}+\frac{1}{R_3}+\frac{1}{R_4} & -\frac{1}{R_4} \\ 0 & -\frac{1}{R_4} & \frac{1}{R_4}+\frac{1}{R_5} \end{bmatrix} \begin{pmatrix} V_1 \\ V_2 \\ V_3 \end{pmatrix} = \begin{pmatrix} I_{IN} \\ 0 \\ 0 \end{pmatrix}$$

Frequency Domain Analysis

$$I = \frac{dQ}{dt}(V) \quad => \quad i(j\omega) = j\omega \frac{dQ}{dV} v(j\omega)$$

$$V = \frac{d\phi}{dt}(I) \quad => \quad v(j\omega) = j\omega \frac{d\phi}{dI} i(j\omega)$$

Frequency Domain Analysis

- In DC Analysis, there are *n-1* real linear equations to solve
- In AC (Frequency Domain) Analysis, there are *n-1* complex linear equations to solve, or *2(n-1)* real linear equations to solve

The Nodal Equations

$$Y_n V_n = I_s$$

Y_n : Nodal Admittance Matrix

V_n : Node Voltage Vector

I_s : Current Excitation Vector

Modified Nodal Analysis

- Works with circuits with both current-defined branches and voltage-defined branches
- Include in the unknown vector *n-1* node voltages and the current in each voltage-defined branch
- Write *n-1* equations for KCL and a BCR for each voltage-defined branch

Modified Nodal Analysis

$$\begin{bmatrix} Y_n & 0 \\ 1 & Z_{bv} \end{bmatrix} \begin{pmatrix} V_n \\ I_{bv} \end{pmatrix} = \begin{pmatrix} I_s \\ V_s \end{pmatrix}$$

- Z_{bv} : V-def branch impedance matrix
- I_{bv} : V-def branch current vector
- V_s : Voltage Excitation Vector

Linear Solution Techniques

- Solve the Circuit Equations (usually Nodal Equations)
  - Gaussian Elimination
  - LU Factorization
- Circuit Equations are very sparse if chosen properly, so Sparse-Matrix techniques are utilized
- Nodal Equations are well conditioned, other formulations may need pivoting or heuristics

Nonlinear Solution Techniques

$$F(x+\delta x) \approx F(x) + \frac{dF}{dx}(x)\,\delta x$$

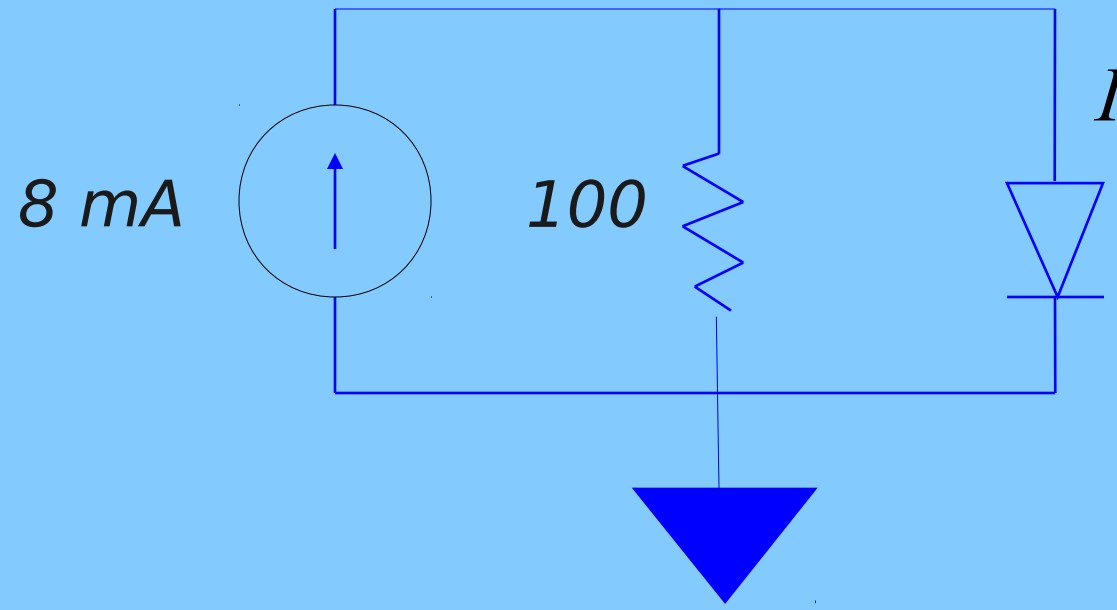
- Each nonlinear BCR is approximated by a linearized equivalent
- Newton-Raphson algorithm linearizes using Taylor Series expansion
- Process continues until $\delta x = 0$

Ideal Diode Example

$$I_d = I_S\left[\exp\left(\frac{V_d}{V_T}\right) - 1\right]$$

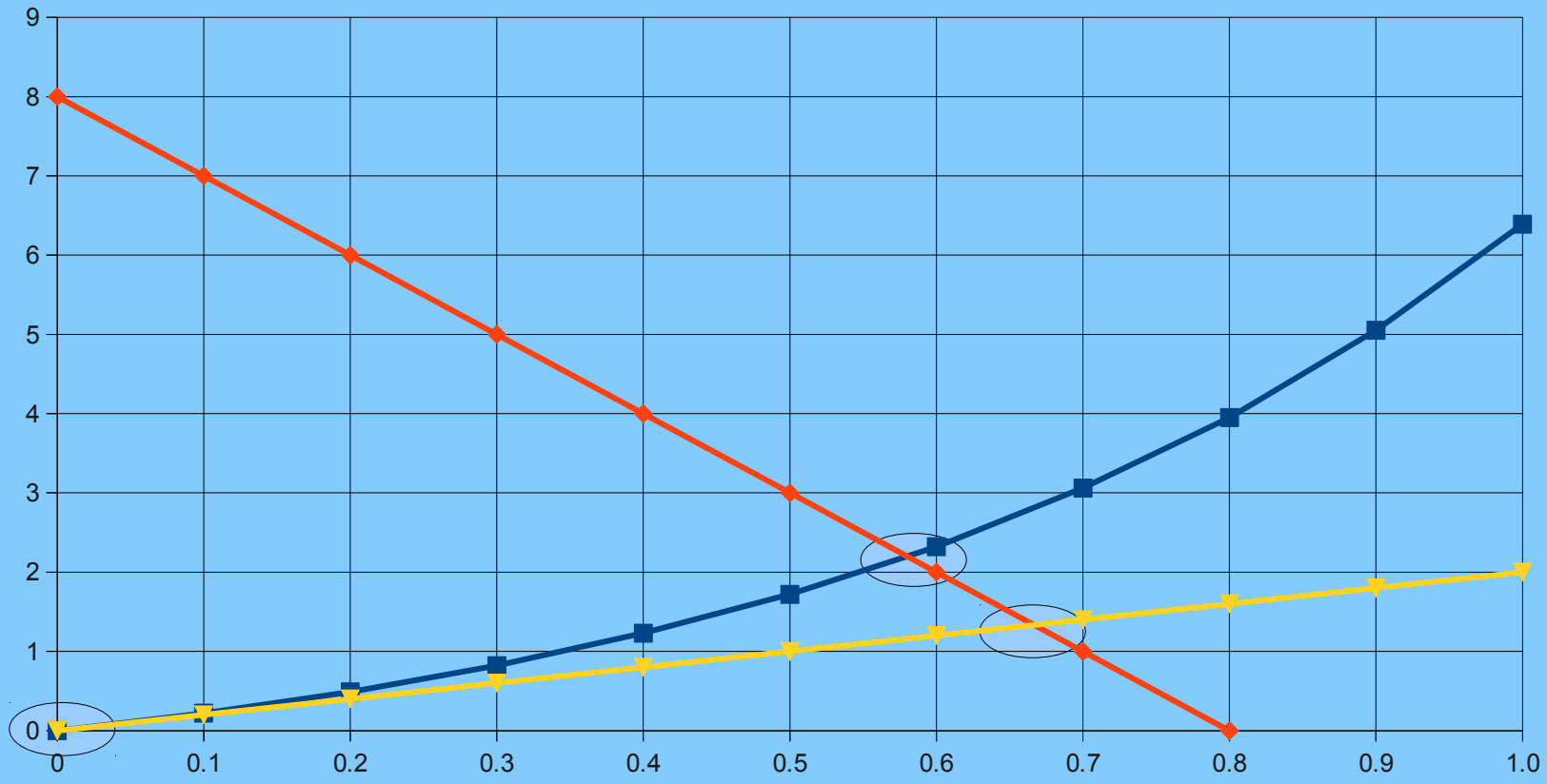
$$I_d + \delta I_d \approx I_S\left[\exp\left(\frac{V_d}{V_T}\right) - 1\right] + \frac{I_S}{V_T}\exp\left(\frac{V_d}{V_T}\right)\delta V_d$$

Ideal Diode Example



$$I_D = \exp(2V_D) - 1$$

Ideal Diode Example



Nonlinear Analysis Issues

- Quadratic Convergence
  - SPICE will always converge, as long as the initial conditions are close enough to the correct solution
- Device models must have continuous derivatives (at least to second order)
- Device models must give reasonable results for unreasonable conditions
- Convergence aids
  - Source stepping, GMIN stepping, pseudo-transient analysis, homotopy

Numerical Integration

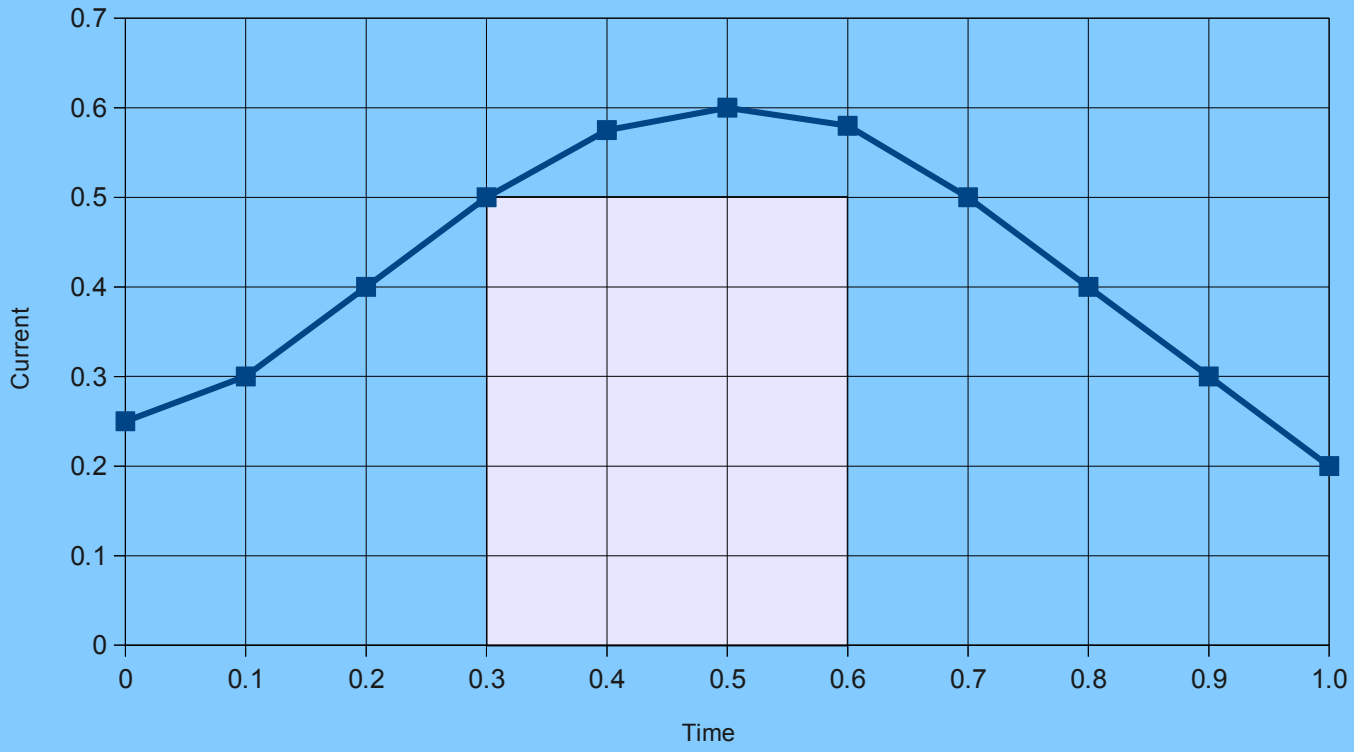
- Approximate the integral function with a polynomial
- Accuracy of the approximation, known as truncation error, depends upon the timestep and the order of the polynomial

Explicit Forward Euler Integration

$$\int_{t}^{t+h} i(t)dt \approx h\, i(t)$$

- Explicit: Integral approximation includes only past function value(s)

Forward Euler Integration

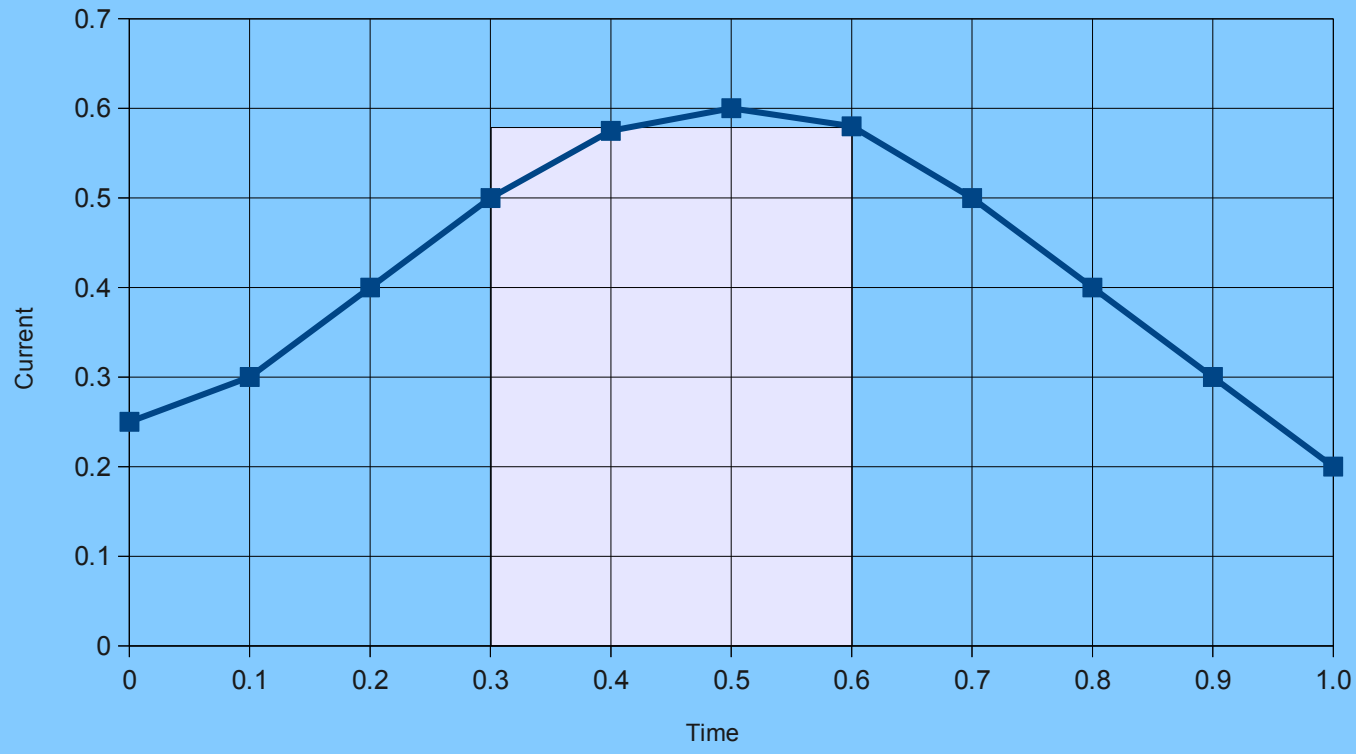


Implicit Backward Euler Integration

$$\int_{t}^{t+h} i(t)dt \approx h\, i(t+h)$$

- Implicit: Integral approximation includes both past and future function value(s)

Backward Euler Integration



Backward Euler Integration for Ideal Capacitor

$$i(t) = C \frac{dv}{dt}(t)$$

$$\int_{t}^{t+h} i(t)\,dt = C\left[v(t+h) - v(t)\right] \approx h\, i(t+h)$$

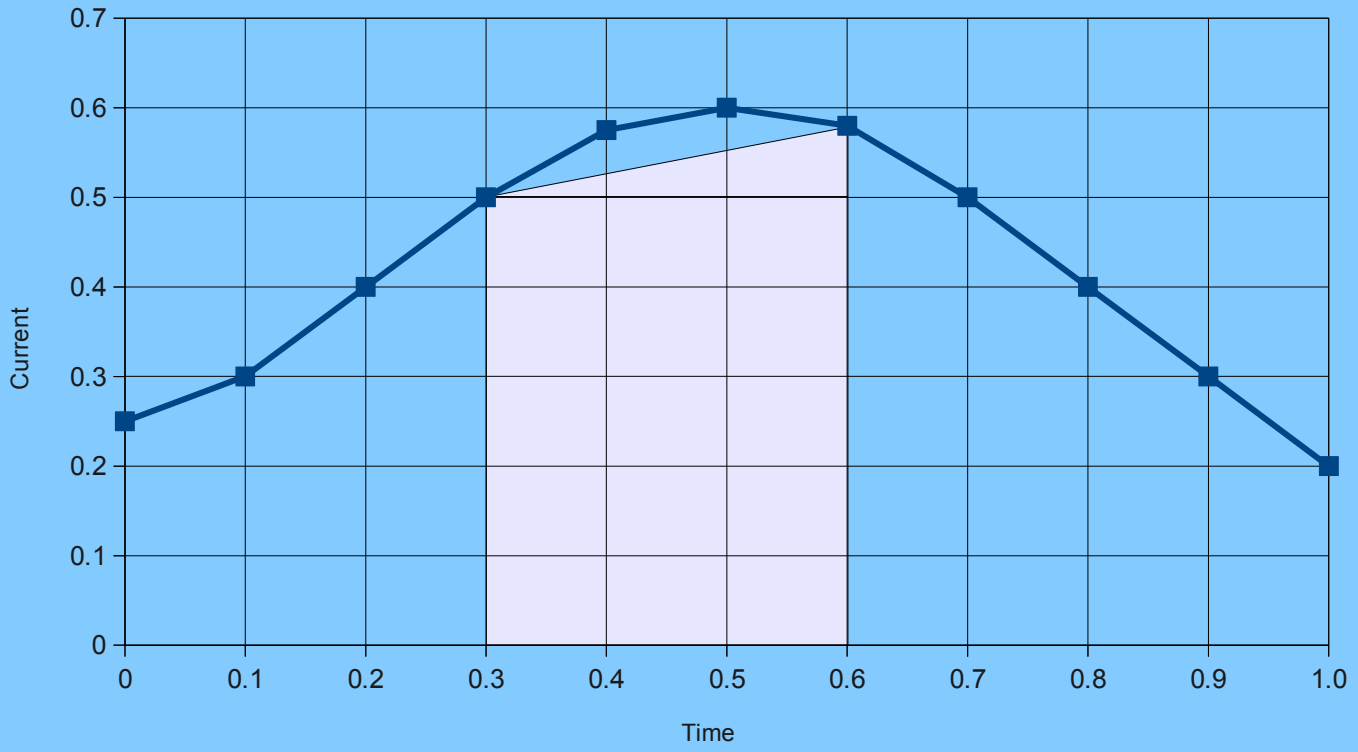
$$i(t+h) \approx \frac{C}{h}\left[v(t+h) - v(t)\right]$$

Trapezopidal Integration

$$\int_{t}^{t+h} i(t)dt \approx \frac{h}{2}\left[i(t)+i(t+h)\right]$$

- Implicit: Integral approximation includes both past and future function value(s)

Trapezoidal Integration



Trapezoidal Integration for Ideal Capacitor

$$i(t) = C\frac{dv}{dt}(t)$$

$$\int_{t}^{t+h} i(t)dt = C\left[v(t+h) - v(t)\right] \approx \frac{h}{2}\left[i(t) + i(t+h)\right]$$

$$i(t+h) \approx -i(t) + \frac{2C}{h}\left[v(t+h) - v(t)\right]$$

Numerical Integration Issues

- Accuracy
  - Higher order yields better accuracy
  - Smaller timestep yields better accuracy
- Stability – Stable circuits should yield stable solutions, and *vice versa*
  - Higher order methods are less stable
- Timestep control
  - Need algorithms to control timestep to maintain accuracy

SPICE Transient Algorithm (Recap)

- Outer loop for each time point (100 -1000)
  - Evaluate sources
  - Calculate integration coefficients
  - Timestep Newton Iteration (2-5 per timepoint)
    - Evaluate equations for each device (most of effort)
    - Construct Nodal Admittance matrix
    - Solve Nodal Admittance matrix
  - Evaluate time point truncation error and select next timestep

SPICE CPU Time

SPICE CPU = Timepoints

* (Newton Iterations / Timepoint)

* (CPU / Newton Iteration / Transistor)

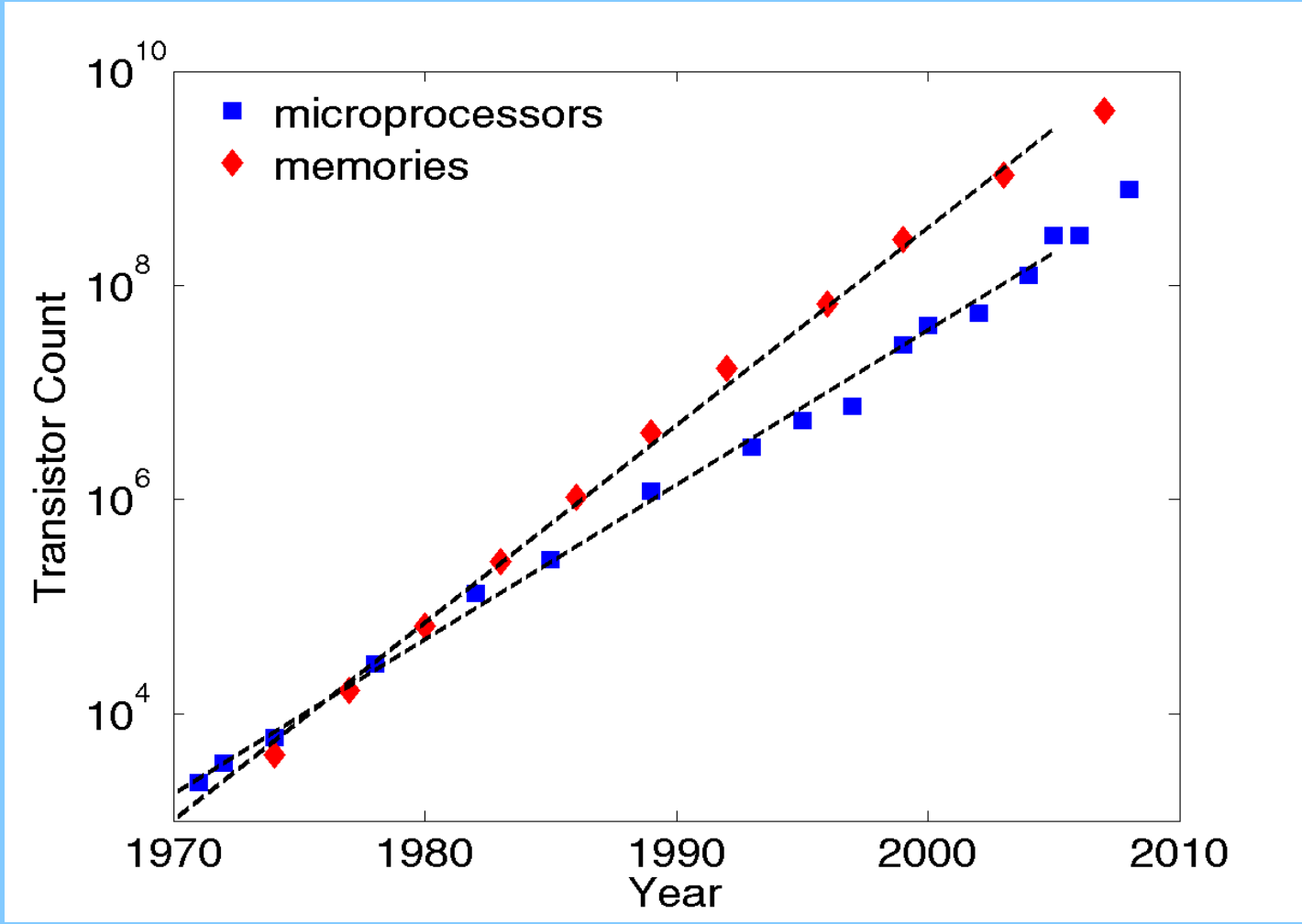
* (Transistors)

Moore's Law

"The complexity for minimum component costs has increased at a rate of roughly a factor of two per year ... Certainly over the short term this rate can be expected to continue, if not to increase. Over the longer term, the rate of increase is a bit more uncertain, although there is no reason to believe it will not remain nearly constant for at least 10 years. That means by 1975, the number of components per integrated circuit for minimum cost will be 65,000. I believe that such a large circuit can be built on a single wafer."

Gordon Moore, *Electronics Magazine*, 1965

Moore's Law


microprocessors
memories
Transistor Count
10^{10}
10^{8}
10^{6}
10^{4}
1970
1980
1990
2000
2010
Year

SPICE Corollary of Moore's Law

- Timepoints increase by at least $\sqrt{2}$ every two years
- Newton Iterations / Timepoint is constant
- CPU / Newton Iteration / Transistor is simply the CPU required to evaluate a device model. This has been relatively constant.
- Transistors increase by at least $\sqrt{2}$ every two years
- This is at least an N^2 Process!!!

SPICE Corollary of Moore's Law

- Fortunately, computer CPUs get faster by $\sqrt{2}$ every two years

  Still...

- SPICE CPU consumption doubles every four years!!!

Segue Into How To Speed Up SPICE

- Reduce device evaluation time (table look up, simpler models)
- Exploit the inherent latency of many circuits (Fast SPICE)
- We can simulate different blocks in different domains (RF, analog, digital) using different algorithms
- Exploit parallel processing techniques