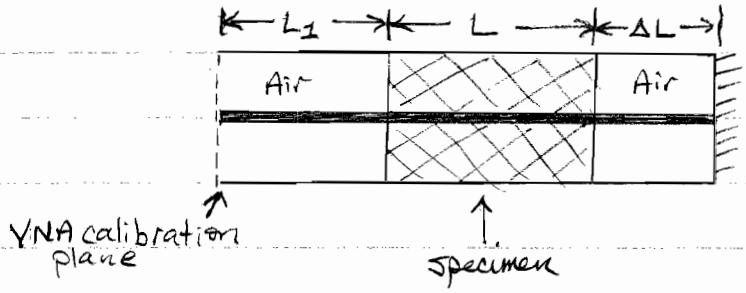
# Extracting Permittivity from Measured Data



For the one-port measurement fixture, let's image a specimen has been inserted, the VNA calibrated to the end of the cable, and the $S_{11}$ has been measured over a frequency range, $S_{11}^m$.




First, translate VNA cal. plane to front face of sample:

$$S_{11}^{m\prime} = S_{11}^{m} e^{+2\gamma_0 L_1} \tag{1}$$

An analytical expression for the reflection coeff. at the front face of the specimen, $\Gamma = C_1$, was determined in the previous lecture.

To determine $\epsilon = \epsilon' - j\epsilon''$ of the sample at a freq., we make the equality

$$S_{11}^{m\prime} = \Gamma \tag{2}$$

This is a nonlinear equation in $\epsilon$, which has no explicit solution for $\epsilon$.

Instead, we will use numerical methods to vary $\epsilon$ until (2) is satisfied: at least sufficiently accurately so.

# Root Finding

One way of approaching the solution to (2) is to find a value for $\epsilon$ that satisfies

$$S_{11}^{m\prime} - \Gamma = 0 \qquad (3)$$

This is called <u>root</u> finding.

There are a number of algorithms for root finding. Some overall observations are that:

(1) Root finding invariably proceeds by iteration in locating a solution to (3), except for linear problems.

(2) For equations such as (3), if only one real unknown quantity, excellent methods for finding solution, or solutions if more than one real unknown, there are <u>no</u> good general methods to find a sol'n.

Methods for root finding include:

- Bisection Method.
- Newton-Raphson method (also called Newton's method).
- Secant method.

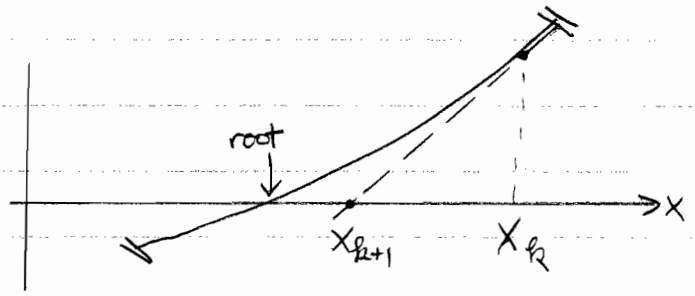
<u>Bisection Method</u>: $f(x) = 0$.

- ✓ $f$ is continuous
- ✓ Requires two initial guesses $f(a)$ & $f(b)$ that must be of opposite sign, i.e., the root is bracketed.

- ✓ If interval contains a root, algorithm will find it.
- ✓ Slow algorithm
- ✓ If more than one root in interval $[a, b]$, will find one root
- ✓ If interval contains a singularity, this method converges onto singularity.

- **Newton-Raphson Method**: $f(x) = 0$.

This method finds the slope of $f$ at the current point $x_k$, and uses the zero of this tangent line as the next evaluation point, $x_{k+1}$.




Starting with the forward-difference definition of the first derivative

$$\frac{f(x_k) - f(x_{k+1})}{x_k - x_{k+1}} = f'(x)\Big|_{x_k} \equiv f'(x_k) \qquad (4)$$

Referring to the sketch, we set $f(x_{k+1}) = 0$ in (4) and solving for $x_{k+1}$ we find

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} \qquad (5)$$

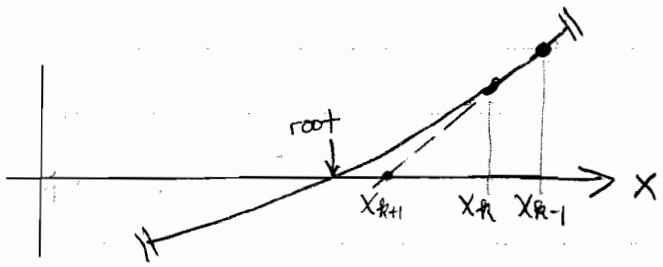
This is the algorithm for iteratively determining root.

✓ One initial guess.

✓ Method requires derivative of the fct.
✓ Fast convergence.
✓ May not converge if initial guess too far from root (to local minima) or other reasons!
✓ Readily generalized to higher dimensional root finding.
✓ con

- **Secant Method**: $f(x) = 0$.

Very similar to Newton-Raphson, except using numerical differentiation (i.e. finite difference approx. to derivative).




From (5): ~~&~~ using backward diff. approx.

$$X_{k+1} = X_k - \frac{f(X_k)}{\frac{f(X_k) - f(X_{k-1})}{X_k - X_{k-1}}} \qquad (6)$$

✓ Two initial guesses.
✓ Good convergence, but slower than Newton-Raphson.
✓ May not converge if $f$ not sufficiently smooth or inital guess too far from root.

These are all methods that can work well for one-dimensional root finding. For multi-dimensional root finding, $f(\bar{x}) = 0$, there are no good methods. The secant method and Newton-Raphson method can

be extended to multi-dimensional root finding. Often find that initial guess(es) are important to locating the root you're searching for. May find other sol'n's, perhaps not physical. May not converge to a sol'n at all.

---

## Function Optimization

Another approach to extracting material parameters from measured data is to approach (3) from an optimization approach. That is, we seek to determine the $\epsilon$ that minimizes

$$|S_{11}^{m\prime} - \Gamma| = 0 \qquad (7)$$

or maximizes $\quad -|S_{11}^{m\prime} - \Gamma| = 0$ .

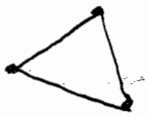
In this material parameter extraction context, the objective of root finding and function minimization are the same. Their approaches couldn't be more different.

### Methods of Optimization

- Exhaustive search – Evaluate the function on a fine grid in the multi-dimensional space of the unknowns. Permittivity – 2, Permeability – 2, Sample location – 1, Sample thickness – 1, etc.

Very slow method, and computational expensive.

- Downhill Simplex Method (not to be confused with the Simplex Method in linear programming). A simplex is the most elemental geometrical object that can be formed from N+1 points in an N-dimensional space.

| 1-space | 2-space | 3-space | 4-space |
|---|---|---|---|
| N=1, N+1=2 | N=2, N+1=3 | N=3, N+1=4 | N=4, N+1=5 |
|  |  |  | ? |
| Line | Triangle | Tetrahedron | |

The algorithm moves "downhill" in an N-dim. space moving the "highest" pt. through the "opposite" face to a lower pt (reflections) while preventing "degeneracy" called in which a dimension of the simplex is lost when two or more pts. of the simplex coincide.

- ✓ Fairly straightforward algorithm. Easy to code.
- ✓ No analytical derivatives needed
- ✓ Slow
- ✓ Will get stuck on local minima. This is a problem shared by all downhill methods. The algorithm is able to minimize the fct + converge but this solution is not the global minimum: The best answer

- Line Minimization Methods - Steepest Decent Method, Conjugate Gradient method.

The general idea is to start at some point $\bar{P}$ in the N-dimensional space of unknowns and perform a succession of function minimizations along lines in this N-dim space using 1-D minimization algorithms.

How one chooses the directions of the line minimizations varies from one method to another.

- Nature-Based Methods — These are methods based on physical and biological processes that have produced — or tend to produce — optimal outcomes.

— Simulated Annealing (SA): If a liquid is cooled at a sufficiently slow rate, the material will solidify into a crystalline lattice solid; which is the state of minimum energy. This is called annealing. If liquid cooled too quickly, material doesn't reach lowest energy but rather winds up in a polycrystalline or amorphous state.

Metropolis et al (ca 1953) developed a mathematical model of the annealing process: simulated annealing.

A "thermodynamic" system will change configuration from energy $E_1$ to $E_2$ w/ probability

$$P = e^{-\frac{(E_2 - E_1)}{kT}} \qquad (8)$$

If $E_2 < E_1$ then $P > 1$ and the change will be made. If not, then the change of state may be made w/ prob. $P$.

In our problem of extracting $\epsilon$ from $S_{11}^{m\prime}$ in (7), our "energy" fct might be:

$$E(\epsilon', \epsilon'') = \left| S_{11}^{m\prime} - \Gamma(\epsilon', \epsilon'') \right| \qquad (9)$$

Create random changes in $\epsilon' \& \epsilon''$ within a predetermined range. Evaluate $E$ in (9). Accept this new "state" (i.e., $\epsilon' \& \epsilon''$) if a random # generated is $< P$. Can "jump" out of local minima

The temperature $T$ evolves over time according to a predetermined method. Must start high enough to "melt" the system, then cool slowly enough to reach minimum energy state where from (9)

$$E(\epsilon', \epsilon'') \approx 0 \qquad (10).$$

If done correctly, can reach global or nearly global minima. Much practice on Range of $T$,

9/10

ref: Haupt & Haupt

- Genetic algorithm (GA) - This algorithm is based on biological processes that tend towards improving a species. GA employs natural selection & mutation concepts for optimization!

Quick outline of simple binary encoded GA:

- A chromosome = array of variables ($N_{var}$): $[\epsilon', \epsilon'']$
- Genes = binary version of quantized variables $[\epsilon' \; \epsilon'']$
  s.t. chromosome = [10110110 (gene 1) 00101101 (gene 2)]

  Chromosome has a total of $N_{gene} \times N_{var} = N_{bits}$

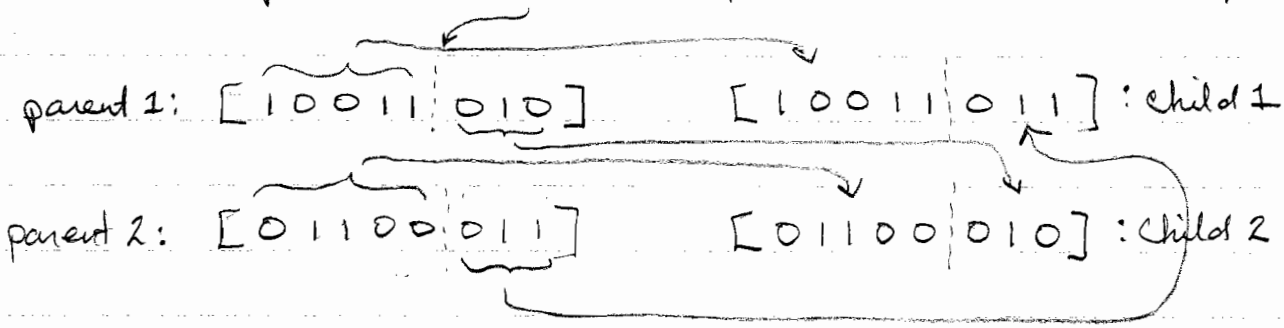
- Choose number of "individuals" that will exist in the population $N_{pop}$. Fill chromosomes w/ genes randomly chosen as 1 or 0.

- Compute "cost fct" associated w/ each individual (chromosome). Rank order from smallest to largest.

- Natural selection is implemented by keeping only a percentage of the population - the "best".
  $N_{keep} = X_{rate} \times N_{pop}$ (e.g., $X_{rate} = 0.5 \Rightarrow 50\%$ of chromosomes survive.)

- The survivors "mate", which is implemented by selecting two individuals (many methods to do this) and having them exchange a randomly determined # of genes to create two new "offspring"

Randomly choose crossover pt. = 5 in this example:




- reintroduce random mutation – cause change in gene in a certain percentage. (Except for the "best" individuals - called elitism.)

This important to allowing GA to escape local minima.

Other biologically-based optimization algorithms:
- ✓ Particle swarm optimization
- ✓ Ant colony optimization