EE5900

Parallel VLSI CAD Algorithms

Lecture 6
RF Circuit Simulation Methods
Zhuo Feng
Neither ac analysis nor pole / zero analysis allow nonlinearities

Harmonic balance incorporates nonlinear effects into ac analysis via quasi-periodic analysis

For linear ckt's, output frequency is the same as input frequency

For nonlinear ckt's, output is quasi-periodic

\[ \cos(\omega_0 t) \rightarrow \text{Nonlinear ckt} \rightarrow \text{Output can contain frequencies other than } \omega_0 \]
Quasi-periodic: a linear combination of sinusoids at the sum and difference frequencies of a finite set of fundamental frequencies and their harmonics

- Fundamental frequencies are input frequencies and sometimes their multiples, (e.g. mixers) sums, differences

- A fundamental frequency can also include a self oscillation rather than an input stimulus
The output of a mixer, which multiplies two signals, is considered quasi-periodic.

In general, responses of nonlinear circuits driven by periodic inputs are quasi-periodic.
RF Mixer

500kHz to 1.36GHz

RF

RF to 100k

LO

Mixer

RF-LO & RF+LO

Filter

► Ex: Transient analysis might require timesteps ~ $10^{-10}$s, but capturing 10 cycles would require a simulation to 100’s of $\mu$s’s
Most analog design specs of interest are best measured in the frequency domain:

- Harmonic distortion
- Power
- Noise
- Frequency
- Gain as a function of frequency
- Impedance as a function of frequency

Lot’s of different degrees of nonlinearity

A mixer is an example of a highly nonlinear analog ckt for which we seek the frequency response
Two well-known approaches for solving for periodic steady state solution of nonlinear ckt's:

1. Time Domain: Shooting Methods [1]


Time domain approaches:

\[ \ddot{x} = f(\dot{x}(t), t) \text{ for } [0, T] \]

Search for the initial condition for the system, \( \ddot{x}(0) \), such that the solution at time T equals the solution at time t=0:

1. Simulate with initial conditions
2. Compare result at time T with initial conditions
3. Adjust initial conditions
4. Repeat 1 until convergence

The challenge is in Step 3
Harmonic Balance Approach

Nonlinear Diff. Equations

Convert to nonlinear equation of Fourier coefficients

“Approximate” steady state solution

- It’s an approximation since we consider only the first several dominant sinusoids
- Harmonic balance refers to balancing the current between linear and nonlinear portions
We can’t compute Fourier coefficients for general nonlinearities.

But there have been some attempts at doing so for simple nonlinearities. 


General nonlinearities require an FFT to convert from freq. to time domain and vice versa during nonlinear iterations.
Simple example: nonlinear LC ckt with sinusoidal input current

\[ \frac{d^2v}{dt^2} + \frac{v}{LC(v)} = A_1 \cos(\omega_0 t) \]

\[ C(v) = \frac{1}{\lambda^2 + \mu v^2}; L = 1 \]

2\textsuperscript{nd} order nonlinear diff. equation:

\[ \ddot{v} + \lambda^2 v + \mu v^3 = A_1 \cos(\omega_0 t) \]
The degree of nonlinearity depends on $\mu$ and $\lambda$.

With nonlinearities, the (single sinusoid input) steady state solution will be of the form (Fourier Series):

$$x(t) = \sum_{k=0}^{\infty} a_k \cdot \cos(k\omega_0 t)$$

For a linear ckt the solution would be a fct of $\omega_0$ only!

It can be shown that the response for this simple example is an even fct with half-wave symmetry:

$$a_k = 0 \quad \text{for} \quad k = 0, 2, 4 \ldots$$
- We can’t keep an infinite number of terms for harmonic balancing.

- The less severe the nonlinearities the faster the higher order coefficients get smaller.
  - For a linear system, all higher order terms are zero.

- For our simple example we’ll consider using just the first two harmonics:

  \[ \hat{x}(t) = a_1 \cos(\omega_0 t) + a_3 \cos(3\omega_0 t) \]
\[ \mu (a_1 \cos(\omega_0 t) + a_3 \cos(3\omega_0 t))^3 + \lambda^2 (a_1 \cos(\omega_0 t) + a_3 \cos(3\omega_0 t)) - \omega_0^2 a_1 \cos(\omega_0 t) - 9 \omega_0^2 a_3 \cos(3\omega_0 t) = A_1 \cos(\omega_0 t) \]

▶ New harmonics are created by the nonlinear capacitor:

\[ \omega_0, 3\omega_0 \Rightarrow 5\omega_0, 7\omega_0, 9\omega_0 \]
From Fourier analyses we know that sinusoids at different harmonics are orthogonal.

So we can equate all of the $k\omega_0$ coefficients.

This yields 5 equations in terms of 2 unknowns.

We can solve just the first 2 equations exactly, or try to perform a least-squares solution for all of them.

Generally we focus on only those in our assumed solution form.
Equating the $\omega_0$ and $3\omega_0$ in terms results in a nonlinear problem in terms of Fourier coefficients

\[
\cos(\omega_0 t): \quad \frac{1}{4} \mu \left(3a_1 a_3 + 6a_1 a_3^2 + 3a_1^3\right) + \left(\lambda^2 - \omega_0^2\right)a_1 = A_1
\]

\[
\cos(3\omega_0 t): \quad \frac{1}{4} \mu \left(3a_3^3 + 6a_1^2 a_3 + a_1^3\right) + \left(\lambda^2 - 9\omega_0^2\right)a_3 = 0
\]

Solve for $a_1$ and $a_3$ by N-R or some similar means.
Solution:

\[ \hat{x}(t) = a_1 \cos(\omega_0 t) + a_3 \cos(3\omega_0 t) \]

When we plug this into:

\[ \ddot{x} + \lambda^2 x + \mu x^3 = A_1 \cos(\omega_0 t) \]

There is an error because we ignored higher order terms that were created by this assumed solution

\[ \cos(5\omega_0 t): \quad \frac{3}{4} \mu(a_1^2 a_3 + a_1 a_3^2) \neq 0 \]
\[ \cos(7\omega_0 t): \quad \frac{3}{4} \mu a_1 a_3^2 \neq 0 \]
\[ \cos(9\omega_0 t): \quad \frac{1}{4} \mu a_3^3 \neq 0 \]
These terms represent an error in our solution; we’ve actually solved:

\[ \hat{x} + \lambda^2 \hat{x} + \mu \hat{x}^3 = A_1 \cos(\omega_0 t) + \]
\[ A_5 \cos(5\omega_0 t) + \]
\[ A_7 \cos(7\omega_0 t) + \]
\[ A_9 \cos(9\omega_0 t) \]

These terms represent the dominant error component for such an approximation.

Clearly our solution form has to contain a sufficient number of coefficients so that errors are acceptable.
- This simple example could be solved via symbolic algebra

- In general we require a discrete Fourier transform

- Several ways of solving general problem using DFT, one of which is Gauss-Jacobi Newton
For our example:

1. Assume an initial guess for $a_1$ and $a_3$

2. Compute $\hat{x}(t)$ and $f(t)$ at several timepoints

   $$f(t) = \ddot{x} + \lambda x + \mu x^3 - A \cos(\omega_0 t) \Rightarrow \text{error}$$

3. Convert $f(t)$ to frequency domain via DFT
   - - error coefficients tell us how to adjust $a_1$ and $a_3$

4. Repeat 1 thru 3 in G-J manner
At each iteration we have a guess for $a_1$ and $a_3$, hence a complete waveform assumption at each node.

Using Newton part of iteration process:

- Find linearized model for nonlinear $C$.

Alternatively without Newton part -- splitting relaxation:

- Weaker convergence properties.
- Works for slight nonlinearities.
In general formulation, circuit unknowns are in the form of Fourier Coefficients of node voltages, etc.

Dimension of the problem increases by a factor of $M$, the number of frequencies considered.

e.g. KCL at a node

$$\sum_{m} I_m(t) = \sum_{k} e^{jk \frac{2\pi}{T} t} \sum_{m} I_{m,k} \Rightarrow \sum_{m} I_{m,k} = 0$$
Matrix-form HB equations look like the following

\[ I(V) + \Omega Q(V) + YV = U \]

- **Y**: Same as in AC analysis admittance matrix, but evaluated at multiple frequencies here

- **\( \Omega \)**: Frequency-domain differentiation operator to convert nonlinear cap. currents from charges in steady-state
- KCL/KVL equations are enforced for each frequency

- But due to the nonlinearities, there are couplings between different frequency components

  - In general, cannot solve for the solution individually for each frequency

\[
\begin{align*}
\text{freq. 1:} & \quad F_1(\cdot) [\begin{bmatrix} V(\omega_1) \\ F_2(\cdot) \\ \cdots \\ F_m(\cdot) \end{bmatrix}] + [\begin{bmatrix} U(\omega_1) \\ U(\omega_2) \\ \cdots \\ U(\omega_m) \end{bmatrix}] = 0 \\
\text{freq. 2:} & \\
\text{freq. m:} &
\end{align*}
\]
- Another simple example

![Circuit Diagram]

\[ I_1 = A_1 \cos(\omega_0 t) \]

\[ C_1(v) = f(v) \]

- If we consider 4 frequencies, the problem dimension is a 4x2 (complex equation formulation)
Matrices look like the following:

- frequency: major index, node: minor index

\[
F(\cdot) = \begin{bmatrix}
V_1(0) \\
V_2(0) \\
V_1(\omega_0) \\
V_2(\omega_0) \\
V_1(2\omega_0) \\
V_2(2\omega_0) \\
V_1(3\omega_0) \\
V_2(3\omega_0)
\end{bmatrix} = \begin{bmatrix}
0 \\
0 \\
\frac{A_1}{2} \\
0 \\
0 \\
0 \\
0 \\
0
\end{bmatrix}
\]

- KCL at DC for node 1
- KCL at DC for node 2
- KCL at \(\omega_0\) for node 1
- KCL at \(\omega_0\) for node 2

\[\cdots\]

\[\cdots\]
- Stamping of linear elements – Admittance matrix $Y$
- Resistor
  - $I_R(f) = GV_R(f)$ same for every frequency

![Admittance Matrix Diagram]

\[ V_i(\omega_1) \]
\[ V_j(\omega_1) \]
\[ V_i(\omega_2) \]
\[ V_j(\omega_2) \]
\[ \cdots \]
\[ V_i(\omega_m) \]
\[ V_j(\omega_m) \]
Linear Capacitors

Just as in AC analysis

\[ I(\omega) = j\omega \cdot C \cdot V(\omega) \]

\[ y(\omega) = j\omega \cdot C \]

\[
\begin{array}{cccc}
  i & j & i+n & j+n \\
  y(\omega_1) - y(\omega_1) & y(\omega_1) & y(\omega_2) - y(\omega_2) & -y(\omega_2) y(\omega_2) \\
  j+n & i+(m-1)*n & j+(m-1)*n \\
  y(\omega_m) - y(\omega_m) & -y(\omega_m) y(\omega_m) \\
\end{array}
\]

\[ V_i(\omega_1) \]
\[ V_j(\omega_1) \]
\[ V_i(\omega_2) \]
\[ V_j(\omega_2) \]
\[ \ldots \]
\[ V_i(\omega_m) \]
\[ V_j(\omega_m) \]

Node i

Node j
Linear Inductors

Just as in AC analysis

\[ V(\omega) = j\omega \cdot L \cdot I(\omega) \quad z(\omega) = j\omega \cdot L \]

Using admittance will cause problem at DC – using modified nodal analysis

For \( p \)-th frequency

\[ \begin{array}{ccc}
  i+(p-1)*n & j+(p-1)*n & k+(p-1)*n \\
  i+(p-1)*n & 1 & -1 \\
  j+(p-1)*n & -1 & -j\omega_p L \\
  k+(p-1)*n & 1 & -j\omega_p L \\
\end{array} \]
Independent Sources

In harmonic balance, independent sources are specified on a frequency basis:

\[ V = V(\omega_p) e^{j\omega_p t} \]

Stamping of Vsources very similar to inductors, but also need to stamp the VHS

For \( p \)-th frequency:

\[
\begin{array}{ccc}
  i+(p-1)n & j+(p-1)n & k+(p-1)n \\
  \hline
  1 & & 1 \\
  -1 & -1 & \\
  \hline
\end{array}
\]

Node i

Node j

Branch current

\[ V(\omega_p) \]
**Evaluation of Nonlinear Devices**

- **Freq->Time:** terminal voltage waveforms
- **Time domain:** evaluate current (derivative) waveforms
- **Time->Freq:** currents(derivatives) in freq. domain

- In Matlab FFT/IFFT transform a series of discrete points back and forth between time and frequency domain

---

6.31 Z. Feng  MTU EE5900
Nonlinear HB equations can be solved using Newton methods as usual.

Must deal with a potentially dense Jacobian matrix:
- Costly to form and difficult to solve.

Various alternatives have been tried:
- Splitting method, Gauss-Jacobi-Newton, Successive-Chord.

Splitting method is the easiest to implement, but will generally work only for weakly nonlinear circuits.
■ Splitting

- New solution updates are based on linear problem solution to the linear network

- Nonlinear elements only generate excitations to the linear network

\[ YV^{i+1} = -I(V^i) - \Omega Q(V^i) + U \]
Previous example

\[ C_1(v) = f(v) \]

\[ I_1 = A_1 \cos(\omega_0 t) \]

\[ Y \text{ due to the linear elements} \]

\[
Y = \begin{bmatrix}
G_1 + G_2 & -G_2 \\
-G_2 & G_2 \\
G_1 + G_2 & -G_2 \\
-G_2 & G_2 + j\omega_0 C_2 \\
-G_2 & G_2 + j2\omega_0 C_2 \\
G_1 + G_2 & -G_2 \\
-G_2 & G_2 + j3\omega_0 C_2
\end{bmatrix}
\]

\[
U = \begin{bmatrix}
0 \\
0 \\
0 \\
A_1 / 2 \\
0 \\
0
\end{bmatrix}
\]
Splitting Iteration

- Start from some initial guess and set \( i = 0 \)

\[
X^0 = [V_1^0(0), V_2^0(0), V_1^0(\omega_0), V_2^0(\omega_0), V_1^0(2\omega_0), V_2^0(2\omega_0), V_1^0(3\omega_0), V_2^0(3\omega_0)]^T
\]

- Until convergence
  - Convert current nonlinear cap. terminal voltage from freq. to time domain via inverse FFT:

\[
V_{c,t} = [v_{c1}^{t_1}, v_{c2}^{t_2}, \ldots, v_{ct}^{t_7}]
\]

- Evaluate the nonlinear charge at each sample point in time:

\[
Q_{c,t} = [q_{c1}^{t_1}, q_{c2}^{t_2}, \ldots, q_{ct}^{t_7}]^T = [f(v_{c1}^{t_1}), f(v_{c2}^{t_2}), \ldots, f(v_{ct}^{t_7})]^T
\]
\(\nabla\) Convert the charge back to frequency via FFT:

\[
Q_{c_f} = [q_c(0), q_c(\omega_0), q_c(2\omega_0), q_c(3\omega_0)]^T
\]

\(\nabla\) Form the input excitations to the linear network:

\[
U_{\text{Non}} = -\Omega \cdot Q = \begin{bmatrix}
0 & 0 & j\omega_0 & j\omega_0 & j2\omega_0 & j2\omega_0 & j3\omega_0 & j3\omega_0
\end{bmatrix}
\begin{bmatrix}
q_c(0) & 0 & q_c(\omega_0) & 0 & q_c(2\omega_0) & 0 & q_c(3\omega_0) & 0
\end{bmatrix}
\]

\(\nabla\) Update the solution by solving the linear network:

\[
Y \cdot X^{i+1} = U + U_{\text{Non}}
\]
More general and robust way is to use Newton method

True Jacobian is formed and the resulting linearized network is solved at each iteration

To form the Jacobian, compute derivatives of Fourier coefficients of (quasi)periodic current waveforms w.r.t Fourier coefficients of controlling node voltages
  ► Somewhat messy for nonlinear elements

Can form the equations in terms of real and imaginary parts of Fourier coefficients – using real solvers
We talked about the single-tone Harmonic Balance last time

Since the frequencies used were a set of harmonics, transformations between time and frequency could be done efficiently via FFT/IFFT

In reality, we often need to solve Harmonic Balance problems with multiple-tone excitations

- Input tones can be arbitrary
- Often input tones are closely spaced: e.g. intermodulation tests
Frequency-truncation makes computation tractable by working on a finite number of frequencies – physical systems have limited bandwidth.

In the single-tone case, the frequency set used in the simulation is chosen to include sufficiently number of harmonics.

For multi-tone inputs, frequency truncation is done to limit not only the number of harmonics for each individual tone, but also the integer linear combinations of different tones.

People commonly employ “box” or “diamond” truncation.
2-tone Box Truncation

Diamond truncation is usually considered more efficient
Directly converting a multi-tone problem to a single-tone problem might not always be possible or efficient.

Example: 2-tone intermodulation test of a low noise amplifier

- $f_1=930\text{MHz}$, $f_2 = 930.8\text{MHz}$
- $3^{rd}$ intermodulations @ $|2*f_1-f_2|=929.2\text{MHz}$ or $|2*f_2-f_1|=931.6\text{MHz}$
- $\text{GCD}(f_1, f_2) = 0.4\text{MHz} \rightarrow$ if choose $f_0=0.4\text{MHz}$ as the base tone, then $f_2 = 2329*f_0$, need to simulate the circuit using more than 4,000 frequencies!!
Recall that for HB problem

The Jacobian is

\[ I(V) + \Omega Q(V) + Y \cdot V = U \]

\[ J = Y + \Gamma \cdot G \cdot \Gamma^{-1} + \Omega \cdot \Gamma \cdot G \cdot \Gamma^{-1} \]

\[ \begin{bmatrix}
    \frac{\partial f}{\partial x} \bigg|_{t_1} \\
    \frac{\partial f}{\partial x} \bigg|_{t_2} \\
    \vdots \\
    \frac{\partial f}{\partial x} \bigg|_{t_s}
\end{bmatrix} \quad \begin{bmatrix}
    \frac{\partial q}{\partial x} \bigg|_{t_1} \\
    \frac{\partial q}{\partial x} \bigg|_{t_2} \\
    \vdots \\
    \frac{\partial q}{\partial x} \bigg|_{t_s}
\end{bmatrix} \]
The exact Jacobian is usually dense

- e.g.

\[
\Gamma \cdot G \cdot \Gamma^{-1} = \begin{bmatrix}
G_1 & G_2 & \cdots & G_s \\
G_s & G_1 & \ddots & \vdots \\
\vdots & \ddots & \ddots & \vdots \\
G_2 & \cdots & G_s & G_1
\end{bmatrix}
\]

- \([G_1, G_2, \cdots, G_s]^T\) is DFT of \([g_1, g_2, \cdots, g_s]^T\), where \(g_i = \frac{\partial f}{\partial x}\bigg|_{t_i}\)

Simple splitting method, Guass-Jacobi-Newton and Successive-Chord cope with the dense linear problem by utilizing a pruned (approximated) sparse Jacobian

People have also tried Krylov-subspace implicit iterative methods
In Krylov-subspace methods, linear problem solution is linked to an optimization problem (least-square) in a Krylov subspace.

- Jacobian matrix needs not to be formed explicitly, only matrix-vector product with arbitrary vector is required.
- Examples of these type of methods are Generalized Minimum Residual (GMRES) method and Quasi Minimum Residual (QMR) method.
- In matlab, commands `gmres` and `qmr` are available for these two methods.
GMRES: solve $Ax = b$

1. Choose initial guess $x^0$
2. Set $k = 1$ and set the initial search direction $p^0 = b - Ax^0$
3. do {
   ▼ Compute the new direction $p^k = Ap^{k-1}$
   ▼ Orthogonalize $p^k$ with all previous directions \{\(p^0, \ldots, p^{k-1}\)}
   ▼ Doing a line search to minimize \( \| r^k \| = \| b - Ax^k \| \),
     where \( x^k = x^{k-1} + a_k p^k \)
4. } until \( \| r^k \| < \) tolerance
To apply GMRES/QMR to Harmonic Balance, we only need to compute product of the Jacobian and any arbitrary vector.

No need to explicitly form the (dense) Jacobian.

Matrix-vector products can be computed efficiently by using FFT and exploiting the sparsity of the problem.

■ Recall the Jacobian is

\[ J = Y + \Gamma \cdot G \cdot \Gamma^{-1} + \Omega \cdot \Gamma \cdot G \cdot \Gamma^{-1} \]

■ Matrix-vector product \( J \cdot V \) has terms look like

\[
\begin{bmatrix}
\frac{\partial f}{\partial x} \bigg|_{t_1} \\
\frac{\partial f}{\partial x} \bigg|_{t_2} \\
\vdots \\
\frac{\partial f}{\partial x} \bigg|_{t_s}
\end{bmatrix}
\]

Inverse FFT

\( \Gamma^{-1} \cdot V \)

A vector in time

FFT: get back to frequency
However, there is no free lunch: use of implicit method often requires proper preconditioning.

The inverse of a preconditioning matrix $M$ ideally should be close to $A^{-1}$:
$$ (M^{-1}A)x = M^{-1}b $$

At the same time $M$ has to be sparse and easy to factorize, otherwise defeating the purpose of implicit solution.

For strongly nonlinear circuits, simple block-diagonal preconditioner does not work very well.

Multi-level preconditioning has been tried to improve the efficiency.

Iterative Krylov-subspace methods can also be used for time-varying systems under harmonic balance context.

The time-variance of circuit parameters causes the circuit response to be a function of the input launch time.

The special case of linear periodically time-varying (LPTV) systems are encountered in common large-signal small-signal type analyses in analog circuits.

Circuit parameters of a LPTV system vary periodically and consequently input signal frequencies are "linearly" translated by multiples of the system’s frequency.
Example: RF mixer

- Apply only LO, solve the circuit response using single-tone HB
- Upon convergence G, C, Y matrices tell us the circuit sensitivity about the periodically varying operating condition due to LO
- To solve frequency translations of RF input (LPTV transfer functions), linear problems with a structure similar to HB need to be solved
- Can apply essentially same Krylov-subspace techniques

In autonomous circuits such as oscillators, oscillation frequency is not known.

The problem is formulated by adding oscillation frequency as one extra unknown.

We also need to have one more equation: can be done e.g. by enforcing the signal in the circuit has a real phasor.

The circuit response as well as oscillation frequency is solved iteratively using Newton method, for example.
Some Comments on different methods

Shooting Method (Mixed Frequency-Time Method)

- Employed by Cadence SpectreRF
- Primarily a time-domain method, convergence property is enhanced by robustness of transient analysis used in the inner iteration loop
- Converge fast if the state-transition is near linear – strong nonlinearities can be hidden within the shooting period
- Cannot easily handle distributed devices
Harmonic Balance

- Employed by Agilent ADS
- Converge to exact solution if the circuit is linear
- Very accurate for weakly nonlinear circuits
- Need to use many frequencies to accurately represent a highly nonlinear waveform -- less efficient for strongly nonlinear circuits
- Convergence property is less satisfying
- Can easily handle distributed elements in terms of transfer functions