

# Möglichkeiten und Grenzen der Schaltungssimulation für Amateure

Von Gunthard Kraus, DG8GB

## 1. Sinn und Zweck der Schaltungssimulation

Jeder, der bereits Schaltungen ausgetüftelt hat, kennt dieses Problem in Form der Fragen, die bei dieser Arbeit quälen:

- Wird das überhaupt so funktionieren, wie ich mir das vorstelle?
- Schaffe ich die geforderten Daten?
- Wie finde ich das Optimum?
- Welche Effekte -- an die ich noch gar nicht denke -- beeinflussen das Ergebnis?
- Wird die Schaltung sich anständig benehmen und z. B. nicht schwingen?
- Wie wirken sich Bauteil- und Fertigungsstreuungen aus?
- Welche Alternativen habe ich, wenn der vorgesehene Weg in eine Sackgasse führt?
- usw.

Die Schaltungssimulation soll und kann die meisten dieser gestellten Fragen beantworten, bevor überhaupt ein Prototyp gefertigt oder ein Lötkolben angefasst wird. Allerdings sind genaue Kenntnisse des gewünschten Schaltungsverhaltens unbedingt erforderlich und das Arbeiten mit der Simulation sowie irgendwelche Änderungen hängen IMMER am Entwickler -- kein Programm kann ihn ersetzen (selbst wenn es die Werbung manchmal so suggeriert!).

**Der Einsatz von Schaltungssimulation bedeutet immer, dass damit kostbare Arbeitszeit gespart oder eine Alternative bzw. Idee sehr schnell geprüft oder das Optimum aus dem Entwurf herausgeholt werden soll.**

## 2. Einteilung der Simulator-Programme

Man unterscheidet:

- a) Simulatoren, die im **Zeitbereich** (= in der Time Domain) arbeiten. Das sind grundsätzlich alle SPICE und PSPICE-Programme. **„SPICE“ selbst steht für „Simulation Program with Integrated Circuit Emphasis“ und PSPICE ist dann einfach „SPICE for the PC“.**

Sie liefern aufgrund ihres Prinzips (= Zerlegung des ansteuernden Signals in sehr kurze, aber dann praktisch lineare Teilstücke) auch alle Informationen über Anschwing- und Startvorgänge sowie über das Verhalten beim Anlegen von Signalen mit beliebiger Kurvenform. Zusätzlich werden alle Nichtlinearitäten in der Schaltung erfasst. Weitere Optionen sind der **AC-Sweep** (= das Wobbeln zur Bestimmung eines Frequenzganges), die **Rauschsimulation** und die **FFT** (= Fast Fourier Transformation) zur Analyse eines Signals in der Frequency Domain („Spektraldarstellung“). Damit bieten sie die umfassendsten Möglichkeiten, aber die Simulationsgenauigkeit hängt stets an der Qualität der „SPICE-Modelle“. Diese werden entweder mitgeliefert oder von Bauteil-Produzenten zur Verfügung gestellt und sie legen letztlich fest, wie genau Simulation und Wirklichkeit übereinstimmen.

- b) Simulatoren, die im **Frequenzbereich** arbeiten, verwenden oft die **S-Parameter** (...manche auch z. B. die Y-Parameter und rechnen dann auf S-Parameter um). Dadurch können zur Ergebniskontrolle -- aber auch zur Eingabe von Bauteil-Eigenschaften in die Simulation -- die bekannten vektoriellen Network-Analyzer eingesetzt werden. Da hierbei oft keine durch Formeln beschriebenen Modelle, sondern Dateien mit **GEMESSENEN Eigenschaften** benützt werden, erreicht man damit im HF- und Mikrowellengebiet (bis 100GHz und mehr) eine gute Übereinstimmung zwischen Simulation und gefertigtem Prototyp. Ein Nachteil ist hierbei, dass es sich grundsätzlich um eine lineare Simulation (sozusagen im Kleinsignalbetrieb) handelt und Informationen über Verzerrungen bei größerer Aussteuerung meist nur mit patentierten Tricks des Programmanbieters gewonnen werden können (z. B. mit „Harmonic Balance“).
- c) **EM-Simulatoren** gehen auf die Form, die Verteilung und die Beziehungen der elektrischen und magnetischen Felder los und dienen zur Analyse von „Strukturen“. Sie können in zwei Gruppen eingeteilt werden:  
Für Gebilde, die aus **Flächen** bestehen (Patchantennen, Koppler, Stripline-Filter etc.) zerlegt man die Gesamtstruktur in sehr kleine rechteckige oder dreieckige „Zellen“, berechnet für jede Zelle die Einzelfelder und addiert am Ende alles auf.  
Für **Drahtantennen** heißt das Stichwort dagegen **„NEC“**. Auch hier zerlegt man die Struktur in kleine Elemente -- in diesem Fall eben kurze **Drahtstücke** mit linearer Strom- und Spannungsverteilung, die sich leichter untersuchen lassen -- und erhält das Gesamtverhalten der Anordnung durch Addition aller Einzelberechnungen.

### 3. SPICE-Simulation für den Amateur

Hier hat sich die Situation sehr entspannt, denn die Einschränkung „für den Amateur“ bedeutet meist Einschnitte beim Geldbeutel. Es ist kein Problem, bis zu 50 000 € oder \$ für eine Simulationssoftware auszugeben, aber das sind dann riesige Maschinen, mit denen man gleich die zugehörige Leiterplatte entwerfen und das EMV-Verhalten untersuchen kann usw. Deshalb ersetzt man als Privatmann einfach Geld durch Zeit. In der Praxis bedeutet das einen Verzicht auf Komfort sowie eine Trennung von Simulation und Platinen-CAD, wobei es für den Platinenentwurf sogar schon sehr preisgünstige und erschwingliche Lösungen gibt, die gleich die EMV-Simulation und auch eine einfache SPICE-Simulation mit einschließen (Beispiel: „Target“).

Zwar liefern die großen PSPICE-Software-Hersteller auch immer kostenlose Test- oder Demoversion ihrer Produkte, aber wer zum ersten Mal mit den Zähnen knirscht, weil mitten in der Arbeit die Meldung: „alle Bauteile aufgebraucht“ oder „Memory restricted in this version“ erscheint, der lässt sehr schnell die Finger davon. Vor allem deshalb, weil man diesen angefangenen Entwurf leider nie in eine aufwendigere (und teurere) Programmvariante übernehmen kann.

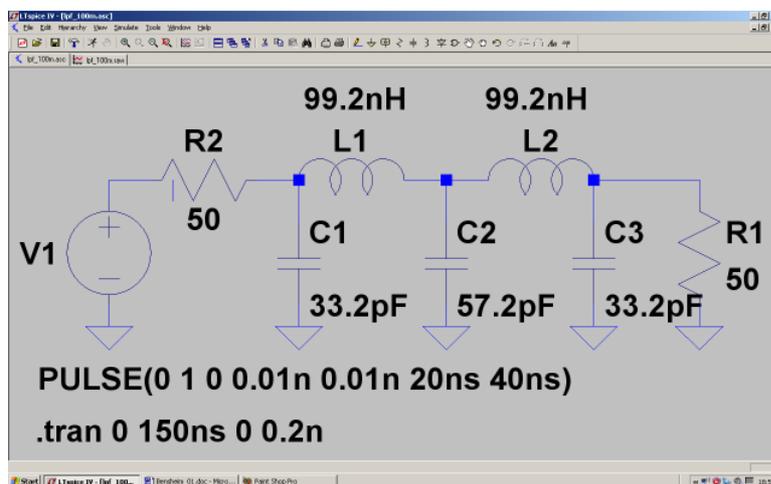
Ein Programm für Amateurzwecke (oder Lehrzwecke!) soll dagegen

- möglichst wenig oder gar nichts kosten
- keine Einschränkungen oder Begrenzungen aufweisen, also voll leistungsfähig sein
- eine problemlose Übernahme neuer Bausteinmodelle aus dem Internet ermöglichen.
- Anbindung an eine Fangemeinde (Community) aufweisen, damit man bei Fragen oder Problemen nicht alleingelassen oder auf einen teuren Support angewiesen ist.

Glücklicherweise hat sich hier die Firma Linear Technologies verdient gemacht, da sie für ihre Kunden und deren Entwicklungsingenieure die kostenlose PSPICE-Software „LTSpice IV“ in ihrer Homepage ([www.linear.com](http://www.linear.com)) zur Verfügung stellt. Da gibt es wirklich keine Begrenzung und sogar die Registrierung vor dem Download kann man verweigern. Die Einbindung neuer Modelle aus dem Internet ist problemlos möglich und es gibt dafür extra einen leicht bedienbaren „Symbol Editor“.

Als einzige Kehrseite hat man eine sehr schlichte Bedieneroberfläche vor sich und die Bedienung ist eine gewöhnungsbedürftige Mischung aus Tastenkommandos, Mausclicks und direkt einzutippenden Anweisungen.

**Hinweis: aus der Homepage des Autors ([www.elektronikschule.de/~krausg](http://www.elektronikschule.de/~krausg)) kann für dieses Programm ein Tutorial (wahlweise in Deutsch oder Englisch) kostenlos heruntergeladen werden. Es enthält eine umfangreiche Sammlung an Praxisprojekten zur Einarbeitung!**



Doch sehen wir uns nun mal den Einsatz von LTSpice am Beispiel eines 110MHz-Tschebyschef-Tiefpasses für einen Systemwiderstand von 50Ω an, dem ein 25MHz-Rechtecksignal zugeführt wird. Dazu muss der Anwender erst diesen Bildschirm hinkriegen.

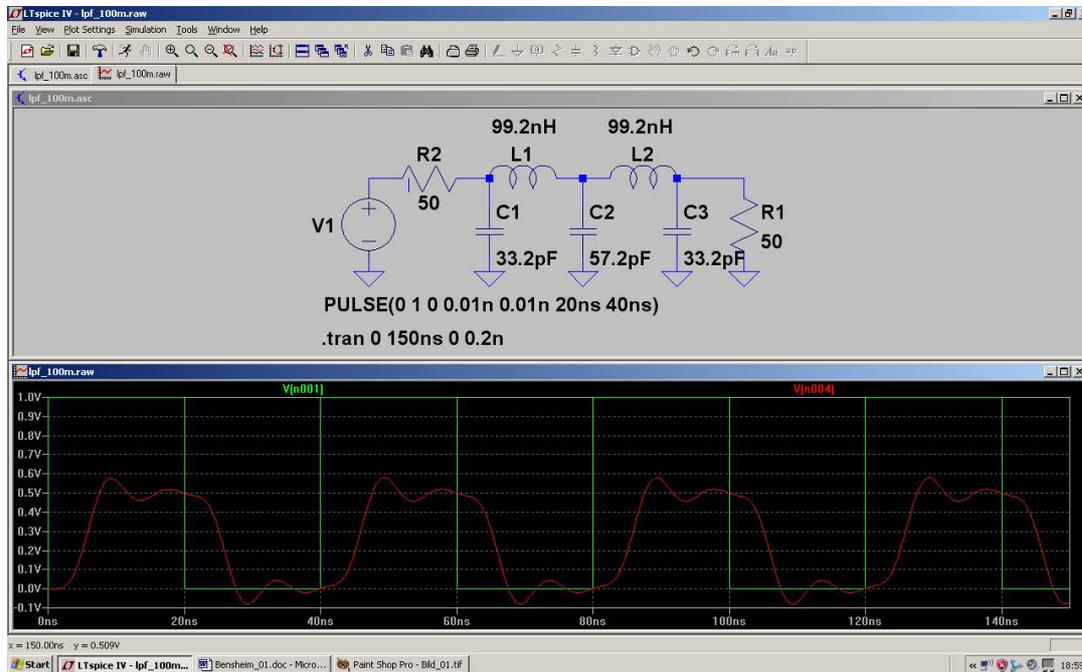
#### Erläuterungen:

Die Zeile „Pulse...“ unterhalb der Schaltung programmiert bei der Spannungsquelle ein Rechtecksignal mit  $U_{min} = 0V$  und  $U_{max} = 1V$ . Die Anstiegs- und Abfallzeit betragen je 0,01ns. Wir arbeiten mit einer Pulslänge von 20ns und einer Periodendauer von 40ns. Das Signal

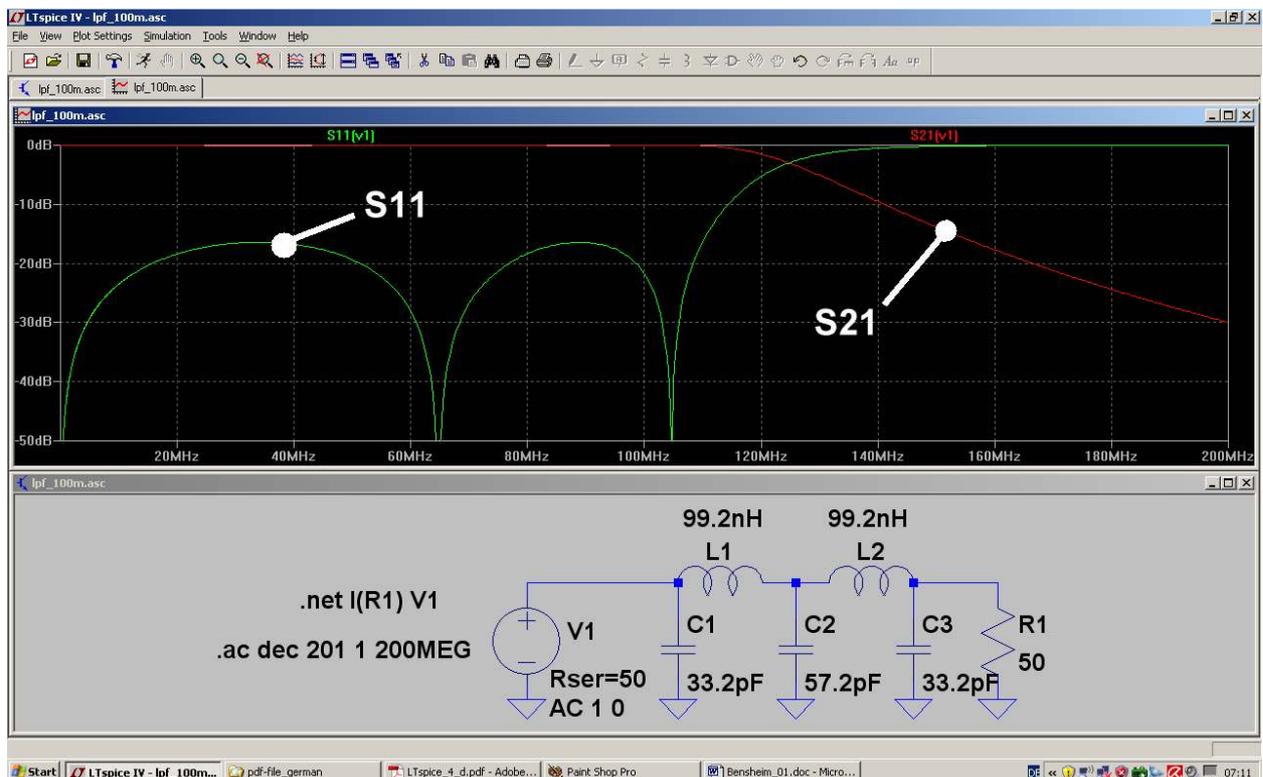
beginnt sofort beim Zeitpunkt  $t = 0s$ .

Die Zeile „tran.“ schreibt vor, dass in der Time Domain der Zeitabschnitt 0...150ns simuliert wird, die Daten sofort ohne Verzögerung weitergegeben werden und der größte zugelassene „Zeitschritt“ (= Time Step) 0,2ns betragen darf.

Nach dem Start der Simulation wird der Bildschirm geteilt und es erscheint zusätzlich ein leeres Ergebnisdiagramm für die vorgesehene Simulationszeit. Nun klickt man mit der Maus an den Punkt der Schaltung, bei dem der Spannungs- oder Stromverlauf interessiert und sofort wird diese Kurve im Diagramm erscheinen (hier sind es der Eingangs- und Ausgangspunkt der Schaltung). Durch verschiedene weitere Mausklicks und Untermenüs kann man sich dann das Ergebnis so hintrimmen, wie es einem am besten gefällt (z. B. mehrere Einzeldiagramme, andere Wertebereiche, andere Farben für die Kurven...)



Sehen wir uns noch die **S-Parameter-Simulation** an. Dazu muss erst der Innenwiderstand der Quelle (= 50Ω) in ihr Property Menü übernommen und ein „**AC-Sweep**“ mit einem „**SPICE Simulation Command**“ programmiert werden (= Wobbeln von 1Hz bis 200MHz mit 201 Punkten pro Dekade, dekadischer Sweep). Die **Spice-Direktive** „**net I(R1) V1**“ bewirkt dann die Bereitstellung der S-Parameter:



### 3. S-Parameter-Simulationen für den Amateur

#### 3.1. Etwas Grundwissen über S-Parameter

Die Systembeschreibung und -berechnung geht ab etwa 1MHz von Größen aus, die auch bei höchsten Frequenzen (HEUTE MEHR ALS 100GHz) leicht messbar sind und man benützt deshalb ein völlig anderes Denkmodell:

Überall im System wird derselbe „**Systemwiderstand**“ verwendet (üblich sind  $75\Omega$  bei der Unterhaltungselektronik und Videotechnik, dagegen  $50\Omega$  bei den meisten anderen Gebieten). Er gilt für den Innenwiderstand der Speisespannungsquellen, die Ein- und Ausgangswiderstände der verwendeten Baugruppen, den Wellenwiderstand aller Verbindungskabel und für alle Abschlusswiderstände.

*Der Kern dieser Sache ist also der Versuch der perfekten Leistungsanpassung (mit  $R_i = R_a$ ) überall im kompletten System!*

Durch geeignete Messgeräte (= Richtkoppler) misst man nun die Abweichungen von diesem Idealfall am Eingang und Ausgang (Fachausdruck: „Ports“) und drückt sie durch „**Reflektionsfaktoren**“ aus (= S-Parameter S11 und S22).

Außerdem misst man die Ausgangsleistung, wenn man dem Eingang Leistung zuführt und drückt das durch die „**Forward Transmission**“ (= S-Parameter S21) aus.

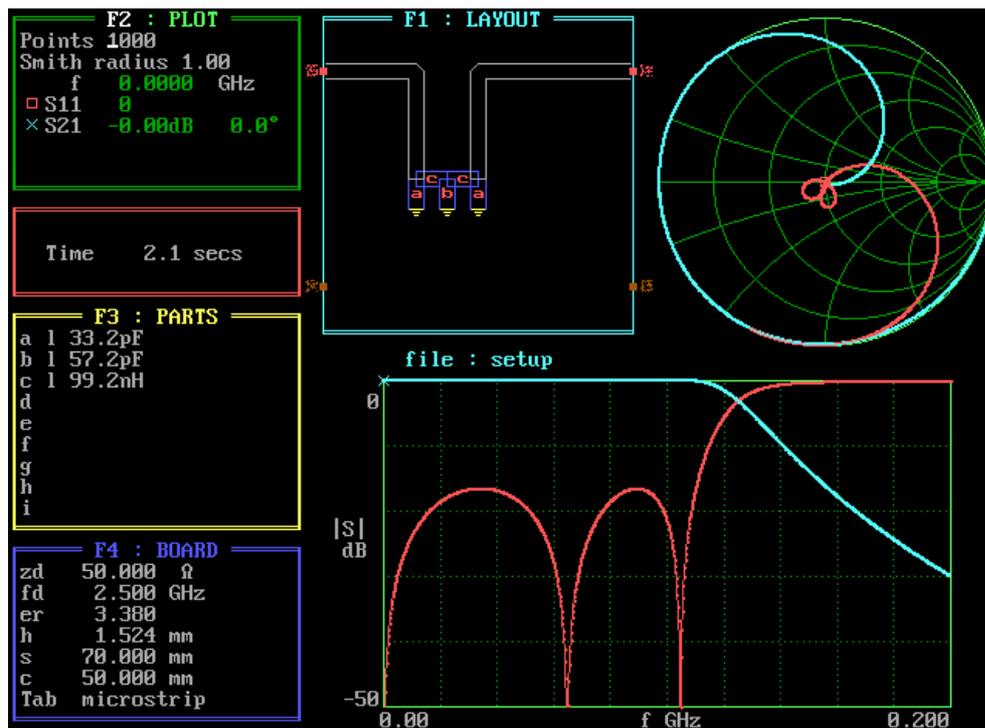
Für die Rückwirkungen in einem Baustein vom Ausgang zurück zum Eingang gibt es dann noch die „**Reverse Transmission**“ (= S-Parameter S12).

Bei einem kompletten System sind dann viele Bausteine in Reihe geschaltet und mit Hilfe geeigneter Software lässt sich dann leicht das Gesamtverhalten bestimmen.

#### 3.2. Einsatz von PUFF

Dieses altbekannte DOS-Programm, erhältlich bei der Zeitschrift „**UKW-Berichte**“ ([www.ukw-berichte.de](http://www.ukw-berichte.de)), war wohl das „Standardprogramm der Amateure“ und hat auch nach fast 20 Jahren nicht ausgedient. Niedriger Preis, Minimaler Speicherplatzbedarf, hohe Stabilität und Simulationsgenauigkeit, eine einfache und einprägsame Bedienung über Tastenkommandos, das Vorhandensein der wichtigsten Grundmodelle und ein klarer übersichtlicher Bildschirm machen es zum „**HF-Taschenrechner des Entwicklers**“, mit dem man schnell neue Ideen austesten kann. Dazu wurde es immer wieder aufgepäppelt, gepflegt und sogar an Windows XP angepasst.

**Hinweis: die Homepage des Autors ([www.elektronikschule.de/~krausg](http://www.elektronikschule.de/~krausg)) enthält hierzu alle Unterlagen und Hilfen (einschließlich eines ausführlichen Katalogs der Bedienkommandos mit Erläuterungen und einer Anleitung bzw. einem „PUFF\_XP-Servicepack“ zum störungsfreien Betrieb unter Windows XP).**



So sieht nun die Simulation des 110MHz-Tiefpasses aus.

Bauteil-Eingabe, Ergebnis-Ausgabe per Smithchart oder Kartesischem Diagramm, Platinen-Layout und die wichtigsten Werkstoffdaten werden komplett auf EINEM Bildschirm präsentiert!

Die Nachteile seien nicht verschwiegen: sowohl die Eingabe wie auch die Grafikausgabe sind arg in die Jahre gekommen (besonders aber das Ausdrucken). **Man hilft sich da über eine zusätzlich installierte und aufgerufene PCX-Ausgabe des Bildschirm-Screenshots.** Es fehlen wichtige Modelle für Störstellen z. B. bei Microstrip-Leitungen usw -- und es wird sicher nicht mehr auf eine Windows-Bedieneroberfläche umgestellt: Trotzdem: für schnelles Testen einer Idee von der Geschwindigkeit und der Bedienung her unschlagbar -- bei gleicher Genauigkeit wie große, teure Maschinen! Und die Beschränkungen muss man halt hinnehmen.

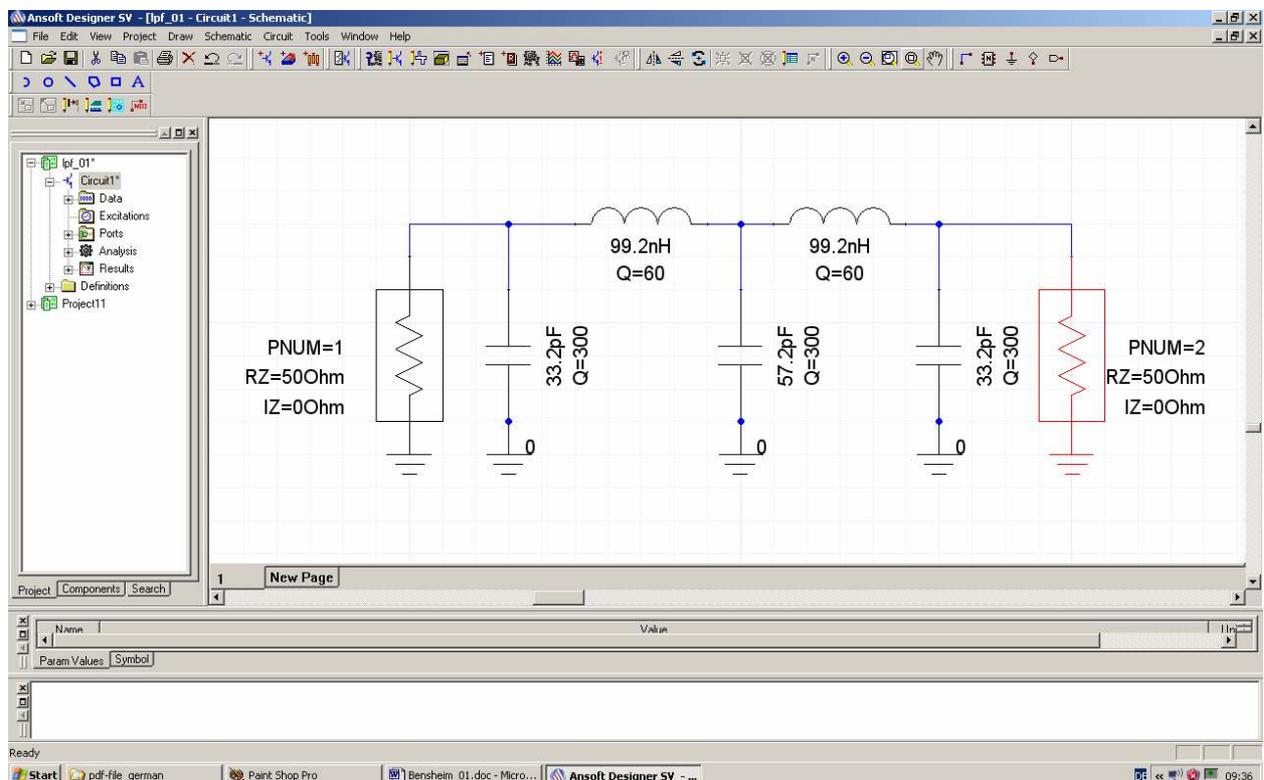
### 3.2. Der Ansoft Designer SV

Das ist natürlich modernster Stand der Simulationstechnik: ein Windows-Programm vom Feinsten und mit allen nur denkbaren Möglichkeiten. Hunderte von Bauteil- und Störstellenmodellen decken alle denkbaren Anwendungsfälle bis hin zum Hohlleiter ab, ein integrierter Filtercalculator erspart die Suche nach einem geeigneten Filter-Berechnungsprogramm, die gesamte lineare Simulation ist ohne Einschränkungen freigegeben -- wo liegt das Problem, wenn Ansoft eine solch teure Software für Studienzwecke (SV bedeutet: Studentenversion...) kostenlos freigibt und zum Download ([www.ansoft.com](http://www.ansoft.com)) bereitstellt?

Nun, da stecken natürlich sehr raffinierte Marketing-Überlegungen dahinter: wenn nämlich Studenten ein solches Programm kennen und lieben lernen, werden sie mit großer Sicherheit am späteren Arbeitsplatz dasselbe verwenden wollen. Also setzt man die Begrenzungen exakt da an, wo es einer Firma weh tut, die heimlich mit dieser Version arbeiten möchte: man sperrt die direkte Übernahme eines Entwurfs in einen Platinentwurf, man sperrt die (sehr gute, selbst ausgefüllte und patentierte) nichtlineare Simulation, man reduziert sehr stark die Möglichkeiten der automatischen Optimierung usw.

Für Amateuranwendungen bleibt das trotzdem ideal und perfekt, aber gegenüber PUFF kommt doch kurze Wehmut auf: ein solch aufwendiges Programm mit so vielen fantastischen Möglichkeiten erfordert halt einfach größeren Einarbeitungs- und Bedieneraufwand. So ist die Eingabe samt Simulation von der Ergebnisausgabe völlig getrennt, da man dem Programm hinterher erst sagen muss, was man gerne sehen möchte. Dafür gibt es nichts, was man sich nicht zeigen lassen kann: seien es Stabilitätswerte, S-Parameter, Rauschzahlen usw....

**Hinweis: aus der Homepage des Autors ([www.elektronikschule.de/~krausg](http://www.elektronikschule.de/~krausg)) kann dazu ein umfangreiches Tutorial (in Deutsch oder Englisch) kostenlos heruntergeladen werden. Es enthält viele Einsteigerprojekte mit aktiven oder passiven Schaltungen.**



So sieht Teil 1 der Arbeit, also die Eingabe der Schaltung für die Simulation, aus....

... und dieser Bildschirm erscheint, wenn man nach der Simulation unter „Results“ die Ausgabe von **S11** und **S22** in kartesischen Koordinaten verlangt.



## 4. EM-Simulationen für den Amateur

### 4.1. Simulation von flächigen Strukturen mit Sonnet Lite

Auch hier gilt wie im vorigen Ansoft-Kapitel die Bemerkung über dieselbe gute Marketing-Strategie der Firma Sonnet. Zwar wird bei diesem Programm eine echte Begrenzung über den maximal belegbaren Speicherplatz vorgenommen, aber im Endergebnis kann man gut damit leben und muss höchstens eine etwas höhere Ungenauigkeit beim Simulationsergebnis akzeptieren.

Dabei gilt:

- a) wer nur das Programm (kostenlos) herunter lädt und installiert, bekommt 1 Megabyte Speicherplatz genehmigt.
- b) Wer sich anschließend online registrieren lässt und eine Lizenz beantragt, kann plötzlich mit 16 Megabyte arbeiten.

Da dauernd Verbesserungen zum Einsparen von Speicherplatz in den Simulator einfließen, wird die Situation für die „Lite-Version“ immer günstiger....

Natürlich ist die Einarbeitung in diese Materie deutlich schwieriger als der Umgang mit PSPICE oder PUFF. Das geht schon los mit der erwähnten Aufteilung der Flächen in einzelne kleine Zellen und die Einbindung der verschiedenen verwendeten Materialien. Dann muss die Signaleinspeisung korrekt vorgenommen und immer wieder mal der bereits verbrauchte Speicherplatz kontrolliert werden.

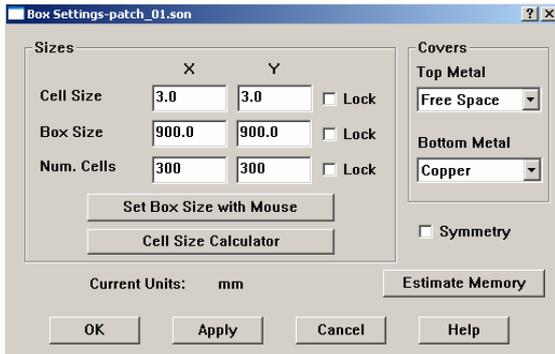
**Hinweis: in der Homepage des Autors ([www.elektronikschule.de/~krausg](http://www.elektronikschule.de/~krausg)) gibt es eine Rubrik „Meine Veröffentlichungen in der Zeitschrift UKW-Berichte“. Darin findet sich der eingescannte Abdruck des Artikels „Das interessante Programm -- Heute: SonnetLite 9.51“. Er enthält eine genaue Gebrauchsanleitung zur Bedienung des Programms und zwei unterschiedliche Patchantennen-Simulationen als Anwendungsbeispiele. Diese Anleitung kann auch noch für die modernste Programmversion verwendet werden.**

Zur Einarbeitung und zur Hilfe bei Problemen: es gibt sehr viele Beispiele, Application Notes und Tipps in der Homepage ([www.sonnetsoftware.com](http://www.sonnetsoftware.com)) bzw. im Internet und die FAQ-Abteilung ist wirklich erste Klasse. Insgesamt ist alles prima, aber eine Sache tut weh: absichtlich ist die Simulation der Strahlungsdiagramme sowie die Antennengewinn-Berechnung gesperrt. Schade, schade...

Doch sehen wir uns mal in konzentrierter Form die Simulation einer quadratischen Patchantenne aus FR4-Material für die Frequenz  $f = 435 \text{ MHz}$  an, durch die der Eingangswiderstand der Antenne an der Patchkante ermittelt werden soll. Dabei muss man folgendes beachten:

**Sonnet ist nur ein Analyseprogramm und KEIN Entwurfsprogramm. Es testet nur die fertigen Ideen des Anwenders!**

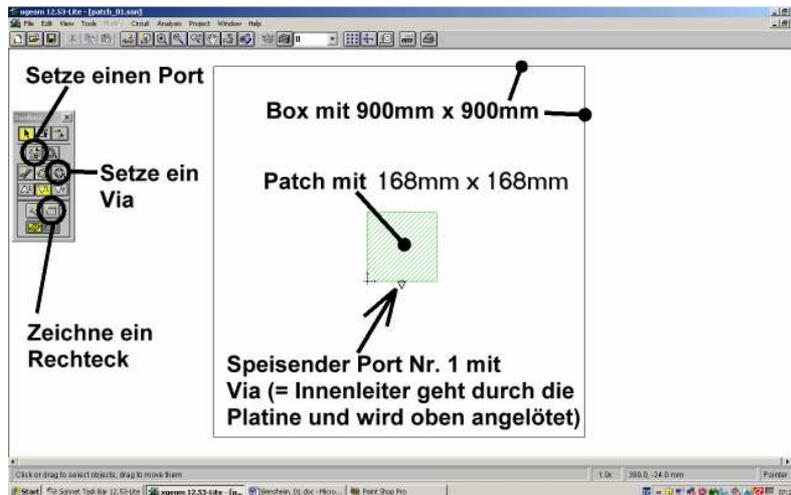
Folglich muss man den „Grundentwurf“ der Antenne z. B. mit dem **kostenlosen DOS-Programm „patch16“** erstellen, das im Internet (**nach der Eingabe „patch16.zip“ in die Suchmaschine**) leicht zu finden ist und durch seine einfache und leichte Bedienung glänzt. Als Trägermaterial dient eine beidseitig kupferbeschichtete FR4-Platte mit den Abmessungen eines DIN A4-Blattes (210mm x 297mm) und einer Dicke von 1,52mm ( $= 0,06$ ). Die Dielektrizitätskonstante liegt bei 4,35 und als Verlustfaktor ist  $\tan\delta = 0,01$  zu verwenden. Das Patch-Programm schlägt nun **6,61“ x 6,61“ = 168mm x 168mm** als Antennenabmessungen vor und wir wollen mit Sonnet die Brauchbarkeit des Vorschlags prüfen.



Nach der Eingabe des verwendeten Platinenwerkstoffs und der beteiligten Metalle muss man sich intensiv um die „Box“-Daten kümmern. Sonnet verwendet als Grundlage der Simulation eine rechteckige Kiste, in die unsere Platine hineingelegt wird. Die Feldverteilung in einer solchen Box ist bekannt und berechenbar. Damit muss der Anwender nun die Details programmieren.

Der erste Schritt ist die Eingabe der **Box-Abmessungen** (mit einer halben Wellenlänge Abstand der Wände in jeder Richtung von der zu untersuchenden Struktur) und **die Länge bzw. Breite sowie die Anzahl der Zellen**, in die wir unsere Struktur zerlegen müssen. Mehr als 5% der Wellenlänge dürfen es bei der Zellenlänge und der Zellenbreite nicht sein, sonst leidet deutlich die Genauigkeit.

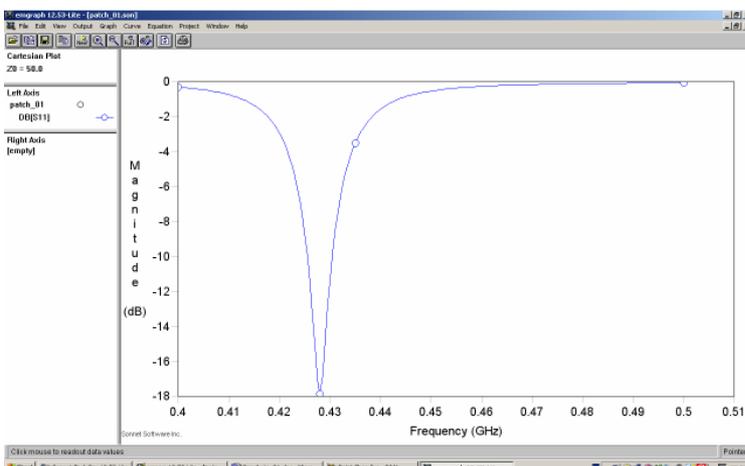
Viel weniger als 1% sollen es aus demselben Grund auch wieder nicht sein und so liegt das Optimum bei 1...2%.



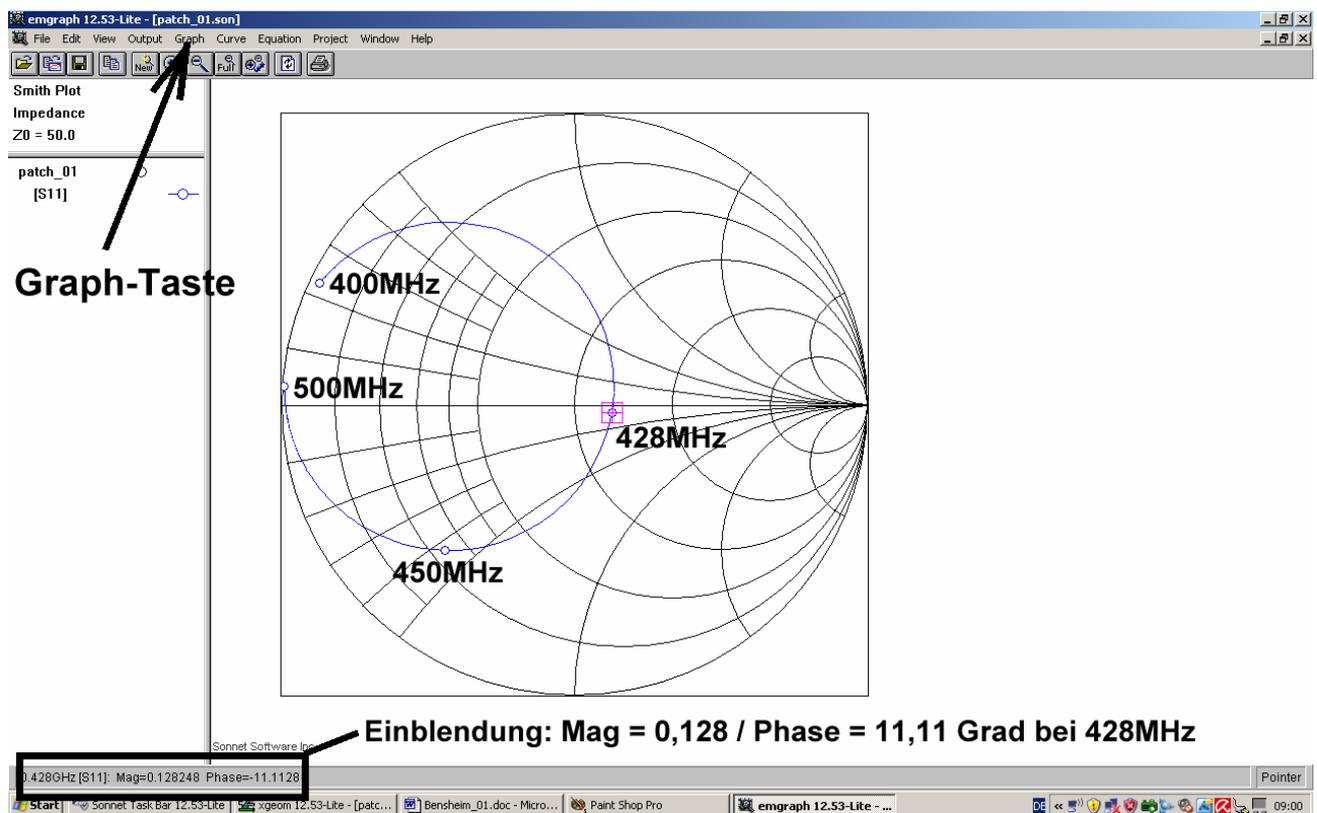
Der nebenstehende Screenshot zeigt den Sonnet-Bildschirm nach Fertigstellung der Eingabearbeit, er wird gleich gespeichert.

Als nächstes wird der zu sweepende Frequenzbereich programmiert -- hier sind 400 bis 500 MHz vernünftig. Erst dann kann man die Simulation starten und bekommt (wie bei großen Programmen üblich) erst mal **NICHTS zu sehen**.

Die Programmierung der Ergebnispräsentation ist nämlich eine eigene Prozedur (View Response / Add to graph) und erzeugt z. B. diesen Bildschirm:



Man sieht sehr schön, dass die Resonanzfrequenz der Antenne noch ca. 7 MHz zu niedrig ist. Außerdem erkennt man bei dieser Darstellung nicht, ob der Eingangswiderstand, (der bei Resonanz als Parallelschaltung aus dem Strahlungswiderstand und den Verlusten des Platinenmaterials gebildet wird) nun größer oder kleiner als 50 Ohm ist. Dazu brauchen wir das Smithchart und dafür gibt es einen Button „Graph“ in der Menüleiste. Darnach kommt die Auswahl zwischen „Cartesian“ und „Smith“, der Rest erledigt sich von selbst.



Wir haben also direkt bei der Resonanz mit einem Reflektionsfaktor  $r$  (= „Mag“) von etwa +0,128 zu rechnen. Da kann man nun sehr leicht mit dem Taschenrechner den zugehörigen Widerstand bestimmen:

$$R_{IN} = 50\Omega \cdot \frac{1+r}{1-r} = 50\Omega \cdot \frac{1+0,128}{1-0,128} = 64,67\Omega$$

Wer sich **nur für den reinen Strahlungswiderstand** an der Patchkante interessiert, der wiederholt die Simulation, setzt dabei die **Platinenverluste auf Null** und schaltet zusätzlich auf „**lossless**“ beim verwendeten Metall. Dann bleibt nur noch die Abstrahlung als „Energiefresser“ übrig.

Schon eine schöne Sache und die Erfahrung zeigt, dass die Simulation recht genau mit den Messergebnissen am gefertigten Prototyp übereinstimmt. Nur die Resonanzfrequenz wird immer etwas zu hoch ausgegeben (meist zwischen 1 und 3%) und **das ist leider typisch für alle EM-Simulatoren**.

Also sinniert man nicht lange herum, sondern fertigt eine Musterplatine, ermittelt die Abweichungen von den Vorgaben und korrigiert mit dem Taschenrechner die Abmessungen im entsprechenden Frequenzverhältnis (dabei ergeben größere Abmessungen eine größere Wellenlänge und folglich eine niedrigere Frequenz).

Und bei der Sache mit den Strahlungsdiagrammen der Antennen kann man nur hoffen....

## 4.2. Simulation von Drahtantennen mit NEC

Mit NEC (= Numerical Electric Code) bezeichnet man eine vom Lawrence Livermore Labor im Jahre 1981 für die Navy entwickelte Simulationsmethode für Drahtantennen. Dabei wird die Antenne in sehr kurze Stücke = „segments“ zerlegt, auf denen sich der Strom und die Spannung (fast) nur linear ändern (Siehe SPICE). Damit lassen sich erstaunlich genaue Simulationen durchführen.

**Der Standard -- für den es sehr viele, auch kostenlose, Anwendungssoftware gibt -- ist NEC2.**

Die Entwicklung ist natürlich weitergegangen und die Schwächen von NEC2 (z. B. falsche Berechnung von Strukturen mit sich sehr nahe kreuzenden oder im Boden verbuddelten Drähten) wurden erst mit NEC4 ausgebügelt.

Aber:

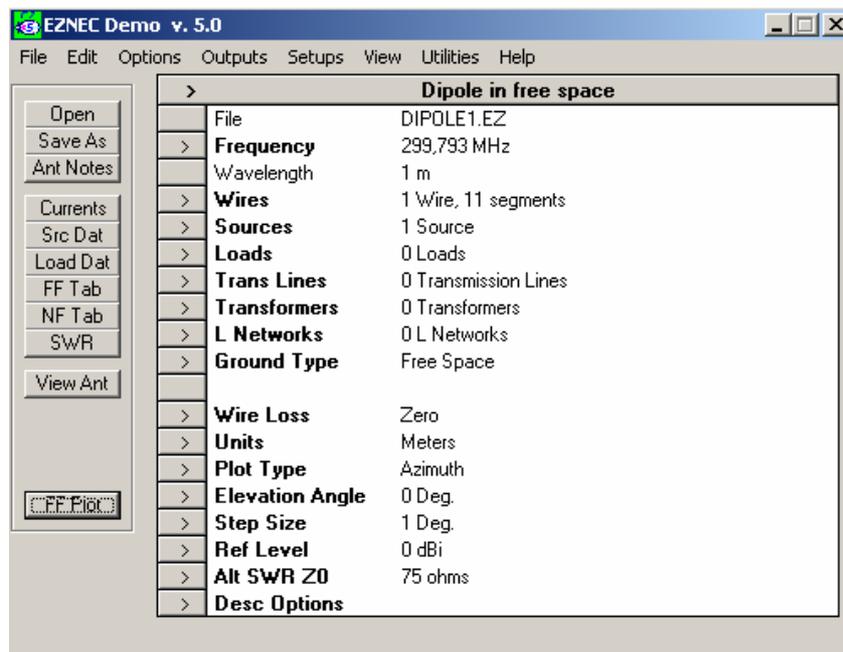
**NEC4 war bis vor kurzer Zeit für den Export gesperrt und galt früher als geheim. Heute ist es auch außerhalb der USA erhältlich, kostet aber eine hübsche Stange Geld (normalerweise so zwischen 2000 und 4000\$).**

Also begnügt man sich als normaler Mensch (= kleiner Geldbeutel) mit **NEC2** und hat dafür die Qual der Wahl bei der Auswahl des geeigneten Programms aus dem Internet (ehrlich!).

Da ist es nun sinnvoll, etwas zu wählen, das

- a) schon weit verbreitet und beliebt ist
- b) eine ansprechende und leicht zu bedienende Eingabe-Oberfläche besitzt
- c) eine leicht verständliche Ergebnisausgabe aufweist
- d) regelmäßig und gut gepflegt wird
- e) eine Nah- und Fernfeld-Simulation durchführen kann
- f) von einer Fangemeinde mit vielen Beispielen und Vorlagen und Hilfsbereitschaft unterstützt wird
- g) ein gutes Handbuch und eine gute Einsteigeranleitung vorweist
- h) den Amateur nicht finanziell ruiniert.

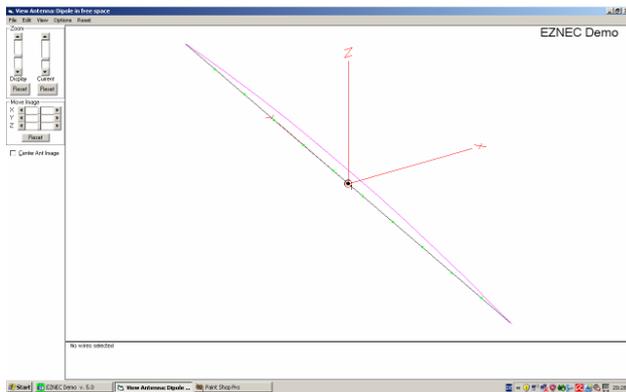
Aus diesen Gründen greifen viele Amateure und Antennenentwickler seit Jahren zu **EZNEC** (Autor: Roy Lewallen, [www.eznec.com](http://www.eznec.com)) und der aktuelle Preis von 89\$ beim direkten Download und Bezahlung per Kreditkarte ist mehr als fair,



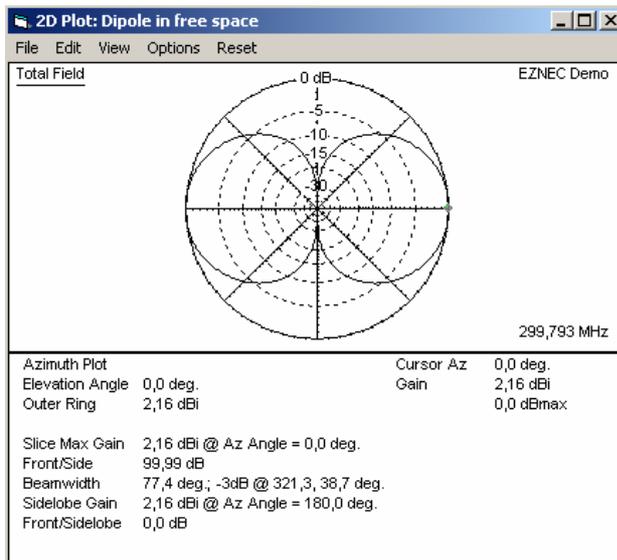
Werfen wir mal einen Blick auf die Bedieneroberfläche der neuesten Version 5.0 dieses Programms, wenn aus dem mitgelieferten Beispielvorrat ein einfacher Lambda-Halbe-Dipol für 299,793MHz ausgewählt wird:

(Für die Untersuchung eigener Antennen sollte man einen großen Vorrat an verschiedenen simulierten Antennentypen sammeln oder erstellen. EZNEC hat nämlich eine etwas andere Philosophie und zeigt kein „leeres Blatt“ beim Start, sondern immer das letzte geladene Beispiel. Also holt man sich einfach aus dem Archiv eine passende Antenne und ändert der Reihe nach in den einzelnen Zeilen des Menüs die Werte. Mal was anderes, nämlich „Re-

Engineering“, aber deswegen nicht schlechter..)



So sieht der Bildschirm nach einem Klick auf „**View Antenna**“ aus...



..und so nach einem Klick auf „**FF Plot**“, da kann man sogar mit dem Cursor die dargestellte Kurve abfahren.

Natürlich lässt sich auch ein anderer Elevationswinkel wählen und unter „**Setups**“ im Hauptmenü findet man sogar die Option „**Near Field**“

**Hinweis:** wer jedoch Strahlungsdiagramme lieber in räumlicher (= 3D - ) Darstellung und in poppigen Farben sehen möchte, der hole sich mal das Programm „**4nec2**“ aus dem Internet!

## 5. Abschließende Hinweise zur Antennensimulation

Es gibt natürlich -- wieder im Internet -- Antennenberechnungsprogramme, die direkt mit passend zugeschnittenen Näherungsformeln arbeiten. Das Programm „**patch16**“ haben wir im Sonnet-Kapitel bereits erwähnt, aber es gibt ein DOS-Programm im Web, mit dem eine Vielzahl von Antennenformen (von Draht- und Patchantennen, von Antennen-Arrays über Log-Periodic-Antennas bis hin zu Parabolspiegeln und Hornantennen) durch Einträge der geforderten Daten am Bildschirm dimensioniert werden können. Dieser kostenlose Alleskönner heißt „**pcaad21.zip**“ und hat selbst als DOS-Version noch heute seinen Reiz nicht verloren.

Und nun: viel Spaß und Erfolg mit eigenen Projekten, bei denen die Simulation nicht nur zum Entwurf, sondern auch anschließend zur Fehleranalyse und zur Fertigung entwickelt wird!